

Учебно-методический комплекс

Web технологияны қолдана отырып, web паркшасын құру



МАЗМҰНЫ

Теориялық бөлімі

1 бөлім. Интернет-бағдарламалаудың технология негіздері.	5
Web-сайтты ұйымдастыру (Web-құжаттар).....	5
Гипермәтіндік құжаттарды құру технологиялары.....	6
Динамикалық бағдарламалау тілдері.....	8
Сервер тұжырымдамасы.....	10
Серверді орнату.	12
HTML –мен танысу... ..	14
HTML: жалпы құжат құрылымы	15
HTML: тізімдер.....	18
HTML: кестелер,фреймдер.....	20
HTML: формалар.....	25
CSS-пен танысу(стильдердің каскадты кестелері).....	28
CSS: тізімдермен жұмыс істеу.....	29
CSS: фонмен жұмыс істеу.....	35
CSS: элементтердің орналасуы және көрінуі	38
Браузер деңгейінде CSS-ті қолдау	41
JavaScript кіріспе	41
HTML құжатына JavaScript енгізу.	43
Мәліметтер типі, айнымалылар, өрнектер	47
JavaScript-тегі арифметикалық операторлар	51
Функция және әдістер ұғымы. Объектілер және қасиеттер.	54
Объектілер және қасиеттер.	57
JavaScript-тегі құжаттың объектілік моделі.	61
2 бөлім. XML негіздері.	64
XML шолу	64
XML синтаксисі	66
DTD сызбалары – құжат түрін анықтау	67
XML - XSD сызбасы	70
XML атаулар кеңістігі	73
XML сілтемелері – Xpath тілі	77
XML — DOM құжаттың объектілік моделі	79
XML үшін Simple API.	81
3 бөлім. PHP бағдарламалау.	84
PHP-ге кіріспе	84
Негізгі синтаксис. Деректер типі. Айнымалылар. Өрнектер.	86
Операциялар.	90
PHP-де жолдар мен массивтермен жұмыс. Функциялар.	92
Кластар мен объектілер.	94
PHP деректер базасымен жұмыс.	96
Сервер жағында бағдарламалау.	98
MySQL деректер қоры.	99
Оңтайлы жұмыс істеу үшін веб-жүйені теңшеу.	100

Коммерциялық емес жоспар жүйесінің модульдерімен жұмыс істеу әдістері.	101
LDAP сервері арқылы пайдаланушыларды импорттау әдістері.	102
Сайтты рұқсатсыз кіруден қорғау құралдары.	105
Жоғары жүктеме кезінде сайттың өміршендігін қамтамасыз ету құралдары.	106
Іздеуді онталайндыру және сайтқа кіруді бақылау құралдары.	108
Әр түрлі CMS мысалында контентті басқару жүйесі үшін үлгілер жасау.	111
Зертханалық-практикалық бөлім	115
1бөлім. Интернет-бағдарламалаудың технология негіздері.	115
Web-сайтты ұйымдастыру	115
Гипермәтіндік құжаттарды құру технологиялары.....	117
Динамикалық бағдарламалау тілдері.	119
Сервер тұжырымдамасы.	121
Серверді орнату.	123
HTML –мен танысу.	127
HTML: жалпы құжат құрылымы	131
HTML: тізімдер	136
HTML: кестелер,фреймдер	139
HTML: формалар	143
CSS-пен танысу(стильдердің каскадты кестелері)	153
CSS: тізімдермен жұмыс істеу	157
CSS: фонмен жұмыс істеу.	160
CSS: элементтердің орналасуы және көрінуі	163
Браузер деңгейінде CSS-ті қолдау	168
JavaScript кіріспе	169
HTML құжатына JavaScript енгізу	171
Мәліметтер типі, айнымалылар, өрнектер	174
Функция және әдістер ұғымы.	177
Объектілер және қасиеттер.	181
JavaScript-тегі құжаттың объектілік моделі.	184
2 бөлім. XML негіздері.	188
XML шолу	188
XML синтаксисі	189
DTD сызбалары – құжат түрін анықтау	191
XML - XSD сызбасы	192
XML атаулар кеңістігі	196
XML сілтемелері – Xpath тілі	199
XML — DOM құжаттың объектілік моделі	201
XML үшін Simple API.	203
XML деректерін HTML элементтерімен байланыстыру	205
3 бөлім. PHP бағдарламалау.	209
PHP-ге кіріспе	209

Негізгі синтаксис. Деректер типі. Айнымалылар. Өрнектер.	211
Негізгі синтаксис. Деректер типі. Айнымалылар. Өрнектер.	216
Операциялар.	221
PHP-де жолдар мен массивтермен жұмыс. Функциялар.	224
Кластар мен объектілер.	228
Кластар мен объектілер.	230
PHP деректер базасымен жұмыс.	232
PHP деректер базасымен жұмыс.	237
Сервер жағында бағдарламалау.	238
Сервер жағында бағдарламалау.	241
MySQL деректер қоры.	243
MySQL деректер қоры.	243
MySQL деректер қоры.	244
Оңтайлы жұмыс істеу үшін веб-жүйені теңшеу.	245
Коммерциялық емес жоспар жүйесінің модульдерімен жұмыс істеу әдістері.	247
LDAP сервері арқылы пайдаланушыларды импорттау әдістері.....	251
Сайтты рұқсатсыз кіруден қорғау құралдары.	255
Жоғары жүктеме кезінде сайттың өміршеңдігін қамтамасыз ету құралдары.....	260
Іздеуді оңталайндыру және сайтқа кіруді бақылау құралдары.	262
Іздеуді оңталайндыру және сайтқа кіруді бақылау құралдары.	265
Әр түрлі CMS мысалында контентті басқару жүйесі үшін үлгілер жасау.	267
Әр түрлі CMS мысалында контентті басқару жүйесі үшін үлгілер жасау.	268
Сайттың жұмыс қабілетін тексертеді	268
ДИАГНОСТИКО-КОНТРОЛИРУЮЩИЙ БЛОК	269

1 бөлім. Интернет-бағдарламалаудың технология негіздері.

Кіріспе. Web-сайтты ұйымдастыру (Web-құжаттар).

Пәннің мақсаты мен міндеттері Web-технологияларды қолданып Web-сайтты ұйымдастырып білу.

Қазіргі таңда бүкіл әлемде болып жатқан кең ауқымды қайта құрулар ақпараттық-коммуникациялық технологиялардың (бұдан әрі - АКТ) қарқынды дамуына байланысты.

Жаңа ақпараттық қоғамның негізін қалаушы сипаттамаларының бірі оның ғаламдық сипаты болып табылады. Web – бағдарламалау негіздерін меңгеру үшін бүкіләлемдік желіні қолдана отырып корпоративті құрылымдардың жұмыс принциптерін, динамикалық сайттар мен Web –порталдарды әзірлеуді және оларды бағдарламалық қамтамасыз етуді, Интернет - бизнесті ұйымдастыру моделдерін, Интернет сервистері мен Web-қосымшаларын құру технологияларын білу қажет.

Интернетте қолданатын ұйымдастыру принциптерін, қызмет ету мен ақпаратты өңдеу технологияларын білуі, Қазіргі интернет технологиялары негізінде программалық қосымшаларды құруды білуі, Интернеттің қазіргі перспективалары мен дамуы беталыстары туралы түсінгі болуы қажет.

Барлық WEB-беттер гипермәтіннің арнайы белгілеу тілдерінде (HTML, CSS, JS) жазылады. Бұл Web – бағдарламалау тілдерімәтінді, графикалық кескінді, дауыстық файлды, видеоклиптерді шығару әдістерін көрсете алады. Компьютер экранындағы көрінісі браузер программасымен анықталатын, құжаттардың құрылымы мен гипербайланыстары HTML тілінің көмегімен сипатталады.

Web сервер арқылы миллиондаған электрондық құжаттардың бір-бірімен байланыстарының бірігуі. Web- кеңістігін құрайтын жекелеген құжаттар Web-беттер деп аталады. Тақырып бойынша біріктірілген Web –беттер тобы Web – түйіндер деп аталады (басқаша атауы – Web-сайт немесе жай ғана сайт). Web – беттерді көруге арналған программа браузерлер деп аталады. Электрондық пошта: немесе e-mail, Интернеттің барлық желілерінде қол жеткізімді. Ол адамдарға файлдарды бүкіләлем бойынша тез жіберуге мүмкіндік береді. Бұл қарапайым мәтіндік белгілеу немесе ішіне графика мен басқа да мазмұнды алып жатуы мүмкін. Файлдарды жіберу: файлдарды бір компьютерден екінші компьютерге көшіреді. Интернет арқылы файлдарды тасымалдау хаттамасы (File Transfer Protocol, FTP) сияқты мәліметтерді тасымалдайтын арнайы басқарушы қағидалар немесе хаттамалар бар: белгілі бір тақырып бойынша мақалалармен алмасатын, программалық өнімдерді қолдауды қамтамасыз ететін немесе сұрақтардың түрлі категорияларына жауаптар ұсынатын компьютерлер желісі. Жаңалықтық топтар (newsgroups): USENET-тегі желілік конференция, ол дискуссиялар мен жаңалықтар алмасуды жүргізу үшін ұйымдастырылған. Мәлімдемелерді оқу мен жіберу тұтынушы компьютерінде жүктеліп, жаңалықтар серверімен байланысатын программа арқылы жүзеге асады. WEB браузер. WEB сайттар HTML атты программалау тілі арқылы құрылады. (Hyper Text Markup Language). WEB парақтарды сіздер арнайы программалау арқылы көре аласыздар. Оның аты WEB браузер деп аталады (Web browser).. Ең кеңінен тараған веб-браузерлер: Internet Explorer (Microsoft.com), Netscape Navigator (Netscape.com), Opera (Opera.com), Mozilla (Mozilla.org), AOL (aol.com). HTML тілінде құрылған браузер сіздерге ақпараттың барлық түрлерін,

оның ішінде мәтіндерді, графиканы, дыбысты, анимацияны, слайд – шоууды, бейне және басқа көптеген ақпараттарды көруге болады. Сіздер WEB браузерді пайдалануды білесіздер, ақпарат алуды, оны іздегенде белгілі уақыт аралығы жеткілікті. WEB браузерді пайдаланылуы WEB браузер сіздерге жеке WEB сайттардың, жеке WEB парақтардан көрсетеді. WEB сайттар былай ұйымдастырылады: ақпараттардың байланысы бірге жиналады, олар

WEB парақтарға топталады, яғни біздің көз алдымызға немесе журналды елестетеді, мысалы, сіздің мектебіңізде WEB сайт бар деп есептейік, олардың көптеген парақтары бар делік. Ал жергілікті кітапханалардың басқа WEB сайты болады, яғни олардың парақтары да бөлек болады немесе олардың парақтары бөлек беттерден тұрады. Әрқашанда WEB сайттардың басы болады, ол үй парағы деп аталады. (HOME PAGE). Үй парағы дегеніміз: ол журналдың сыртқы беті немесе газеттің бірінші бетімен пара-пар. Әдетте ол бетте адамдарды еліктіретін картинкалар болады, яғни ол дегеніміз сайттың мазмұнын ашатын тақырып немесе идея. Оның құрамында сайттардың картасы немесе дәлелдеуіш панель болады. Олар сайттардағы басқа беттердің қандай екенін білдіреді. Браузер беттен –бетке секіруге мүмкіндік береді. Бұны желінің ішін аралау деп айтады. Web сайттар ұйымдастыру. Бүкіл әлемдік тордың мүмкіндіктері өте көп және өте қолайлы. Сілтемелер көмегі арқылы сайттарға оңай шолу жасауға болады. Кейбір жағдайда Web сайтта адасып қалуға да болады. Ол қандай жағдайда? Егер сіз ақпараттарды өз ретімен дұрыс пайдаланбасаңыз. Сіз оңай болу үшін сайттың ұйымдасу құрылымын жақсы білгеніңіз жөн. Web сайттар әдетте 3 түрлі жағдайда ұйымдастырылады: 1)сызықты Web сайт; 2)ағаш тәрізді құрылымды Web сайт; 3)жүйесіз Web сайт (ретсіз). Сызықты Web сайт – кітап бойынша әдетте 1-ші парақтан 2-ші парағын ашу т.с.с. ашу арқылы құрылады. Сіз 1-ші парағын 1) (үй парағын) ашу арқылы бастайсыз да 2-ші параққа көшесіз, одан кейін 3-ші параққа т.с.с кетеді. Ағаш тіріздес құрылымды Web сайты – бұл ағаштың гендік таралымы тәріздес болып келген әр түрлі опциясы бар және сайттағы ақпараттарды көретін құрылымы бар ұйымдастырылуды айтады. Сіз 1-ші парақтан бастайсыз, ал одан әрі бірнеше парақтың біреуін ғана таңдайсыз. 2.3 схемада көрсетілген. Жүйесіз Web сайт – практика жүзінде ұйымдастырылымы жоқ сайт. Ол ақпаратты массивтен тұрады. Ол әр түрлі жүйесіз сілтемелермен біріккен. Сіз әр түрлі парақтарға барып одан қайту жолы өте қиынға соғатын сайттың түрі болып есептеледі. Іздеу жүйесі. Іздеу жүйесі дегеніміз- кілттік сөздерді пайдалану арқылы өзіңізге қызықты, қажетті ақпараттарды табу, іздеу WEBсайтын айтады. Іздеу жүйесінің көптеген түрлері бар, оны көбінесе 4 компонент арқылы ақпаратты іздеу және ұйымдастыру кеңінен қолданылады.

Гипермәтіндік құжаттарды құру технологиялары.

Web беттерді динамикалық қалыптастырудың мүмкіндігін жүзеге асыру үшін серверге мынадай бағыт берілуі керек, қандай файлдар “кәдімгі” болады, ал қандайы оның программалық өңделуіне нұсқаулары болады. Бұл жағдайда бетті сервердің өзі құрастырады (арнайы командалар көмегімен немесе ішкі бағдарлама). Беттердің динамикалық қалыптастыратын командасы болатын программа мәтіні скрип деп аталады. Тұтынушы компьютерінде қалыптастыру. Бұл жағдайда программа мәтіні динамикалық web- парағын қалыптастыру үшін алдымен

тұтынушының локальдық компьютеріне беріледі, бұнда браузер web- парағын алу және өңдеу үшін оған сәкес заттар шақыру керек. Динамикалық беттерді қалыптастыру үшін бірнеше технологиялар бар. Web-парақтарды кэштау. Егер сервер үлкен жойылуда болса, Интернетке шығу жылдамдығы төмен болса немесе қайта қосылу режимінде жұмыс істесе, көбнесе ақпарат кешіктіріліп беріледі.

Бұл кезде қосылуды жылдамдату сұрағы туындайды. Осы сұрақты шешудің бір жолы, мәліметтерді жойылған серверлерден бірнеше рет алуға болмауында. Web навигациясы кезінде көбінесе кітаптағындай “парақты артқа парақтау” сияқты, яғни параққа қайта қосылу қажеттілігі туындайды. Браузерде Артқа пернесін басып, алдында көріп өткен параққа қайта оралғанда, оны серверден қайта алудың мағынасы жоқ екені анық. Сондықтан көріп өткен құжатты компьютердің дискінің арнайы буферлік аймағында (кэште) сақтаса, қолданушы ссылжамен қайта оралғысы келген жағдайда тез ашып көруге мүмкіндігі болады.

Қазіргі заманға сай браузерлер дәл осылай істейді. Мәселен сіз парақтарды 1-2-3 кезегімен қарап, 3-ші парақ 4-ге ссылжасы бар деп ойлайық. Осы жағдайда 3-ші парақтан 4-ге өтумен (Жүйеден жүктеу) салыстырғанда 2-ге өту тез (кэштан жүктеу) орындалады.

Қазіргі заманға сай браузерлерде (Internet Explorer және Netscape Navigator) кэштелген құжаттарды сақтау үшін папканың көлемін өзгерту мүмкіндігі бар. Прокси-сервер. Кэштеу технологиясында өзінің шектеулері бар. Мысалы, сіз және сіздің әріптесіңіз Интернетке бір провайдер арқылы қосылған болсаңыз. Бір кезде сіз, алдында сіздің әріптесіңіз алған құжатты сол провайдердің каналынан сұрайсыз. Жүйеден қайта-қайта бір құжатты сұрау дұрыс емес. Мәліметтерді сервер провайдер деңгейінде кэштау дұрыс. Бұл мәселені шешу провайдерлер прокси-серверге (ағылшынның проху-орынбасар деген сөзінен) жүктейді.

Прокси-сервер – бұл қауіпсіздікті, әкімшілік бақылауды және кэштау функцияларын жоғарылатуға мүмкіндік беретін, қолданушының жұмыс станциясы мен Интернет арасындағы посредник ролін атқаратын сервер. Прокси-сервер қолданушыдан қандай да бір интернет-сервисті орындауға сұраныс алады, мысалы, web-парақты көріп өтуге сұраныс. Егер прокси кэштау функциясын орындаса, ол сервердің локальді кэшінде сұрап отырған парақтың жоқтығын қарайды. Егер бұл парақ бар болса, ол қолданушыға сұраныс Интернетке берілмей қайтарылады. Ал, егер кэшта бұл парақ болмаса, прокси-сервер клиент ролінде қолданушының атынан, өзінің IP-адресін қолданып, Интернеттен керек параққа сұраныс жібереді. Парақ қайта оралғанда, прокси-сервер оны қолданушыға жібереді.

Прокси-сервердің кэштау функциясы Интернетпен жұмысты едәуір жылдамдата алады және жүйелік трафиктің көлемін азайта алады. Web-парақтарды кэштеу Жүйенің жұмысын жақсартудың бір жолы болып табылады, себебі ол web-түйіннің уақытты бұзылыстарын компенсирует. Әдетте, прокси-сервер программасы жұмыс істейтін компьютерде едәуір дисктік аула бөлінеді. Үлкен провайдерлар құжаттарды кэштеуге ондаған және жүздеген гигабайт память бөле алады. Осылайша, үнемі ізделіп отыратын құжат тек құжатты шығарушының серверінде ғана сақталып қоймай, сонымен қатар “қолданушыға жақынырақ”, провайдер прокси-серверінде немесе қолданушының өз компьютерінде сақтала алатындықтан, іздеген құжаттарды тез көре алу мүмкіндігі бар. Біз мындаған клиенттері бар провайдерлар жөнінде айтқана, жүйедегі құжаттардың белгілі бір бөлігін ғана сұрайтынын айтып өткен жөн. Мындаған клиенттермен жұмыс істейтін үлкен ISP-дің прокси-сервері, сұраныстардың 50%-ға жуығын өз дискінен орындайды. Әдетте ISP сервері регионның басқа интернет провайдерлардың серверлерімен байланысты. Сондықтан, егер клиентті қызықтырып отырған құжат провайдердің прокси-сервер кэшінде табылмаса, ол басқа прокси-серверлерде табылуы мүмкін. Осылайша, кэш-

ауласындағы қолданушылардың көлемі көп есе көбейеді. Кэштейтін прокси-серверлерді тек ISP-провайдерлер ғана емес, сонымен қатар жүйеге деген нағрузканы азайтқысы келетін және сұраныстарды оптимально обрабатывать үшін ірі компаниялар орната алады. Web-тен іздеу - Интернетте миллиондаған сайттар бар, соның ішінде өзекті ақпаратпен қоса көптеген ескі қорлар орналыстырылған.

Интернет – белгілі бір басқарушысы жоқ демократиялық ақпарат көзі болып табылады. Кез келген адам желіге өзінің қорын орналастыра алады. Қорытындылап келгенде, интернетте ақпараттың қайталанбауына, оның стандартқа сай келуіне көп адамдар мән бере бермейді. Желіде барлығы бар екені белгілі, бірақ желіден қажетті ақпаратты алу қиын. Яғни, мәліметті табу үшін, оны жақсы іздей білу керек. Осы бөлімде интернет желісімен жұмыс істейтін іздеу аспаптары сипатталып, іздеу жүйесінің жұмыс механизмі түсіндірілген, іздеу оптимизациясына практикалық түсініктеме берілген. Веб сервер (ағылшын. Web server) веб контентке (мәліметтер қоры, құжаттар т.б) интернет арқылы қол жеткізуге мүмкіндік беретін компьютер немесе бағдарламалық жасақтама. Веб серверлер веб сайттардың хостингінде, мәліметтерді сақтау және т.б. мақсаттарда кеңінен қолданылады.

Веб — сервер-клиенттерден, әдетте веб-браузерлерден HTTP-сұрауларды қабылдайтын және HTTP-жауаптарды HTML-бет, сурет, файл, медиа-ағын немесе басқа да деректермен бірге беретін сервер. Веб-сервер веб-сервер функцияларын орындайтын бағдарламалық жасақтама ретінде, сол сияқты тікелей компьютер (осы бағдарламалық жасақтама жұмыс істейтін Сервер (аппараттық жасақтаманы) қараңыз) деп аталады. Әдетте веб-браузер болып табылатын Клиент веб-серверге URL-адресімен белгіленген ресурстарды алуға сұрау жібереді.

Ресурстар-Бұл HTML-беттер, суреттер, файлдар, медиа ағындары немесе клиентке қажетті басқа да деректер. Жауап ретінде веб-сервер клиентке сұралған деректерді береді. Бұл алмасу HTTP протоколы бойынша жүреді. Бұл үшін HTML тілінде арнайы дизайн-пішіндер бар. Нысандар пайдаланушыдан ақпарат алуға арналған. Мысалы, сайттың қай беттеріне жол беруге болатынын анықтау үшін пайдаланушының логині мен паролін білу қажет. Немесе сізге байланысуға мүмкіндік болу үшін пайдаланушының жеке деректері қажет. Нысандар осындай ақпаратты енгізу үшін қолданылады. Оларға мәтінді енгізуге немесе тізімнен қажетті опцияларды таңдауға болады. Пішінге жазылған деректер серверде арнайы бағдарламаны (мысалы, PHP скриптасы) өңдеу үшін жіберіледі. Пайдаланушы енгізген деректерге байланысты бұл бағдарлама әр түрлі web-беттерді қалыптастырып, деректер қорына сұраныстарды жіберуі, әр түрлі қосымшалар мен т. б. іске қосуы мүмкін.

Динамикалық бағдарламалау тілдері.

Динамикалық тіл - мәлімет типтерін анықтауға, синтаксистік анализ жасауға және инструкцияларды (кодты) қолма-қол компиляция жасауға мүмкіндік береді. Динамикалық бағдарламалау тілдері жасақтамаларды (приложение) тез-арада дамытуға мүмкіндік береді.

Динамикалық бағдарламалау тілдеріне: Perl, Tcl, Python, PHP, Ruby, Smalltalk, JavaScript жатады.

Visual Basic кейбір Динамикалық бағдарламалау тілдері сапаларына ие.

Бағдарламалау тілі (орыс. *язык программирования*; гр. *programma* - нұсқау, хабарлау) — хабарларды есептеуіш машиналардың көмегімен сұрыптайтын жасанды тілдер тобы.

Бағдарламалық тілдерде символдарды бірізділікпен өңдеуге қажет мәтіндер алдын ала жазылып қояды, мәтіндерді бұйрықты тапсырма деп қарауға болады. Семантикалық шарттары бойынша мәтіндер мынадай деңгейлерде сұрыпталады: дербес мағынасы жоқ әліпби таңбалары; шағын тіл бірліктерінің екі жақты атаулары; мағынасы бұйрық түрінде емес атаулардың қосындысын білдіретін сөйлемшелер; белгілі әрекетті білдіретін (бұйрық мағынасында) синтаксистік құрылымдардан тұратын операторлар тағы басқа. Бағдарламалық тілдердің өзіне тән сипаты — символдық, сөздік, сөйлемдік, мәтіндік деңгейлердің ара жігі белгісіздігінде мәтіндерді жазып, ЭЕМ-геенгізуде ондағы жетімсіздік басқа символдармен толықтырылады. Мысалы, (СОТО сөйлемшесі бір символ, бір сөз немесе бір сөйлем болып есептелуі мүмкін. Бағдарламалық тілдер қолданылу тұрғысынан екіге бөлінеді: алгоритмдік әр түрлі типтерді алдын ала жазуға болатын универсалдық бағдарламалық тілдер және неғұрлым шағын мақсатты жұмыстарға (модель жасау, мәтінді автомат өңдеу, белгілі бір процесті басқару) арналған мамандандырылған бағдарламалық тілдер. Бұлардың ішінде ЭЕМ-мен пайдаланушының қарым-қатынасын дамытуда диалогтік бағдарламалық тілдер маңызды орын алады. Бағдарламалық тілдердің дамуы олардың деңгейін көтеру, адам мен машинаның қарым-қатынас процесін жеңілдету, бағдарламалардың тиімділігін арттыру арқылы іске асырылып отырады. 20-ғасыр 70-жылдары бағдарламалық тілдердің жалпы саны 1 мыңға жақындады. Ең жиі қолданылатын, деңгейі жоғары тілдік бағдарламалар: фортран, кобол, ПЛ/1, алгол, бейсик, паскаль, мектепте кең тараған — паскаль тілі.

БАҒДАРЛАМАЛАУ ТІЛІ, бағдарламалау тілі — мәліметті (деректі, ақпаратты) және олардың алгоритмін (бағдарламасын) ЭЕМ-де өңдеуге арналған формальды (жасанды) тіл. Бағдарламалау тілінің негізін алгоритмдік тіл құрайды. Алғашқы Бағдарламалау тілі ішкі машиналық тіл болып есептеледі. Қазіргі Бағдарламалау тілі машиналық-бағдарланған, процедуралық-бағдарланған және проблемалық-бағдарланған тілдер болып ажыратылады. Машиналық-бағдарланған Бағдарламалау тілі өзінің мәліметтерінің түрі мен алгоритмінің құрылуы бойынша белгілі бір ЭЕМ-нің (не ЭЕМ класының) құрылымын бейнелейді. Мұның үстіне оның бағдарламалау процесін жеңілдету және автоматтандыру сияқты ерекшеліктері бар. Ол машиналық тілге жақын тіл. Машиналық-бағдарланған бағдарламалау тіліне автокод, алмо, эпсилон, т.б. тілдер жатады.

1 Синтаксис

Бағдарламалау тілінің сыртқы формасы синтаксис ретінде белгілі. Бағдарламалау тілдерінің басым көпшілігі текстік.

2 Процедуралық-бағдарланған бағдарламалау тілі

Процедуралық-бағдарланған бағдарламалау тілі — белгілі бір мәселелер класын шешу бағдарламасын сипаттауға арналған тіл. Бұл тілде мамандар кәсіптік терминдер түрінде ЭЕМ үшін тапсырмалар құрастыра алады, әрі онда мәселені шешу үшін орындалатын нақты қадамдар міндетті түрде көрсетіледі. Мұндай тілге фортран, Паскаль, алгол, ПЛ1, кобол, бейсик, дельфи, СИ, т.б. жатады. Мұның ішінде фортран ғылыми-техникалық есептерді шығару үшін, Паскаль өндіріс пен оқуда, бейсик дербес компьютерде жұмыс істеу үшін, т.б. әр түрлі мақсатта қолданылады.

3 Проблемалық-бағдарланған бағдарламалау тілі

Проблемалық-бағдарланған бағдарламалау тілінде функциялық терминдер ЭЕМ-ге орындалу реті көрсетілмей-ақ беріледі. Мұндай тілге әр түрлі қолданбалы бағдарламалар тобын (пакетін), ақпарат-іздеу жүйесінің сұраныстарын жатқызуға болады. Бағдарламаны сипаттаудың толықтығына қарай Бағдарламалау тілі төменгі және жоғарғы дәрежелі Бағдарламалау тілі болып бөлінеді. Төменгі дәрежедегі

Бағдармалау тіліне нұсқау беру кезіндегі қадамдардың толықтығы тән. Мұндай тілдер әдетте машиналық тілге жақын болып келеді. Бұл ЭЕМ-ге берілетін тапсырмадағы кәсіптік қызметте пайдаланылатын терминдердің қысқа әрі сиымды берілуіне мүмкіндік береді. Қазақстанда Паскаль, дельфи, СИ, ассамблер бағдарламау тілінде көптеген жаңа бағдарламалар жазылды. Олар республикада кеңінен пайдаланылып, шет елдерге де сатылды

Сервер тұжырымдамасы.

Интернеттегі миллиондаған өзара байланысты компьютерлерден немесе виртуалды VPS серверлерінен тұрады. Дегенмен, бірыңғай веб-сервердің әрқайсысының идеяларынан қалай көрінетіні және жұмыс істейтіні осында, сондықтан бұл туралы бірнеше сөз айту пайдалы болар еді.

Жалпы айтқанда, веб-сервер осы кешеннің аппараттық және бағдарламалық қамтамасыз ету бөлігі деп аталады. Сервер техникалық әдеттегі компьютер болуы мүмкін, бірақ бұл жағдайда ол мағынасы аз. Іс жүзінде: оның міндеті - мыңдаған компьютерлерге қызмет көрсету, сондықтан ол, кем дегенде, кіре берісте кіретін мекен-жайлардың аппараттық бөлігін бөліп шығаруды бағдарламалық қамсыздандыруға бөлетін пулды енгізу қажет. Бұл әртүрлі компаниялар мен жеке тұлғалардың сайттары сол физикалық дискке сақталатын конфигурацияларды іске асыруға мүмкіндік береді. Олардың барлығы бір-бірінен тәуелсіз, бірақ олар бірдей IP мекенжайын пайдаланады. Сонымен, мысалы, Narod.ru техникалық ресурстар шектеулері бар көптеген хостингтермен жұмыс істейді. Әрбір сайттың астында, мысалы, 1 Гбайт дискілік кеңістікті және процессор жүктемесінің 1-2% артық емес бөлігін бөліңіз. Егер бұл квота көбіне асып кетсе, онда сайт иелері жұмсақ болған жағдайда қымбат хостингке көшу кезінде нақтыланады. Бірақ мұндай шектеулі ресурстар тіпті, бұл виртуалды хосттарда қуатты аппараттық базасы бар, мысалы, 32 гигабайтқа дейін немесе үлкен дискілік массивке дейін. Сервер - бұл «көп жады болмаса». Және, әрине, осы жақсы жұмыс Pentium II процессорынан алыс - 8-16 физикалық ядросы артық болмайды. Осындай күрделі мазмұнды басқаратын OS Linux, Unix немесе Windows жүйесінің серверлік нұсқасы болып табылады, бірақ соңғы өте сирек. Әрбір сайт үшін клиенттермен диалогты жүзеге асыратын, сондай-ақ сервер тілдерінің аудармашыларын және PHP, Java, SSI және т.б. технологияларды қолданады.

Пайдаланушы үшін бағдарламалық жасақтама бөлігі қарапайым, себебі ол браузер технологиясына негізделген. Браузер URL-мекенжайлар арқылы сұрауларды қабылдайды, шертіп, пішіндерді жібереді және POST немесе GET әдістерімен серверге енгізеді, онда енгізілген деректер толтырылған айнымалы тізімге бөлінеді. Егер деректер дұрыс болса, олар сценарийлер жиынтығы немесе тұрақты беттер болып табылатын сайтқа жіберіледі. Соңғы браузерге өзгеріссіз нысанда жіберуге болады, ал динамикалық сайттар үшін деректер бірінші рет өңделеді және өңделеді. Басқаша айтқанда, әр парақ, қордағыдай, дерекқорда жинақталмаған және сұрау бойынша әр уақытта жиналады. Сонымен қатар PHP түріндегі конфигурациялық сценарийлер орындалады және дайын құжат пайдаланушыға беріледі. Беттерді жасау үрдісінде құралдар жұмыс жасауды жеделдету және беттерді онтайландыру, дерекқор өрістерін кәштау және құжаттарды қалыптастырудың бассейндерін пайдалану, есептеу үшін SSI бағдарламаларын пайдалану және тағы басқалар үшін белсенді қолданылады.

Сондай-ақ, жергілікті компьютерде веб-сервердің үлгісін, мысалы, Apache сервері, MySQL DBMS және сервер тілін аудармашыларды орнату арқылы жүзеге асыра алатыныңызды ескеріңіз. Ал, Apache кеңейтімінің әдістерін жақсы білетін болсаңыз, тіпті сценарийлік бағдарламалау тілін де жасай аласыз.

"Веб-сервер" ұғымы серверге де, физикалық сақтау орнына да, Бағдарламалық жасақтамаға да қатысты болуы мүмкін. Темір тұрғысынан, веб-сервер-бұл сайт ресурстарын сақтайтын компьютер (HTML құжаттар, CSS стильдер, JavaScript файлдар және т.б.) және оларды соңғы пайдаланушының құрылғысына жеткізетін компьютер (веб-браузер және т. б.). Әдетте Интернет желісіне қосылған және домендік атау арқылы қол жетімді болуы мүмкін, мысалы mozilla.org ескерту. БҚ тұрғысынан, веб-сервер серверде орналастырылған файлдарға пайдаланушылардың кіруін бақылайтын кейбір нәрселерді қамтиды, бұл HTTP серверінің минимумы. HTTP сервері-URL (веб-мекенжай) және HTTP (веб-станицаларды көру үшін браузерді пайдаланатын хаттама) шифрын ашатын БҚ бөлігі. Қарапайым сөзбен айтқанда, браузерге веб-серверде орналастырылған файл қажет болғанда, браузер оны HTTP арқылы сұрайды. Сұрау Қажетті веб-серверге (темір) жеткен кезде HTTP сервері сұрау салынған құжатты кері, сондай-ақ HTTP арқылы жібереді.

Веб-сайтты жариялау үшін статикалық немесе динамикалық веб-сервер қажет. Статикалық веб-сервер немесе стек, HTTP сервері бар компьютерден (темір) тұрады. Слэнгте бұл "статика" деп аталады, өйткені сервер онда орналастырылған файлдарды браузерге "бар" жібереді. Динамикалық веб-сервер статикалық веб-серверден және қосымша бағдарламалық жасақтамадан тұрады. Біз оны "динамикалық" деп атаймыз, себебі бағдарлама сервері HTTP арқылы браузерге жіберер алдында бастапқы файлдарды өзгертеді.

Мысалы, шолғышта көретін қорытынды бетті алу үшін, Бағдарлама сервері HTML үлгіні деректер базасынан толтыра алады. MDN (Mozilla Developer Network) немесе Уикипедия сияқты сайттар мындаған веб-беттен тұрады, бірақ олар нақты HTML құжаттары емес, тек бірнеше HTML үлгілері мен Үлкен Деректер Базасы. Бұл құрылым веб-қосымшаларды және мазмұнды жеткізуді жеңілдетеді және жеделдетеді.

Веб-серверлердің негізгі функциялары

Бастапқыда веб гипермәтіндік құжаттарды тарату үшін жасалды [3]. Уақыт өте келе, функционалдық бөлік өсті, веб-мүмкіндіктері көп болды. Енді веб-сайт ондаған функционалдылыққа ие, ең алдымен, веб-сервермен қамтамасыз етіледі. Веб-серверлердің негізгі функциялары:

- HTTP арқылы веб-браузерлерден ТРС / IP желілік хаттамалары арқылы сұраулар алу;

HTTP протоколы арқылы файлдарды гипермәтінге және / немесе кез келген құжаттарды веб-шолғышқа іздеу және жіберу;

- Қызмет көрсету және сұратуды өңдеу, мысалы: mailto, telnet, ftp және басқалары;

CGI интерфейсі арқылы өңдеу параметрлерін беру және қайтару арқылы веб-серверде бағдарламаларды іске қосу;

- навигацияның имидждік карталарын жасау және қызмет көрсету (сурет картасын);

Java-қосымшаларды жүктеу;

- Серверді басқару және операциялық басқару;

Пайдаланушыларды авторландыру және олардың аутентификациясы;

- әртүрлі ресурстарға қол жеткізуді қамтамасыз ету журналын жүргізу;

Динамикалық беттерге қолдау көрсету;

- Пайдаланушылармен сенімді байланыстар үшін қауіпсіз HTTPS протоколына қолдау көрсету.

Веб-браузерлер мен веб-серверлер HTTP арқылы байланысады, бұл қарапайым сұрау және жауап беру хаттамасы, TCP / IP арқылы ақпаратты жіберу үшін. Ол келесідей жұмыс істейді: веб-сервер сұрауды алады, файлды табады, браузерге жібереді, содан кейін қосылымнан ажыратады. Содан кейін веб-шолғыш жүктелген HTML құжатын пайдаланушының экранында көрсетеді.

Веб-сервер кез-келген деректерді сақтай алады, бұл мәтін, суреттер, бейне және т.б. болуы мүмкін. Жалпы алғанда, іздеу процессорлары және дерекқордың байланыс құралдары сияқты қолданбалы бағдарламалар веб-серверде жұмыс істей алады. Оларды жасау үшін стандарттарды әдеттегі шлюз интерфейсі (CGI - CommonGatewayInterface), сценарийлік тілдер (мысалы, JavaScript), сондай-ақ толыққанды бағдарламалау тілдері (Java, Basic және т.б.) пайдаланылады.

Веб-Сервердің мақсаты

Веб-Сервердің мақсаты қарапайым - hardware тиімді пайдалана отырып, бір уақытта көптеген клиенттерге қызмет көрсету.

Веб Сервердің басты міндеті пайдаланушылардың HTTP сұрауларын қабылдау, оларды өңдеу, сандық компьютерлік кодқа аудару. Содан кейін HTTP-жауаптар беру, оларды миллион нөлдер мен бірліктерден бейнелерге, медиа ағындарға, әріптерге, HTML беттерге түрлендіру.

Кез келген веб-сервердің пайдаланушылары оны пайдалану ыңғайлылығы үшін ыңғайлы веб-браузер болуы тиіс. Ол веб-серверге интернет - ресурстардың URL-мекенжайларына түрлендірілген сұрауларды береді.

Стандартты мүмкіндіктермен қатар, кейбір веб серверлер қосымша. Мысалы, тиісті бағдарламалық қамтамасыз ету қандай да бір ресурсқа пайдаланушылардың өтініштерінің санын тіркеп, оларды жеке журналға жаза алады. Сонымен қатар, олар HTTPS-ді қолдай алады, бұл сайттар мен пайдаланушылар арасында қорғалған байланыс үшін маңызды емес. Жиі веб-сервер мейл-сервермен бірге орнатылады. Бұл пайдаланушыларға пошта бетіне тек бір гиперсілтемеге басу арқылы тікелей сайттан тез өтуге мүмкіндік береді.

Серверді орнату

Windows ОЖ басқарылатын компьютерге Прокси-серверді орнату үшін:

1.Дистрибутив файлын іске қосыңыз. Сізге орнатылып жатқан өнім жайлы ақпарат көрсетін **InstallShield Wizard** терезесі ашылады. **Next** батырмасын басыңыз.

2.Лицензиялық келісімнің мәтіні бар терезе ашылады. Лицензиялық келісімнің шарттарымен танысқаннан кейін, **I accept the terms of the license agreement** пунктін таңдап, **Next** батырмасын басыңыз.

3.Прокси-сервердің негізгі параметрлерін күйге келтіруге арналған терезе ашылады:

•**Listen to** өрісінде Прокси-сервермен "тыңдалатын" IP-адресі беріңіз. Үндеместік бойынша any (0.0.0.0) – барлық интерфейстерді "тыңдайды".

Адрестер **Қосымшалар** документінде, **E Қосымшасы. Желілік адресстер спецификациясы** көрсетілген, желілік адресстер форматында беріледі.

•**Port** өрісінде Прокси-серверді "тыңдайтын" порттың нөмірін көрсетіңіз. Үндеместік бойынша – ол 2193 порты.

•Серверді имитация жасау режимін қосу үшін **Enable discovery** жалаушасын орнатыңыз. Бұл режим Желі сканеріне Прокси-серверді Dr.Web Сервері ретінде анықтауға мүмкіндік береді. Серверді имитация жасау режиміне келесідей күйге келтірулер қолжетімді:

▫Прокси-сервер Серверге бағытталған кең таратулардың барлығына жауап беру үшін **Enable multicasting** жалаушасын орнатыңыз.

▫**Multicast group** өрісінде, құрамына Прокси-сервер кіретін көпадресті тобтың IP-адресін көрсетіңіз. Көрсетілген интерфейс, желіде активті Dr.Web Серверлерін іздестіру кезінде, Желілік инсталляторламен қарымқатынас жасау үшін Прокси-сервермен тыңдалатын болады. Егер өрісті бос қалдыратын болса, Прокси-сервер көпадресті тобтардың ешқайсысына кірмейтін болады. Үндеместік бойынша құрамына Сервер - 231.0.0.1 кіретін көпадресті тоб.

•**Compression mode** құлама тізімінде Прокси-сервер мен қызмет көрсетілетін клиенттердің арасындағы канал үшін сығу режимін таңдаңыз: Агенттер мен Агент инсталляторларының. **Level** өрісінде сығу деңгейін көрсетіңіз. 1 ден 9 -ге дейінгі аралықтағы бүтін сандарды көрсетіңіз.

Негізгі күйге келтірулерді бергеннен кейін **Next** батырмасын басыңыз.

4.Прокси-сервердің кэштауды күйге келтіру терезесі ашылады:

Прокси-сервермен жіберілетін деректер кэшталу үшін, **Enable caching** жалаушасын орнатып, келесідей параметрлерді беріңіз:

•Үндеместік бойынша орнатылған кэшталған ақпараттарды сақтау каталогін өзгерту үшін, **Browse** батырмасын басып, файлдық жүйе бойынша браузерде жаңа каталогті көрсетіңіз.

•**Maximum revisions number** өрісінде сақталатын ревизиялардың максималды санын көрсетіңіз. Үндеместік бойынша соңғы 3 ревизия сақталады, ал одан ескі ревизиялар жойылады.

•**Cleanup interval** өрісінде ескі ревизияларды жою аралығының минуттар бойынша уақыттық интервалын көрсетіңіз. Үндеместік бойынша - 60 минут.

•**Unload interval** өрісінде қолданылмайтын файлдарды жадыдан шығарып тастау аралығының минуттар бойынша уақыттық интервалын көрсетіңіз. Үндеместік бойынша - 10 минут.

•**Integrity check mode** құлама тізімінде кэш бүтіндігін тексеру режимін таңдаңыз:

▫**At startup** - Прокси-серверді іске қосқан уақытта (ұзақ уақыт алуы мүмкін).

▫**Idle** - Прокси-сервердің фондық режимде жұмыс істеуі кезінде.

Кэштау күйге келтірулерін бергеннен кейін **Next** батырмасын басыңыз.

5.Қосылуларды қайта бағыттауды күйге келтіру терезесі ашылады:

Redirection settings блокында Прокси-сервермен орнатылып жатқан қосылаулар, қайта бағытталаатын Dr.Web Серверінің адресін немесе адресстер тізімін көрсетіңіз.

Адресстер **Қосымшалар** документінде, Е Қосымшасы. Желілік адресстер спецификациясына көрсетілген, желілік адресстер форматында беріледі.

Compression mode құлама тізімінде Прокси-сервер мен берілген Dr.Web Серверлернің әрбірінің арасындағы канал үшін трафикті сығу режимдерін таңдаңыз.

Қайта бағыттау күйге келтірулерін бергеннен кейін **Next** батырмасын басыңыз.

6.Орнату каталогін таңдау терезесі ашылады. Егер үндеместік бойынша орнатылған каталогті өзгерту қажет болса, **Change** батырмасын басып, қажетті каталогті таңдаңыз.

Next батырмасын басыңыз.

7.Прокси-серверді орнатуға дайын екендігін хабарлайтын терезе ашылады. Прокси-серверді орнатуды бастау үшін **Install** батырмасын басыңыз.

8.Орнату процессі аяқтағаннан кейін **Finish** батырмасын басыңыз.

Орнатуды аяқтағаннан кейін, қажет болған жағдайда, сіз Прокси-сервердің жұмыс істеу параметрлерін өзгерте аласыз. Ол үшін келесідей орнатылу каталогінде орналасқан, drwcsd-proxy.xml конфигурациялық файлы қызмет етеді:

•Windows ОЖ: C:\ProgramData\Doctor Web\drwcsd-proxy\

•Linux ОЖ және Solaris ОЖ: /var/opt/drwcs/etc

•FreeBSD ОЖ: /var/drwcs/etc

Конфигурациялық файлдың күйге келтірулері **Қосымшалар** құжатында, п. G4 Қосымшасы берілген.

UNIX жаңұясының ОЖ басқарылатын компьютерге Прокси-серверді орнату үшін:

Келесі команданы орындаңыз:

```
sh ./<дистрибутив_файлы>.run
```

БҚ **FreeBSD** ОЖ басқарылатын компьютерге орнату процессінде /usr/local/etc/rc.d/0.dwcp-proxy.kұрылады.

Келесі командаларды қолданыңыз:

- usr/local/etc/rc.d/0.dwcp-proxy.sh stop - Прокси-серверді қолмен токтату үшін;
- usr/local/etc/rc.d/0.dwcp-proxy.sh start - Прокси-серверді қолмен іске қосу үшін.

БҚ **Linux** ОЖ және **Solaris** ОЖ басқарылатын компьютерге орнату процессі барысында, Прокси-серверді іске қосу және токтатуға арналған init құрылатын болады init-скрипт құрылатын болады /etc/init.d/dwcp-proxy.

HTML –мен танысу. Синтаксис.

HTML (HyperTextMarkupLanguage) – бұл құжаттарды кодтау үшін қолданылатын гипертекстік белгілеу тілі. HTML ді көбі программалау тілі деп ойласа да, бұл программалау тілі емес. HTML – мәтінді белгілеу тілі.

Бұл тілде компьютердің программалық және ақпараттық қамтуларының айырмашылықтарын ойламайтындай құжат құруға болатындай жасалынған. HTML технологиясы кәдімгі мәтіндік құжатқа басқарушы символдар (тәги - tage) орналасқан құрылымнан тұрады. Тәгтер мәтіннің өлшемін, сызылымын, түсін, мәтіннің орналасуын, беттәгі графикалық бейнені мәтіндік құжатқа графикалық бейнені, жасанды қозғалысты, дыбыс, видео енгізу мүмкіндігін береді, сонымен қатар осындай Web-беттермен мағыналы байланыс орнатады.

HTML құжаттарды көру үшін браузерларды қолданамыз. Браузер-программалардың саны өте көп, мысалы көп таралғандар Netscape Communicator, Microsoft Internet Explorer, Opera. Web-бет браузерде HTML тәгі арқылы берілген түрде қарастырылады. Мұндай Web-бетті HTML-файл деп атаймыз. HTML-файлының кеңейтілуі .htm немесе .html.

HTML-файлдарының жетістігі: аз ақпараттық көлем; кез келген дербес компьютерден көру мүмкіндігі; интерактивтілігі.

HTML гипермәтіннің тіл белгілеу дамуы жалғасуда. Осылайша DHTML (Dynamic HTML), XML, VRML және т.б. тілдер пайда болды. Белгілеуі кеңейтілген XML (Extensible Markup Language) тілі HTML гипермәтін белгілеуінің кеңейтілуі болып табылады, сондай-ақ кеңейтілген басқарушы символдарды енгізу мүмкіндігі және де құжаттың дұрыстығын тексеру мүмкіндігі бар. Виртуалды шынайы тіл VRML (Virtual Reality Modeling Language) HTML гипермәтін тілінің белгілеуімен құрылған, ол көлемі 3D-графикті пайдалану мүмкіндігін береді.

Тәг - ол браузерге тәг ішінде орналасқан мәтіндегі бейнелеуді көрсететін басқарушы символ. Тәг "кіші" (<) белгісінен басталады және "үлкен" (>) белгісімен аяқталады (<tage>). Тәг жазылуында бас және кіші әріптердің айырмашылықтары болмайды. Тәгтер жұп және жұп емес болады. Жұп тәг тек ашылатын <tage> және слэш </tage> символымен жабылатын тәгтерді қамтиды. Жұп емес тәг тек ашылатын тәгті қамтиды. Жұп тәг ашылатын тәгтің басынан жабылатын тәгтің арасындағы мәтінге әсер етеді.

HTML - тәг атрибуттар тізімінен тұрады. Тәг тексті үшбұрышты жақша ішінде жазылады. Мысалы:

Менің бірінші бетім - бұл жерде - тәг, face – атрибут, arial – атрибуттың мәні.

Тэгтер: құрылымдық, абзацты форматтау, символдар, гипермәтінді анықтау, графиканы қою, тізім құру және сол сияқтылар болады.

Көрсетілген мысалдарды компьютерде көру үшін мәтінді Блокнот программасына теріп, оны htm түрінде сақтаңыз. Файлды браузерде ашыңыз.

HTML: жалпы құжат құрылымы.

HTML құжаттың құрылымы

HTML құжаттың негізгі құрылымы:

<HTML>

<HEAD>

<TITLE> құжат тақырыбы </TITLE>

</HEAD>

<BODY>

құжат тұлғасы

</BODY>

</HTML>

Кез келген HTML құжат <HTML> тәгімен басталып, </html> тәгімен аяқталу керек. Бұл тэгтер браузерға HTML құжатын көрсетеді. Құжат болса қарапайым ASCII кодындағы мәтіндік файл. <HTML>, </html> тэгтер болмаса браузер программалары құжатты танымай қалуы мүмкін.

<HTML>, </html> тэгтерінің ортасына құжат денесі келеді. Құжат екі бөлімнен тұрады: тақырып бөлімі <HEAD> тәгімен басталынатын және <BODY> тәгімен басталынатын негізгі бөлімнен.

■ Құжаттың тақырыбы

Тақырып бөлімі міндетті емес, бірақ ол жерде браузерге қажетті көп ақпарат болуы мүмкін. <TITLE>, </title> тэгтерінің арасында құжаттың аты жазылады, сол сөз терезенің тақырып жолына шығады. Әрбір HTML құжатта тақырып қатары болу керек. Бұл қатар құжатты теңестіреді және негізгі терезенің тақырыбында ғана емес, басқа жерлерде де навигатор арқылы бейнеленеді, мысалы, қатысқан беттердің тізімінде.

Құжаттың денесі

<BODY> тәгімен Web-беттің негізгі бөлімі басталады. Бұл бөлімге мәтін, графика, кесте, аудио және видео ақпараттар енгізіледі. HTML құжаттың денесі <BODY> және </BODY> операторларымен шектеліп, навигатордың ішкі аймағында бейнеленеді. Біздің қарапайым мысалда дене бір жолдан ғана тұрады:

<BODY>

WWW серверлар әлеміне қош келіңіздер!

</BODY>

<HEAD> бөліміне тағы <META> тэгтерді қолдануға болады, олар сайтты интернет желісінде тез табылу үшін пайдалынады.

<meta name="Language" content=" kz"> - сайт тілі

<meta name="Autor" content="Zhanar"> - сайт авторы

<meta name="Keywords" content="информатика, интернет, финансы, компьютер, университет, менеджмент"> - іздеу қызметіне арналған сөздер

<meta name="Generator" content="блокнот"> - қай программада жасалынды

Тапсырма – 1. Тек құрылымдық тэгтен тұратын бос Web-бет құру.

HTML\1.html

<HTML>

<HEAD>

```
<TITLE>Қазақстан Республикасының мемлекеттік символдары</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

HTML құжаттарын редактрлеу

Мәтіннің параграфтарын туралау

Абзац енгізу үшін <P>, </p> тәгтерін қолданады, мәтіннің сол жақ, ортада, оң жақта орнату үшін align атрибуты пайдалынады. Бұл атрибут келесі мәндерге ие болады:

Мәні	Орнату типі
LEFT	Сол шекара бойынша
CENTER	Орталандыру
RIGHT	Оң шекара бойынша

Тапсырма – 2. Мәтінді туралау

HTML\2.html

```
<HTML>
<HEAD>
<TITLE>XYZ Incorporation</TITLE>
</HEAD>
<BODY>
<P ALIGN=RIGHT> оң жақтан орнату
<P ALIGN=LEFT> Бұл мысал параграфтың сол жақ шекарада орнатылуы
<P ALIGN=CENTER>Мәтінді ортада бірінші тәсілмен орнату
<P><CENTER> Мәтінді ортада екінші тәсілмен орнату </CENTER>
</BODY>
</HTML>
```

size

Бұл элемент мәтіннің белгісінің биіктігін көрсетуге мүмкіндік береді. Қаріп өлшемі 1-ден 7- ге дейінгі шартты бірлікпен көрсетіледі және олкәдімгі стильдегі өлшемі 3 болатын қаріпке негізделген. SIZE атрибутын екі тәсілмен қолдануға болады: қаріптің абсолют өлшемін көрсету, мысалы SIZE=5, немесе салыстырмалы өлшемін көрсету

арқылы, мысалы SIZE=+2. Екінші тәсіл көбіне негізгі қаріп basefont көрсетілсе қолданылады. Төменде мысалда қаріптің абсолют өлшемді қоюкөрсетілген:

```
<font size=1> size=1 </font>
<font size=2> size=2 </font>
<font size=3> size=3 </font>
<font size=4> size=4 </font>
<font size=5> size=5 </font>
<font size=6> size=6 </font>
<fontsize=7>size=7 </font>
```

Бұл программа нәтижесі: size=1 size=2 size=3 size=4 size=5 size=6 size=7

Келесі мысал қаріптің негізгі қаріп 3 болғандағы салыстырмалы өлшем көрсетілген: size=-4 size=-3 size=-2 size=-1 size=+1 size=+2 size=+3 size=+4

Сол сияқты негізгі қаріп 6 ға тең болғанда:

size=-4 size=-3 size=-2 size=-1 size=+1 size=+2 size=+3 size=+4

қаріп өлшемін өзгерту мысалдары:

```
<HTML>
<HEAD>
```



```

<TITLE>қаріп өлшемін тандау мысалы</TITLE>
</HEAD>
<BODY>
<FONT SIZE=1>өлшем 1</FONT><BR>
<FONT SIZE=-1>өлшем 2</FONT><BR>
<FONT SIZE=3>өлшем 3</FONT><BR>
<FONT SIZE=4>өлшем 4</FONT><BR>
<FONT SIZE=+2>өлшем 5</FONT><BR>
<FONT SIZE=6>өлшем 6</FONT><BR>
<FONT SIZE=+4>өлшем 7</FONT><BR>
</BODY>
</HTML>

```

color

Мәтін түсін оның қаріп атын бергендей беруге болады. COLOR атрибуты RGB оналтылық мәнін немесе стандартты түс аттарын қабылдайды. Төменде BODY элементінде BGCOLOR атрибутында қолданылуына ұқсас түсті анықтау:

```

<HTML>
<HEAD>
<TITLE>түсті тандау мысалы</TITLE>
</HEAD>
<BODY>
<FONTCOLOR="#FF0000">бұл мәтін түсі қызыл</FONT><BR>
<FONTCOLOR="GREEN">бұл мәтін түсі жасыл</FONT><BR>
</BODY>
</HTML>

```

HTML мәтін фрагменттерін қаріптік белгілеуді екі түрлі жағынан қарастыруға болады. Бірінші жағынан, мәтіннің кейбір аймағындағы қаріп **қою**, *көлбеу* немесе асты сызылған болуы қажет деп тікелей нұсқауға болады, яғни мәтіннің **физикалық** стилін өзгерту. Екінші жағынан, мәтіндегі кейбір фрагментті басқа фрагменттерден өзгеше логикалық стилі бар деп белгілеуге болады, бұл стильдің түсіндірмесін браузерге қалдырады. Мысалдарда түсінік берейік.

Физикалық стильдер

Физикалық стиль браузерге ағымдағы қаріптің түрленуі туралы тікелей нұсқауды түсіндіреді. Мысалы, **** және **** белгілерінің арасында жазылған нәрсенің барлығы **қою қаріппен** жазылады.

<I> және **</I>** белгілерінің арасындағы мәтін *көлбеу қаріппен* жазылады.

<U> және **</U>** белгілерінің арасындағы мәтін *асты сызылған қаріппен* жазылады.

<TT> және **</TT>** қос белгілері негіз болып табылады. Осы белгілердің арасында орналасқан мәтін жазба машинасымен жазылып жатқандай қаріппен жазылады, яғни символдардың бекітілген ені болады.

Логикалық стильдер

Логикалық стильді автор қолдану барысында оқитын адамның экран бетінен не көретінін алдын ала біле алмайды. Логикалық стильдің таңбаларын түрлі браузерлер түрлі түрде түсіндіреді. Кейбір браузерлер кейбір таңбаларды жалпы ескермейді және логикалық стильмен белгіленген мәтінді жай мәтін ретінде көрсетеді. Ең көп таралған логикалық стильдер мыналар.

**** ... **** Ағылшын тілінен *emphasis* — *акцент*, яғни *курсив (көлбеу шрифт)* дегенді білдіреді.

**** ... **** Ағылшын тілінен *strong emphasis* — күшті *акцент*, яғни *жартылай қою*.

<CODE> ... </CODE> Нәтижедегі мәтіннің фрагменттері үшін қолдануға ұсынылады.

<SAMP> ... </SAMP> Ағылшын тілінен *sample* — үлгі. Программалар арқылышығарылатын хабарламалар үлгілерін көрсету үшін қолдануға ұсынылады.

<KBD> ... </KBD> Ағылшын тілінен *keyboard* — пернетақта. Пернетақтадан не енгізу

керек, соны көрсету үшін қолдануға ұсынылады.

<VAR> ... </VAR> Ағылшын тілінен *variable* — айнымалы. Айнымалылардың аттарын

жазу үшін қолдануға ұсынылады, сонымен қатар бұл қаріп көлбеу қаріпті еске түсіреді.

1 мысалы арқылы логикалық және физикалық стильдер туралы жинаған білімдерді қорытайық. Сонымен қатар, логикалық стильдердің түрлерін сіздің браузеріңіз қалай көрсететінін көруге болады.

HTML: тізімдер

HTML-де тізімдердің келесі түрлері бар:

- Таңбаланған
- Нөмірленген
- Көп деңгейлі тізімдер

Таңбаланған тізімдер қою үшін және тегтері пайдаланады. тегі бүкіл тізімнің басы мен соңын белгілейді. тегі жеке тізім элементінің басы мен соңын белгілейді.

Әдепкі бойынша тізім элементтері қара шеңбермен белгіленеді. *Type* атрибуты көмегімен таңбаның төлсипатын өзгертуге болады.

Таңбаланған тізімдерге ұқсас, және тегтерін пайдаланып нөмірленген тізімдер жасай аласыз. Нөмірлеу стилін таңдау үшін *type* атрибутын қолданамыз және тізім нөмірлеуін қай таңбадан бастауды көрсету үшін *start* қолданылады. тегі үшін *value* атрибуты ағымдағы тізім элементінің санын орнатуға мүмкіндік береді.

Анықтау тізімі - екі байланыс жиынтықтардан тұратын мәтіннен тұрады: терминдер тізімі мен терминдердің анықтамалары.

Анықтау тізімінің құрылымы

Термин 1

Термин анықтамасы 1

Термин 2

Термин анықтамасы 2

Тізім <dl> тегі арқылы беріледі, термин - <dt>, термин анықтамасы <dd>тегі арқылы беріледі.

Сайт дизайнына жалпы тұрғылар. Бет макетті әзірлеу .

Кез келген HTML-құжатында орналасатын ең бірінші тэг – бұл <HTML>... </HTML>. Бұл тэг арқылы біз құжатта HTML- мәтіні бар екендігін білеміз. Барлық жазылып отырған құжаттағы мәліметтер осы тэгтардың ішінде орналасуы қажет.

< HTML >

...

Құжат мәтіні

...

</HTML >

Сіздің құжатта шынымен HTML-командалары бар екендігін білдірдіңіз. Құжат негізі екі бөліктен: "тақырыбынан" және "құжат денесінен" тұрады.

Бұл 3 тэг стандарт бойынша қатысты қарастырылғанымен, кейбір браузерлер құжатты олардың көмегінсіз-ақ бейнелей алады. Бірақ бұл тэгтарды сілтеп кету керек, себебі келешекте тэгтар қажет болған жағдайда құжатты редакциялау қажет болады.

1-жаттығу. Қарапайым HTML файлдарын құрастыру.

1. Құрастырылған Web-парақты сақтау үшін KURS-бумасын ашыңыз.
2. Блокнот (Notepad) стандарт программасын жүктеңіз.
3. Қарапайым HTML-файлының мәтінін редактор терезесінде теріңіз.

<HTML >

<HEAD>

<TITLE> HTML оқу файлы </TITLE>

</HEAD>

<BODY>

Сейсенбіге арналған сабақ кестесі

</BODY>

</HTML >

4. RASP.HTM атымен файлды сақтаңыз.

5. Құрастырылған Web-парақты көру үшін Microsoft Internet Explorer браузерін жүктеңіз.

6. Файл (File), Открыть (Open), Просмотр (Обзор, Browse) командаларын орындап KURS бумасының ішінен RASP.HTM файлын тауып, оны жүктеңіз. Web-парақтың ("HTML оқу файлы") атауы браузердің жоғарғы статусы жолында бейнеленгенін көріп тексеріңіз.

HTML-құжатын бейнелеген кезде браузерлер автоматты түрде мәтінді экранда орналастырады. Ол файлда кездесетін символдар мен бос жол адымдарына мән бермейді.

Жолдың тасымалдауын орындайтын және жаңа абзацтың басталуын беретін арнайы командалар бар. Одан қалса браузер программасына форматтауды тыйым салатын және экранға дәлме-дәл мәтін файлының үзіндісін бейнелейтін командалар бар.

 - жолды тасымалдаушы тэг жолды келесі мәтіннен немесе графикадан бөліп тұрады. <P> тәгі де жолды бөледі, сонымен қатар абзацты айқындап тұратын бос жол қалдырады. Бұл 2 тэг бірэлементті болып табылады.

Мәтін үзінділерін ерекшелеу

Мәтін үзінділерін ерекшелеу тэгтері жеке символдар мен сөздердің бейнелеуін басқаруға мүмкіндік береді. Осы мақсатта үш түрлі тэгтар қолданылады:

... - **жартылай қалың қарамен ерекшелеу (полужирный)**

<I>...</I> - *курсивпен ерекшелеу*

<U>...</U>-астын сызу арқылы ерекшелеу

Құрама ерекшелеу қолданылуы мүмкін. Мысалы, <I> Сейсенбіге арналған </I> <I> сабақ </I> <U> кестесі </U> деп өзгерту енгізіңіз.

Бірақ бұл жағдайда құрама тэгтардың жазылу ережесін есте сақтау керек.

<1-тэг> <2-тэг>...</ 2-тэг> </ 1-тэг> - дұрыс жазу
<1-тэг> <2-тэг>...</1-тэг> </2-тэг> - қате жазу.

Символ өлшемдерінің берілуі

Браузер арқылы бейнеленген мәтінді басқарудың 2 тәсілі бар.

- Тақырып стилін қолдану;
- Негізгі құжат шрифтінің өлшемін беру немесе ағымдағы шрифтінің өлшемін беру;

Тақырып стилін қолдану. Мұнда б-тақырып тәгі қолданылады (<H1>-ден <H6>-ға дейінгі). Әрбір тәгке браузер параметрлерінің берілген нақты стили сәйкес келеді.

Ағымдағы шрифтінің өлшемін беру. шрифтік тәгі 1-ден 7-ге дейінгі диапазонда мәтіннің кез келген жерінде ағымдағы шрифтінің өлшемін беруге мүмкіндік береді.

Гарнитура мен шрифт түсі

 тәгі гарнитура пішіні мен мәтін түсін өзгерту мүмкіндіктерін береді. Шрифт гарнитурасын өзгерту қарапайым тәгіне FACE атрибутын қосу арқылы жүзеге асады. Мысалы, мәтінді Arial шрифтімен жазу үшін деп жазу керек.

Шрифт түсін өзгерту үшін тәгінде COLOR = "X" атрибутын қолдануға болады. X-тің орнына түстің атын (тырнақшада) қоямыз немесе 16-лық мәндегі сандар жиынтығын жазамыз. Түсті 16-лық сандармен беру үшін, бұл түсті әрқайсысы О-дан FF-қа дейінгі мәнге ие болатын, жіктелген үш құрамға бөліп көрстеуге болады: қызыл (R-Red), жасыл (G-Green), көк (B-Blue). Бұл жағдайда біз RGB деп аталатын форматпен жұмыс істейтін боламыз.

Мәтінді туралау

HTML-тәгтарының көмегімен мәтіннің туралауды басқаруға болады. Егер туралау тәсілі көрсетілмесе, құжаттағы барлық элементтер сол шеттен тураланып, оң жағы ретсіз орналасады.

Қазіргі браузерлер ALIGN = : мәтінді туралау атрибутын қолданады.

ALIGN = CENTER – центрі бойынша туралау

ALIGN = RIGHT- оң шеті бойынша туралау

ALIGN = LEFT – сол шеті бойынша туралау.

HTML: кестелер, фреймдер

Күрделі кестелермен жұмыс жасау

Тапсырма - 1. Тақырыбы бар кестелер

```
<HTML> <HEAD>  
<TITLE>Тақырыбы бар кестелер</TITLE>  
</HEAD>  
<BODY BGCOLOR="#FFFFFF">  
<TABLE BORDER>  
<TR>  
<TH>Тақырып 1</TH>  
<TD>000</TD><TD>001</TD><TD>002</TD>  
</TR>  
<TR>  
<TH>Тақырып 2</TH>  
<TD>010</TD><TD>011</TD><TD>012</TD>  
</TR>  
<TR>
```

```

<TH>Тақырып 3</TH>
<TD>100</TD><TD>101</TD><TD>102</TD>
</TR>
</TABLE> </BODY></HTML>

```

Кесте ұяшықтарын біріктіру

Бір бағанада орналасқан кесте ұяшықтарын біріктіру үшін COLSPAN параметрін қолдануға болады. Сол сияқты бір жолда орналасқан кесте ұяшықтарын біріктіру үшін ROWSPAN параметрі қолданылады.

Төмендегі мысалда келтірілген HTML құжатының бастапқы мәтіні кестенің екі бағанасына ортақ тақырып құру үшін ұяшықтарды біріктіруді қалай көрсетуге болатынын көруге болады.

Тапсырма - 2. Кесте ұяшықтарын біріктіру

```

<HTML> <HEAD>
<TITLE>Кесте ұяшықтарын біріктіру</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<TABLE BORDER>
<TR>
<TH>Ұяшық 1</TH>
<TH COLSPAN=2>Ұяшық 2 және 3</TH>
</TR>
<TR><TD>000</TD><TD>001</TD><TD>002</TD></TR>
<TR><TD>010</TD><TD>011</TD><TD>012</TD></TR>
<TR><TD>100</TD><TD>101</TD><TD>102</TD></TR>
</TABLE> </BODY> </HTML>

```

Төменде келтірілген HTML құжатының бастапқы мәтіні кестенің екі жолына ортақ тақырып құру үшін ұяшықтарды біріктіруді қалай көрсетуге болатынын көруге болады.

Тапсырма - 3. Кесте ұяшықтарын біріктіру

```

<HTML> <HEAD>
<TITLE> Кесте ұяшықтарын біріктіру </TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<TABLE BORDER>
<TR>
<TH ROWSPAN=2>Ұяшық 1 және 2</TH>
<TD>000</TD><TD>001</TD><TD>002</TD>
</TR>
<TR>
<TD>010</TD><TD>011</TD><TD>012</TD>
</TR>
<TR>
<TH>Ұяшық 3</TH>
<TD>100</TD><TD>101</TD><TD>102</TD>
</TR>
</TABLE> </BODY></HTML>

```

Көпқабатты тақырыптар
Кестенің жолдары мен бағаналарында бірнеше тақырыптары болу мүмкін.
HTML құжатының бастапқы мәтінде соған мысал келтірілген.

Тапсырма - 4. Көпқабатты тақырыптар

```
<HTML> <HEAD>
<TITLE>Көпқабатты тақырыптар</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<TABLE BORDER>
  <TR>
    <TH COLSPAN=3>негізгі ұяшық </TH>
  </TR>
  <TR>
    <TH COLSPAN=2>Ұяшық 1 және 2</TH>
    <TH>Ұяшық 3</TH>
  </TR>
  <TR>
    <TD>000</TD><TD>001</TD><TD>002</TD>
  </TR>
  <TR>
    <TD>010</TD><TD>011</TD><TD>012</TD>
  </TR>
  <TR>
    <TD>100</TD><TD>101</TD><TD>102</TD>
  </TR>
</TABLE> </BODY> </HTML>
```

Қолтаңбасы бар кестелер

Кестеде қолтаңба болу үшін <CAPTION> операторын қолдану керек. Төмендегі мысалда HTML құжатының бастапқы мәтін қолтаңбасы бар кестесін келтірілген. Қолтаңбаны бейнелеу үшін мұнда Microsoft Internet Explorer навигаторының <CAPTION> операторының VALIGN параметрінің кеңейтуін қолданылды.

Тапсырма - 5. Қолтаңбасы бар кестелер

```
<HTML> <HEAD>
<TITLE>Қолтаңбасы бар кестелер </TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<TABLE BORDER WIDTH=100%>
<CAPTION VALIGN=TOP ALIGN=RIGHT>Кесте 1.1. Қолтаңбасы бар кесте
</CAPTION>
<TR>
<TH COLSPAN=3>Негізгі тақырып</TH>
</TR>
<TR>
<TH COLSPAN=2>Ұяшық 1 және 2</TH>
<TH>Ұяшық 3</TH>
</TR>
<TR>
```

```

<TD>000</TD><TD>001</TD><TD>002</TD>
</TR>
<TR>
<TD>010</TD><TD>011</TD><TD>012</TD>
</TR>
<TR>
<TD>100</TD><TD>101</TD><TD>102</TD>
</TR>
</TABLE> </BODY><HTML>

```

Төменде келтірілген қолтаңба кестенің астында орналасқан HTML құжатының бастапқы мәтіні көрсетілген. Ол үшін Netscape Navigator навигаторымен және Microsoft Internet Explorer навигаторымен де үлескен тәсілді қолдандық.

Тапсырма - 6. Қолтаңбасы бар кестелер

```

<HTML> <HEAD>
<TITLE>Қолтаңбасы бар кестелер</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF>
<TABLE BORDER WIDTH=100%>
<CAPTION ALIGN=BOTTOM>Кесте. 1.2. Қолтаңбасы бар кесте</CAPTION>
<TR>
<TH COLSPAN=3 >Негізгі тақырып</TH>
</TR>
<TR>
<TH COLSPAN=2>Ұяшық 1 және 2</TH>
<TH>Ұяшық 3</TH>
</TR>
<TR>
<TD>000</TD><TD>001</TD><TD>002</TD>
</TR>
<TR>
<TD>010</TD><TD>011</TD><TD>012</TD>
</TR>
<TR>
<TD>100</TD><TD>101</TD><TD>102</TD>
</TR>
</TABLE> </BODY><HTML>

```

Кесте ұяшықтарындағы мәтінді туралау

ALIGN және VALIGN параметрлері көмегімен кестелер ұяшықтарында орналасқан мәтінді туралауға болады. Төменде көрсетілген кестеден сол және үстінде орналасқан көпқабатты тақырыптары бар күрделі кестесі HTML құжатының бастапқы мәтінінде келтірілген.

Тапсырма - 7. Кесте ұяшықтарындағы мәтінді туралау

```

<HTML> <HEAD>
<TITLE> Кесте ұяшықтарындағы мәтінді туралау </TITLE>
</HEAD>
<BODY BGCOLOR= #FFFFFF>

```

```

<TABLE BORDER>
<TR>
<TD>
<TH ROWSPAN=2></TH>
<TH COLSPAN=3>Пәндер тізімі</TH>
</TD>
</TR>
<TR>
<TD>
<TH>Ақпарат қауіпсіздігі</TH>
<TH>Ықтималдық теориясы</TH>
<TH>Интернет технологиясы </TH>
</TD>
</TR>
<TR>
<TH ROWSPAN=2>Студент</TH>
<TH>Туралау түрлері</TH>
<TD>Сол жақ шетке туралау</TD>
<TD ALIGN=CENTER>Ортаға туралау</TD>
<TD ALIGN=RIGHT>Оң жақ шетке туралау</TD>
</TR>
<TR>
<TH>Ақылбекова М.З. </TH>
<TD VALIGN=TOP>Бес</TD>
<TD ALIGN=CENTER>Төрт</TD>
<TD VALIGN=BOTTOM ALIGN=RIGHT>Үш</TD>
</TR>
</TABLE> </BODY></HTML>

```

Кестелер ішіндегі кестелер

Кестелер өздерінің ұяшықтарында HTML құжатында келтірілген көптәген объектілер болу мүмкін. Сонымен бірге кесте ұяшықтарының ішіне басқа кестелер орнатуға болады.

Тапсырма8. Кестелер ішіндегі кестелер

```

<HTML> <HEAD>
<TITLE>Кесте ішіндегі кесте </TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF>
<TABLE BORDER>
<TR>
<TD>
<TABLE BORDER>
<TR>
<TD>01</TD>
<TD>02</TD>
</TR>
<TR>
<TD>11</TD>
<TD>12</TD>
</TR>
</TABLE>

```



```

</TR>
<TD>001</TD>
<TD>002</TD>
</TR>
<TR>
<TD>100</TD>
<TD>101</TD>
<TD>102</TD>
</TR>
<TR>
<TD>200</TD>
<TD>201</TD>
<TD>202</TD>
</TR>
</TABLE>
</BODY></HTML>

```

HTML: формалар

CGI бұл WWW сервердің бақылауындағы сырты программаларды іске қосу үшін арналған стандарттың көмейлік интерфейсі. Өндеуге арналған ақпаратты енгізу үшін құрылған HTML құжатында пішіні орналасқан. Бұл форма қажетті басқару органдарынан: мәтіндік ақпаратты редакциялау өрістерінен, ауыстыру қозғыштардан, тізімдерден және т. б тұрады. Енгізу пішіні бізге белгілі Microsoft Windows операциялық жүйенің диалогтық панелдерге ұқсайды. Әрбір басқару органына керікті ат меншіктеледі. Сонымен бірге, бұл формада пішін толғаннан кейін басатын батырма болу қажет.

Қолданушы пішінді толтырып және сол батырманы басқан кезде форманың тақырыбында жолы көрсетіліп, CGI қолданбасына мәліметтер беріледі. Бұл қолданба өрістер/мәндер атты түрде берілген пішіндер өрістерінен HTTP протоколы арқылы мәліметтер алады.

Алынған мәліметтерді өндегеннен кейін CGI қолданбасы HTML құжатын құрайды және оны stdout шығару құрылғысына жазады. Бұл құжат автоматты түрде қашықтығы қолданушыға беріледі.

HTML формалары web-құжат жариялаушылар мен оқырмандар арасында ақпарат алмасуға мүмкіндік жасайды. Бұған дейін біз web-құжаттарды тек экранға шығару әдістерін талқылап келген болсақ, енді, керісінше оларға мәлімет енгізу әрекеттерін қарастырамыз. Формалар арқылы тұтынушыдан жалпы мәтін түрінде ақпарат сұрап алуға, "иә/жоқ" деген жауаптың бірін таңдауға немесе бірнеше жолдың біріне тоқтауға болады.

Формаларды әртүрлі мақсаттарда қолдану мүмкіндігі бар. Мысал ретінде, сайтқа кірген оқырмандардың ол туралы пікірлерін формаға енгізуі туралы айтуға болады, жалпы HTML формаларын пайдалану аймағы көп түрлі болып келеді.

<FORM> тәгі

Әрбір форма осы тәгпен басталады. Оның қолданылатын форманы өңдеу программасы (скрипт) және мәліметтерді жөңелту әдісін көрсететін екі атрибутын анықтап алу қажет.

- Атрибуты
- Атқаратын қызметі

ACTION

Форма мәліметтерін қабылдап алып, оны өңдейтін URL -ды анықтайды.

Егер бұл атрибут анықталмаса, онда мәліметтер форма орналасқан web парақ адресіне жіберіледі.

METHOD

Форманы оңдцеу программасына қалайша ақпарат жөнелтілетінін көрсетеді.

Әдетте, оны мәні POST болады, мұндайда форма мәліметі URL-дан бөлек, жеке жөнелтіледі. Ал оның мәні GET болса, онда мәлімет URL мен бірге жіберіледі.

Мысал:

```
<FORM METHOD="post" ACTION='7 cgi-bin/comment^script" >
```

.....

```
</FORM>
```

Бұл мысалда броузерге мынадай нұсқау берілген: толтырылған форманы *post* әдісін қолдана отырып жөнелтіп, оны web-құжат орналасқан сервердің *cgi-bin* каталогында орналасқан *comment script* скрипті көмегімен өңдеу керек екендігі көрсетілген. Бір web-параққа орналасатын формалар саны шектелмейді, бірақ та бір форманы екінші бір форманың ішіне кіріп кетпеуін қадағалап отыру керек.

Форма жасау тегтері

HTML, тілінде формадағы әр түрлі өріс типтерін жасау үшін <TEXTAREA>, <SELECT> және <INPUT> сыяқты үш түрлі тэг қолданылады. Олардың кез келген саны <FORM> ... </FORM> тәрізді қосарланған тәггер ішінде, яғни контейнерде орналаса алады.

Төмендегі кестеде солардың қысқаша сипаттамалары көрсетілген, кейінірек олар толық қарастырылады.

- Тэг
- Атқаратын қызметі

<NEXTAREA>

Кез келген тұтынушы бірнеше жолдан тұратын мәтіндік ақпарат енгізе алатын өрісті анықтайды.

<INPUT>

Мәлімет енгізуді кейбір өзге түрлерін бір жол мәтін енгізу, жалаушаларды көтеріп қою және түсіру, ажыратып қосқышты таңдау және мәліметтерді жөнелтуге немесе форманы тазартуға арналған батырманы басу сыяқты әрекеттерді орындауды қамтамасыз етеді.

<SELECT>

Тұтынушыға жылжымалы жолағы бар терезедегі немесе суырылып шығатын меню ішіндегі бір жолды таңдау мүмкіндігін береді.

<TEXTAREA> тәгі

Бұл тэг бірнеше жолдан тұратын мәтіндік ақпарат енгізуге арналған өріс құру үшін қажет.

<TEXTAREA> ...</TEXTAREA> тәгері арасына, алдын ала келісім бойынша, енгізу өрісіне шығарылатын кез келген мәтін жолдарын орналастыруға болады. Бұл төгтің атрибуттары:

NAME - өріс атын анықтайды, міндетті түрде болуы тиіс;

ROWS — өріс биіктігі бойынша орналасатын жолдар санын береді;

COLS — өріс енін, яғни жол ұзындығын символдар санымен береді.

Мысалы, әр жолына 15 символ мәтін сиятын, екі жолдан тұратын өріс былай жасалады:

```

<HTML>
<BODY>
<FORM>
TEXTAREA Name-Студент ROWS=2 COLS=15>
        Қабдиева Әйгерім
    </TEXTAREA>
</FORM>
</BODY>
</HTML>

```

Мұның нәтижесі суретте көрсетілген. ROWS және COLS атрибуттары арқылы кез келген көлемдегі өріс құруға болады. Бұл атрибуттар міндетті түрде қажет болмағанмен, олардың алдын ала келісім бойынша бекітілген белгілі бір мәні жоқ, сондықтан олардың мәнін орқашанда ашық көрсетуге тырысу керек.

SELECT тәгі

Бұл тәг суырылып шығатын меню немесе айналдыру жолағы бар опциялар (командалар) тізімін құру үшін қолданылады. Опциялар тізімі мен меню пунктері SELECT контейнерінің ішінде орналасады. <TEXTAREA> тәгі сияқты <SELECT> тәгінде де берілген атты анықтайтын NAME атрибуты міндетті түрде болуы тиіс. Опциялар саны SIZE атрибутында көрсетіледі. Төменгі кестеде <SELECT> тәгінің атрибуттарының атқаратын қызметтері жайлы айтылған. Атрибут атқаратын қызметі NAME ақпарат атын анықтайды SIZE таңдау опциялары үшін терезенің биіктігін анықтайды. Егер атрибут көрсетілмесе немесе оның мәні 1- ге тең болса, онда айнымалы опциялар тізімі шығады. Егер оның мәні 1- ден артық, балса, онда опциялар айналдыру жолай бар терезеде көрінеді. Ал егер атрибут мәні тізімі элементтерінің нақты санынан артық болса, онда бос жолдар қосылады. Оларды таңдаған кезде, бос жолдар қайтарылады.

MULTIPLE тәгі

Бірден бірнеше опциялар таңдауға мүмкіндік береді.

Опциялар тізімі <OPTION> тәгі арқылы <SELECT> контейнеріне кіргізіледі. Бұл тәгтің екі атрибуты болады, олар төмендегі кестеде сипатталған.

- Атрибут
- Атқаратын қызметі

VALUE

Тұтынушы опцияны таңдаған жағдайда өңдеу бағдарламасына қайтарылатын мәнді көрсетеді.

SELECTED

Алдын ала кеісім бойынша таңдалған опцияны көрсетеді.

INPUT тәгінің төмендегі кестеде көрсетілгендей бірсыпыра атрибуттары болады.

- Атрибут
- Атқаратын қызметі

MAXLENGTH

Енізу өрісінің символдар санымен берілген енін көрсетеді.

Өріске енгізуге болатын ең үлкен символдар санын анықтайды-

VALUE

Мәтіндік өріс үшін алдынала келісім бойынша шығарылатын мәтінді анықтайды. Жалаушалар мен ажыратып- қосқыштар үшін өңдеу программасына қайтарылатын мәнді көрсетеді. Формаларды жөнелту мен тазарту батырмалары үшін батырма үстіндегі жазуды анықтайды.

CHECKED

Жалауша немес ажыратып қосқыштың алдын ала келісім бойынша, іске қосылған жағдайын орнатады. Басқа тәг түрлерінде INPUT қолданылмайды TYPE Енгізу өрісі типін орнатады. Енгізу өрісі түрі, <TYPE> атрибуты <INPUT> тәгінің <TYPE> атрибуты мәндері кестеде көрсетілген.

CSS-пен танысу(стильдердің каскадты кестелері)

Стильдік анықтамаларды құруда қолдануға болатын 70-тен артық қасиеттер түрі бар. Алдымен бірнеше қарапайым оқу жаттығуларын жасау қажет.

Стильдік анықтамалардың қасиеттерін келесі бөлімдерге бөліп қарастырайық:

- қаріп;
- түс;
- мәтін;
- өрістер мен рамкалар;
- кескін.

CSS қасиеттерінің ішінде стильдік нұсқауларда қолданылатын өлшем бірліктерінен бастайық. Өлшем бірліктері Стильдік нұсқауларда қолданылатын өлшем бірліктер

2.1 кестесінде көрсетілген.

2.1 кесте

Белгіленуі	Атауы	Мысал
in	Дюйм	<P style="font-size: 1in">
cm	Сантиметр	<P style="font-size: 1cm">
mm	Миллиметр	<P style="font-size: 1mm">
px	Пиксел	<P style="font-size: 1px">
pt	Пункт	<P style="font-size: 1pt">
pc	Пикалар	<P style="font-size: 1pc">
%	Пайыз	<P style="font-size: 1%">

Пункттер мен пикалар — қаріп өлшемі немесе кегльді белгілейтін типографиялық бірліктер.

Wordта мысалы, бұл параметр арнайы терезеде 8- ден 72-ге дейінгі сандармен беріледі. Бұл кегльдің пункттердегі шамасы. Бір типографиялық пункт 0,375 мм-ге тең.

Кітаптарды баспадан шығару кезінде негізгі мәтін үшін 10 немесе 12 пункттердегі кегль таңдалады. Тақырыптар үшін – үлкенірек кегльдер, ал ескертулер мен сілтемелер үшін ұсақ кегльдер (әдетте 8 пункттер).

Пика — тым ірі өлшем бірлігі. Бір пика 12 пунктке тең.

Пайыздық есеп қандай да бір базалық мәннен жүргізіледі. Қаріп жағдайында — ағымдағы өлшемнен.

Пайызды қолдану әзірлеуші үшін өте қызықты. Қолданушы браузерде қаріп өлшемін өзгерткен кезде, стильдік нұсқауларда пайыздық түрде берілген өлшемдер пропорционалды түрде өзгереді.

font-family

қаріп Бұл стильдік қасиет қаріптің (мысалы, Arial) гарнитурасының атауын немесе тектік атауын береді:

serif – кертпесі бар қаріп (серифті);

sans-serif – кертпесіз қаріп (ұсақталған);

monospace – біренді қаріп (таңбалардың ендері бірдей болады).

2.6—2.8 суреттерінде қаріптердің мысалдарф көрсетілген

Бұл кертпесі бар қаріп.

2.6 - сурет. Серифті қаріп

Бұл ұсақталған қаріп.

2.7 - сурет. Ұсақталған қаріп

Бұл біренді қаріп.

2.8 - сурет. Біренді қаріп.

Гарнитура — бір қаріптің таңбалар кескіндерінің жинағы.

Бұл жинақта тура және көлбеу кескін болуы мүмкін, қоюлық деңгейі (штрихтың қалыңдығы), литер (ықшамдалған, қалыпты, созылған) және кегль ені бойынша ажыратылуы мүмкін.

Кез келген графикалық операциялық жүйеде үш стандартты гарнитура жинағы болады, Windows үшін бұл:

Times Roman — серифті қаріп;

Arial — ұсақталған қаріп;

Courier — біренді қаріп.

Браузер қалыпты жағдайда қарапайым мәтінді шығару үшін Times Roman қарпін және біренді үшін Courier қарпін қолданады.

Егер гипермәтіндік өнім — сайт болса, өзінің гарнитурасын бермеген жөн.

font-family қасиетінің көмегімен бір ғана емес, бірнеше қаріптерді үтір арқылы қажетті тәртіппен көрсетуге болады:

N1,N2,N3,N4,N5,N6

{

font-family:"Arial Cyr", Geneva, Helvetica, sans-serif;

}

Алдымен браузер Arial Cyr қарпін табуға тырысады, сосын Geneva, сонан соң Helvetica және, ақырында, сәтсіздік жағдайында қандай да бір ұсақталған қаріпті іздейді (sans-serif нұсқауы).

Егер қаріптің атауы бірнеше сөзден тұрса, ол тырнақшаға алынып жазылады.

font-size қасиеті ағымдағы қаріптің абсолютті немесе салыстырмалы өлшемін беру үшін қолданылады. Салыстырмалы өлшем ретінде пайыздық (ағымдағы қаріптің өлшеміне қатысты) беру немесе кілттік сөздер қолданылады:

larger – ірірек;

smaller – ұсақтау.

Абсолютті өлшем ретінде санның келесі өлшем бірліктерінің біреуімен қолданылады: in, cm, mm, px, pt, pc. Және де мынадай кілттік сөздер қолдануға болады:

CSS: тізімдермен жұмыс істеу

Бұл list-style-type сипатын қарастырғыңыз келетін бірінші сипат, ол тізім элементінің маркер түрін көрсетеді. Шарлау мәзірі, мысалы, әдетте, әдеттегі таңбаланған тізімдерден (HTML элементі) жасалады, олардың әдепкі бойынша маркер боялған шеңбер түрінде көрсетіледі, оны алып тастау үшін, None мәні бар list-style-type CSS қасиетін пайдалану қажет:

```
ul {
list-style-type : none; /* убираем маркер у списка */
}
```

Таңбаланған (реттелмеген) тізімдер үшін (HTML элементі) маркерлердің үш түрі қарастырылған: боялған үйірме түрінде (disc - бұл әдепкі мән), қуыс шеңбер түрінде (circle) және квадрат түрінде (square), ал нөмірленген (реттелген) тізімдер үшін (элемент) барлық қалған опциялар. Толық тізім келесі кестеде орналастырылған:

Мәні	Сипаттамасы
none	
Тізім элементі Тізім элементі Тізім элементі	Маркер көрінбейді.
disc	
<ul style="list-style-type: none"> • Тізім элементі • Тізім элементі • Тізім элементі 	Боялған үйірме түріндегі Маркер. Бұл мән әдепкі болып табылады.
circle	
<ul style="list-style-type: none"> ○ Тізім элементі ○ Тізім элементі ○ Тізім элементі 	Шеңбер түріндегі Маркер.
decimal	
<ol style="list-style-type: none"> 1. Тізім элементі 2. Тізім элементі 3. Тізім элементі 	Сандық маркер (1-ден басталатын ондық араб сандары).
decimal-leading-zero	
<ul style="list-style-type: none"> • Тізім элементі • Элемент списка • Элемент списка 	Сандық маркер (1 - ден басталатын және бастапқы нөлмен толықтырылған ондық араб сандары-01, 02, 03...).
lower-latin	
<ul style="list-style-type: none"> • Тізім элементі • Тізім элементі • Тізім элементі 	Латын әріптері (a, b, c, d,...z).
lower-roman	
<ol style="list-style-type: none"> i. Тізім элементі ii. Тізім элементі iii. Тізім элементі 	Төменгі регистрдегі рим сандары (i, ii, iii, iv, v және т.б.).
square	
<ul style="list-style-type: none"> ▪ Тізім элементі ▪ Тізім элементі • Тізім элементі 	Квадрат түріндегі Маркер.

upper-roman

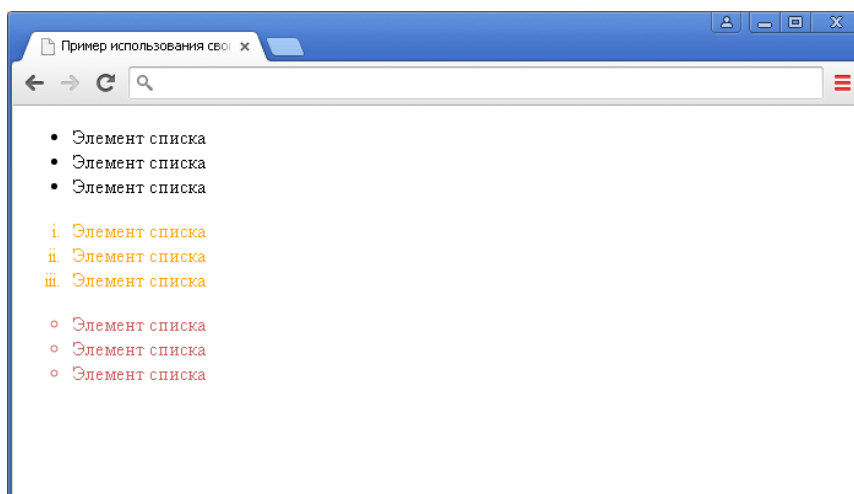
- I. Тізім элементі
 - II. Тізім элементі
 - Тізім элементі
 - III.
- Жоғарғы регистрдегі рим сандары (I, II, III, IV, V және т. б.).

Мәтінде list-style-type қасиеттерін пайдалану мысалын қарастырайық:

```
<!DOCTYPE html>
<html>
<head>
  <title>Пример использования свойства list-style-type</title>
<style>
.test {
list-style-type : lower-roman; /* устанавливаем тип маркера - римские числа в
нижнем регистре */
color : orange; /* обратите внимание, что текст цвета соответствует цвету
маркера */
}
.test2 {
list-style-type : circle; /* устанавливаем тип маркера в форме круга */
color : IndianRed; /* обратите внимание, что текст цвета соответствует цвету
маркера */
}
</style>
</head>
  <body>
    <ul> /* список со значением типа маркера по умолчанию
(disc). */
      <li>Элемент списка</li>
      <li>Элемент списка</li>
      <li>Элемент списка</li>
    </ul>
    <ol class = "test"> /* нумерованный список с типом маркера
lower-roman */
      <li>Элемент списка</li>
      <li>Элемент списка</li>
      <li>Элемент списка</li>
    </ol>
    <ul class = "test2"> /* маркированный список с типом маркера
circle */
      <li>Элемент списка</li>
      <li>Элемент списка</li>
      <li>Элемент списка</li>
    </ul>
  </body>
</html>
```

Бұл мысалда біз екі стильді құрдық, біріншісі маркер түрін белгілейді - төменгі регистрдегі Рим сандар (lower-roman мәні), біз оны нөмірленген тізімге (HTML элементі) қолдандық, ал таңбаланған тізімге (HTML элементі) біз толық шеңбер түрінде маркер түрін белгілейтін стильді қолдандық (circle мәні).

Біздің мысалымыздың нәтижесі:



Сур. 69 list-style-type қасиеттерін пайдалану мысалы (маркер түрін орнату).

Сіздің назар аударыңыз, list-style-type сипатын тіпті бөлек алынған тізім элементіне (HTML элементі) қолдануға болады, немесе келесі мысалда қарастырылғандай белгілі бір таңбаны жұп немесе тақ тізім элементтеріне қолданатын мәнерлерді жасау:

```

<!DOCTYPE html>
<html>
<head>
  <title>Пример чередования стилей свойства list-style-type</title>
</head>
<body>
  <ul>
    <li>Элемент списка</li>
    <li>Элемент списка</li>
    <li>Элемент списка</li>
    <li>Элемент списка</li>
    <li>Элемент списка</li>
    <li>Элемент списка</li>
    <li>Элемент списка</li>
    <li>Элемент списка</li>
  </ul>

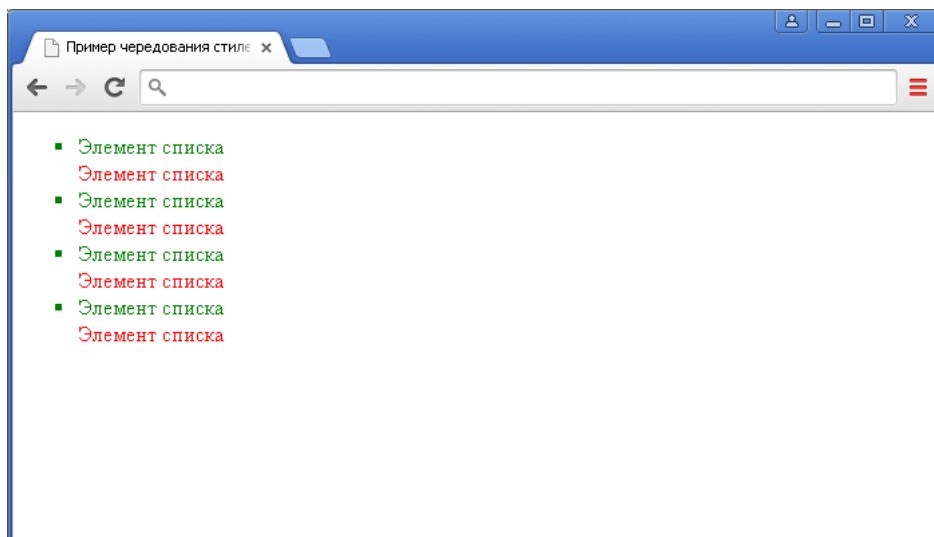
```



```
</body>  
</html>
```

Бұл мысалда біз тізімнің барлық тақ элементтерін стильдедік - маркер түрін шаршы көрсетті, ал мәтін түсі жасыл. Тізімнің жұп элементтері келесі мәнерді алды: мәтіннің қызыл түсі және маркер болмауы.

Біздің мысалымыздың нәтижесі:



Сур. 70 list-style-type қасиеттері мәнерлерін кезектестіру үлгісі.

Таңбашаның тізім элементіне қатысты орналасуы

List-style-position сипаттары CSS арқылы таңбаның орналасуын немесе сандарды тізім элементіне қатысты анықтай аласыз. Таңбаны (санды) мазмұнмен бірге тізім элементінің ішіне орналастыру үшін inside мәнін пайдалану керек, ал тізім элементінің шетіне орналастыру үшін outside мәнін пайдалану керек.

Көрнекілік үшін list-style-position қасиеттерінен басқа элементтің шекарасын орнатуға мүмкіндік беретін тағы бір жаңа мүлікті қолданамыз.

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Пример использования свойства list-style-position</title>  
<style>  
li {  
border : 1px solid orange; /* устанавливаем сплошную границу размером 1px  
оранжевого цвета для всех элементов списка*/  
}  
.test {  
list-style-position : outside; /* указываем, что маркер располагается слева от  
текста за границей элемента списка */  
background-color : khaki; /* устанавливаем цвет заднего фона */  
}
```

```

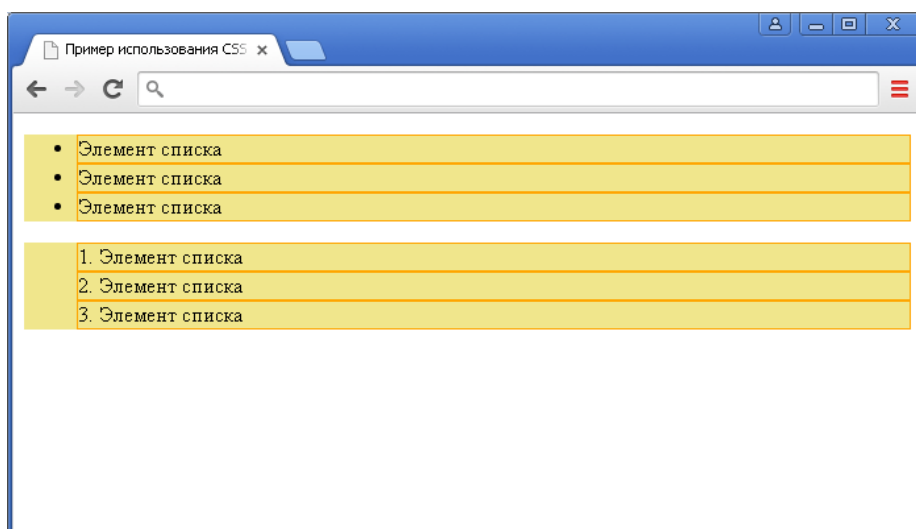
.test2 {
list-style-position : inside; /* указываем, что маркер располагается слева от
текста внутри элемента вместе с содержимым */
background-color : khaki; /* устанавливаем цвет заднего фона */
}
</style>
</head>
  <body>
    <ul class = "test">
      <li>Элемент списка</li>
      <li>Элемент списка</li>
      <li>Элемент списка</li>
    </ul>
    <ol class = "test2">
      <li>Элемент списка</li>
      <li>Элемент списка</li>
      <li>Элемент списка</li>
    </ol>
  </body>
</html>

```

Бұл мысалда таңбаланған тізім үшін (HTML элементі) біз маркерді мазмұнмен бірге list-style-position қасиетін outside мәніне қолдана отырып, тізім элементінің ішінде орналастырдық, ал нөмірленген тізім үшін (HTML элементі) цифрды элементтің ішінде мазмұнмен бірге (inside мәніне list-style-position қасиеті) орналастырдық.

Тізімнің барлық элементтері үшін шекараны орнатуға мүмкіндік беретін border әмбебап қасиеті list-style-position қасиеттерін жақсы түсіну үшін қолдандық. Элементтердің шекараларымен жұмыс жақын арада оқулықта "CSS элементінің шекаралары"мақаласында егжей-тегжейлі қаралатын болады.

Біздің мысалымыздың нәтижесі:



Сур. 71 list-style-position қасиеттерін пайдалану мысалы (тізімдегі маркер / санның орналасуы).

CSS: фонмен жұмыс істеу

CSS-те түс пен фонның негізгі сипаттары мыналар:

color
background-color
background-image
background-repeat
background-attachment
background-position
background

Енді ретімен айтып көрейік:

COLOR сипаты

Қандай да бір элемент негізгі түсті береді (алдыңғы планның түсін). Мысалы, егер біз бірінші деңгейдегі тақырыптардың барлық түсін қызыл, ал параграфтарың мәтінінің түсін жасыл жасағымыз келсе, онда стилдер кестесі былайша беріледі:

```
H1 {  
color: red ;  
}  
P {  
color: green ;  
}
```

Мұнда P орнына H2 алуға да болады. Кодта өзгерту керек

BACKGROUND-COLOR сипаты

Бұл фондық түс береді. Фондық түсті тек бет пен ячейка кестесі үшін (bgcolor атрибуты барларға) пайдалануға болады. Фондық түсті не болса, соған беруге болады: кесте, тақырып, параграф, сілтеме тағы басқалар. Мұнда басты мәселе бізге не керек екенін анықтау керек болады.

Мына мысалда барлық беттің фондық түсін өзгерту үшін BODY элементінің сипатын беру керек, өйткені ол бүкіл бетке жауап береді, яғни.

```
BODY {  
background-color : #FFEE8C ;  
}  
H1 {  
color: red ;  
background-color :blue ;  
}  
P {  
color: green ;  
}
```

BACKGROUND-IMAGE сипаты

Бұл сипат фондық бейнені беру үшін қолданылады. Төмендегі мысалда, мен барлық беттің фондық бейнесін көрсетемін, яғни BODYэлементі үшін.

```
BODY {
background-color : #FFEE8C ;
background-image : url(smile.png) ;
}
H1 {
color: red ;
background-color :blue ;
}
P {
color: green ;
}
```

Сипаттың мәні ретінде бейненің жолы көрсетіледі, бірақ html-дегідей емес. Әуелі URL, сосын дөңгелек жақшаның ішіне сурет орны жазылады. Егер ол сол қалтада тұрса, онда сурет атауын жазамыз, жоғарыдағы мысалдай. Егер img қалтада болса, онда жазамыз url(img/gul.png).

BACKGROUND-REPEAT сипаты

Сіз байқаған боларсыз, әдепкі фондық бейне жоғарғы сол бұрыштан бастап, тігінен де , көлденеңінен де, бүкіл экран толғанша қайтланады. Сонымен background-repeat сипатының көмегімен біз осы қайталауды бақылай аламыз.

Бұл сипаттар төрт мән қабылдайды:

Background-repeat: repeat-x ; - Көлденеңінен қайталау

Background-repeat: repeat-y ; - Тігінен қайталау

Background-repeat: repeat ; - Тігінен және көлденеңінен (әдепкі мән)

Background-repeat: no-repeat ; -қайталанбайды

Егер сипатты көрсетпесеңіз, онда әдепкі мән қолданылады, яғни фонды бейне тігінен де көлденеңінен де қайталанатын болады.

BACKGROUND-ATTACHMENT сипаты

Фондық сурет бар болса, бұл сипат қояды, фондық сурет бет мазмұнымен бірігеді, немесе бұғатталады, яғни қозғалмайды. Екі мән қабылдайды:

SCROLL - фон мәнімен бірге айналады;

FIXED - фон қатаң бекітілген.

```
BODY {
background-image : url(img/gul.png) ;
background-repeat: no-repeat;
background-attachment: scroll ;
}
```

BACKGROUND-POSITION сипаты

Бейненің фондық позициясын береді. Мәнді процентпен беруге болады, ұзындық бірлігі кілттік сөз арқылы беріледі. Есептеу браузердің сол жоғарғы

бұрышынан басталады және онда әдепкі фондық сурет орналасады. Суретте позициялау мысалы келтірілген (нүкте бұл бейне типі, ал оның астында – жуықтау координатасы)

Әуелі көлденеңінен координатаны (X осі бойынша), сосын пробел арқылы тігінен координатаны (Y осі бойынша) көрсетеміз. Координатаны браузер терезесінің ені бойынша процентпен, тіпті пикселмен беруге болады. Сантиметрмен де беруге болады, бұл қолайсыздау болады.

Орынды арнайы сөздермен де беруге болады:

left - солға, right - оңға, center -ортасына, top - жоғары, bottom – төмен .

Төмендегі стилдердің біреуіне ғана тоқталамыз. Қалғанын өздеріңіз байқап көріңіздер:

```
BODY {
background-image : url(img/gul.png) ;
background-repeat: no-repeat;
background-position: top right;
}
```

Жазбаның қысқартылған түрі - BACKGROUND

BACKGROUND сипаты барлық жоғарыда қарастырылған қысқартылған жазбалар үшін қызмет етеді.

Бұл элементтің сипатының реті мынадай:

```
background-color_background-image_background-repeat_background-attachment
_background-position
```

Яғни пробел арқылы бес мән жазылады. Төмендегі мысалды қарастырсаңыз Сізге бәрі түсінікті болады:

Былай жазылғандар:

```
BODY {
background-color:#ffee8c ;
background-image : url(smile.png) ;
background-repeat: no-repeat;
background-attachment: fixed;
background-position: top right;
}
```

Бір жолмен жазуға болады:

```
BODY {
background: #ffee8c url(smile.png) no-repeat fixed top right ;
}
```

Өзіңіз көріп отырғандай, сипатты пробел арқылы жазамыз. Егер қандай да бір сипат көрсетілмесе, онда оған автоматты түрде әдепкі мән беріледі.

CSS: элементтердің орналасуы және көрінуі

Орналасу-веб-беттегі элементтің орналасу орнын (позициясын) басқару, ол position қасиетінің CSS арқылы бақыланады.

Көрсету үшін нақты орналасқан жерін позиционированных элементтерін пайдаланылады CSS қасиеттері: top, right, bottom және left. Олар статикалық элементтерден басқа барлық позициялық элементтермен жұмыс істейді.

Position сипаты 4 мәні бар: static, fixed, relative және absolute. Осы мәндердің әрқайсысы пайдалану мысалымен төменде көрсетіледі.

Беттегі элементтерді жайғастырудың барлық түрлерін егжей-тегжейлі талдау алдында, біз құжат ағынын қарастыруымыз керек.

Құжат ағыны

Әдепкі бойынша, Веб-беттегі элементтер HTML-құжатта ұсынылған ретпен көрсетіледі, яғни блоктық элементтер оларға қол жетімді барлық енін алады және бір-бірінің астына тігінен қойылады. Жол элементтері барлық ені бос болғаннан кейін, барлық ені бос болғанша, жолдың тасымалдануы жасалады және бәрі жаңаға барады. Элементтердің орналасуының мұндай тәртібі қалыпты ағын деп аталады (оны құжаттың ағыны немесе жалпы ағын деп атайды).

Float немесе position қасиеттері арқылы элементті қалыпты ағыннан алып тастауға болады. Егер элемент қалыпты ағыннан "түссе", онда осы элементтің төменгі кодында орналасқан элементтер веб-бетте оның орнына ауысады.

Статикалық позициялау

Статикалық деп веб-беттегі барлық элементтер үшін әдепкі орналасу деп аталады. Егер элементке position сипатын қолданбаса, онда ол статикалық болады және элементтердің жалпы ағынына сәйкес веб-бетте пайда болады.

Жоғары, left, right немесе bottom қасиеттерінің CSS-ін статикалық орналасқан элементке қолданғанда, олар еленбейді.

Қажет болса, static мәнімен стильдер кестесінде статикалық орналасуды орнатуға болады:

```
1
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <meta charset="utf-8">
6 <title>Название документа</title>
7 <style>
8 p { position: static; }
9 </style>
10 </head>
11 <body>
12 <p>Первый абзац.</p>
13 <p>Второй абзац.</p>
14 </body>
15 </html>
```

Тіркелген позициялау

Тіркелген орналасу элементтері браузер терезесіне қатысты бетте орналасады. Мұндай элементтер жалпы ағыннан жойылады, тіркелген элементтің ағынында келесі элементтер оны елемейді, жылжып, веб-бетте оның орнын алады.

Бекітілген орналасу элементтері басқа элементтерді толығымен немесе ішінара жасырып, жаба алады. Ұзын беттерді айналдырған кезде, олар бір жерде қалып, қозғалмайтын нысандардың әсерін жасайды:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Название документа</title>
6 <style>
7     img {
8         position: fixed;
9         top: 5%;
10        left: 40%
11    }
12 </style>
13 </head>
14 <body>
15 
16 <pre> Мәтін
17     Мәтін
18     Мәтін
19     Мәтін
20     Мәтін
21     Мәтін     Мәтін
22     Мәтін
23     Мәтін
24     Мәтін </pre>
25 </body>
</html>
```

Салыстырмалы орналасу

Статикалық элементтер сияқты салыстырмалы орналасу элементтері жалпы ағында қалады. Жоғары, left, right немесе bottom сипаттарын салыстырмалы түрде орналасқан элементтерге қолданғанда, олар элемент бастапқыда орналасқан жерде бос кеңістікті қалдырып, өз орнына қатысты ауысады.

Мұндай элементтер қоршаған элементтердің орналасуына әсер етпейді, қалған элементтер өз орындарында қалады және позицияланған элементпен жабылуы мүмкін:

Абсолютті позициялау

Абсолютті орналасу элементтері жалпы ағыннан толығымен жойылады, қалған элементтер толығымен позицияланған элементтерді елемей, бос кеңістікке ие болады. Содан кейін элементті top, left, right немесе bottom сипаттарын пайдаланып, кез келген веб-беттің орнына қоюға болады.

Барлық абсолютті позицияланған элементтер браузер терезесіне немесе position сипаты absolute, fixed немесе relative мәні бар жақын орналасқан ата-бабаға (егер бар болса) қатысты орналастырылады.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Название документа</title>
6 <style>
7     img {
```

```

6         position: absolute;
7         right: 0px;
8         top: 0px;
9     }
10    </style>
11    </head>
12    <body>
13        
14        <p>Изменим расположение логотипа с помощью абсолютного позиционирования.<br>
15        Теперь логотип будет располагаться в правом верхнем углу страницы.</p>
16    </body>
17 </html>

```

Жабу элементтері

Элементтер беттің жалпы ағынынан тыс болғанда, олар бір-бірін жаба алады. Әдетте элементтердің орналасу тәртібі олардың бет HTML-кодындағы тәртібіне сәйкес келеді, бірақ z-index қасиетінің CSS көмегімен жабу тәртібін басқару мүмкіндігі бар, оның мәні неғұрлым көп болса, элемент соғұрлым жоғары болады.

```

1    <!DOCTYPE html>
2    <html>
3        <head>
4            <meta charset="utf-8">
5            <title>Название документа</title>
6            <style>
7                div {
8                    position: absolute;
9                    width: 100px;
10                   height: 100px;
11                   }
12                .div1 {
13                    background-color: #66FFFF;
14                    left: 70px;
15                    top: 0px;
16                    z-index: 1;
17                }
18                .div2 {
19                    background-color: #FFFF66;
20                    left: 0px;
21                    top: 30px;
22                    z-index: 0;
23                }
24                .div3 {
25                    background-color: #66FF66;
26                    left: 33px;
27                    top: 60px;
28                    z-index: 2;
29                }
30            </style>
31        </head>
32        <body>
33            <div class="div1">z-index: 1;</div>
34            <div class="div2">z-index: 0;</div>
35            <div class="div3">z-index: 2;</div>
36        </body>
37    </html>

```

Браузер деңгейінде CSS-ті қолдау

Тапсырма: HTML мен CSS қолданып портфолио құрастыру

Портфолио деген – белгілі бір маманның орындаған жұмыстарының сәнді де мәнді етіп жасалған тізбесі.

Мысалы,

копирайтердің (жалға мәтін жазғыш) портфолиосы – оның өзі жазған, тақырыптар немесе мәтін түрлері бойынша (коммерциялық ұсыныс, сайт контенті, слогандар) сұрыпталған мәтіндерінен тұрады.

Модельдің портфолиосы – модельдің түрлі образдардағы бейнесін көрсететін фотосуреттерінен құралады.

Веб-дизайнердің портфолиосы – ол жасаған сайттардың, баннерлердің немесе дизайнерлік өнімдерінің суреттерінен жасалады.

Фотографтың портфолиосы – оның түрлі бағыттарда (тойлар, портреттер, репортаждар) түсірген фотосуреттерінің жиынтығынан жасалады.

Бағдарламашының портфолиосы – өзі жазған CMS, скрипттер, программалардың интерфейс суреттері мен олардың жұмысын көруге болатын сілтемелерден тұрады.

Портфолиодағы жұмыстардың сапасы мен деңгейіне қарап клиенттер оны орындаушы туралы пікір қалыптастырады – мұнымен жұмыс істеуге бола ма, жоқ па. Сондықтан портфолионы жасауға өте үлкен мән беру керек.

Портфолио қалай жасалады?

1. Портфолионың бағыттарын немесе бөлімдерін белгілеп алыңыз. Мысалы, копирайтерде бұл: коммерциялық ұсыныстар, сайттарға арналған мәтіндер, нейминг, слогандар. Фотографта: репортаждық түсірілімдер, портреттер, той түсірілімдері және т.б.,

2. Әр бағыт бойынша ең жақсы жұмыстарыңыз бен орташа сапалы жобаларыңызды таңдап алыңыз. Портфолиоға әр бағыт бойынша 10-20 жұмыс қосу оптималды болады. Осының өзі сіздің кәсіби деңгейіңізді клиентке жеткілікті дәрежеде көрсетеді.

3. Портфолионы жақсылап сәндеуді ойластырыңыз. Жұмыстарыңызды ыңғайлы ретпен және көрнекі түрде орналастырыңыз.

4. Портфолионы Интернеттегі жеке сайтыңызға орналастырыңыз. Қосымша компьютеріңіз бен флешкаға да сақтап қойған жөн.

Портфолиоға қандай жұмыстарды қосу керек?

Портфолио жасағанда алғашқы туындайтын сұрақ – қандай жұмыстарымды көрсету керек? Егер Сіз кәсібіңізді енді-енді бастаған болсаңыз, онда барлық тындырған істеріңізді көрсеткен дұрыс. Сізге сұраныс беруші клиенттеріңіз еңбекқор, ізденгіш, тындырымды екеніңізді және жеткілікті тәжірибе жинақтағаныңызды көреді. Ал егер өз кәсібіңізбен ежелден айналысып келе жатқан, өте тәжірибелі шебер болсаңыз, онда портфолиода ең сапалы да қызықты жұмыстарыңызды іріктеп көрсеткен жөн. Сіздің үлкен шеберлігіңізді сол жоғары сапалы жұмыстарыңыз бірден-ақ көрсетеді.

JavaScript кіріспе

Бастапқыда JavaScript "веб-беттерді тірі ету" үшін жасалды. Бұл тілдегі бағдарламалар скрипттер деп аталады. Олар HTML ішіне кіріп, веб-бетті жүктеу кезінде автоматты түрде орындалуы мүмкін. Скрипттер қарапайым мәтін ретінде таратылады және орындалады. Оларға арнайы дайындық немесе іске қосу үшін компиляция қажет емес. Бұл JavaScript басқа тіл – Java ерекшелендіреді.

JavaScript құрылғанда, оның басқа атауы – "LiveScript" болды. Алайда, Java тілі сол уақытта өте танымал болды және Java "кіші ағасы" ретінде JavaScript позициялау пайдалы болады деп шешілді. Уақыт өте келе, JavaScript өзінің ECMAScript деп аталатын өзіндік ерекшелігімен толық тәуелсіз тіл болды және қазір Java-ға ешқандай қатысы жоқ.

Қазіргі таңда JavaScript тек браузерде ғана емес, серверде немесе JavaScript "қозғалғыш" деп аталатын арнайы бағдарламасы бар кез келген басқа құрылғыда орындалуы мүмкін. Браузерде "JavaScript виртуалды машинасы" деп аталатын өз қозғалтқышы бар. Әртүрлі қозғалғыштарда әртүрлі "кодтық есімдер" бар. Мысалы: V8-Chrome және Opera. Spidermonkey-Firefox. Сонымен қатар, Trident "және" Chakra "IE," ChakraCore "Microsoft Edge," Nitro "және" Squirrelfish " Safari және т. б. Бұл атауларды білу пайдалы, себебі олар әзірлеушілер үшін мақалаларда жиі қолданылады. Біз оларды да пайдаланамыз. Мысалы, егер "x функционалдығы V8 болса", онда "X" Chrome мен Opera-да жұмыс істейді.

Қозғалтқыштар қалай жұмыс істейді? Қозғалтқыштар күрделі. Бірақ негіздер түсіну оңай. Қозғалтқыш (кірістірілген, егер бұл браузер болса) скрипт мәтінін оқиды ("парсит"). Содан кейін ол скриптті машина тіліне айналдырады. Осыдан кейін машина коды іске қосылады және тез жұмыс істейді. Қозғалтқыш әр кезеңде оңтайландыруды қолданады. Ол өзі арқылы өтетін деректерді талдай отырып, оның жұмысы кезінде компилденген скриптті тіпті көреді және алынған білімге сүйене отырып, машина кодына оңтайландыруды қолданады. Нәтижесінде скрипттер өте тез жұмыс істейді.

JavaScript браузерде не болуы мүмкін?

Қазіргі JavaScript-бұл "қауіпсіз" бағдарламалау тілі. Ол жадқа немесе процессорға төмен деңгейлі қатынауды қамтамасыз етпейді, себебі бастапқыда оны қажет етпейтін браузерлер үшін жасалған.

JavaScript мүмкіндіктері ол жұмыс істейтін қоршаған ортаға қатты байланысты. Мысалы, Node.JS еркін файлдарды оқу/жазу, желілік сұраныстарды және т. б. орындау функцияларын қолдайды.

JavaScript үшін браузерде веб-беттерді манипуляциялаумен, пайдаланушымен және веб-сервермен өзара әрекеттесумен байланысты барлық нәрсе қол жетімді.

Мысалы, JavaScript браузерінде:

Жаңа HTML кодын бетке қосу, бар мазмұнды өзгерту, стильдерді өзгерту.

Пайдаланушы әрекетіне, тінтуірге, көрсеткішті жылжытуға, пернелерді басуға жауап беру.

Желілік сұраныстарды қашықтағы серверге жіберу, файлдарды жүктеу және жүктеу (Ajax және COMET технологиялары).

Куки алу және орнату, келушіге сұрақтар қою, хабарламаларды көрсету.

Клиенттің жағында деректерді есте сақтау ("local storage").

Браузерде JavaScript не мүмкін емес?

Браузердегі JavaScript мүмкіндіктері пайдаланушының қауіпсіздігі үшін шектелген. Мақсат жосықсыз веб-беттің жеке ақпаратқа қол жеткізуін немесе пайдаланушының деректеріне зиян келтіруін болдырмау болып табылады.

Мұндай шектеулердің мысалдары қамтиды:

JavaScript веб-бетте еркін файлдарды қатты дискіге оқуға/жазуға, оларды көшіруге немесе бағдарламаларды іске қосуға болмайды. Ол ОЖ жүйелік функцияларына тікелей қол жеткізе алмайды.

Қазіргі браузерлер оған файлдармен, бірақ шектеулі қатынаспен жұмыс істеуге мүмкіндік береді және пайдаланушы файлдың браузер терезесіне "сүйреу"

немесе <input>тегімен оны таңдау сияқты белгілі бір әрекеттерді орындаған жағдайда ғана оны ұсынады.

Камерамен/микрофонмен және басқа құрылғылармен өзара әрекеттесу тәсілдері бар, бірақ олар пайдаланушының айқын рұқсатын талап етеді. Осылайша, JavaScript қолдауы бар бет веб-камераны елеусіз қосуға, болып жатқан оқиғаларды бақылауға және ақпаратты ФСБ-ға жіберуге болмайды.

Әр түрлі терезелер / қойындылар дос туралы білмейді. Кейде JavaScript көмегімен бір терезе басқа тілқатысу терезесін ашады. Бірақ тіпті бұл жағдайда да JavaScript бір беттен басқа сайттарға (басқа доменнен, протоколдан немесе порттан) келген жағдайда қол жеткізе алмайды.

Бұл "бірдей дерек көзі саясаты" (Same Origin Policy) деп аталады. Бұл шектеуді айналып өту үшін, екі бет де осы келісілуі тиіс және мәліметтерді арнайы түрде алмасатын JavaScript коды болуы тиіс.

Бұл шектеу пайдаланушының қауіпсіздігі үшін қажет. Бет <http://anysite.com> пайдаланушы ашқан басқа браузер қойындысына кіруге болмайды <http://gmail.com> және сол жерден ақпаратты ұрлауға.

JavaScript ағымдағы бет келген сервермен оңай өзара әрекеттесуі мүмкін. Бірақ оның басқа сайттардан/домендерден деректерді алу қабілеті шектеулі. Бұл мүмкін болса да, алыстағы жағы бар айқын келісім (HTTP тақырыбында көрсетілген) қажет. Тағы да, бұл қауіпсіздікті шектеу.

HTML құжатына JavaScript енгізу

JavaScript программалау тілін Netscape пен Sun Microsystems фирмсы бірлесіп ұсынған, ол интерактивті HTML-құжаттарды құруға арналған. JavaScript тілін қолданылатын аймақтары:

- Динамикалық, яғни құжат жүктелгеннен кейін мазмұнын өзгертуге болатын парақтарды құру.

- Серверге жөнелтілгенге дейін пайдаланушының форманы дұрыс толтырылғандығын тексеру.

- Сценарийлер көмегімен «жергілікті» (локальный) және басқа да мәселелерді шешеу.

JavaScript клиент және сервер жағында да орындалатын қосымшаларды құру мүмкіндігін береді. Екі түрлі қосымшаларды құру кезінде де, стандартты объектілерді анықтаулардан тұратын, ядро қолданылады. Пайдалынушы компьютерінде клиенттік қосымшаларды браузер орындайды.

JavaScript тіліндегі программа (сценарий) браузерге ендірілген интерпретатор арқылы өңделеді. Сценарийлерді кез-келегн брузерде орындалатындай түрде жазуға тырысу керек.

JavaScript тілінде жазылған программалар (сценарий) операторлар тізбегінен тұрады. Егер бір жолға бірнеше операторлар жазылса олардың арасына нүктелі үтір (;) қойылады. Егер әрбір оператор жеке жолға жазылсағ онда ешнәрсе қоймауға болады. Бір операторды бірнеше жолға жазуға болады.

Құрылымдық програмалау принципіне сәйкес, программаны бөліктік құрылым болатындай түрде жазу ұсынылады. Бұл программаны зерттеу мен қателерді іздеуді жеңілдетеді.

JavaScript программаларында түсініктемелерді қолдануға болады. Бір жолға сиятын түсініктеме беру үшін екі қиғаш сызықты қолдануға болады (//). Егер түсініктеме мәтіні бірнеше жолға жазылса, онда оларды /* және */ символдарының

ішінде жазу керек. JavaScript тілінде бас әріптер мен кіші әріптер әр түрлі символ болып саналады.

JavaScript тілінде айнымалыларды сипаттау. Өрнектер. комментарийлер.

Айнымалылар ақпаратты сақтау үшін қолданылады. Айнымалылар сценарийлерде идентификатор латын әріптерінен, цифрлардан, астын сызу белгісінен тұрады және ол міндетті түрде латын әрпінен астын сызу белгісінен басталуы керек, мысалы, `_my_test`, `test_1`. Айнымалының типі сценарийде сақталаған ақпараттың типіне байланысты болып, сол мәліметтің типі өзгерсе, айнымалының типі де өзгереді. Айнымалы `var` операторы арқылы анықталуы тиіс, мысалы: `var test1`

Мұнда `test1` айнымалысының типі әлі анықталмағандықтан, ол айнымалыға белгілі бір мән меншіктеген кезде ғана айқындалады.

`Var` операторын айнымалыны инициализациялағанда да қолдануға болады. Мысалы `var test2=276`

Мұнда `test2` айнымалысының мәні өзгереді. Меншіктеу операторы программаның кез келген жерінде тұра береді және ол айнымалының тек мәнін ғана емес, типін де өзгерте алады. Меншіктеу операторы мынадай түрде беріледі: `a=b`

Мұндағы `a`-біз бір мән бергіміз келетін айнымалы, `b`- айнымалының жаңа мәнін анықтайтын өрнек. `var n=3725`

`var x=2.75 var p=true var="Программа орындалды"`

`n` және `x` айнымалыларының типі `number`, `p` айнымалысының типі-логикалық, `s`-айнымалысының типі `string`. JavaScript тілінде барлық стандартты және тұтынушының өзі анықтайтын функциялар үшін `object` типі бар. `Object` типті айнымалылар объектілерді сақтайтындықтан, оларды жай объектілер деп атай береді.

Өрнектер

Өрнектер операциялар (амалдар) таңбаларымен біріктірілген, литералдарда, айнымалылардан және жақшалардан тұрады. Өрнектерді есептеу нәтижесінде сан (бүтін не нақты), сөз тіркесі немесе логикалық типте болатын бір ғана мән шығады. Өрнектерде қолданылатын айнымалылар оған дейін инициалданған болуы керек. Өрнектерді есептеу кезінде анықталмаған немесе инициалданбаған айнымалылар кездесе қате шығады. JavaScript мәнді белгілеу үшін `null` мәнін меншіктесе, ол инициалданған болып саналады.

Өрнек операндтардан (мәндер мен айнымалылардан) және операциялар таңбаларынан (+,-,*,/) тұрады. Мысалы, `a*b` формуласында `a` және `b` операндтар, `*` таңбасы –көбейту операциясы.

Операциялар унарлық (бірорындық) немесе бинарлық (екіорындық) болып екіге бөлінеді. Өрнек `+A` түрінде жазылса. Ондағы `+` унарлық операцияны белгілейді, ал егер ол `A+B` түрінде берілсе, ондағы `+` бинарлық операцияның орындалатындығын көрсетеді. `+A` өрнегін есептеу `A` операндының мәнін есептеп, оған `+` операциясын ұолдану дегенді білдіреді. Ал `A+B` өрнегін есептеу төмендегі әрекеттерден тұрады.

1. `A` және `B` мәндері есептеледі

2. Операция 1-ші қадамда табылған операндтар мәндеріне қолданылады, `h`.

Операторлар `X+=Y=>X=X+Y`

`X-=Y=>X=X-Y` `X*=Y=>X=X*Y` `X/=Y=>X=X/Y` `X%=Y=>X=X%Y`

Қатынас операциялары кез-келген типтегі операндқа қолданыла береді. Келесі мысалда:

`var x=1;`

```
var y;  
y = (x += 2) + 1;
```

у айнымалысының мәні 4, ал x айнымалысының мәні-3. Мынадай тізбекті түрлегі меншіктеулерді де пайдалануға болады. $x = y = z = t =$ өрнек;

24.JavaScript тілінде меншіктеу операторларының түрлерін пайдалану.

Операциялар унарлық (бірорындық) немесе бинарлық (екіорындық) болып екіге бөлінеді. Өрнек +A түрінде жазылса. Ондағы +- унарлық операцияны белгілейді, ал егер ол A+B түрінде берілсе, ондағы +- бинарлық операцияның орындалатындығын көрсетеді. +A өрнегін есептеу A операндының мәнін есептеп, оған + операциясын қолдану дегенді білдіреді. Ал A+B өрнегін есептеу төмендегі әрекеттерден тұрады.

1. A және B мәндері есептеледі

2. Операция 1-ші қадамда табылған операндтар мәндерінеіқолданылады,h.

Операторлар $X+=Y \Rightarrow X=X+Y$

$X-=Y \Rightarrow X=X-Y$ $X*=Y \Rightarrow X=X*Y$ $X/=Y \Rightarrow X=X/Y$ $X\%=Y \Rightarrow X=X\%Y$

Қатынас операциялары кез-келген типтегі операндқа қолданыла береді. Келесі мысалда:

```
var x=1;  
var y;  
y = (x += 2) + 1;
```

у айнымалысының мәні 4, ал x айнымалысының мәні-3. Мынадай тізбекті түрлегі меншіктеулерді де пайдалануға болады. $x = y = z = t =$ өрнек;

Мұнда бірнеше айнымалының бәріне бір ғана мән меншіктеледі.

«++» және «--»операциялары тек айнымалыларға тіркеледі, оларды өрнектерге қосып жазуға болмайды.

Мынадай командалардың ++x және x++ (--x және x--) айырмашылығы бұлар басқа командалар құрамына кіріп тұрғанда ғана есепке алынады. Алғашқысында (олар айнымалы алдында) операция айнымалыны пайдаланғанға дейін орындалады, ал соңғысында –пайдаланғанға дейін орындалады, ал соңғысында –пайдаланғаннан кейін орындалады.

x=5; меншіктеулерінен кейін:

y=++x; x және y-тің мәні бға тең.

x=5; Ал мына меншіктеулерден кейін:

y=x++; x-тің мәні 5-ке, ал y-тің мәні 6-ға тең болады.

Операциялардың бұлай жазылуы соңғы кезде шығып жатқан, C/C++ тілдерінен алынған программалау синтаксисіне негізделген. Олар айнымалыларды бірге арттыру және бірге кеміту сияқты операцияларды қысқаша жазу үшін қолданылады.

Операция нәтижесі –салыстыру мәндері дұрыс болса, логикалық true,ал оған кері жағдайды- false мәні болып табылады.

< (кіші) <= (кіші немесе тең-артық емес) == (тең) != (тең емес)

>= (үлкен немесе тең-кіші емес) > (үлкен)

! логикалық емес операциясы логикалық типтегі операндтарға қолданылады, егер операнд мәні true болса, онда a!=true болады. && (логикалық және) пен || (логикалық немесе) операцияларының қолдану ережесі:

A&&B өрнегінің мәні –екі операнд мәндері де ақиқат болғанда ақиқат, қалған жағдайларда жалған болады. A||B өрнегінің мәні –кем дегенде, бір операнд ақиқат болса, ақиқат, ал басқа жағдайларда –жалған болып саналады.

Символдар тізбегінен тұратын тіркестік мәндер үшін символдарды конкатенциялау операциясы анықталған. Операция «қосу» таңбасымен белгіленеді. Бұл операцияның екі операндтар тізбегінен құралатын жаңа сөз тіркесі болады,

төмендегі меншіктеу операторларының орындалуы нәтижесінде: st = “озат ”
”+”студент”

St айнымалысы «озат студент» мәнін меншіктейді.

.кестеде орындалу басымдығы кему реті бойынша орналасқан операциялар тізімі қарастырылған.

Инкремент

++

Декремент

--

Терістеу

!

Унарлы минус

-

Көбейту

*

Бөлу,қалдық табу

/,%

Қосу, азайту

+,.-,=

Салыстыру

<,>,<=,

Теңдік

==

Теңсіздік

!=

Логикалық және

&&

Логикалық немесе

||

Меншіктеу

+=,+=,-=,*=,/=,%=,! =

25.JavaScript тілінің стандартты константалары мен функцияларын пайдалану.

Олардың түрлері, жазылу форматтары.

Барлық стандартты математикалық функциялар мен жиі қолданылатын тұрақтылар Math-математикалық объектісінің тәсілдері мен қасиеттері түрінде беріледі.

Math объектісі JavaScript тілінің ішкі объектісі болып табылады. Мысалы, PI тұратысын пайдалану үшін Math.PI түрінде жазу керек. JavaScript тұрақтыларының дәлдігі үтірден кейінгі –цифрды қамтиды. Math функцияларын да тәсіл ретінде пайдаланып, мысалы, синус функциясы - Math.sin(argument), түрінде жазылады, мұндағы argument функция аргументі болып табылады.

Стандартты тұрақтылар

1. Натуралдық логарифм негізі –E болып жазылады.мысалы Alert(Math.E)
2. 10 санының натуралдық логарифмі- LN10. Мысалы. Alert(Math.LN10)
3. 2 санының натуралдық логарифмі- LN2. M: Alert(Math.LN2)

Мәліметтер типі, айнымалылар, өрнектер

Айнымалылар

Айнымалы мән берілген идентификатор. Айнымалыға бағдарлама арқылы оған берілген мәнмен жұмыс жасай аласыз.

JavaScript-те өзі сақталатын мәндердің түрі туралы ақпаратты қамтымайды. Бұл айнымалыға, мысалы, жолға жазу арқылы, кейін оған сан жазуға болады. Мұндай әрекет бағдарламада қате шақырмайды. Сондықтан JavaScript кейде "сипатталмаған" тіл деп аталады.

Кілт сөз *var*

ES2015 стандарты пайда болғанға дейін var кілт сөзін пайдалану айнымалыларды хабарландырудың жалғыз жолы болды.

```
var a = 0
```

Егер бұл конструкцияда var төмен болса, мәні жарияланбаған айнымалы болады. Бұл операцияның нәтижесі бағдарлама қандай режимде орындалатынына байланысты.

Егер қатаң режим (strict mode) қосылса, бұл қате тудырады. Егер қатаң режим қосылмаған болса, ауыспалы хабарландыру пайда болады және ол жаһандық объектіге тағайындалады. Атап айтқанда, бұл функция жұмысты аяқтағаннан кейін де қолжетімді болады. Әдетте функцияларда жарияланған айнымалылар олардың шегінен тыс "шықпайды"деп күтілуде. Бұл көрінеді:

```
function notVar() {  
  bNotVar = 1 // жақсы емес  
}  
notVar()  
console.log (bNotVar)
```

Консоль 1 түседі, мұндай мінез — құлықты әдетте ешкім күте алмайды, bNotVar = 1 өрнегі айнымалыны жариялау және инициализациялау әрекеті емес, көріну аймағының сыртқы функциясына қатысты айнымалыға жүгіну әрекеті ретінде көрінеді (бұл-қалыпты жағдай). Нәтижесінде, айнымалыларды анық жариялау кодты кім оқиды және күтпеген мінез-құлыққа әкелуі мүмкін түсініктен ауытқиды. Кейінірек айтамыз және функциялары туралы, мен туралы облыстарында көру, әзірге тырысыңыз әрқашан болған, мағынасы қазанның білдіру болып табылады жариялау ауыспалы, пайдалануға мамандандырылған негізгі деген сөздермен ауыстырылсын. Егер осы мысалда var bNotVar = 1 түрінде жазу функциясы болса, онда жоғарыда келтірілген код фрагментін іске қосу әрекеті қате туралы хабардың пайда болуына әкеледі (оны браузер консолінен көруге болады).

Мысалы, ол көрінуі мүмкін: Uncaught ReferenceError: bNotVar is not defined. Оның мағынасы бағдарлама жоқ айнымалымен жұмыс істей алмайды. Бағдарламаны бірінші рет іске қосқан кезде, күтпеген жерден өзін-өзі ұстай алатын түсініксіз кодты жазуға қарағанда, қате туралы хабарламаны көру әлдеқайда жақсы.

Айнымалы пайдаланбас бұрын, оны var немесе let кілт сөзін пайдаланып жариялау керек. Егер бұл тұрақты болса, const кілт сөзі қолданылады. Айнымалыны жариялау және оған осы кілт сөздерді пайдаланбай-ақ мән беруге болады, бірақ мұны істеу ұсынылмайды.

Егер айнымалы хабарланған кезде оны бастамашылық етпесе, оған қандай да бір мән бермесе, оған undefined мәні автоматты түрде беріледі.

```
var a // typeof a == 'undefined'
```

Var кілт сөзін пайдаланып жарияланған айнымалылар оларға жаңа мәндерді тағайындау арқылы қайта жариялауға болады (бірақ ол кодты оқитын адамды шатастыруы мүмкін).

```
var a = 1
```

```
var a = 2
```

Бір көріністе бірнеше айнымалыны жариялауға болады:

```
var a = 1, b = 2
```

Облыс көріну айнымалы (scope) деп атайды учаскесі бағдарламасын, онда қол жетімді (видима) бұл айнымалы.

Кез келген функциядан тыс var кілт сөзі арқылы іске асырылған айнымалы, жаһандық объектіге тағайындалады. Ол жаһандық көріну аймағы бар және бағдарламаның кез келген жерінен қол жетімді. Егер айнымалы функцияның ішінде var кілт сөзін пайдаланып жарияланса, ол жергілікті айнымалы болып табылатын осы функцияның ішінде ғана көрінеді.

Егер var пайдалану функциясында аты жаһандық көріну аймағындағы айнымалы есімімен сәйкес келетін айнымалы болса, ол жаһандық айнымалыны "ауыстырады". Яғни, функцияның ішінде мұндай айнымалыны қолданған кезде оның жергілікті нұсқасы қолданылады.

Блоктар (фигуралық жақшаларға салынған код аумағы) көрінудің жаңа салаларын құрмайтынын түсіну маңызды. Функцияны шақырғанда жаңа көріну аймағы жасалады. Var негізгі сөзі блокты емес, көрінудің функционалдық аймағы деп аталады.

Егер функцияның кодында айнымалы болса, ол функцияның бүкіл кодынан көрінеді. Егер айнымалы функция кодының соңында var арқылы жарияланған болса да, кодтың басында да оған жүгінуге болады, өйткені JavaScript айнымалы көтеру механизмі жұмыс істейді (hoisting). Бұл механизм айнымалы хабарландыруларды "көтереді", бірақ оларды инициализациялау операциялары емес. Бұл шатастырудың көзі болуы мүмкін, сондықтан функцияның басында айнымалыларды жариялау ережесін алыңыз.

Кілт сөз *let*

Let негізгі сөзі ES2015-да пайда болды, оны "блокты" var нұсқасы деп атауға болады. Let кілт сөзінің көмегімен жарияланған айнымалылардың көріну аймағы ол жарияланған блоктармен, оператормен немесе өрнектермен, сондай-ақ салынған блоктармен шектеледі.

Егер "let" сөзі өте түсінікті болып көрінсе, оның орнына "болсын"сөзі қолданылады. Сонда let color = 'red' өрнегін ағылшын тіліне аударуға болады: "let the color be red", ал орыс тіліне: "түс қызыл болсын".

Let кілт сөзін қолданғанда var негізгі сөзіне сәйкес келетін бір мағынадан құтылуға болады (мысалы, let арқылы екі рет бірдей айнымалыны жариялау мүмкін емес). Let функциядан тыс пайдалану, айталық, циклдарды инициализациялау кезінде, жаһандық айнымалыларды құруға әкелмейді.

Мысалы, мұндай код қате тудырады:

```
for (let i = 0; i < 5; i++) {
```

```
    console.log (i)
```

```
}
```

```
console.log (i)
```

Егер циклді инициализациялау кезінде, i есептегіші var кілт сөзін пайдалану арқылы жарияланса, онда i ол жұмысты аяқтағаннан кейін циклдан тыс қол жетімді болады.

Қазіргі уақытта, қазіргі заманғы стандарттар негізінде JS-бағдарламаларды әзірлеу кезінде var-дан толық бас тартуға және тек let және const кілт сөздерін пайдалануға болады.

Кілт сөз **const**

Var немесе let кілт сөздерін пайдаланып жарияланған айнымалы мәндер қайта жазылуы мүмкін. Егер осы кілт сөздердің орнына const пайдаланылса, оның көмегімен жарияланған және инициалданған константаға жаңа мәнді беруге болмайды.

```
const a = 'test'
```

Осы мысалда a константасында жаңа мәнді беруге болмайды. Бірақ, егер a-бұл қарапайым мән емес, сандар сияқты, ал объект, const кілт сөзін пайдалану бұл нысанды өзгертуден қорғайды.

Айнымалы нысан жазылған деп айтады кезде, шын мәнінде, айнымалы объектіге сілтеме сақталады екенін білдіреді. Бұл сілтемені өзгерту мүмкін емес, сілтеме жасайтын нысанның өзі өзгерте аласыз.

Кілт сөз const объектілерді иммутабельді емес. Ол жай ғана тиісті константаларға жазылған сілтемелерді өзгертуден қорғайды. Бұл қалай көрінеді:

```
const obj = {}  
console.log(obj.a)  
obj.a = 1 // жұмыс істейді  
console.log(obj.a)  
//obj = 5 // қатені тудырады
```

Obj константасында, инициализация кезінде, жаңа бос нысан жазылады. Оның a қасиетіне әрекет ету әрекеті, жоқ, қателерді тудырмайды. Консольге undefined түседі. Содан кейін біз объектіге жаңа сипат қосып, қайтадан оған жүгінуге тырысамыз. Бұл жолы консольге бұл сипаттың мәні — 1. Егер мысалдың соңғы жолын қарастырсаңыз, осы кодты орындау әрекеті қатеге әкеледі.

Const кілт сөзі let-ке өте ұқсас, атап айтқанда, ол блокты көріну аймағына ие.

Қазіргі жағдайда мәні өзгертілмейтін барлық мәндерді жариялау үшін const негізгі сөзін қолдануға болады, тек ерекше жағдайларда ғана let-ке жүгінгенде. Неліктен? Барлық мәселе, бағдарламаны қиындатпай және қателіктерден аулақ болу үшін қол жетімді құрылымдардан мүмкіндігінше қарапайым пайдалануға ұмтылу керек.

Деректер түрлері

JavaScript кейде "сипатталмаған" тіл деп аталады, бірақ бұл істің нақты жағдайына сәйкес келмейді. Бірақ, JavaScript-те деректер түрлері бар. Атап айтқанда, бұл қарапайым және объективті деректер түрлері туралы болып отыр.

Деректер түрін анықтау үшін typeof операторын пайдалануға болады. Ол операндтың түрін көрсететін жолды қайтарады.

Деректердің қарапайым түрлері

Мұнда қарапайым JavaScript деректер түрлерінің тізімі:

number (саны)

string (жол)

boolean (логикалық мәні)

null (null арнайы мәні)

undefined (арнайы мағынасы undefined)

symbol (ерекше жағдайларда қолданылатын таңба ES6-да пайда болды)

Мұнда деректер түрлерінің атаулары оларды typeof операторы қайтарады.

Осы тізімдегі ең жиі қолданылатын деректер түрлері туралы сөйлесейік.

Number түрі

JavaScript-те number типінің мәндері 64-биттік сандар түрінде екі есе дәлдікпен өзгермелі үтірмен берілген.

Кодта сандық литералдар бүтін және Бөлшек сандар түрінде ондық санау жүйесінде берілген. Сандарды жазу үшін басқа жолдарды пайдалануға болады. Мысалы, егер сандық Литер басында 0x префиксі бар болса-ол он алтылық санау жүйесінде жазылған сан ретінде қабылданады. Сандарды экспоненциалды көріністе де жазуға болады (мұндай сандарға E әрпін табуға болады).

Мұнда бүтін сандарды жазу мысалдары:

10

5354576767321

0xсс // он алтылық Сан

Міне, Бөлшек сандар.

3.14

.1234

5. 2e4 //5.2 * 10⁴

Сандық литералдар (мұндай мінез-құлық кейбір басқа примитивті типтерге де тән), оларға объектілер ретінде қарауға әрекет жасаған кезде операцияны орындау уақытында автоматты түрде "Объектілік орама"деп аталатын тиісті объектілерге айналады. Бұл жағдайда number Объектілік орамасы туралы.

Мұнда, мысалы, Google Chrome консолиінде объект сияқты сандық Литер жазылған а айнымалысына жүгіну әрекеті қалай көрінеді.

String түрі

String түріндегі мәндер символдардың кезектілігі болып табылады. Мұндай мәндер бір немесе екі еселенген тырнақшаға жасалған жол литералдары түрінде беріледі.

'A string'

"Another string"

Жол мәндерін бірнеше бөлікке бөлуге болады, кері қиғаш сызық (backslash).

"A \
string"

Жол консольге жол шығарған кезде түсіндірілетін escape тізбектері деп аталады. Мысалы, \n тізбегі жолды аудару белгісін білдіреді. Кері қиғаш сызық таңбасын сол тырнақшаға жасалған жолдарға тырнақшаларды қосу үшін де пайдалануға болады. Тырнақшаның таңбасын экрандау / жүйе оны арнайы таңба ретінде қабылдамайды .

'I\' m a developer'

Жолдарды + операторымен байланыстыруға болады.

"A" + " string"

Үлгі литерлері

ES2015-те үлгі немесе үлгі жолдары деп аталады. Олар кері тырнақшаға (`) жасалған жолдар болып табылады және кейбір қызықты қасиеттерге ие.

`a string`

Мысалы, шаблонды литерлерге JavaScript-өрнектерді есептеу нәтижесі болып табылатын мәндерді орнатуға болады.

`a string with \$ {something}`

'a string with \$ {something+somethingElse}'

'a string with \${obj.something ()}'

Кері тырнақшаларды пайдалану жол литерлерін көп жолды жазуды жеңілдетеді:

'a string

with

`{something}``

Boolean түрі

JavaScript — те логикалық мәндермен жұмыс істеу кезінде пайдаланылатын резервтелген сөздердің жұбы бар-бұл шын (шындық), және false (өтірік). Салыстыру операциялары, мысалы, ==, ===, <, >, true немесе false қайтарады.

Логикалық өрнектер if және while сияқты конструкцияларда бағдарламаның орындалу барысын басқаруға көмектеседі.

Сонымен қатар, true немесе false мәні күтілетін жерде тіл арқылы шынайы (truthy) немесе жалған (falsy) деп автоматты түрде бағаланатын басқа да мәндерді қолдануға болады.

Атап айтқанда, жалған мәндер::

0

-0

NaN

undefined

null

//бос жол

Қалған мәндер шынайы.

Null түрі

JavaScript-те мәні жоқ екенін көрсететін арнайы null мәні бар. Мұндай мәндер басқа тілдерде қолданылады.

Undefined түрі

Кейбір айнымалыға жазылған undefined мәні бұл айнымалының бастамалмағанын және ол үшін мәні жоқ екенін көрсетеді.

Бұл мән автоматты түрде return кілт сөзін пайдалана отырып, нәтижесі анық қайтарылмайтын функциялардан қайтарылады. Егер функция қоңырауда көрсетілмеген параметрді қабылдаса, ол undefined бағдарламасында да орнатылады.

Мәнді undefined тексеру үшін келесі құрылымды пайдалануға болады.

```
typeof variable == 'undefined'
```

JavaScript-тегі арифметикалық операторлар

Арифметикалық өрнектер

Бұл санатқа есептеу нәтижесі сандар болып табылатын өрнектер жатады.

```
1 / 2
```

```
i++
```

```
i - = 2
```

```
i * 2
```

Параметрмен, жолдық өрнектер

Мұндай өрнектерді есептеу нәтижесі жолдар болып табылады.

```
'A' + 'string'
```

```
'A' += 'string'
```

Бастапқы өрнектер

Бұл санатқа литералдар, константтар, идентификаторларға сілтемелер кіреді.

```
2
```

```
0.02
```

```
'something'
```

```
true
```

```
false
```

```
this/ / орындалу контексті, ағымдағы нысанға сілтеме
```

```

undefined
I // қайда i айнымалы немесе тұрақты
Мұнда JavaScript кейбір кілт сөздер мен дизайндарды жатқызуға болады.
function
class
function* // генераторы
yield // генератордың жұмысын тоқтату/жаңарту командасы
yield * // басқа итераторға немесе генераторға жіберу
async function* // асинхронды функционалдық өрнек
await // асинхронды функцияны орындауды күтуді ұйымдастыру
/ pattern / i // тұрақты өрнек
() // топтастыру
Массивтер мен объектілерді инициализациялау өрнегі
[] //массива литері
{} // объектілі Литер
[1,2,3]
{a: 1, b: 2}
{A: {b: 1}}

```

Логикалық өрнектер

Логикалық өрнектерде логикалық операторлар қолданылады, оларды есептеу нәтижесі логикалық мәндер болып табылады.

```

a & b
a // b
!a

```

Қасиеттерге қол жеткізу өрнектері

Бұл өрнектер объектілердің қасиеттері мен әдістерін қолдануға мүмкіндік береді.

```

object.property // объектінің сипаттарына (немесе әдісіне) қарау
object[property]
object ['property']

```

Объектілерді құру өрнектері

```

new object()
new a(1)
new MyRectangle ('name', 2, {a: 4})

```

Функцияларды жариялау өрнектері

```

function() {}
function (a, b) { return a * b }
(a, b) => a * b
a => a * 2
() => { return 2 }

```

Шақыру өрнегі

Мұндай өрнектер элементтерді немесе нысандарды шақыру үшін қолданылады.

```
a. x(2)
window.resize()
```

Объектілермен жұмыс

Жоғарыда біз объектілі литералдар, олардың әдістерін шақыру, олардың қасиеттеріне қол жеткізу туралы айта отырып, объектілермен бетпе-бет келдік. Мұнда біз объектілер туралы егжей-тегжейлі әңгіме, атап айтқанда, прототипті мұра тетігін және class кілт сөзін пайдалану қарастырайық.

Прототипті тұқым қуалау

JavaScript қазіргі заманғы бағдарламалау тілдері арасында прототипті мұраны қолдайды. Көптеген объектілі-бағытталған тілдер сыныптарға негізделген мұра моделін пайдаланады.

Әрбір JavaScript-объектіде оның прототипі болып табылатын басқа объектіні көрсететін ерекше сипат (`__proto__`) бар. Объект прототиптің қасиеттері мен әдістерін зерттейді.

Мысалы, бізде объектілі Литер көмегімен құрылған нысан бар.

```
const car = {}
```

Немесе Object конструкторы арқылы нысанды құрдық.

```
const car = new Object()
```

Осы жағдайлардың кез келгенінде Car прототипі Object болады.prototype.

Егер объект болып табылатын массив құру болса, оның прототипі Array нысаны болады.prototype.

```
const list = []
// немесе
const list = new Array()
```

Мұны келесідей тексеруге болады.

```
car.__proto__ = Object.prototype /true
car.__proto__ === new Object().__proto__ // true
list.__proto__ = Object.prototype /false
list.__proto__ = Array.prototype /true
list.__proto__ === new Array().__proto__ // true
```

Мұнда біз __ proto __ қасиеттерін пайдаландық, ол міндетті түрде әзірлеушіге қол жетімді болмауы керек, бірақ әдетте оған жүгінуге болады. Объект прототипін алудың сенімді тәсілі Object жаһандық объектісінің getPrototypeOf() әдісін пайдалану болып табылады.

```
Object.getPrototypeOf(new Object())
```

Прототиптің барлық қасиеттері мен әдістері осы прототипі бар нысанға қол жетімді. Міне, мысалы, массив үшін олардың тізімі қалай көрінеді.

Барлық нысандар үшін негізгі прототип Object болып табылады.prototype.

```
Array.prototype.__proto__ = Object.prototype
```

Object Бар.prototype прототип жоқ.

Жоғарыда көргеніміз прототип тізбегінің үлгісі болып табылады.

Объектінің қасиетіне немесе әдісіне қарау әрекеті кезінде, егер мұндай сипат немесе әдіс объектінің өзінде болмаса, оларды іздеу оның прототипінде, содан кейін — прототип прототипінде және сол сияқты — ізделгенге дейін немесе прототиптердің тізбегі аяқталғанша орындалады.

Жаңа операторды пайдалану арқылы нысандарды жасау және объект литералдарын немесе массивтердің литералдарын қолданудан басқа, нысан данасын Object әдісі арқылы жасауға болады.create (). Бұл әдіске берілетін бірінші дәлел оның көмегімен жасалатын объектінің прототипі болатын объект болып табылады.

```
const car = Object.create(Object.prototype)
```

Қандай да бір объект басқа нысанның прототипінің тізбегіне кіретінін isPrototypeOf () әдісін пайдалана отырып тексеруге болады.

```
const list = []
Array.prototype.isPrototypeOf(list)
```

Функция және әдістер ұғымы.

Конструкторлар-функциялары

Жоғарыда біз тілде бар конструкторларды (оларды шақырғанда new кілт сөзі пайдаланылады) пайдалана отырып, жаңа нысандарды құрдық. Мұндай функцияларды дербес жасауға болады. Мысал қарастырайық.

```
function Person (name) {
  this.name = name
}

Person.prototype.hello = function() {
  console.log(this.name)
}
```

```
let person = new Person ('Flavio')
person.hello()

console.log(Person.prototype.isPrototypeOf(person))
```

Мұнда біз конструктор функциясын жасаймыз. Оны шақырған кезде жаңа нысан құрылады, ол конструктор денесіндегі `this` кілт сөзін көрсетеді. Біз бұл нысанға `name` қасиетін қосып, оған конструкторға берілгенін жазамыз. Бұл нысан автоматты түрде конструктордан қайтарылады. Конструктор функциясының көмегімен, `name` сипаттары конструктор жасаған кезде берілген элементтерді қамтитын көптеген нысандарды жасауға болады.

Конструктор жасағаннан кейін, біз оның прототипіне консольге осы функцияны пайдаланып жасалған нысанның `name` сипатының мәнін шығаратын функцияны қосамыз. Осы конструктордың көмегімен жасалған барлық объектілер бірдей прототипке ие болады, яғни `hello ()` бір функциясын пайдаланады. Бұл `Person` типті тағы бір нысанды жасап, оның `hello()` функциясын мысалда бар нысан функциясымен салыстыра отырып, тексеру қиын емес (бұл жағдайда функцияның аты жақшасыз жазылады).

І Сыныптар

JavaScript-те ES6 стандартында "класс" деген ұғым келді.

Осыған дейін JavaScript-те прототипті тұқым қуалаудың жоғарыда сипатталған механизммен ғана пайдалануға болады. Бұл механизм басқа тілдерден JS келген бағдарламашылар үшін ерекше көрінді. Сондықтан тілде және тұқым қуалаудың прототипті механизмі үшін "синтаксистік қант" болып табылатын сыныптар пайда болды. Яғни, дәстүрлі тәсілмен құрылған объектілер мен кластарды пайдалану арқылы құрылған объектілер прототиптерге ие.

Сынып хабарламасы

Сынып хабарландыруы қалай көрінеді.

```
class Person {
  constructor (name) {
    this.name = name
  }
  hello() {
    return 'Hello, I am' + this.name + '!'
  }
}
```

Класты жаңа `ClassIdentifier ()` конструкциясын қолдана отырып жаңа нысандарды жасау үшін пайдалануға болатын идентификатор бар.

Жаңа объектіні құру кезінде `constructor` әдісі шақырылады, оған параметрлер беріледі.

Сыныпта әдістерді жариялауға болады. Біздің жағдайда hello() — класс негізінде жасалған барлық нысандарды тудыруы мүмкін әдіс. Мұнда Person класын пайдаланып жаңа нысанды жасау сияқты көрінеді.

```
const flavio = new Person ('Flavio')
```

```
flavio.hello()
```

Класстарға негізделген мұрагерлік

Класстар басқа сыныптарды кеңейте алады. Мұндай сыныптар негізінде құрылған Нысандар кеңейтілген сыныпта берілген әдістер мен бастапқы кластарға ие болады.

Егер басқа классқа кеңейтетін сынып (осы сыныптың мұрагері) ата-ананың сыныбындағы аты бар әдіске сәйкес келетін болса, бұл әдіс бастапқы әдіс алдында артықшылығы бар.

```
бағдарламашы классытұлғаны кеңейтеді {
```

```
  Сәлем() {
```

```
    қайтару супер.Сәлем () + ' мен бағдарламашы.'
```

```
  }
```

```
}
```

```
const flavio = жаңа бағдарламашы ('Flavio')
```

```
Флавио.Сәлем()
```

Шақыру кезінде әдісін құрметті() қр вышеприведенном мысалында жол қайтарылады Здравствуйте, меня зовут Флавио. Мен бағдарламашы.

Класстарда ауыспалы (қасиеттер) бар болуы қарастырылмаған, кластар көмегімен жасалатын объектілердің қасиеттері конструкторда теңшеу керек.

Класс ішінде ата-ана класына кілт сөз super.

Статикалық әдістер

Класста сипатталатын әдістерді осы сынып негізінде құрылған объектілерге қарап, бірақ сынып өзіне емес, атауға болады. Статикалық (статика) әдістерді класқа тікелей қаратып айтуға болады.

Жеке Әдістер

JavaScript тілінде жеке (жеке, жабық) әдістерді жариялауға мүмкіндік беретін кіріктірілген тетік жоқ. Бұл шектеу, мысалы, тұйықталу арқылы айналып өтуге болады.

Геттерлер мен сеттерлер

Класста әдістерді негізгі сөздермен алдын ала сипаттауға немесе жиынтықты жасауға болады. Бұл класс негізінде құрылған объектілердің қасиеттеріне қатынауды басқару үшін пайдаланылатын геттерлер мен сеттерлер-функцияларды жасауға мүмкіндік береді. Геттер псевдо-сипат мәнін оқуға әрекеттенгенде, ал сеттер — оған жаңа мәнді жазуға әрекеттенгенде шақырылады.

```
person класы {
```

```
  конструктор (аты) {
```

```
    бұл.пайдаланушы аты = аты
```

```
  }
```

```
  теру аты (мәні) {
```

```
    бұл.пайдаланушы аты = мән
```

```
  }
```

```
  аты алу() {
```

```
    оны қайтарыңыз.пайдаланушы аты
```

```
  }
```

```
}
```


Объектілер және қасиеттер.

Тілдің құрамындағы объектілер. Олардың қасиеттері мен тәсілдерін пайдалану. Date объектісі Объект — бұл мәліметтер мен функциялар жиынынан тұратын бірыңғай конструкция немесе, JavaScript терминологиясында, қасиеттер мен тәсілдер жиыны. JavaScript тілінің ішкі объектілерін қарастыруды өте пайдалы болып саналатын Date объектісінен бастайық. Бұл объект күн-ай мерзімімен (дата) және уақытпен жұмыс істеу үшін керек. Объект экземплярын жасау үшін (Date объектісінің ғана емес, одан басқасының да) JavaScript тілінде new түйінді сөзі қолданылады: `var now = new Date();` Енді now айнымалысы Date объектісі экземпляры болып табылады да, ол үстіміздегі дата мен уақытты береді. Жалпы экземпляр жасау былай орындалады: `var айнымалы = new Date(параметрлер);`

Мысал:

```
var now = new Date();
var birthday = new Date(1954,1,8);
var bell = new Date(2003, 0,14,12,20,0);
```

Date объектісі экземпляры құрылғаннан кейін, оның ішкі мәліметтерін көруге болады, оны өзгерту мүмкіндігі де бар. Ол үшін көптеген тәсілдер бар, олардың тізімі тілге арналған кітап қосымшаларында келтіріледі. Объект тәсілі атауы (тәсіл – JavaScript терминологиясында функция) экземпляр атауынан нүктемен бөлініп жазылып тұрады. Былай жазуға болады: `var year = bell.getYear();` year айнымалысы мәні 2003 болады.

Date объектісінің бірнеше қарапайым скриптерін қарастырайық. Ағымдағы күн және уақыт 3) `var now = new Date();`

```
alert("Бүгін:"+now.getDate()+"'"+(now.getMonth()+1)+"'"+ now.getYear()+"\n
```

```
Қазір: "+now.getHours() +"':"+ now.getMinutes());
```

Осы кодтарды жазған кезде скрипті орындау мынадай хабарлама шығуына себепші болады. Date ішкі объектісі және оның тәсілдері 3) Тәсіл түрі – `setYear()`. Жыл нөмірін тағайындайды.

Мысалы:

```
var d = new Date();
d.setYear(2004);
alert(d.getYear());
```

3) Тәсіл түрі – `getMonth()`. Ай нөмірі мәнін береді.

```
Var d = new Date(2005,1,2);
var m = d.getMonth();
alert(m);
```

Объект — бұл мәліметтер мен функциялар жиынынан тұратын бірыңғай конструкция немесе, JavaScript терминологиясында, қасиеттер мен тәсілдер жиыны. Функция = тәсіл (метод). Айнымалы = қасиет (свойства). Инкапсуляция термині «қара жәшік» ретінде қарастырылатын объектінің ішкі құрылымын жасыру деген сөз. Объектінің қасиеттері белгілі болып саналады, яғни олар – сырттан қол жеткізуге болатын айнымалылар. Бірақ бұл функциялар қалай құрылған, олар қандай алгоритммен жұмыс істейді, ол туралы программалаушыға айтылмайды. Программалаушы немесе объектіні тұтынушы адам объектінің қосымша ішкі функциялары мен айнымалылары бар ма, олар қол жеткізуге болатын қасиеттер мен тәсілдермен қалай байланысқан, ол жағын білмейді. JavaScript жаңа объектілерді программалауға және олардың ішкі объектілерін өзгертуге мүмкіндік береді. Енді бірнеше ішкі объектілер жұмысын қарастырайық. JavaScript тілінің ішкі объектілерін қарастыруды өте пайдалы болып саналатын Date объектісінен

бастайық. Бұл объект күн-ай мерзімімен (датамен) және уақытпен жұмыс істеу үшін керек. Объект экземплярын жасау үшін (Date объектісінің ғана емес, одан басқасының да) JavaScript тілінде new түйінді сөзі қолданылады: var now = new Date(); Енді now айнымалысы Date объектісі экземпляры болып табылады

2) Array объектісі Бұл объект мәліметтер жиымын (массивін) жасау үшін керек. Массив — элементтердің реттелген жиыны. Жеке элементті көрсету оның атауы мен индексін (нөмір) көрсету арқылы орындалады. JavaScript тілінде элементтерді нөмірлеу нөлден басталады. Массивті үш түрлі тәсілмен құруға болады: 1-тәсіл.

```
Item = new Array();
item[0] = "жексенбі";
item[1] = "дүйсенбі";
item[2] = "сейсенбі";
item[3] = "сәрсенбі";
```

2-тәсіл.

```
Var dayNames = new Array(«жексенбі», «дүйсенбі», «сейсенбі», «сәрсенбі»);
```

3-тәсіл.

```
Var item = [«жексенбі», «дүйсенбі», «сейсенбі», «сәрсенбі»];
```

Жиымның жеке элементтерін пайдалану үшін былай жазылады: массив_атауы [индекс]

Төмендегі скрипт:

```
var dayNames = new Array(«жексенбі», «дүйсенбі», «сейсенбі», «сәрсенбі», «бейсенбі», «жұма», «сенбі»);
```

```
alert(dayNames[0]);
```

 жұмысы нәтижесінде alert терезесіне «жексенбі» мәтіні шығады.

Кесте 3. Array объектісінің әдістері: Array әдісі Сипаттамасы concat Екі жиымды бір жиымға біріктіреді.

Мысалы: a = new Array(0,1,2,3,4);

//дайын мәндері бар b = new Array(5,6,7,8,9);

//жиым құрады c = a.concat(b);

join Жиымның барлық элементтерін бір жолға біріктіреді. Мысалы: a = new Array(0,1,2,3,4); b = a.join("-");

/"0-1-2-3-4" жолдарын қайтарады.

pop Соңғы элементті жиымнан жояды және осы элементті қайтарады.

push Жиым соңына бір не одан көп элементті қосады және соңғы қосылған элементті қайтарады. reverse Жиымды нөлдік элемент соңғы болатындай және т.б етіп төңкереді.

shift Жиымның нөлдік элементін жояды және оны қайтарады.

slice Жиымның бір бөлігін ерекшелейді. Синтаксисі: slice(start, [end]).

Мысалы:

```
newArray = myArray.slice(0, -1);
```

splice Жиымға элементтер қосады және ондағыны жояды.

sort Жиым элементтерін сұрыптайды. toString Жиымның жолдық берілу түрін қайтарады.

unshift Жиымның бас жағына бір не бірнеше элементтерді қосады.

3) Math объектісі Math математикалық классы — математик-программисттерге тригонометриялық және логарифмдік функциялар жиынын береді. Яғни ол тұрақтылар мен әдістерді қамтитын класс. Олардың шақырылу түрі: Math.константа Math.функция(i..) Кесте 4. Math класының тұрақтылары: Тұрақтылар p Түсініктемесі E LN10 LN2 LOG10E LOG2E PI SQRT1_2 SQRT2 E саны (натурал логарифмдерге негізделген) Натурал логарифм (ln10) Натурал логарифм (ln2) 10

негізді е логарифмі (log10e) 2 негізді е логарифмі (log2e) Р тұрақтысы ("пи" саны) 0,5-тің квадрат түбірін есептейді (1/2 саны) 2-нің квадрат түбірін есептейді 4) Window объектісі Window иерархияның төбесінде орналасады. Бұл қалған объектілердің одан төмен орналасып, соның қасиеттері түрінде қарастырылатынын білдіреді. Window объектісінде браузер құжатты көрсетіп тұрған терезенің қасиеттері мен тәсілдері де болады. Үш тәсіл бізге бұған дейін белгілі болған, олар: alert, prompt және confirm. Window иерархияның төбесінде орналасады. Бұл қалған объектілердің одан төмен орналасып, соның қасиеттері түрінде қарастырылатынын білдіреді. Window объектісінде браузер құжатты көрсетіп тұрған терезенің қасиеттері мен тәсілдері де болады. Үш тәсіл бізге бұған дейін белгілі болған, олар: alert, prompt және confirm. Егер былай жазылса: Window.alert ("alert функциясы – бұл window объектісінің әдісі"); — онда браузер осы хабарламаны шығаратын қосалқы терезе ашады (суретті қ.) Бірақ бұрын былай жазатынбыз:

```
alert("alert функциясы – бұл window объектісінің әдісі ");
```

Басты объект болып саналатын window объектісінің атауын жазбаса да болады. Көбінесе тұтынушыға бір нәрсе көрсету үшін бұрынғы терезе ашық тұрғанда, жаңа терезе ашуға тура келеді. Жаңа терезе ашу үшін window объектісінің open тәсілін қолданамыз. Оны мына тәсілдердің бірімен аша аламыз: var айнымалы = open(); var айнымалы = open(файл); var айнымалы = open(файл,терезе_атауы); var айнымалы = open(файл, терезе_атауы, терезе_параметрлері); Мұндағы: айнымалы — айнымалы. Ашылатын терезеге нұсқауыш; файл — сөз тіркесі (строка). Ашылған терезеде көрсетілетін файл аты. Егер бұл параметр берілмесе, бос терезе ашылады; терезе_атауы — сөз тіркесі. Терезе аты сол терезеге мәлімет шығару үшін керек (window объектісінің name қасиеті мәні); терезе_параметрлері — сөз тіркесі. Терезе қасиеттерін сипаттау. Егер параметр берілмесе, келісім бойынша қасиеттер қолданылады. Параметрлер арасындағы үтірден соң, бос орын қоймаған дұрыс, браузерлер онсыз да жұмыс істей береді. Кесте 5. Терезе параметрлерін сипаттау

Параметр Мәні Сипаттамасы Width Сан Пиксель, терезе ені. Минимал мәні – 100 Height Сан Пиксель, терезе биіктігі. Минимал мәні – 100 scrollbars yes, no,1, 0 Айналдыру жолағын береді resizable yes, no,1, 0 Терезе көлемін өзгертуді көрсету үшін қажет Menubar yes, no,1, 0 Меню өрісі бейнелетінін көрсету үшін қажет Location yes, no,1, 0 Адрес енгізу өрісі бейнелетінін көрсету үшін қажет Status yes, no,1, 0 Статус жолағы бейнелетінін көрсету үшін қажет Toolbar yes, no,1, 0 Батырмалар (саймандар) тақтасы бейнелетінін көрсету үшін қажет Мысал. Ит суреті бар dog01.jpg терезе ашу. <HTML>

```
<HEAD>
<TITLE>терезе</TITLE>
</HEAD>
<BODY bgcolor=white text=black>
<H2>Терезе</H2>
<HR>
<SCRIPT language=JavaScript>
<!-- var win = open("./pic/dog01.jpg"," ", "width = 320, height =
260" + "resizable=0, scrollbars=1" + "menubar=0, location = 1" +"status = 0, toolbar =
no");
//--> </SCRIPT> <P>
Негізгі мәтінге қайта өту үшін браузер тақтасындағы <EM>Артқа</EM>
батырмасын шертіңіз.
</BODY>
</HTML>
```

Терезеде көрсетілетін файл ретінде мыналарды көрсете аламыз: HTML-файл, мысалы, file1.htm; □ сурет файлы, мысалы, ./pic/fish.gif; мәтіндік файл, мысалы, 010401.txt. Close тәсілі ашық терезені жабады. Мұндағы close() немесе window.close() ағымдағы терезені жабады. Ал win.close() немесе window, win.close() — осы терезеден open тәсілі арқылы ашылған win нұсқауышы бар терезені жабады: var win = window.open(...); Терезенің өлшемі мен орнын анықтау. moveBy(X,Y) – терезені ағымдағы терезе орнына қарағанда санақ нүктесі сол жақ жоғарғы бұрыштан X шамаға (көлденеңінен, пикселмен) және Y шамаға (тігінен) жылжытады. moveTo(X,Y) – терезені санақ нүктесі сол жақ жоғарғы бұрыштан (X,Y) нүктесіне жылжытады. resizeBy(X,Y) – терезе өлшемін оның ағымдағы өлшеміне қарағанда X шамаға (көлденеңінен, пикселмен) және Y шамаға (тігінен) өзгертеді. resizeTo(X,Y) – терезенің ені X (пикселмен) және биіктігі Y болатындай абсолюттік өлшемін береді. scroll(X, Y) – терезенің жүйелік аумақтағы деректерін (X,Y) нүктесіне ауыстырады, яғни айналдыру жолақтарын X пен Y-ке ығыстырады. scrollBy(X,Y) - терезенің жүйелік аумақтағы деректерін ағымдағы орнына қарағанда көлденеңінен X-пикселге және тігінен Y-пикселге айналдырады. scrollTo(X,Y) - терезенің жүйелік аумақтағы деректерін (X,Y) нүктесіне айналдырады. 5) Document класы Document объектісі (ол document объектісі қасиеттері болып табылады) <html>...</html> блоктарында орналасқан тегтерді модельдейтін объектілер үшін қажет.

Document объектісінің кең тараған тәсілі — write тәсілі. Бұл тәсіл құжат терезесіне мәлімет жазу үшін керек. Write тәсілін қолдану алдында жазба ашу керек (open тәсілі), ал шығарған соң, оны жабу керек (close тәсілі). Open тәсілі форматы: айнымалы.document.open();

Мұндағы: айнымалы — бұл мәлімет жазылатын терезеге нұсқауыш. Write тәсілі форматы: document.write(жол);

немесе айнымалы.document.write(жол);

Мұндағы: айнымалы — бұл шығарылатын мәлімет көрсетілетін терезеге нұсқауыш. Егер айнымалы берілмесе, онда мәлімет ағымдағы терезеге жіберіледі;

жол — шығарылатын мәлімет көрсетілетін сөз тіркесі. document.close();

немесе айнымалы.document.close();

айнымалы — жазба шығарылмай жабылатын терезеге нұсқауыш. document.open тәсілін қолдану міндетті емес, өйткені document.write тәсілі жабылған құжатқа мәлімет жазу кезінде оның ішіндегі ескі мәліметті өшіріп, оны жаңа жазба үшін ашады. Window объектісінің document өрісіндегі Document класының бір түрі. Шақырылуы: терезе атауы.document немесе жай document (ағымды терезе үшін) Кесте 6. Window объектісінің әдістері: Әдістер Мағынасы document.clear()

document.close()

document.open ()

document.writeln(...) document.write(1мән,2мән,...)

W атаулы терезені немесе фреймді тазалау. Буфердан қалған ақпаратты экранға шығарады және құжаттан шығу жолын жабады. write(...) операторы көмегімен экранға мәтін жазу үшін, құжатқа кіру жолын ашады. Құжатқа қалған мәндерді жазу. write(...) ұқсас, бірақ жаңа жолға өту символын кірістіру арқылы.

JavaScript-тегі құжаттың объектілік моделі.

JavaScript Объектілік моделі

JavaScript қарапайым объектілі-бағытталған мысалға негізделген. Объект-бұл JavaScript айнмалысы болып табылатын қасиеттері бар конструкция. Қасиеттері басқа объектілер болуы мүмкін. Объектімен байланысты функциялар объект әдісі ретінде белгілі.

Navigator client және LiveWire server құрылған нысандарға қосымша, сіз өз объектілерін анықтай аласыз.

Объектілер және қасиеттері

Функциялары мен әдістері

Жаңа Нысандарды Құру

Объектілер және қасиеттері

JavaScript объектісі онымен байланысты қасиеттерге ие. Сіз нысанның сипаттарына келесі қарапайым белгілеу жүйесімен жүгінесіз:

```
objectName.propertyName
```

Объектінің аты мен сипаттың аты регистрге сезімтал. Сіз мәнін қосып, қасиеттерді анықтайсыз. Мысалы, myCar атымен объект бар болсын (біз объектілерді кейінірек қалай құру туралы талқылаймыз - енді, тек объект бар екенін қабылдаймыз). Сіз make, model және year деп аталатын қасиеттерді бере аласыз:

```
myCar.make = "Ford"
myCar.model = "Mustang"
myCar.year = 69;
```

Сіз сондай-ақ кестені белгілеу жүйесін пайдалана отырып, осы қасиеттерге қарай аласыз:

```
mycar ["make"] = "Ford"
myCar ["model"] = "Mustang"
myCar ["year"] = 69;
```

Кестенің бұл түрі ассоциативті кесте ретінде белгілі, себебі әр индекс элементі жолдың мәнімен байланысты. Бұл қалай жасалғанын түсіндіру үшін, келесі функция элементті және нысанның атауын функцияның аргументтері ретінде өту кезінде объектінің қасиетін көрсетеді:

```
function show_props(obj, obj_name) {
    var result = ""
    for (var i in obj)
        result += obj_name + "." + i + " = " + obj[i] + " \n"
    return result;
}
```

Show_props(myCar, "myCar") функциясына жүгіну келесілерді қайтарады:

Мысалы, реттік сандар арқылы сипаттарды анықтай аласыз:

```
[0] = 34
temp[1] = 42
temp[2] = 56
```

Бұл тұжырымдар temp нысанының үш қасиеттерін жасайды және сіз бұл қасиеттерге temp[i] ретінде қарап шығуыңыз керек, мұнда i - 0 мен 2 арасындағы бүтін сан.

Функциялары мен әдістері

Функциялар-JavaScript-те іргелі кіріктірілген блоктардың бірі. Функция-JavaScript процедурасы-белгілі бір тапсырманы орындайтын тұжырымдар жиынтығы. Функцияны анықтау function негізгі сөзден тұрады

Функция Атауы

Дөңгелек жақшаларда берілген және үтірлермен бөлінген функцияның аргументтер тізімі

JavaScript фигуралық жақшаларда берілген функцияны анықтайтын тұжырымдармен, {...}

Ағымдағы бетте анықталған кез келген функцияларды пайдалана аласыз. HEAD беттегі барлық функцияларды анықтау жақсы. Пайдаланушы бетті жүктеп алған кезде, алдымен мүмкіндіктер жүктеледі.

Функциялардағы бекітулер функцияларға басқа да өтініштерді қамтуы мүмкін.

Мысалы, pretty_print атымен функция бар:

```
function pretty_print(string) {
  document.write (""+ string)
}
```

Бұл функция жолды аргумент ретінде қабылдайды, кейбір HTML тегтерін (+) қосып, ағымдағы құжатта нәтижені көрсетеді.

Функцияны анықтау оны орындамайды. Ол үшін оны орындау үшін функцияны шақыруыңыз керек. Мысалы, pretty_print функциясын келесідей атауға болады:

```
<SCRIPT>
pretty_print ("This is some text to display")
< / SCRIPT>
```

Функция аргументтері тек жолдар мен сандармен шектелмейді.

Функцияның дәлелдері кестеде сақталады. Функцияның ішінде параметрлерді келесі түрде жібере аласыз:

```
functionName.arguments [i]
```

Мұнда functionname-функция Атауы, және I-нөлден басталатын Аргументтің реттік саны. Осылай, myfunc атымен бірінші дәлел myfunc болады.arguments [0]. Аргументтердің жалпы саны arguments айнымалысымен белгіленген.length.

Функция тіпті рекурсивті болуы мүмкін, яғни ол өзін тудыруы мүмкін. Мысалы, факториалдарды есептейтін функция бар:

```
function factorial(n) {
  if ((N == 0) / / (n == 1))
    return 1
  else {
    result = (n * factorial(n-1) )
    return result
  }
}
```

Сіз бір-беске дейінгі факториалдарды келесідей көрсете аласыз:

```
for (x = 0; x < 5; x++) {
  document.write(x, "factorial is", factorial (x))
  document.write(" ")
}
```

Айнымалы сандық аргументтері бар функциялар

Arguments массивін пайдалана отырып, формальды жарияланған қарағанда көп аргументтер функциясы деп атауға болады . Сіз мүмкін болатын қанша дәлел алдын ала білмеген кезде бұл жиі пайдалы. Сіз arguments пайдалана аласыз.функциядағы аргументтер санын анықтау үшін length, содан кейін arguments массивін пайдалану арқылы әрбір аргументке хабарласыңыз .

Мысалы, HTML тізімін жасау үшін белгілі бір функцияны қарастырайық. Функцияның жалғыз ресми аргументі-тізім реттелмеген болса, "U" немесе тізім реттелген (нөмірленген) болса, "O" болып табылатын жол. Функция келесідей анықталды:

```
function list(type) {
  document.write("<" + type + " L>")
  // тізім басталады
  for (var i = 1; i < list.arguments.length; i++)
  // Аргументтер арқылы қайталау
  document.write("" + list.arguments[i])
  document.write("</" + type + " L>")
  // тізім аяқталады
}
```

Сіз осы функцияның кез келген аргументтер санын өтіп, содан кейін әрбір аргументті тізімнің белгіленген түрінде әрбір жеке тармақ ретінде көрсете аласыз. Мысалы, келесі функцияға сұрау:

```
list ("o", "one", 1967, "three", "etc, etc...")
```

Әдістерін Анықтау

Әдіс-объектіге байланысты функция. Сіз стандартты функцияны анықтайсыз, өйткені сіз әдісті бірдей анықтайсыз. Содан кейін, функцияны бар элементпен байланыстыру үшін келесі синтаксисті пайдаланыңыз:

```
object.methodname = function_name
```

Мұнда object - бар нысан, methodname - әдіске берілетін атау және function_name-функция Атауы.

Нысан контексінде әдісті келесідей атауға болады:

```
object.methodname (params);
```

Нысан сілтемелері үшін this пайдалану

JavaScript ағымдағы нысанға жүгіну үшін пайдалануға болатын арнайы кілт сөзі бар this. Мысалы, сіз validate атымен функция бар болсын, бұл нысан мәнінің дұрыстығын тексереді, және high және low мәндер:

```
function validate(obj, lowval, hival) {
  if ((obj.value < lowval) || (obj.value > hival))
    alert ("Invalid Value!")
}
```

Келесі мысалда көрсетілгендей, this пайдаланып, validate әр onChange іс-шаралар өңдегішінің элементіне қоңырау шалуыңызға болады:

```
<INPUT TYPE = " text"
  NAME = " age"
  SIZE = 3
  onChange= " validate (this, 18, 99)">
```

2 бөлім. XML негіздері

XML шолу

XML markup (белгілеу) акцентімен Extensible Markup Language білдіреді. Сіз әр сөзді, сөйлемді немесе фрагментті сәйкестендірілетін, сұрыпталатын ақпаратқа айналдыра отырып, мәтін жасап, оны тегістейтін тегтермен белгілей аласыз. Сіз жасаған файлдар немесе құжат даналары элементтерден (тегтерден) және мәтіннен тұрады, бұл ретте элементтер құжатты қағазға оқу кезінде дұрыс түсінуге немесе тіпті оны электрондық түрде өңдеуге көмектеседі. Сипаттама элементтері көп болған сайын, құжаттың көп бөліктерін анықтауға болады. Таңбаның алғашқы күнінен бастап оның артықшылықтарының бірі компьютерлік жүйе жоғалған жағдайда басып шығарылған деректер тегтердің арқасында әлі де оқылатын болады.

Белгілеу тілдері компаниялар мен мемлекеттік мекемелер құрған алғашқы нысандардан бастап жалпыланған белгілеу стандартты тіліне (Standard Generalized Markup Language - SGML), гипермәтіндік белгілеу тіліне (Hypertext Markup Language - HTML) дейін және ақырында XML дейін өтті. SGML күрделі болып көрінуі мүмкін, ал HTML (шын мәнінде, алдымен элементтердің жиыны болды) ақпаратты сәйкестендіру үшін жеткіліксіз қуатты болып шықты. XML қолдану оңай және белгілеу тілін кеңейту үшін ыңғайлы ретінде жасалды.

XML-да өз элементтерін жасауға болады, бұл деректер фрагменттерін дәл көрсетуге мүмкіндік береді. Құжаттарды тек абзацтар мен тақырыптарға бөліп қана қоймай, құжаттың ішінде кез келген фрагменттерді де бөлуге болады. Бұл тиімді болу үшін, оның элементтерінің соңғы тізімін анықтау және оны ұстану керек. Элементтерді құжат түрінің сипаттамасында (Document Type Definition - DTD) немесе төменде қысқаша талқыланатын схемада анықтауға болады. Сіз XML пайдалана бастағанда, нақты файлдарды жасау арқылы элементтердің аттарымен тәжірибе қорықпаңыз.

XML құжатын құру

Жоғарыда айтылғандай, XML файлдары мәтіннен және белгілеулерден тұрады. Мәтіннің көп бөлігі мәтін тегімен қоршалған элементтерге орналастырылады. Мысалы, XML форматында ас кітабын жасау керек. Бізде Ice Cream Sundae деп аталатын рецепт бар, ол XML түрлендіру керек. Рецепт атауын белгілеу үшін оның мәтінін тегпен басталатын және аяқталатын элементке жазамыз. Бұл элементті `recipeName` деп атауға болады. Элементтің бастапқы тегін белгілеу үшін, оның атын бұрыштық жақшаларға `< >` қойыңыз, осылай: `<recipeName>`. Содан кейін Ice Cream Sundae мәтінін енгізіңіз. Кейін мәтінді қоямыз тұйықталу тег, ол білдіреді аты элемент бұрыштық жақшада, плюс-қисық сызық аяқталу элементі (`/`) атының алдына элемент, сондықтан: `</recipeName>`. Бұл тегтер мәтінді және басқа да элементтерді енгізуге болатын элементті құрайды.

Элементтердің аттарын жеке құжаттар үшін де, құжаттар топтары үшін де жасауға болады. Сіздің талаптарыңызға сәйкес элементтер үшін сақталуы тиіс ережелерді көрсетуге болады. Элементтер қатаң арнайы немесе жеткілікті ортақ болуы мүмкін. Ереже сондай-ақ әрбір элементке енгізуге рұқсат етілетіндігін анықтауы тиіс. Олар қатаң, еркін немесе аралық болуы мүмкін. Тек сіздің ойыңызша маңызды құжаттың бөліктерін анықтайтын элементтерді жасаңыз.

XML файлын жасауды бастау

XML құжатының бірінші жолы XML декларациясы болуы мүмкін. Файлдың бұл қосымша бөлігі оны XML файлы ретінде анықтайды, бұл автоматты құралдар мен адамдарға SGML емес, XML ретінде файлды тануға немесе белгілеудің басқа тәсілі.

Декларация тек <XML> немесе XML нұсқасын (<XMLversion="1.0">) және тіпті, мысалы, <?xml version=" 1.0"encoding="utf-8"?> Unicode үшін. Бұл декларация файлдың басында болуы керек болғандықтан, Егер сіз ұсақ XML файлдарын ірі файлға біріктіруді жоспарласаңыз, бұл қосымша элементті жіберіп алған жөн.

Түбірлік элементті жасау

Негізгі элементтің бастапқы және тұйықталған тегтері XML-құжаттың бүкіл мәтінін қоршайды. Файлда тек бір тамыр элементі болуы керек және ол үшін қажетті "мұқаба". 1 листингінде мен мұнда қолданатын мысал фрагменті көрсетілген, түбірлік элемент <recipe> (рецепт). (Толық XML файлы жүктеуде көрсетілген.)

Листинг 1. Тамыр элементі

```
<?xml version=" 1.0"encoding="UTF-8"?> >
<recipe>
< / recipe>
```

Құжаттың жасалуына қарай сіз мәтін мен қосымша тегтерді <recipe> және </recipe>арасында орналастырасыз.

Элементтердің атауы

Тіркелімді тегтерде сақтау

XML жасағанда бастапқы және соңғы тегтердің регистрлері сәйкес келуі тиіс. Әйтпесе, XML-ді пайдалану немесе көру кезінде қате туралы хабарды алуға болады. Мысалы, Internet Explorer регистрлер сәйкес келмеген жағдайда мәтінді көрсетпейді. Оның орнына ол бастапқы және соңғы тегтердің сәйкессіздігі туралы хабар шығарады.

Сонымен, бізде тамыр элементі бар <recipe>. XML-де алдымен элементтердің аттары таңдалады, содан кейін осы атаулардың негізінде DTD немесе схеманың тиісті сипаттамасы анықталады. Аттарда әріптер, сандар және астын сызу белгісі (_) сияқты арнайы белгілер болуы мүмкін. Міне, есте сақтау керек есімдер туралы бірнеше ережелер:

Элементтер атауларындағы бос орындарға жол берілмейді.

Атаулар саннан немесе белгіден емес, әріппен басталуы керек. (Осы бірінші әріптен кейін әріптерден, цифрлардан және рұқсат етілген таңбалардан кез келген комбинацияны пайдалануға болады.)

Регистр маңызды емес, бірақ шатастырмау үшін оны сақтаңыз.

Алдыңғы мысалда <recipe> деп аталатын элементті қоссаңыз, оның бастапқы тегі <recipe>, ал тиісті тұйықталған тегі —</recipe>болуы керек.

2 Листинг. Басқа элементтер

```
<?xml version=" 1.0"encoding="UTF-8"?> >
<recipe>
<recipe>Ice Cream Sundae< / recipe>
<preptime>5 minutes< / preptime>
< / recipe>
```

XML құжат ішінде ештеңе жоқ және ашушы және тұйықтаушы тегтерден жұбы емес, бір тегпен білдірілуі мүмкін бос тегтерді қамтуы мүмкін. Мысалы, HTML . Ол ешқандай еншілес элементтер немесе мәтін жоқ, сондықтан бұл бос элемент және оны (соңында бос орынмен және таныс тұйықтайтын қисық сызықпен).

XML синтаксисі

Тіркеме-элементтерді басқа элементтердің ішінде орналастыру. Бұл жаңа элементтер еншілес элементтер деп аталады, ал оларды қоршаған элементтер — олардың ата-аналық элементтері. 3 листингінде тамыр элементіне <recipe> бірнеше элемент салынған. Бұл ішкі еншілес элементтер <recipename>, <ingredlist> және <preptime>. <Ingredlist> элементінің ішінде <listitem>бірнеше бірдей еншілес элементтері бар. Тіркеме XML-құжатты көп деңгейлі ете алады.

Типтік синтаксистік қате бас және еншілес элементтердің салымымен байланысты. Әрбір еншілес элемент өзінің ата-аналық элементінің ашушы және тұйықтаушы тегтерінің арасында тұтас орналасуы тиіс. Еншілес элементтер келесі еншілес элемент басталғанға дейін аяқталуы тиіс.

Дұрыс салым мысал 3 листингінде келтірілген. Тегтер басқа тегтермен тоқусыз басталады және аяқталады.

3 Листинг. XML элементтерінің дұрыс тіркемесі.

```
<?xml version=" 1.0"encoding="UTF-8"?> >
<recipe>
<recipename>Ice Cream Sundae< / recipename>
<ingredlist>
<listitem>
<quantity>3</quantity>
<itemdescription>chocolate syrup or chocolate fudge</itemdescription>
< / listitem>
<listitem>
<quantity>1</quantity>
<itemdescription>nuts< / itemdescription>
< / listitem>
<listitem>
<quantity>1</quantity>
<itemdescription>cherry< / itemdescription>
< / listitem>
< / ingredlist>
<preptime>5 minutes< / preptime>
< / recipe>
```

Атрибуттарды қосу

Элементтерге кейде атрибуттар қосылады. Атрибуттар Қос атрибуттардан тұрады, мұнда мән қос тырнақшаға алынады ("), мұнда: type="dessert". Атрибуттар элементпен бірге осы параметрлердің мәндерін элементтен бір құжаттағы элементке ауыстыра отырып, қосымша параметрлерді сақтауға мүмкіндік береді.

Атрибут —немесе тіпті бірнеше атрибуттар-элементтің бастапқы тегінің ішінде көрсетіледі: <recipe type="dessert">. Бірнеше төлсипаттарды қосқан кезде олар бос орындармен бөлінеді: <recipename cuisine="american" servings="1">. 4 листингінде XML файлын көрсетеді, ол қазір көрінеді.

Листинг 4. Элементтер мен атрибуттар бар XML файлы

```
<?xml version=" 1.0"encoding="UTF-8"?> >
<recipe type="dessert">
<recipename cuisine="american" servings=" 1 " >Ice Cream
Sundae</recipename>
<preptime>5 minutes< / preptime>
```

< / recipe >

Кез келген атрибуттарды қолдануға болады. Сіздің құжатыңызға қандай мәліметтерді қосуға болады деп ойлаңыз. Құжаттар сақталса, атрибуттар әсіресе пайдалы-мысалы, рецепт түрі бойынша. Атрибуттардың аттарында элементтердің аттары сияқты таңбалар болуы мүмкін.

Дұрыс және дұрыс салынған XML

Егер сіз құрылымда анықталған ережелерді ұстансаңыз, дұрыс құрылған XML кодын оңай жасай аласыз. Дұрыс XML-барлық XML ережелерін сақтай отырып жасалған XML коды: элементтерді дұрыс атау, тіркеме, атрибуттар және т. б.

XML-мен не істеп жатқаныңызға байланысты, Сіз дұрыс салынған XML-мен жұмыс істеу қажет болуы мүмкін. Жоғарыда келтірілген рецепт түрі бойынша сұрыптау мысалын қарастырайық. <Recipe> элементтері типе атрибутын қамтуы керек. Кодты сәтті тексеруге және осы атрибут мәнінің тұрақты болуына кепілдік беру өте маңызды.

Тексеру (validation) деп құжаттың құрылымын ол үшін белгіленген ережелерге және әрбір бас элемент үшін еншілес элементтерді анықтауға сәйкестігін тексеру түсініледі. Бұл ережелер құжат түрінің сипаттамасында (DTD) немесе схемада айқындалады. Мұндай тексеру үшін DTD немесе схема жасау керек, содан кейін өзінің XML файлдарында DTD файлына немесе схемаға сілтеме беру керек.

Тексеруді рұқсат ету үшін, XML-құжаттарының басына жақынырақ құжат түріндегі декларацияны (DOCTYPE) орналастыру керек. Бұл жол DTD сілтемесін немесе осы құжатты тексеру үшін пайдаланылатын схеманы (элементтер мен ережелер тізімін) қамтиды. DOCTYPE жолы 5 листингінде сияқты болуы мүмкін.

Листинг 5. DOCTYPE

1

```
<!DOCTYPE MyDocs SYSTEM " filename.dtd">
```

Бұл мысал filename деп аталатын элементтер тізімінің файлын білдіреді.dtd сіздің компьютеріңізде (яғни жалпы PUBLIC каталогында емес, SYSTEM каталогында).

DTD сызбалары – құжат түрін анықтау

Қазіргі уақытта Бүкіләлемдік Интернет Консорциум (W3C) екі схеманы қолдайды: DTD (Document Type Definition) және XML Schema.

Бұл дәрісте екі схеманың бірінші, ертерек қарастырылады, DTD. XML схемасы келесі дәрістерде талқыланады.

DTD сипаттамасы «Кеңейтілетін белгілеу тілі (XML) 1.0» ұсынымында, яғни орындалады. XML құжатына қойылатын талаптарды сипаттайтын бірдей ұсыныста.

DTD анықтау міндеті:

XML құжатында қолданылатын элементтердің құрамы;

мазмұн үлгілерінің сипаттамасы, яғни бір элементтердің басқасына кіру ережесі;

құжаттың XML элементтерімен қолданылатын атрибуттардың құрамы; элементтерде атрибуттар қалай қолданылуы мүмкін;

XML құжатқа қосылатын мәндерді сипаттау. Бұдан әрі, осы дәрістің материалын баяндау үшін біз ақпараттық технологиялар бойынша кітаптар кітапханасын сипаттайтын төменде берілген XML құжатты пайдаланатын боламыз:

```
<?xml version="1.0" encoding="Windows-1251"?>
  <library>
    <books>
      <book id="N1" isbn="978-5-8459-1118-6" year="2008"
issue="translation">
      <udk value="004.541.47"/>
      <authors>
        <author country="US">С. Граннеман</author>
      </authors>
      <title>Linux. Карманный справочник</title>
      <cover border="1" align="center" width="350" height="200"
href="www.williamspublishing.com/covers/linux.gif"/>
      <publishing>Вильямс</publishing>
      <annotation>Область применения <scope>операционные
системы</scope></annotation>
    </book>
    <book id="N2" isbn="5-314-00189-6" year="2000"
issue="translation">
      <udk value="681.3" />
      <authors>
        <author country="US">К. Уилмер</author>
        <author country="US">А. Хоумер</author>
      </authors>
      <title>Dynamic HTML. Справочник</title>
      <cover border="1" align="center" width="350" height="200"
href="www.piter-press.ru/covers/dhtml.gif"/>
      <publishing>Питер</publishing>
      <annotation>Область применения
<scope>Интернет</scope></annotation>
    </book>
    <book id="N3" isbn="978-5-9775-0104-0" year="2007"
issue="original">
      <udk value="681.3.06"/>
      <authors>
        <author country="RU">Д.Н. Колисниченко</author>
      </authors>
      <title>Ubuntu Linux</title>
      <cover border="1" align="center" width="350" height="200"
href="www.bhv.ru/covers/ubuntu.gif"/>
      <publishing>БХВ-Петербург</publishing>
      <annotation>Область применения ОС Linux</scope></annotation>
    </book>
  </books>
</library>
```

1 XML құжатында DTD хабарландыру

DTD құру ережелерін зерттеуді бастамас бұрын, XML құжат DTD жариялайды деп қарастырайық. Бұл DTD зерттеу кезінде компьютерде жасалған мысалдарды сынап көруге мүмкіндік береді.

XML құжатқа қатысты DTD екі түрі болуы мүмкін: сыртқы және ішкі. Сондай-ақ, құжатта екі DTD түрін де бір уақытта пайдалануға болады.

XML құжатында DTD жариялау үшін құжаттың XML түбірлік элементінің алдында орналасуы тиіс DOCTYPE ұсынысы пайдаланылады, ал онда көрсетілген DTD атауы түбірлік элементтің атымен сәйкес келуі тиіс. DOCTYPE ұсынысы көрініс бар:

```
<!DOCTYPE _dtd орналасқан жері _dtd >
```

мұнда: имя_dtd-құжаттың XML түбірі элементінің атымен сәйкес келетін DTD сызбасының атауы;

DTD – түрі DTD түріне байланысты болатын DTD схемасының орналасу орнының сипаттамасы.

1.1 Сыртқы DTD

Сыртқы DTD кеңейтімі болуы тиіс бөлек файлда.dtd, мысалы, library.dtd. DOCTYPE ұсынысының синтаксисі сыртқы байланыс жағдайында келесі түрді қабылдайды:

```
<!DOCTYPE атым_dtd >
```

мұнда: URI-DTD схемасының орналасқан жерін көрсету.

SYSTEM және PUBLIC-келесі алгоритмде қолданылатын альтернативті қолданылатын кілт сөздер.

SYSTEM көрсеткен кезде, анализатор нақты көрсетілген URI бойынша DTD табу керек, мысалы:

```
<!DOCTYPE library SYSTEM "http://www.library.org/dtd/library.dtd">
```

PUBLIC көрсеткенде құжаттың XML өңдеушісіне "жақсы таныс" жария DTD қолданылады деп саналады. Мысалы, кітапхана жүйелері үшін Интернетке белгілі бір сайтта орналасқан DTD жария схемасы бар. Егер кітапханалық XML құжаттарымен жұмыс істеу үшін стандартты емес, мамандандырылған талдағыштар пайдаланылса, онда ол өзіне қажетті DTD сұлбасы қайда екенін өзі білуі мүмкін. Бұл жағдайда DTD схемасы бар файлдың дәл URI орналасуын көрсету міндетті емес, мысалы:

```
<!DOCTYPE library PUBLIC "world/library.dtd">
```

PUBLIC сипаттамасын файлдың нақты орналасуын DTD сызбасымен қоса беруге болады. Бұл жағдайда, егер талдағыш жария схеманы таппаса, ол нақты мекен-жай бойынша көрсетілген схеманы пайдаланатын болады, мысалы:

```
<!DOCTYPE library PUBLIC "world/library.dtd"
```

```
"http://www.library.org/dtd/library.dtd">
```

1.2 ішкі DTD

Ішкі DTD құжаттың XML денесіне тікелей қосылады. DOCTYPE ұсынысы ішкі байланыс жағдайында келесі түрге ие:

```
<!-- DTD -->
```

```
<!DOCTYPE имя_dtd
```

```
[
```

```
    содержание DTD
```

```
]>
```

```
<!-- XML документ -->
```

```
<library>
```

```
...
```

```
</library>
```

```
>
```

DTD мазмұны шаршы жақшаның ішінде орналасқан.

```
<!ELEMENT book (udk, authors?, publishing, annotation?)>
```

Мәтіндік түр тек мәтінді қамтиды. Берілген мысалда осы түрге author, title, publishing сияқты элементтер жатады. Бұл түрі мазмұн моделі өрісінде #PCDATA (parsed character data) сөзімен сәйкестендіріледі. Мәтін түріндегі элементті сипаттау мысалы:

```
<!ELEMENT author (#PCDATA)>
```

Аралас түрі мәтінді де, элементтерді де қамтуы мүмкін. Ұсынылған мысалда бұл түрге annotation элементі жатады. Аралас түрдегі элементті сипаттау мысалы:

```
<!ELEMENT annotation (#PCDATA | scope)*>
```

Аралас типті пайдалануға бірқатар шектеулер қойылады:

аралас түрге кіретін элементтердің жүру тәртібін анықтау мүмкін емес;

аралас түрге кіретін элементтердің қайталау санын анықтау мүмкін емес;

бір атау аралас түрдегі хабарландыруда бір реттен артық ұсынылуы мүмкін емес.

Еркін түр кез келген мазмұнды қамтуы мүмкін. Мазмұн моделі өрісінде ANY сөзімен сипатталады. Еркін түрдегі элементті сипаттау мысалы:

```
<!ELEMENT имя_элемент ANY>
```

Еркін түрдегі элементтерді жасау және пайдалану әдетте ұсынылмайды.

XML - XSD сызбасы

W3C консорциумы схемаларды сипаттаудың ең танымал тілдерін бір стандартқа біріктіре отырып, XML (XSD) сұлбаларын анықтау тілін ұсынды. Бұл ретте көздеген негізгі мақсат — стандартты алу, оны кеңінен жүзеге асыруға болады және ол платформа-тәуелсіз.

Тіл XML Schema Definition Language, ол сондай-ақ XML Schema Language деп аталады, көп жағдайда сіз бұрын танысқан XDR тіліне ұқсас. XSD схемалары келесі міндеттерді шеше алады:

XML құжатында элементтерді аудару және құжатта тек жарияланған элементтердің болуын тексеру.

Құжат элементтерін түрлендіретін атрибуттарды жариялау және анықтау.

Элементтердің арасындағы бас-еншілес қатынастарды анықтау.

Элементтер мен атрибуттар үшін мазмұнның жай-күйі мен үлгілерін анықтау.

Деректер типтерін анықтау.

Әдепкі мәндерді орнату.

Кеңейту мүмкіндігі.

Атау кеңістігін пайдалануды қолдау.

XML схемасындағы түбірлік элемент схема құжаттарындағы барлық қалған элементтерді қамтитын Schema элементі болып табылады. Xmlns атрибутымен XSD схемасының түбірлік элементі аясында XSD схемасының элементтері мен атрибуттары бар XMLSchema атауларының кеңістігі анықталады.

```
<xsd: schema xmlns: xsd="http://www.w3.org/2001/XMLSchema"> >
```

Барлық XSD элементтері xsd префиксінен басталады:, ол схема данасының түбірлік элементінде жарияланған XSD есімдерінің кеңістігі үшін көрсетіледі.

XML-схема арқылы тексерілетін құжат, сондай-ақ атау кеңістігін жариялау болуы тиіс. Атау кеңістігі әрқашан құжат данасының түбірлік элементінде xmlns атрибуты арқылы көрсетіледі:

xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"

Бұл атау кеңістігі XML құжатына қосуға болатын xmlschema элементтері мен атрибуттарын қамтиды. Жалпы келісім бойынша xsi префиксі осы аттар кеңістігі үшін қолданылады және олардың қос нүктемен бөле отырып, аттар кеңістігіне тиесілі барлық элементтер мен атрибуттар атауларының басында қосылады.

Нақты схемаға сілтеме атрибутта келтіріледі

xsi:schemaLocation="http://kit.znu.edu.ua/scemes/имя_файла.xsd"

XSD элементін және төлсипатын жариялау

Схеманы құру процесі екі қадамды қамтиды — элементтердің типтерін немесе атрибуттардың типтерін анықтау және жариялау. XML-құжаттың элементтері мен атрибуттары xsd:element және xsd:attribute схемаларының элементтерімен жарияланады. XML-құжат құрылымы xsd:simpleType және xsd:complexType схемаларының элементтерімен анықталады.

Элементтің негізгі хабарландыруы аты мен деректер түрінен тұрады

<xsd:element name="имя_элемента" type="xsd:тип_данных"/>

XML құжаттарында пайдаланылатын XSD дескрипторлар екі санатқа бөлінеді-күрделі типтері мен қарапайым типтері. Күрделі типті элементтердің басқа элементтері болуы мүмкін, сондай-ақ белгілі бір атрибуттары бар; қарапайым типті элементтердің мұндай мүмкіндіктері жоқ.

Атрибут-басқа элементтерді қамтымайтын қарапайым түрдегі хабарландыру. Атрибутты хабарландыру элементті хабарландыру сияқты:

<xsd:attribute name="имя_атрибута" type="xsd:үлгі_данных" / >

Қарапайым деректер түрлері

Қарапайым үлгілердің екі негізгі санаты бар:

кірістірілген түрлері;

пайдаланушы анықтаған қарапайым түрлері.

XSD тілінің көп саны енгізілген қарапайым деректер типтері бар. Кірістірілген түрлері примитивті типтерді және туындыларды қамтиды. Қарапайым деректер түрлері басқа деректер түрлерінен алынбады. Мысалы, өзгермелі үтір бар сандар-басқа деректер түрлерінен алынбаған математикалық ұғым. Деректер типтері бар деректер типтерінің терминдерінде анықталған. Мысалы, бүтін сан - деректердің ондық түрінен алынған жеке жағдай.

Келесі кесте XML-сызбалар, деректер түріне және деректер түрін сипаттауға қолданылуы мүмкін аспектілер мәліметтерінің қарапайым түрлерінің тізімін ұсынады.

Деректер түрі	Аспектiлер	Сипаттамасы
string	length, pattern, maxLength, minLength, enumeration, whiteSpace	Символдық жолды білдіреді.
Boolean	pattern, whiteSpace	True немесе false болуы мүмкін логикалық мәнді ұсынады.
decimal	enumeration, pattern, totalDigits, fractionDigits, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Еркін санды білдіреді.

float	pattern, enumeration, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	32-биттік Сан бір дәлдіктегі құбылмалы үтірмен беріледі.
double	pattern, enumeration, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	64-биттік Сан екі есе дәлдіктегі құбылмалы үтірмен беріледі.
duration	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Уақыт ұзақтығын ұсынады. Duration үшін үлгі келесі- PnYnMnDTnHnMnS, мұнда nу жылдар санын білдіреді; nM- ай; nD-күн; T-күн және уақыт бөлгіші; nH - сағат саны; nM - минут; nS-секунд.
dateTime	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Нақты уақытты ұсынады. DateTime үшін үлгі келесі-CCY-mm-DDThh:mm:ss, мұнда CC Жүзжылдықты ұсынады; YY - жыл; MM - ай; DD - күн; T - күн мен уақытты бөлгіш; hh - сағат саны; mm - минут; SS - секунд. Қажет болса, секунд үлесін көрсетуге болады. Мысалы, үлгідегі жүздеген үлес: ss.ss
time	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Күннің нақты уақытын ұсынады. Уақыт үлгісі келесі- hh:mm:ss.SSS (секундтың үлестік бөлігі міндетті емес).
date	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Күнтізбелік күнді ұсынады. Мұндай date үлгісі - CCY-MM-DD (мұнда уақытты білдіретін бөлік міндетті емес).
gYearMonth	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Нақты жылдың нақты айы (CCY-MM).
gYear	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Күнтізбелік жыл (CCYY) ұсынады.
gMonthDay	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Нақты айдың нақты күнін ұсынады (--MM-DD).
gDay	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Күнтізбелік күнді ұсынады (- - - DD).
gMonth	enumeration, pattern,	Күнтізбелік ай (--mm -) ұсынады.

	minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	
hexBinary	length, pattern, maxLength, minLength, enumeration, whiteSpace	Еркін он алтылық-кодталған екілік ақпаратты ұсынады. HexBinary-төрт жұп он алтылық таңбадан тұратын белгіленген ұзындықтағы екілік октеттер жиынтығы. Мысалы, 0-9a-fA-F.
base64Binary	length, pattern, maxLength, minLength, enumeration, whiteSpace	Еркін Base64-кодталған екілік ақпаратты ұсынады. Base64Binary-белгіленген ұзындықтағы екілік октеттер жиынтығы.
anyURI	length, pattern, maxLength, minLength, enumeration, whiteSpace	RFC представляет URI, как определено в 2396. Значение AnyURI может быть абсолютно или сравнительно и может быть необязательным идентификатором фрагмента.
QName	length, enumeration, pattern, maxLength, minLength, whiteSpace	Құрамдас атауды ұсынады. Аты қос нүктемен бөлінген префикстен және жергілікті атаудан тұрады. Префикс және жергілікті атаулар NCNAME болуы керек. Префикс ат кеңістігін хабарландыру арқылы namespace URI сілтемесімен байланысты болуы тиіс.
NOTATION	length, enumeration, pattern, maxLength, minLength, whiteSpace	Белгілер жүйесі төлсипатының түрін ұсынады. QNamef жиынтығы.

XML атаулар кеңістігі

XML аттар кеңістігі элементтердің түрлері және атрибуттардың атаулары ретінде XML құжаттарында қолданылатын URI сілтемесі арқылы белгіленетін аттар жиынтығы болып табылады.

XML аттар кеңістігі 2006 жылдың 16 тамызындағы W3C ұсынысымен анықталған: XML атаулар кеңістігі. XML тегтерінің атаулары жаһандық бірегей, сондай-ақ өнімділігін арттыру үшін қысқа болуы керек. Бұл атау кеңістігіне қатысты W3C ұсынымынан XML элементін толықтыруға мүмкіндік беретін xmlns атрибуты беріледі. Егер ол элементте болса, ол осы элемент үшін атау кеңістігін білдіреді.

Xmlns атрибуты келесі синтаксиспен қолданылады:

`xmlns:префикс= кеңістіг_аты`

кеңістіг_аты бірегей URI (мысалы, `www.ibm.com`) және префикс атау кеңістігін білдіреді және оған көрсеткішті көрсетеді.

Тапсырыс беруші элементінің келесі анықтауында басқа бизнес-қолданбалар жасаған тапсырыс берушілердің жазбаларында пайда болатын тегтерден элементтің тегін айыру үшін тиісті атау кеңістігі беріледі.

```
<acct:customer xmlns:acct="http://www.my.com/acct-REV10">
```

```
  <acct:name>Corporation</acct:name>
```

```
  <acct:order acct:ref="5566"/>
```

```
  <acct:status>invoice</acct:status>
```

```
</acct:customer>
```

Бірінші жолда атау кеңістігін анықтау атау кеңістігі префиксін береді <http://www.my.com/acct-REV10>. Префикс атау сияқты элементтердің аттары үшін, оларды атау кеңістігіне бекіту үшін қолданылады. Мысалы, екінші қолданба - орындау жүйесі, тапсырыс берушілердің элементтеріне басқа аттар кеңістігін бере алады:

```
<ful:customer xmlns:ful="http://www.your.com/ful">
  <ful:name>Corporation</ful:name>
  <ful:order ful:ref="A98756"/>
  <ful:status>shipped</ful:status>
</ful:customer>
```

Деректер түрінің екеуі де қолданба енді Есеп және орындау деректерімен жеке жұмыс істей алады. Әдепкі аттар кеңістігі бар. Атау кеңістігін анықтау үшін жергілікті атау берілмеген болса, ол қойылады:

```
<acct:customer xmlns="http://www.my.com/acct-REV10" xmlns:acct="http://www.my.com/acct-REV10">
  <name>Corporation</name>
  <order acct:ref="5566"/>
  <status>invoice</status>
</customer>
```

Бұл мысалда Тапсырыс берушінің жазбасындағы барлық тегтер атау кеңістігімен анықталған <http://www.my.com/acct-REV10>. префиксті орнатудың қажеті жоқ, себебі әдепкі оп аттар кеңістігі қолданылады. Әдепкі атау кеңістігі барлық атрибуттар анықтамаларына қолданылады.

XML атауларының үлгілері мен кеңістігі

Келесі XML схемасында әдепкі атау кеңістігі XML схема атауларының стандартты кеңістігі ретінде анықталған <http://www.w3.org/2001/XMLSchema>. сондай-ақ арнайы аттар кеңістігі бар <http://www.ibm.com> ескерту.

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.ibm.com"
xmlns:TestSchema="http://www.ibm.com">
  <simpleType name="ZipCodeType">
    <restriction base="integer">
      <minInclusive value="10000"/>
      <maxInclusive value="99999"/>
    </restriction>
  </simpleType>
  <!--element definitions skipped -->
</schema>
```

Мысалы, алдыңғы XML схемасы сақталған C:\temp\TestSchema ескерту.xsd, схемаға арналған XML файлының мысалы болып табылады:

```
<?xml version="1.0"?>
<x:addressList xmlns:x="http://www.ibm.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.ibm.com file:///C:/temp/TestSchema.xsd">
  xsi:schemaLocation="http://www.ibm.com file:///C:/temp/TestSchema.xsd">
  <x:address>
    <x:street>x:Vangerowstrasse</x:street>
    <x:zipCode>69115</x:zipCode>
    <x:city>x:Heidelberg</x:city>
  </x:address>
  <x:address>
    <x:street>x:Bernal Road</x:street>
    <x:zipCode>90375</x:zipCode>
    <x:city>x:San Jose</x:city>
  </x:address>
</x:addressList>
```

Атаулардың мақсатты кеңістігі

Аттардың мақсатты кеңістігі элемент пен оның аты арасында байланыс болған жағдайда аттар кеңістігін сәйкестендіру үшін қызмет етеді. Бұл байланыс XML файлдарындағы элементтердің атауларының орнын анықтайды. Схеманы импорттайтын XML файлы schemaLocation атрибутындағы оның атауларының мақсатты кеңістігіне сілтеме жасауы тиіс. Элемент атауларының мақсатты және нақты кеңістігі арасындағы кез келген сәйкессіздік схеманы тексеру қателері түрінде көрсетіледі. Біздің мысалда аттардың мақсатты кеңістігі болып табылады <http://www.ibm.com>; ол XML схема файлында беріледі және XML файлында екі рет көрсетіледі. Бұл үш атау кеңістігі арасындағы сәйкессіздік тексеру қателерінің пайда болуына әкеледі.

Келесі мысалда атау кеңістігі префикстерінің жұмыс істеу принциптері және XML сұлбалары мен олардың тиісті XML элементтері құжаттарындағы аттардың мақсатты кеңістіктері көрсетілген.

1-мысал-атаулы мақсатты кеңістік және әдепкі атау кеңістігі, сондай-ақ қарапайым жергілікті элементтер схемасы

XML схемасы:

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.ibm.com"
xmlns:x="http://www.ibm.com">
<complexType name="AddressType">
<sequence>
<element name="name" type="string"></element>
</sequence>
</complexType>
<element name="MyAddress" type="x:AddressType"></element>
</schema>
```

Осы схемадан жасалған жарамды XML данасының құжаты келесідей болуы мүмкін. Жергілікті элементтер мен атрибуттар қарапайым.

```
<?xml version="1.0"?>
<x:MyAddress xmlns:x="http://www.ibm.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com x.xsd ">
<name>Peter Smith</name>
</x:MyAddress>
```

Жергілікті элементтер (мысалы, "name") және атрибуттар XML файлында анықталмаған кезде, тек тамыр элементі нақтыланады. Осылайша, бұл мысалда "x" атау кеңістігінің префиксі "MyAddress" түбірлік элементіне берілген. <http://www.ibm.com>", бірақ "x" префиксі "name" жергілікті элементіне берілмеген.

2-мысал-атау мақсатты кеңістігі және әдепкі атау кеңістігі, сондай-ақ толық Жергілікті элементтері бар схема

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.ibm.com"
xmlns:x="http://www.ibm.com" elementFormDefault="qualified">
<complexType name="AddressType">
<sequence>
<element name="name" type="string"></element>
</sequence>
</complexType>
<element name="MyAddress" type="x:AddressType"></element>
</schema>
```

Осы схемадан жасалған жарамды XML данасының құжаты келесідей болуы мүмкін. Жергілікті элементтер мен атрибуттар нақтыланды, себебі elementFormDefault атрибуты XML схемасында орнатылған.

```
<?xml version="1.0"?>
  <x:MyAddress xmlns:x="http://www.ibm.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.ibm.com x.xsd ">
    <x:name>Peter Smith</x:name>
  </x:MyAddress>
```

Бұл мысалда атау кеңістігінің "x" префиксі "MyAddress" түбірлік элементіне және атау кеңістігіне байланысты "name" жергілікті элементіне берілген. <http://www.ibm.com> ескерту.

3-мысал-аттардың мақсатты кеңістігімен және анық берілген xmlns: xsd

Бұл XML схемасы осы төлсіпатты қосады:

xmlns:xsd = "http://www.w3.org/2001/XMLSchema"

Бұл дегеніміз, XML схема тілінің әрбір конструкциясы "xsd" префиксі арқылы нақтылауды талап етеді. Мысалы, xsd:complexType және xsd:string

Ескерту және пайдалану үшін "xs" немесе "foobar" сияқты кез келген басқа префикстерді таңдауға болады.

Бұл префиксті XML схемасының параметрлері бетінде көрсетуге болады. Қосымша ақпарат байланысқан тапсырмалар сипаттамасында келтірілген.

Пайдаланушы анықтаған барлық түрлері аттар кеңістігіне жатады <http://www.ibm.com> targetNamespace атрибутымен анықталғандай, xmlns:x атрибутымен анықталғандай "x" префиксі бар.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.ibm.com"
xmlns:x="http://www.ibm.com">
<xsd:complexType name="AddressType">
<xsd:sequence>
<xsd:element name="name" type="xsd:string"></xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:element name="MyAddress" type="x:AddressType"></xsd:element>
</xsd:schema>
```

Осы схемадан жасалған жарамды XML данасының құжаты келесідей болуы мүмкін. Жергілікті элементтер мен атрибуттар қарапайым. Спецификация семантикасы 1 мысалында сипатталған ұқсас.

```
<?xml version="1.0"?>
  <x:MyAddress xmlns:x="http://www.ibm.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.ibm.com x.xsd ">
    <name>Peter Smith</name>
  </x:MyAddress>
```

Мысал 4-xmlns:xsd анық беретін, жарияланбаған аттардың нысаналы кеңістігі бар схема

Бұл XML схемасы жеке мақсатты кеңістік жоқ. Бұл жағдайда XML схемасының барлық құрылымын "xsd" сияқты префикспен айқын анықтау ұсынылады. AddressType сияқты осы схеманың анықтамалары мен хабарландыруларына сілтемелер атау кеңістігінің префиксі жоқ болғандықтан, аттар кеңістігімен нақтыланбаған.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:complexType name="AddressType">
<xsd:sequence>
<xsd:element name="name" type="xsd:string"></xsd:element>
<xsd:element name="name" type="xsd:string"></xsd:element>
<xsd:element name="name" type="xsd:string"></xsd:element>
</xsd:sequence>
```

```

</xsd:complexType>
<xsd:element name="MyAddress" type="AddressType"></xsd:element>
</xsd:schema>

```

Осы схемадан жасалған жарамды XML данасының құжаты келесідей болуы мүмкін. Барлық элементтер анықталмаған.

```

<?xml version="1.0"?>
<MyAddress xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="x.xsd">
<name>name</name>
</MyAddress>

```

5-мысал-атау арналы кеңістігі әдепкі атау кеңістігі болып табылатын сызба

Бұл мақсатты ретінде пайдаланылатын әдепкі атау кеңістігі бар XML схемасы. Мұнда атау кеңістігі префикссіз көрсетіледі.

```

<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.ibm.com"
xmlns="http://www.ibm.com">
<xsd:complexType name="AddressType">
<xsd:sequence>
<xsd:element name="name" type="xsd:string"></xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:element name="MyAddress" type="AddressType"></xsd:element>
</xsd:schema>

```

Осы схемадан жасалған жарамды XML данасының құжаты келесідей болуы мүмкін:

```

<?xml version="1.0" encoding="UTF-8"?>
<MyAddress xmlns="http://www.ibm.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com NewXMLSchema.xsd">
<name>name</name>
</MyAddress>

```

XML сілтемелері – Xpath тілі

XLink қалай қолданылады XLink XML құжаттар ішінде гиперсілтеме жасау үшін қолданылады Құжаттың кез келген XML элементі сілтеме ретінде өзін ұстай алады XLink қарапайым Сілтемелер (HTML ретінде) және кеңейтілген сілтемелер (бірнеше ресурстарды байланыстыру үшін) XLink арқылы сілтемелер байланысты файлдардан тыс анықталуы мүмкін XLink W3C стандарты XLink Синтаксисі HTML гиперсілтемеде <a>элементінің көмегімен анықталады. XML-да бұл жұмыс істемейді. XML құжаттарында кез келген аттары бар элементтерді пайдалануға рұқсат етіледі, осылайша браузерлер XML құжаттарындағы элементтердің сілтемелер ретінде пайдаланылатынын алдын ала біле алмайды. Төменде XML құжаттағы сілтемелерді жасау үшін XLink пайдалану мысалы бар:

```

<?xml version="1.0" encoding="UTF-8"?>
<homepages xmlns:xlink="http://www.w3.org/1999/xlink">
  <homepage xlink:type="simple" xlink:href="http://msiter.ru">Наш учебный сайт</homepage>
  <homepage xlink:type="simple" xlink:href="http://www.w3.org">Консорциум W3C</homepage>
</homepages>

```

XLink пайдалану мүмкіндігі болу үшін, біз xlink атау кеңістігін декларациялауымыз керек. Xlink атауларының кеңістігі: "http://www.w3.org/1999/xlink"

ескерту. Xlink:type және xlink:href атрибуттары <homepage> элементінде xlink атауларының кеңістігінен алынады. Xlink атрибуты: type= " simple "қарапайым" HTML-ұқсас "сілтемені жасайды ("онда өту үшін мұнда басыңыз" дегенді білдіреді). Xlink атрибуты: href сілтеме URL мекенжайын анықтайды. Xlink шолғыштарды қолдау Браузерлер XML құжаттарында XLink қолдауы жоқ. Дегенмен, барлық негізгі браузерлер SVG қарапайым XLink қолдайды. Мысал XLink Келесі XML құжат xlink функционалын қамтиды:

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore xmlns:xlink="http://www.w3.org/1999/xlink">
  <book title="Harry Potter">
    <description xlink:type="simple" xlink:href="/images/HPotter.gif"
xlink:show="new">
```

As his fifth year at Hogwarts School of Witchcraft and Wizardry approaches, 15-year-old Harry Potter is...

```
</description>
</book>
<book title="XQuery Kick Start">
  <description xlink:type="simple" xlink:href="/images/XQuery.gif"
xlink:show="new">
```

XQuery Kick Start delivers a concise introduction to the XQuery standard...

```
</description>
</book>
</bookstore>
```

Құжаттың басында xlink (xmlns: xlink="http://www.w3.org/1999/xlink") Xlink атрибуты: type= "simple" қарапайым "HTML-ұқсас" сілтемені жасайды Xlink атрибуты: href сілтеме URL мекенжайын анықтайды (бұл жағдайда - сурет) Xlink атрибуты: show= "new" сілтеме жаңа терезеде ашылатынын анықтайды XLink-ары қарай жүреміз Жоғарыда мысалда қарапайым xlink сілтемелері көрсетілді. Алайда, XLink жеке беттерге емес, алыстағы ресурстарға қол жеткізу қажет болған кезде қызықты болады. Егер xlink:show атрибутын "embed" деп орнатсаңыз, онда сілтеме жасалатын ресурс бет кодына кірістіріледі. Егер басқа XML құжатқа сілтеме жасалса, онда құжаттардың XML иерархиясын жасау мүмкіндігі пайда болады. Сондай-ақ,, ресурс көрсетілуі тиіс кезде анықтауға болады. Бұл үшін xlink:actuate атрибуты жауап береді. Xlink Атрибуттары

Атрибут	Значение	Описание
xlink:actuate	onLoad onRequest other none	Қосылған ресурстың оқылуы және көрсетілуін анықтайды: onLoad-ресурс жүктеу және құжатты жүктегеннен кейін бірден көрсету керек onRequest-ресурс жүктеледі және тек сілтемені басқан кезде ғана көрсетіледі
xlink:href	URL	Сілтеме URL мекенжайын анықтайды
xlink:show	embed new replace other	Сілтеменің қайда ашылғанын анықтайды. Үнсіз әрекет етеді " replace"

	none	
xlink:type	simple extended locator arc resource title none	Сілтеме түрін анықтайды

XML — DOM құжаттың объектілік моделі

XML-құрылымдық деректерді сақтау және беру үшін ыңғайлы мәтіндік пішім. Бірақ онымен тікелей бағдарламалау тілінде жұмыс істеу қиын. XML-құжаттан деректерді оңай оқу немесе оны түрлендіру үшін алдымен оны неғұрлым қолайлы нысанға айналдыру керек. Қазіргі уақытта бұл мәселені шешу үшін кез келген дұрыс құжат нысан ағашы түрінде ұсынылуы мүмкін дегенге негізделген DOM бағдарламалық моделі кеңінен қолданылады. DOM және Flash-плеер қолданылады. DOM (Document Object Model — құжаттың Объектілік моделі) W3C Комитетімен әзірленген және көптеген тілдер мен бағдарламалық орталарда іске асырылған. Оның негізгі бәсекелесі — оқиғаларға негізделген SAX моделі-аз таралған. DOM өте танымал болуының себебі оның қарапайымдылығы мен интуитивтілігі. Бұл ActionScript іске асырылды оңайлатылған нұсқасы DOM бар көп ауытқу стандарты. Оны қарауға біз осы бөлімді арнаймыз.

Dom негізгі идеясы-XML құжатын сипаттау үшін Нысандар ағашын салу керек. Әрбір объект бір түйінді бейнелейді (қарапайым айтқанда, тег). Әрбір объектіде сипатталған XML ағаш торабының еншілес тораптарына сәйкес келетін объектілерге сілтеме массиві бар. Олар белгіленген тегтер жазылған тәртіппен орналасқан. Әрбір торап нысаны ата-аналық торап объектісіне сілтеме жасайтын сипат бар. Торап объектісімен байланысты ерекше объектіде қасиеттер түрінде Оның атрибуттары сақталады.

Бұл ActionScript бар арнайы сынып, объектілері, оның рөлінде түйіндер ағашы DOM. Бұл XmlNode сыныбы. Оның қасиеттері мен әдістері ағаш бойынша навигацияны жүзеге асыруға, деректерді оқуға және әртүрлі модификацияларды шығаруға мүмкіндік береді. XmlNode - дан басқа, ActionScript-та XML-мен жұмыс істеуге байланысты тағы бір сынып бар. Бұл класс XML деп аталады және оның объектілері құжатты тұтастай бейнелейді. Бұл класс барлық құжатқа қатысты операцияларды жүргізу үшін қажет. Мұндай операцияларға, мысалы, XML-хабарландырулар мен DDT-хабарландырулар оқу және инициалдау, мазмұнды жүктеу және құжатты серверге жіберу, XML-мәтінді Dom нысандарының ағашына түрлендіру және т.б. жатады.

XML класс объектісі арқылы Dom ағашының тамырлы тегіне сәйкес келетін нысанына қатынасу жүзеге асырылады. Үшін _____үлкен интуитивтілік, бұл операция ата-аналық торап объектісінен еншілес торапқа өтуге болатын сол құралдардың көмегімен жүзеге асырылады. Сондықтан, құжатты сипаттаушы XML класының объектісіне DOM ағаш түйіндерінің нысандарына ұқсас қасиеттер мен әдістер қол жетімді болуы тиіс. Мүмкін, XML класы XmlNode Ішкі сыныбы жасалды. XmlNode прототипінде құжаттың ағашымен жұмыс істеуге арналған

элементтер сақталады. XML прототипінде Жалпы операцияларды жүргізуге арналған әдістер мен қасиеттер жазылған.

Іс жүзінде объектісі сынып XML қолданылады қасиеттері сынып XmlNode арналған кіру объектінің ата-аналар торабының объектісіне торабының еншілес. Бұл әрекет құжаттың түбір түйінін сипаттайтын XmlNode класс объектісіне кіру үшін қажет. Яғни, XML класының формальды нысаны-DOM ағашының XML-құжатын сипаттайтын түбірлік түйін нысаны үшін бас түйін болып табылады.

XML және XmlNode сыныптары арасындағы айырмашылықты есте сақтау керек. Тек XML класына тән әдіс немесе сипатты түйін нысаны арқылы іске қосу әрекеті істен шығады. Сондай-ақ, барлық тораптардың нысандарына қандай да бір әдіс немесе сипат жасау үшін оны XmlNode класс прототипіне жазу керек. Егер оны XML класының прототипінде сақтаса, онда DOM ағашын құрайтын нысандар оны пайдалана алмайды.

Бір қарағанда, Flash-плеер іске асыратын DOM ерекшелігі тег-ге салынған мәтін объектінің түйінін сипаттайтын сипатқа орналастырылмайды, ал оны сақтау үшін жаңа торап нысаны қалыптасады. Бұл түйін мәтін жататын түйіннің еншілес түйіні болып саналады. Тег негізінде қалыптасқан түйін және мәтіннен алынған түйіндер-ELEMENT және TEXT әр түрлі түйіндерге формальды түрде жатады. Олардың нақты сипаттамасы өте ұқсас және олардың арасындағы айырмашылықты тек nodeType қасиеттері арқылы анықтауға болады.

W3C-тен" осы " DOM-де кез келген XML элементінің негізінде торап нысаны құрылады. Бұл нысандар түрі бойынша ерекшеленеді. Әдеттегі _____ басқа емес тег негізінде құрылған торап ELEMENT түріне сәйкес келеді. Тегке салынған мәтінді сипаттайтын түйін TEXT түріне жатады. Атрибутты сақтайтын түйін Attr түріне жатады. DOM 12 түйіндерінің барлық түрлері. Және олардың екеуі ғана — ELEMENT және TEXT — Flash-плеермен жүзеге асырылатын DOM-де бар. Бұл, бұл анализатор XML плеера қолдайды емес, барлық мүмкіндігі осы тілі, сондай-ақ, бұл DOM қолданылатын ActionScript, бірқатар бар ауытқу стандартты (мысалы, әрбір атрибута құрылмайды жеке объект торабының).

Егер торап мәтіні тегпен үзілсе, онда оның әр бөлігі талдағышпен дербес ретінде қарастырылады (сондықтан TEXT типті торап еншілес тораптары болуы мүмкін емес). Мысалы, келесі құжат DOM ағашын береді, онда 4 екінші деңгейлі торап болады: екі ELEMENT түрі және екі TEXT түрі:

```
<text>
<tab / > XML атасы <link addr="http://w3c.org" / > SGML
</text>
```

W3C-тен DOM пайдаланылса, онда қарастырылып отырған құжаттың ағашында үш деңгей болуы мүмкін (Attr түрі түйіні addr тегінің link атрибуты үшін жасалды).

DOM-бұл мінсіз жүйе емес. Оның басты кемшілігі-құжат толығымен бөлшектелуі тиіс, бұрын онымен жұмысты бастауға болады. Және бұл маңызды уақытты талап етуі мүмкін. Сондай-ақ, үлкен XML құжат болған жағдайда Dom ағаш үлкен жадқа ие болуы мүмкін. Нақты уақыт режимінде XML деректерін талдау үшін пайдаланылатын DOM баламасы - SAX (Simple API for XML). Бұл технология оқиғаларға негізделген: SAX кезекті тегті талдаған кезде, ол тиісті хабарлама жібереді. Өкінішке орай, бірақ ActionScript-те SAX моделімен жұмыс істеу мүмкіндігі жоқ.

XML үшін Simple API

SAX (XML үшін Simple API) - XML-DEV тарату тізімінде әзірленген API бар XML-құжаттарды талдау үшін онлайн алгоритм оқиғалар болып табылады. SAX құжаттың Объектілік моделін (DOM) қамтамасыз ететін балама болып табылатын XML құжатынан деректерді оқу механизмін қамтамасыз етеді. Егер DOM жалпы құжатпен жұмыс істесе, яғни пайдаланушының ыңғайлылығы үшін толық AST-XML құжатын құру, SAX-парсерлер құжаттың әрбір бөлігіне XML ретімен жұмыс істейді.

Dom айырмашылығы, SAX үшін ресми спецификация жоқ. Java SAX іске асыру Нормативтік болып саналады. SAS XML-құжаттарды өңдеуге байланысты күй үшін пайдаланылатын DOM-ге қарағанда бір-біріне қарамастан күй туралы құжаттарды өңдейді.

SAX талдағыш тек осы сияқты әрбір талдау оқиға туралы хабарлауға тиіс және, әдетте, бұл ақпаратты дерлік барлық дерлік тастайды (бұл, дегенмен, кейбір нәрселерді ұстап, мысалы, әлі жабық емес барлық элементтердің тізімі, мұндай қате соңғы тегтер сияқты кейінірек қателерді қуып үшін). Осылайша, SAX парсерге қажетті ең аз жады көлемі XML файлының ең үлкен тереңдігіне пропорционалды (яғни. (мысалы, аты және атрибуттары-тег немесе өңдеу нұсқаулығының мазмұны және т.б.).

Бұл жад көп, әдетте, елеусіз болып саналады. DOM парсер, керісінше, құжаттың барлық ұзындығы ұлғайған жады пайдалана отырып, осылайша бастау үшін жады барлық құжаттың көне көрінісін салу керек. Бұл үлкен құжаттар үшін көп уақыт пен орын алады (жады бөлу және осы құрылыс құрылымы уақыт талап етеді). Артықшылығын өтеу, әрине, құжаттың кез келген бөлігін жүктегеннен кейін кез келген тәртіппен алуға болады.

Өйткені SAX оқиғалы сипаты, құжаттарды өңдеу, әдетте, DOM стилінің парсерлері қарағанда әлдеқайда жылдам, сондай-ақ ұзақ, өңдеу соңына дейін басталуы мүмкін. Индекстеу, басқа форматтарға түрлендіру, өте қарапайым пішімдеу және сол сияқты көптеген міндеттер осылайша жасалуы мүмкін. Сұрыптау сияқты басқа да міндеттер, секцияларды ауыстыра отырып, өз мақсатына сілтеме ала отырып, процесс неғұрлым кеш және т. б. көмектесу үшін бір элементте ақпаратты іздеу. күрделі тапсырыстарда құжат құрылымына қол жеткізуді талап етеді және есе SAX өтеді қарағанда DOM әлдеқайда жылдам болады.

Кейбір іске асыру Мұқият сәйкес емес, немесе санат: DOM көзқарас дискідегі өзінің тұрақты деректерін сақтай алады, жылдамдыққа шебер ұйымдастырды (мұндай softquad Автор / Редактор және мұндай DynaText сияқты үлкен браузер / индексаторлар сияқты редакторлар); SAX тәсіл келесі пайдалану үшін ақпаратты кәштеуі мүмкін, ал (кез келген Sax парсер жоғарыда сипатталғандардан артық ақпаратты сақтайды). Мұндай іске асыру Dom / SAX ымыралы, бірақ жиі іс жүзінде өте тиімді.

Dom табиғатынан, дискіден ағынын оқу жалқау бағалау, кәша, виртуалды жады, тұрақты деректер құрылымы немесе басқа да әдістер сияқты әдістерді талап етеді (5557722 АҚШ патентінде сипатталған осындай тәсілдердің бірі). Кейбір DOM парсерлер мүмкіндік бермейді, өйткені XML өңдеу көп RAM кейде мүмкін емес көрінеді. Дегенмен, бұл шектеуді айналып өту үшін жады ретінде дискілік кеңістікті пайдалану негізгі жад қарағанда деректер жинағын сұрыптау артық болуы мүмкін емес.

Кемшіліктер

SAX оқиғалы моделі XML талдау үшін пайдалы, бірақ ол белгілі бір кемшіліктер бар.

XML валидациясының кез келген түрі толық көлемде құжатқа қол жеткізуді талап етеді. Атрибут DTD жарияланған ең қарапайым мысал ID тәсіпаты үшін бірдей мәнді пайдаланатын құжатта тек бір элементті талап етеді. SAX парсерде оны тексеру үшін, барлық ID атрибуттарын қадағалау қажет (олардың кез келгені соңында IDREF атрибутын сілтеме жасай алады); сондай-ақ әрбір IDREF атрибуты шешілгенге дейін. Ұқсас, әрбір элементтің еншілес элементтердің қолайлы тізбегі бар екенін тексеру үшін, баланың элементтердің әрбір ата-анасы үшін байқалғаны туралы ақпарат ата-ана жабылғанша сақталуы тиіс.

Сонымен қатар, XML өңдеудің кейбір түрлері бүкіл құжатқа қол жеткізу қажет. XSLT және XPath, мысалы, өңделген XML ағашында кез келген уақытта кез келген торапқа қол жеткізу мүмкіндігі болуы тиіс. Редакторлар мен шолушылар кез келген уақытта қарауға, өзгертуге және қайта тексеруге мүмкіндігі болуы тиіс. SAX талдағыш бастапқыда мұндай ағашты жасау үшін де пайдаланылуы мүмкін, ал SAX мұндай өңдеу үшін ешқандай көмек бермейді.

оқиғалар синтаксистік талдау кезінде болған кезде пайда болады. Саксада оқиғалар қамтиды (өзгелердің арасында):

XML және мәтін түйіндері

XML элемент басталады және аяқталады

XML өңдеу нұсқаулары

XML-Пікірлер

Кейбір оқиғалар Пікірлер сияқты барлық оңай қайтарылған XML пішіміне сәйкес келеді. Бірақ, XML элементтері XML форматындағы көптеген басқа объектілерді қамтуы мүмкін, сондықтан саксофон оларды XML-кодтың өзі сияқты ұсынады: басында бір оқиға, екіншісі соңында. Шын мәнінде, Sex интерфейсі элементтерде қарастырылмайды, бірақ тегтерге негізінен сәйкес келетін оқиғалар. Синтаксистік талдаудың саксофоны бір бағытты болып табылады; бұрын бөлшектелген деректерді синтаксистік талдау операциясының қайта басталуынсыз қайта оқуға болмайды.

Көптеген саксофон бар-өмір сүруде ұқсас реализация. Іс жүзінде деректер өзгереді, бірақ тұтастай алғанда модель бірдей. Мысалы, XML файлының атрибуттары, әдетте, оқиғаның элементтеріне берілетін атрибуттар және құндық параметрлер ретінде ұсынылады, сонымен қатар жеке оқиғалар түрінде немесе хэш-кесте немесе барлық атрибуттарды ұқсас жинау арқылы орындалуы мүмкін. Екінші жағынан, кейбір іске асыру "init" және "fin" талдаудың басы мен соңы үшін кері шақыруларды қамтамасыз етеді; басқалары жоқ. Осы оқиғалар түрлері үшін нақты атаулар, сондай-ақ іске асыру арасында шамалы өзгереді.

мысал

Келесі XML кодын ескере отырып, құжат:

```
<?xml нұсқасы= " 1.0 "кодтау=" UTF-8"?> >
```

```
<DocumentElement param= "мән">
```

```
<FirstElement>
```

```
    & # xb6; аз мәтін
```

```
< / FirstElement>
```

```
<?some_pi some_attr="some_value"?> >
```

```
<SecondElement param2= "бір нәрсе">
```

Алдын ала мәтін <кірістірілген>кірістірілген мәтін</қатардағы> мәтіннен кейін.

< / SecondElement>

< / Documentelement функциясы>

Бұл XML-SAX парсері арқылы өткен кезде, келесі оқиғалар ретін жасайды:

XML DocumentElement деп аталатын элементті тең жұп атрибутымен бастау

" мәні»

FirstElement деп аталатын XML элементі

XML мәтіні "& # XB6; Some Text " (ескерту: кейбір Бос орындар өзгертілуі мүмкін)

XML элементі firstelement деп аталатын соңы

Мақсатты some_pi және some_attr = "some_value" оқиғаны өңдеу нұсқаулығы(мазмұн мақсаты тек мәтіннен кейін, дегенмен, осы мысал ретінде XML атрибуттарының синтаксисі өте жиі ұқсайды)

Парам2 атрибуты бар SecondElement деп аталатын XML іске қосу элементі "бір нәрсе»

XML мәтіні "Pre-Text" тең деректер түйіні"

Инлайн деп аталатын XML элементі

XML мәтіні деректер түйіні " енгізілетін мәтінге ұқсас»

XML элементі Инлайн деп аталатын соңы

XML мәтіні " Post-мәтінге тең деректер түйіні.»

XML элементі SecondElement деп аталатын соңы

XML элементі DocumentElement деп аталатын соңы

Үлгіден жоғары бірінші жол XML-декларация болып табылатынына және өңдеу нұсқаулығы емес екеніне назар аударыңыз; ол өңдеу нұсқаулығы оқиғасы ретінде хабарланбайды (кейбір SAX жүзеге асыру тек XML хабарландыру үшін ғана жеке оқиғаны ұсынады).

Нәтиже жоғарыда өзгеруі мүмкін: SAX ерекшелігі бұл мәтін фрагменті бірнеше жүйелі мәтіндік оқиғалар түрінде ұсынылуы мүмкін деп әдейі мәлімдейді. Көптеген парсерлер, мысалы, сандық сілтемелер үшін жеке мәтіндік оқиғаларды қайтару. Осылайша, жоғарыда келтірілген SAX мысалында талдаушы әр түрлі оқиғалар серияларын жасай алады, олардың бөлігі қамтуы мүмкін:

FirstElement деп аталатын XML элементі

XML мәтіні "& # XB6; " (Юникода u + 00b6 символы)

XML мәтіндік түйіні, "Some Text" тең деректер»

XML элементі firstelement деп аталатын соңы.

3 бөлім. PHP бағдарламалау

PHP-ге кіріспе

PHP-бұл Web –серверге жіберілетін скриптердің көмегімен Web-беттерінің динамикалық генерациясына қажетті бағдарламалау тілі. Сіз бетті PHP және HTML-дің көмегімен ашасыз. Сайтты пайдаланушы бетті ашқанда, сервер html-код қосылған PHP операторларын орындайды және нәтижені пайдаланушының браузеріне жібереді. Бұл әрекет дәл осылай ASP және Cold Fusion-ның көмегімен жасалады. Дегенмен ASP және Cold Fusionға қарағанда, PHP ашық бастапқы кодты өнім болып табылады және платформалы тәуелсіз. PHP Windows NT және Unix-тің көптеген версияларында жұмыс жасайды. Ол Apache –дегі модуль ретінде жіберіле береді. Егер жіберілу Apache модулі түрінде болса PHP оңай және жылдам жұмыс жасайды. Бұл кезде процессті жасауға байланысты туындайтын қосымша шығындар болмайды. Сондықтан нәтижесі тез шығады және сервердегі сақтаудағы шығынды азайтатын mod_perl-ді орнатудың қажеттілігі болмайды.

Сіздің құжат бетіңіздегі әртүрлі операциялардан басқа сіз PHP көмегімен HTTP-тақырыптарды қалыптастырып HTTP орната аласыз. Сонымен қатар аутентификацияны басқарып пайдаланушыны басқа бетке бағыттай аласыз. PHP берілгендер қорына еруге үлкен мүмкіндіктер ашады. Бұл сізге PDF құжаттарды инерациялаудан, XML-дегі грамматикалық талдауды жасауға мүмкіндік береді.

PHP операторлары сіздің бетіңізге Web-беттерді қойып береді, сондықтан арнайы ортада жүзеге асырудың қажеттілігі туындамайды. Сіз PHP-кодының блогын <?php тегінен бастап, оны ?> тегімен аяқтайсыз. Бұл тегтердің арасындағылардың барлығы PHP код ретінде интерпритацияланады. PHP тілінің синтаксисі Си және Perl синтаксисіне ұқсас келеді сіз ауспалыларды оларды пайдаланудан бұрын жарияламауыңыз керек. Массивтермен Хэмтер оңай жүзеге асырылады. Дегенмен PHP Apache құрылған жағдайда бәрінен жылдам жұмыс жасайды. PHP Web –сайтында оны Microsoft IIS және Netscape Enterprise Server орналастырудың реті бар. Егер сізде PHP ді орнатуға бағдарламалық қамтамасыз етудің көшірмесі жоқ болса, оны сізге ресми Web –сайттан ала аласыз. Ол жерден сіз PHP дің барлық ерекшелігі мен қызмет бейнеленген түсіндірілген жетекшілікті таба аласыз.

Соңғы он бес жылдары интернеттің қиял-ғажайып дамуы және адамдардың арасындағы қарым-қатынастың жаңа әдістерін белгіледі. Бұл құбылыстың алдыңғы өлкелеріне (WWW) World Wide Webде болады.

PHP тарихы 1995 жылдан тәуелсіз бағдарламашы (Rasmus Lerdorf) Рasmus Лердорф сайтта оның онлайндық резюмеcін оқып шыққан келушілердің санын есептеуге арналған Perl/CGI нің сценарийн жазудан басталады. Оның сценарийи екі есепті шешті : келушінің деректерінің тіркелуі және келушілердің санын - web бетке шығару. WWW дамыту әлі тек қана басталды, бұл есептерді шешуге арналған ешқандай да арнайы құралдар болмады, сондықтан авторға көп сұрақтар лап берді. Лердорф (PHP) Personal Home Page немесе (гипермәтін процессоры) Hypertext Processor аталған өз құрал-сайманын тарата бастады.

Лердорфаның PHP құрал-сайманының шулы жетістігі PHP-ның кеңейтулерін өңдеуге әкеп соқты. Кеңейтулердің бірі HTML формада енгізілген мәліметтерді символдық айнымалыларға түрлендірді, осыдан оларды басқа жүйелерге шығаруға мүмкіндік берген. Қойылған мақсатқа қол жеткізу үшін Лердорф өңдеулерді жасауды ендігәрі Perl ден C-ге өтуге шешті. Сол кездегі PHP-ның құрал-сайманының кеңейтуі 2.0-ші PHP пайда болуына, немесе (Personal Home Page Form Interpretator) PHP-FIні келтірді. Дүние жүзінен бағдарламашылар 2.0 версияны жетілдіруге қатысты. Жаңа PHP версиясы ерекше мәлімділікпен пайдаланды, және

өндеушілердің негізгі командасы жақында іріктелді. Олар HTML-ға программалық кодтың енгізуінің бастапқы тұжырымдамасын тікелей сақтады және РНР-дің 3.0-ші версиясының пайда болуына келтірген лексикалық талдаудың механизмін қайтадан жазды.

РНР 4 бұл қозғалыста жұмыс жасаушы ретінде 2000 жылы жарыққа шықты. Өндірісті жоғарылатуда қосымша ретінде РНР 4 тің мүмкіндіктері зор еді. РНР 4-тің шығуымен ол интернет доменінің 20 пайыздан көбінде қолданыла бастады.

2000-2004 жылдар арасында 4 версияны жұмсарту үшін белсенді жұмыстар жалғасты. Шамамен сол кезден бастап ақ РНР Group жаңа версияның мүмкіндігін ойластыруға кірісті. Бірінші кезекте тілдің объектілік мүмкіндіктерін күшейту шешілді. Бұл оның жүзеге асыру үшін ірі жобаларды пайдалануға мүмкіндік берді. 5-ші версияны жарыққа шығару үшін қосымша уақыт бөлінді. Оған көптеген мамандар қатысты олардың ішінен Стерлинг Хьюза және Маркус Бергерді атап өткен жөн болар.

Ақыры 2004 жылдың шілдесінде РНР 5 ресми түрде жарыққа шығады. Бірінші кезекте жоспарланғандай өндеуде объектілермен жұмысты барлық механизм жұмылдырылды. Егер алдыңғы версияларда РНР –дегі объектілік-жобалаушылық бағдарламалау минимальды деңгейде мүмкін болатын болса, ал РНР 5 объектілік бағдарламалауды жүзеге асыруда көптеген потенциалдарға ие. Онымен қоса РНР XML мен жұмыс жасауда бағалы кеңейтулермен, берілгендердің әртүрлі қайнар көзіне байыды.

РНР 4-дегі XML мен жұмыс жасауға арналған барлық негізгі кітапканалар күрделі өндеуге ұшырады. Мынадай танымал кеңітілер SAX, DOM, және XSLT қазір libxml2-нің құрал-сайманын пайдаланады. Сонымен қатар тағы да жаңа екі кеңейтулер қосылды-SimpleXML және SOAP. Simple XML XML- берілгендермен жұмысты айтарлықтай оңайлатады. Бұдан алдын РНР-да XML мен жұмыс бұншалықты қарапайым болмаған еді.

SOAP -тың кеңеюі РНР да XML –хабарламаның көмегімен басқа қосымшалармен ақпарат ауысатын сценарий құруға мүмкіндік береді. Мысалы HTTP. РНР қосымшалы қазіргі таңдағы танымал веб-сервистермен интеграциялауға мүмкіндік туындайды.

РНР 5 My SQL (My SQL Improved) жаңа кеңейтулер My SQL – сервер версиясымен 4.1.2 және жоғарылармен жұмыс жасауға арналған. My SQL - ға тек қана процедуралық емес сонымен қатар объекті жобалаушылық интерфейс жүзеге асады. Бұл модульдің қосымша мүмкіндіктері SSL-ді, транзакцияны бақылауды қосады.

SQLite–тің кеңейтілуі мәліметтерді кәдімгі файлдарда сақтайтын қосымшаны құрастыруға мүмкіндік береді. Ол SQL интерфейсін пайдалануға мүмкіндік тудырады. SQLite – нің маңызды ыңғайлылығы – пайдаланылу қарапайымдылығы. SQLite процедуралықпен қатар мәліметтермен жұмыс жасау үшін мықты объектілік - жобалаушылық мүмкіндіктері бар. SQLite-нің басқа ерекшеліктері –жоғары жылдамдық, басқарудағы күрделі механизмдердің жоқ болуы және оңай тізгіштік. Көптеген эксперттер РНР -нің бұл кеңейюінің үлкен танымалдылыққа ие болатындығын жоспарлап отыр. РНР 5-тің басқа кеңейтілуі өзіне төмендегілерді қосады:

- Tidy –HTML-құжаттарды сараптау және түзету үшін.
- Perl - Perl сценарийлерді пайдалану.
- SPL – кластармен интерфейстердің стандарты жиынтығын анықтайтын ZE2 –нің кеңеюі.
- -PDO -РНР-нің мәліметтеріне әмбебап интерфейс ұсынады.

- -PIMP-GD модульінің орнына келген, графикамен жұмыс жасауға арналған жаңа кітапхана.
- Бұдан басқа PHP 5-те есте сақтаудың әлдеқайда жылдам диспетчері пайдаланылады.
- PHP 5 REAR құрылымын нақты жүзеге асыруға керекті қасиеттермен толықтырылған.

Негізгі синтаксис. Деректер типі. Айнымалылар. Өрнектер.

Негізгі синтаксис

Біріншіден, PHP синтаксисіне қатысты білу керек, ол HTML-кодына қалай кіреді, интерпретатор бұл PHP тілінде код екенін біледі. Алдыңғы дәрісте біз бұл туралы айттық. Қайталанбайтын боламыз, мысалдарда біз көбінесе нұсқаны қолданамыз `<?php ?>`, кейде қысқартылған нұсқасы `<? ?>`.

Нұсқауларды бөлу

PHP бағдарламасы (яғни және басқа бағдарламалау тілінде) – командалар жиынтығы (нұсқаулар). Бағдарламаны өңдеушіге (парсерге) бір команданы екіншісінен ажырату қажет. Ол үшін арнайы бөлгіш таңбалар қолданылады. PHP нұсқауы Си немесе Perl сияқты бөлінеді – әрбір өрнек үтірмен аяқталады.

Жабатын тег"`?>`" сондай-ақ нұсқаудың аяқталуын білдіреді, сондықтан оның алдында үтірмен нүкте қойылмайды. Мысалы, кодтың келесі фрагменттері тең:

```
<?php
echo "Hello, world!"; // нүктелі үтір
                        // команданың соңында
                        // міндетті
?>
<?php
echo "Hello, world!" ?>
<!--нүктелі үтір
жазылмайды "-->
```

Коментарийлер

Көбінесе бағдарламаларды жазу кезінде кодқа ешқандай әсер етпейтін, тек оны түсіндіретін кодқа қандай да бір түсініктеме жасау қажеттілігі туындайды. Бұл үлкен бағдарламаларды жасау кезінде және бірнеше адам бір бағдарламамен жұмыс істеген жағдайда маңызды. Бағдарламада түсініктеме болса, оның кодында түсіну әлдеқайда оңай. Сонымен қатар, егер тапсырманы бөліктер бойынша шешсе, шешімнің бөлінбеген бөліктері бұдан әрі олар туралы ұмытпау үшін түсініктеме беруге ыңғайлы. Барлық бағдарламалау тілдерінде бағдарлама кодына түсініктеме енгізу мүмкіндігі қарастырылған. PHP бірнеше түсініктеме түрлерін қолдайды: Си, C++ және Unix қабығы. Символы `//` және `#` білдіреді начало однострочных пікірлер `/*` және `*/` – сәйкесінше жыл басындағы және аяғындағы многострочных түсіндірмелер.

```
<?php
echo "Менің атым Вася";
// Бұл бір жолдық пікір
// с стилінде
echo " Менің Петров Тегі";
/* Бұл көп жолдық пікір.
Мұнда бірнеше жол жаза аласыз.
Бағдарламаны орындау кезінде барлығы
мұнда (қиып алынған),
болады игнорировано. */
echo "мен PHP оқып INTUIT.ru";
```

```
# Бұл стильдегі пікір
# Unix қабығы
?> >
```

Деректер типі

PHP сегіз қарапайым деректерді қолдайды.

Төрт скалярлық түрі:

boolean (логикалық) ;
integer (тұтас) ;
float (күбылмалы нүктесімен) ;
string (жол).

Екі аралас түрі:

array (алап) ;
object (нысан).

Және екі арнайы түрі:

resource (ресурс) ;
NULL.

PHP-да айнымалы түрлердің айқын хабарлауы қабылданбаған. Бұл айнымалы қолданылатын контекстке байланысты бағдарламаны орындау кезінде интерпретатордың өзі жасаған жөн. Барлық аталған деректер түрлерін ретімен қарастырайық.

Boolean түрі (түйреуіш немесе логикалық түрі)

Бұл қарапайым түрі мәннің шынайылығын білдіреді, яғни осы түрдегі айнымалы тек екі мағынаға ие болуы мүмкін – шындық шын немесе false өтірігі .

Түйреуіш түрін анықтау үшін TRUE немесе FALSE кілт сөзін қолданады. Екі тіркелімге тәуелді.

```
<?php
$test = True;
?> >
```

2.5-мысал. Логикалық түрі

Логикалық айнымалылар әр түрлі басқару құрылымдарында (циклдарда, шарттарда және т. б.) қолданылады. Логикалық түрі бар, яғни тек екі мәнді, ақиқатты немесе өтірікті қабылдай алады, сондай-ақ кейбір операторлар (мысалы, теңдік операторы) да мүмкін. Олар сондай-ақ қандай да бір шарттарды тексеру үшін басқару конструкцияларында пайдаланылады. Мысалы, шартты конструкцияда оператор немесе айнымалы мәнінің шынайылығы тексеріледі және тексеру нәтижесіне байланысты қандай да бір әрекеттер орындалады. Мұнда шарт шын немесе жалған болуы мүмкін, ол айнымалы және логикалық типті операторды көрсетеді.

```
<?php
// Оператор '= =' теңдікті тексереді
// және қайтарады
// булево мағынасы
if ($know = False) {/егер $know
                        // маңызы бар
                        // false

echo " PHP оқы!";
}
if (!$know) {/сол
                // және одан жоғары, яғни тексеру
                // $know мағынасы бар ма
                // false

echo " PHP оқы!";
}
/* оператор ==
   $know айнымалысының мәні
```

```

"PHP оқы". Егер сәйкес келсе, онда
true қайтарады, әйтпесе – false.
Егер true қайтарылса, онда орындалады
фигуралы жақшаның ішінде қандай */
if ($know = " PHP оқыңыз")
{ echo "оқуды бастады"; }
?> >
2.6-мысал. Логикалық типті пайдалану

```

Integer түрі (бүтін)

Бұл түр $Z = \{\text{бүтін сандар жиынынан санды көрсетеді...}, -2, -1, 0, 1, 2, \dots\}$. Бүтін санаудың ондық, он алтылық немесе сегіздік жүйесінде, қалауы бойынша алдыңғы " - "немесе" + "белгісімен көрсетілуі мүмкін.

Егер сіз сегіздік санау жүйесін пайдалансаңыз, 0 санын (нөлді) алдын ала аласыз, он алтылық жүйені пайдалану үшін 0x санына қою керек.

```

<?php
# ондық сан
$a = 1234;
# теріс сан
$a = -123;
# сегіздік Сан (эквивалентті
# 83 ондық жүйеде)
$a = 0123;
# он алтылық Сан (баламалы
# 26 ондық жүйеде)
$a = 0x1A;
?> >

```

Тұтас өлшемі платформаға байланысты, бірақ, әдетте, екі миллиард (бұл 32 биттік белгі) максималды мәні. PHP-ді толық қолдайды емес.

Егер бүтін түр шегінен асатын санды анықтасаңыз, ол өзгермелі нүктемен Сан ретінде түсіндіріледі. Сондай-ақ, егер сіз жұмыс нәтижесі бүтін шегінен асатын Сан болатын операторды пайдалансаңыз, оның орнына қалқымалы нүкте бар сан қайтарылады.

PHP-да тұтас бөлу операторы жоқ. 1/2 нәтижесі 0.5 құбылмалы нүктесімен Сан болады. Сіз әрқашан оны аз жаққа дөңгелектейді немесе стандартты ережелер бойынша дөңгелектейтін round () функциясын пайдалана аласыз. Айнымалыны нақты түрге түрлендіру үшін, айнымалының алдында жақшада қажетті түрді көрсету керек. Мысалы, \$a=0.5 айнымалысын бүтін түрге түрлендіру үшін (integer)(0.5) немесе (integer) \$a жазу немесе қысқартылған жазуды (int)(0.5) пайдалану қажет. Мұндай принцип бойынша типтерді анық келтіру мүмкіндігі барлық деректер түрлері үшін бар (әрине, әрқашанда бір түрдің мәнін басқа түрге аударуға болады). Біз PHP контекстке байланысты автоматты түрде жасайды, өйткені барлық түрлер үйлесімділігіне тереңдей алмаймыз.

Float түрі (құбылмалы нүктелі сандар)

Өзгермелі нүктелі сандар (олар екі дәлдік сандар немесе нақты сандар) келесі синтаксистердің кез келгенінің көмегімен анықталуы мүмкін:

```

<?php
$a = 1.234;
$b = 1. 2e3;
$c = 7E-10;
?> >

```


Өзгермелі нүктелі санның өлшемі платформаға байланысты, бірақ максимум, әдетте, ~1. 8e308 шамамен 14 ондық сан.

String түрі (жолдар)

Жол-таңбалар жиынтығы. PHP-да таңба-бұл байт сияқты, бұл дәл 256 түрлі таңба бар дегенді білдіреді. Бұл сондай-ақ PHP Unicode кіріктірілген қолдауы жоқ дегенді білдіреді. PHP-де жолдар өлшеміне шектеу жоқ, сондықтан олардың ұзындығына алаңдатты ешқандай себеп жоқ.

PHP жол үш түрлі жолмен анықталуы мүмкін:

- бір тырнақшаның көмегімен ;
- Қос тырнақша көмегімен ;
- heredoc-синтаксис.

Айнымалылар

PHP-дегі айнымалы доллар белгісімен белгіленеді, содан кейін оның аты болады. Мысалы:

```
$ my_var
$My_var және $My_var айнымалылары әртүрлі.
```

Айнымалылардың аттары PHP-дегі басқа атаулар сияқты ережелерге сәйкес келеді: айнымалының дұрыс аты әрпінен немесе астын сызу символынан кез келген санда әрпімен, сандармен немесе астын сызу Символдарымен басталуы тиіс.

PHP 3-те айнымалылар әрқашан мән бойынша берілген. Яғни, сіз айнымалы өрнек берген кезде, түпнұсқа өрнектің барлық мәндері осы айнымалыға көшіріледі. Бұл, мысалы, бір айнымалы мән екінші берілгеннен кейін, олардың біреуінің өзгеруі басқасының мәніне әсер етпейді.

```
<?php
$first = ' Text '; // $first тағайындау
                // мәні
                // 'Text '
$second = $first; // $second тағайындау
                // мәні
                // $first айнымалы
$first = 'New text'; // өзгертеміз
                // мәні
                // $first
                // 'New text 'дегенге
echo "first атымен айнымалы" . $first <br>";
    // $first мәнін шығарамыз
Echo "second атымен айнымалы" . $ second";
    // $second мәнін шығарамыз
?> >
```

Мысал 2.2. Мән бойынша тағайындау

Бұл скрипт жұмысының нәтижесі келесідей болады:

First атымен айнымалы Жаңа мәтінге тең

Second атымен айнымалы мәтінге тең

PHP 4, сонымен қатар, айнымалы мәндерді берудің тағы бір жолын ұсынады: сілтеме бойынша тағайындау. Сілтеме бойынша айнымалы мәнін тағайындау үшін, бұл мағынада аты болуы керек, яғни ол қандай да бір айнымалы болуы керек. Бір айнымалының мәні сілтеме бойынша басқа айнымалыға берілгенін көрсету үшін, бірінші айнымалы атының алдында амперсанд & белгісін қою керек.

Жоғарыда бірдей мысал қарастырайық, тек сілтеме бойынша second айнымалы first мәнін белгілейміз:

```
<?php
$first = ' Text '; // $first тағайындау
                // 'Text' мәні
```

```

$ second = & $first;
/ * $Second арқылы $first сілтемені жасаймыз.
  Енді осы айнымалылардың мәндері
  әрқашан сәйкес болады */
// $First мәнін өзгертеміз
// 'New text 'дегенге
$first = ' New text ';
echo "first атымен айнымалы" .
      "тең .$first <br>";
// екі айнымалының мәнін шығарамыз
Echo "second атымен айнымалы" .
      "$second тең";
?> >

```

Мысал 2.3. Сілтеме бойынша тағайындау

Бұл скрипт келесілерді шығарады:

First атымен айнымалы Жаңа мәтінге тең.

Second атымен айнымалы Жаңа мәтінге тең.

Яғни \$first айнымалысымен бірге \$second айнымалы өзгерді.

Операциялар.

Арифметикалық операциялар

PHP-дегі математикалық әрекеттерді орындауға арналған операторлар:

```

<?php
$answer = 5 + 4; /* қосу */
$answer =$answer - 5; /* алу */
$answer =$answer / 2; /* бөлу */
$answer =$answer *$answer; /* көбейту */
$answer =$answer % 3; /* бөліндіден шыққан қалдық */
?>

```

Егер екі аргумент те бүтін сан болса, онда нәтижесі бүтін сан болады. Ал егер операндтардың бірі қалқыма нүктелі сан болса, онда жауабында бүтін сан шықса да нәтижесі қалқыма нүктелі сан болады. Мысалы, $0.5 + 1.5 = 2.0$, бірақ 2 емес.

Бұл бес оператордың қысқартылған нұсқалары да бар. Мысалы $a = a + b$ өрнегін қысқаша түрде былай жазуға болады: $a +=b$. Ал a айнымалысының мәнін екі еселендіру керек болса, $a *= 2$ деп жазса болғаны. Бұл $a = a * 2$ деген өрнекке сәйкес.

Айнымалының мәнін бірге арттыру мен кемітудің қысқартылған операторлары бар: инкремент (++) және декремент (--). Бұл унарлы операторларды жазған кезде орналасуында айырмашылықтар бар. Мысалы келесі скриптті жазып, нәтижесін көріңіз:

```

<?php
$a = 1;
echo($a++."
");
$a = 1;
echo(++$a);
?>

```

Нәтижесінде келесідей екі сан шығады:

```

1
2

```

Бірінші жағдайда \$a айнымалысының мәні echo шығару командасы орындалып болғаннан кейін бірақ бірге артады. Екінші жағдайда \$a айнымалысы 2-тең болып, одан кейін бірақ echo-ның көмегімен экранға шығады.

Логикалық операциялар

Салыстыру операторларына «кіші» (<), «кіші не тең» (<=), «үлкен» (>) және «үлкен не тең» (>=) операторлары жатады. Олар берілген екі мәнді салыстырып, true немесе false логикалық мәндерін қайтарады.

Мысалы:

```
<?php
$a = 3 < 4;
echo($a);
?>
```

Егер \$a-ның мәні ақиқат болса, онда скрипт бірді шығарады, егер жалған болса, ештеңе шығармайды.

Жолдарға қолданылатын операциялар

Жолдарға қолданылатын маңызды операция - оларды біріктіру (конкатенация) операциясы болып табылады. PHP-де жолдарды конкатенациялау үшін нүкте символы пайдаланылады: «(.)»

Мысалы:

```
<?php
$a = "Hello, ";
$b = "World";
$c = $a . $b;
echo($c);
?>
```

Арифметикалық операциялардағы сияқты конкатенацияның қысқартылған нұсқасы бар:

Ол мына өрнекпен сәйкес келеді:

«Нүкте» операторының көмегімен бірнеше жолды конкатенациялауға болады:

Логикалық операциялар

Логикалық операторлар бульдік шарттарды тексереді. Төрт негізгі бульдік шарттар бар:

ЖӘНЕ (and немесе &&), ИЛИ (or немесе ||) , ЕМЕС (!) және БОЛДЫРМАЙТЫН ЕМЕС (xor).

Логикалық операцияларды қолдану нұсқалары:

```
$a = true
$b = false
$c = $a && $b
$d = $a || $b
$e = !$a
$f = $a xor $b
```

Бұдан келесідей нәтижелер аламыз:

\$c – false (операндардың бірі жалған),

\$d – true (операндардың бірі ақиқат),

\$e – false (\$a емес),

\$f – true (операндардың бірі ақиқат, ал екіншісі жалған).

PHP-де жолдар мен массивтермен жұмыс. Функциялар.

Массивтер инициализациясы

PHP-де массивтердің инициализациясының екі түрі бар. Олардың біріншісі мәндері қарапайым меншіктелген массив элементтерінен тұрады:

```
<?
$scar[] = "nissan";
$scar[] = "honda";
echo($scar[1]); // "honda" шығарады
?>
```

Массив индексі ашық көрсетуге болады:

```
<?
$scar[0] = "nissan";
$scar[1] = "honda";
echo($scar[1]); // " honda" шығарады
?>
```

Егер массив элементтерін хабарлау кезінде айнымалылар ашық индексациялармен, индексациясыз араласса, онда индексі көрсетілмеген элементке PHP осыған дейін ең үлкен қолданылған индекске кейінгі бірінші қолжетімді индексті меншіктейді. Мысалы, егер біз индекстері 10, 20 және 30 болатын элементтермен массив құрсақ, содан кейін индексі көрсетілмеген элемент құрсақ, онда автоматты түрде оған 31 индексі меншіктеледі:

```
<?
$scar[10] = " nissan";
$scar[20] = " honda";
$scar[30] = "opel";
$scar[] = "mitsubishi"; echo($scar[31]);
?>
```

Нәтиже mitsubishi.

Массивтерді анықтаудың балама тәсілі **array()** құрылымының қолдануынан тұрады:

```
<?
$scar = array("nissan ","honda"); echo($scar[1]); // "honda" шығарады
?>
```

Индекстерді ашық түрде көрсету үшін мұнда => операторы қолданылады:

```
<?
$scar = array("nissan ", 5 => " honda ", " opel "," mitsubishi ");
echo($scar[0]); echo("<br>"); // " nissan " шығарады echo($scar[5]); echo("<br>");
// " honda " шығарады echo($scar[6]); echo("<br>"); // " opel " шығарады
echo($scar[7]); // " mitsubishi " шығарады
?>
```

Массив индекстері ретінде жолдар да болуы мүмкін:

```
<?
$scar = array("pc" => " nissan ", "lr" => " honda "); echo($scar["lr"]);
echo("<br>"); // " honda " шығарады echo($scar["pc"]); // " nissan " шығарады ?>
```

Массивтерді тексеріп шығуға арналған foreach циклы

PHP4-те циклда массивті тексеріп шығуды **foreach** циклы арқылы ұйымдастыруға болады, оның синтаксисі келесі түрде болады:

```
foreach (array as [$key =>] $value)
```

```
{
statements;
}
```

Бұл циклдың мағынасы қарапайым: әрбір массив элементінің өтуі кезінде осы элементтің индексі **\$key** айнымалысына орналасады, ал

\$value айнымалысына – оның мәні. Осы екі айнымалының атаулары еркін болып келеді. Мысал:

```
<?
$car = array("nissan ", " honda ", " opel "," mitsubishi ");
foreach($car as $index => $val)
{
echo("$index -> $val <br>");
}
?>
0 -> nissan
1 -> honda
2 -> opel
3 -> mitsubishi
```

Синтаксистен көрініп тұрғандай **\$key** айнымалысы міндетті емес және түсіріліп қалуы мүмкін:

```
<?
$car = array("nissan ", " honda ", " opel "," mitsubishi ");
foreach($car as $val)
{
echo("<li>$val</li>\n");
}
?>
```

Нәтиже

- nissan
- honda
- opel
- mitsubishi

Мысалы:

<p>Массивтер деректері: Автомобиль маркасы, түсі, иесінің аты-жөні</p>	<p>Сұратулар:</p> <ol style="list-style-type: none"> 1. Қызыл түсті «Toyota Corolla» автомобильдерінің саны 2. Иванов И.И. иесіне тиесілі автотранспорттық құралдардың тізімі
---	--

Кластар мен объектілер.

PHP тілінде тұрақтылар **define()** функциясы арқылы белгіленеді:

```
define(CONSTANT, value)
```

Осы функцияның бірінші параметрі – константа аты, екіншісі – оның мағынасы. Константаны қолданған кезінде оған келесідегілер сілтеме көмегімен жіберіледі:

```
<?
define(CONSTANT1,15);
define(CONSTANT2,"\x20"); // бос қалдыру коды define(CONSTANT3,"Hello");
echo(CONSTANT1);
echo(CONSTANT2); echo(CONSTANT3);
?>
```

Қате шығарады. Бұл ескерту PHP тілінде қателерді бақылаудың жоғары деңгейінде өту себебінен туындайтын шығар, ол Денверде (E_ALL) үнсіздік бойынша қойылған. Қателерді шығарудың мұндай режимі ұсынылатын болып есептеледі және скриптерді кейінге қалдырғанда өте маңызды болып есептеледі. Бірақ та дайын скрипттердің көбісі қателерді бақылаудың төменгі деңгейін талап етеді.

Сіз қателердің бақылау деңгейін сәл болса да төмендете аласыз. Оны келесі әрекеттер арқылы жүзеге асыра аласыз:

```
Скрипттің ішіне мына жолды жазыңыз: Error_Reporting(E_ALL &
~E_NOTICE); Шығатыны: 15 Hello.
```

Сонымен қатар **defined()** функциясы бар, ол константаның анықталғанын тексереді:

```
<?
define(CONSTANT,"Hello"); if(defined("CONSTANT"))
{
echo("<p>CONSTANT is defined</p>");
}
?>
```

Жоғарыда айтып кеткендей, PHP тілі айнымалылар типтеріне қатысты белгілі бір жеңілдіктер береді, яғни бір айнымалыны қолданып, жолмен де, санмен де жұмыс істеуге болады. Бірақ та оған қарамастан, PHP тілінде айнымалылардың белгілі бір типтері болады, олар айнымалылармен жұмыс жасағанда көрсетілуі мүмкін:

Осы фрагменттің нәтижесі алдыңғы кодтың нәтижесімен бірдей болып шығады.

settype() функциясынан басқа PHP тілінде типтердің түрлендіру әрекетін типтер түрленуін жүзеге асыратын операторлар көмегімен де орындауға болады. Типтердің түрленуі айнымалының алдына жақшаға алынған оның жаңа типін көрсету арқылы жүзеге асырылады:

```
$var = (int)$var;
```

Сәйкесінше, осы кодтың нәтижесінде PHP integer типін қайтарады:

```
<?
```

```
$var = "5"; // string muni
```

```
$var = (int)$var; // int muni түрлендіреміз echo(gettype($var));
```

```
?>Операторлар Арифметикалық операторлар
```

	Қосу
--	------

	Айырма
	Көбейту
	Бөлу
	Модуль бойынша қалдықты есептеу (мысалы: 5 % 2=1)

Логикалық операторлар

or немесе 	НЕМЕСЕ
Xor	ЖОЯТЫН НЕМЕСЕ
and немесе &&	ЖӘНЕ
!	ЕМЕС

Логикалық айнымалылары немесе дұрыс айтатын болсақ **Boolean** типтегі айнымалылар тек екі мағынаға ие: **true**(ақиқат) және **false**(жалған). **true** және **false** мағыналарын кез келген уақытта 1-ге (0 санынан кез келген абсолютті сан) және сәйкесінше 0-ге ауыстыруға болады. **true** және **false** мағыналарын 1 мен 0 сандарына ауыстыра алу мүмкіндігі – бұл арнайы **true** және **false** мағыналары болмаған C тілінен қалған «мұра».

```
<?
$flag = true; // ақиқат if($flag)
{
echo "<p> flag айнымалысы true</p> ақиқат мағынасына ие";
} else {
echo "<p> flag айнымалысы false</p> жалған мағынасына ие";
}
?>
```

Скрипт жұмысының нәтижесі мына түрде болады:

flag айнымалысы true мағынасына ие

Шарт бір болған кезінде нәтиже де жеңіл шығады. Шарт көбірек болған сайын жұмысқа логикалық операторлар енеді:

```
<?
$flag1 = true; // ақиқат
$flag2 = true; // ақиқат
if($flag1 && $flag2) // ЖӘНЕ $flag1 ЖӘНЕ $flag2 ақиқат
{
echo "<p>шарт: true (екі флаг та ақиқат)</p>";
} else {
echo "<p>шарт: false (флагтың біреуі жалған)</p>";
// Жоқ, олар екеуі де жалған
} ?>
```

Бұл скрипт мынаны шығарады: *Шарт: true (екі флаг та ақиқат)*. Егер оператордың тым болмаса біреуі **false** (немесе екеуі де) мағынасына ие болса – онда шығатыны: *Шарт: false (Флағтардың біреуі жалған)*.

- integer;
- string;

- **boolean;**
- **double;**
- **array;**
- **object.**

Сонымен қатар **gettype()** функциясы бар, PHP тілі белгілеген айнымалыға типті қайтарады:

```
<?
$var = "5";
$var1 = 5; echo(gettype($var)); echo "<br>"; echo(gettype($var1));
?>
```

Шығатыны: string integer

Тағы **settype()** дейтін функция бар, ол ашық түрде айнымалының типін анықтайды:

```
<?
$var = "5"; echo(gettype($var)); settype($var,integer); echo "<br>";
echo(gettype($var));
?>
```

PHP деректер базасымен жұмыс.

1. Администрациялау

Ыңғайлы болу тәртібімен ДБ әкімшілендіру тәсілдері:

- phpMyAdmin (өте ұсынамын!)
- Базаны ауыстыратын сценарийді жазу (мысал қараңыз)
- mysql.MySQL пакетінде exe
- mysql_manager.exe (шамандық жағында ғана мүмкін сияқты)

Әсіресе бірінші әдісті ұсынамын. Онымен ALTER TABLE сұрауларын зерттеуге қажеті жоқ, ADD COLUMN және T .p. Оның үстіне, "мұндай сұрақтар, Елші, кондакпен шешілмейді" - егер сіз автоматты түрде база немесе кесте құрылымын өзгерту қажет болған кезде? Екінші әдіс туралы бірнеше сөз. Бұл phpMyAdmin және mysqldump утилитасы туралы білмей, мен қолданған айналма технология деп айтуға болады. Скриптте базаны алып тастайтын және оны қайтадан жасайтын командалар жазылады. Бір кездері көмектесті, бірақ бұл тағы бір рет айтайын, айналма технология, "тіреуіш".

Болашаққа: егер сізде ДБ пайдаланатын бірнеше сайт болса, онда үй сервері шегінде бірнеше база құрыңыз. Бұл сервердің жұмысын жеңілдетеді және кестелердің шатасуы мүмкіндігін болдырмайды. Жалпы, ДБ-мен жұмыс істеу ережелері сайтпен бірдей-басқалардан бөлек директорияда ұстау.

2. ДБ серверімен байланыс

...mysql_connect функциясының көмегімен жүзеге асырылады: \$connect = mysql_connect (, ,); әдепкі бойынша mysql-серверде пайдаланушылар кестесінде root пайдаланушысы бар, ол тек localhost-а-мен ғана кіре алады, онда mysql сервері тұрған компьютерден билесіз. НАЗАР аударыңыз! "Localhost-а-мен қол жеткізу" Сіздің PHP скриптыңыз бар дегенді білдіреді, ал сіз кез келген басқа компьютерден оған жүгіне аласыз.

mysql_connect функциясын шақырған кезде не болады? Сіздің скриптыңыздың басталуымен, php ол және оның айнымалылары туралы ақпарат үшін өз жадында орын бөледі. Орындалатын скрипт туралы ақпаратта, соның ішінде, деректер базасымен қосылыстар туралы ақпарат сақталады. Айнымалы \$connect-бұл ақпарат сақталатын орын көрсеткіш. Айнымалы бұл қалғандары

сияқты бірдей - егер сіз функцияны пайдалансаңыз, онда оған жүгіну үшін жаһандық айнымалыларды жариялау керек.

Нәліктен айнымалы қолданылады? Бұл жұмыс істеу үшін бірнеше дерекқор серверлерін пайдалану қажет болған жағдайда (немесе, мысалы, үлкен қауіпсіздікті қамтамасыз ету үшін Сіз әр түрлі артықшылықтар болуы мүмкін әр түрлі логиндерді пайдаланасыз). Мұндай жағдайларда әрбір сұрауда команда қандай арнада жүретіні туралы айқындық қажет. Егер сіз тек бір ғана қосылымды пайдалансаңыз, оны сұрау мүмкіндіктерінің параметрлерінде (олар туралы - төменде) көрсету қажет емес - php бірінші (және бұл жағдайда жалғыз) орнатылған қосылымды тауып, оны пайдаланады.

3. Сұраныс-нәтижелерді іріктеу және өңдеу

ДБ-ға сұраныс функциясының жұмыс механизмі қосылу функциясындағыдай: функциялар сұраныс параметрлері және (қажет болса) қосылыстар беріледі, ал нәтиже ауыспалы болып жазылады.:

```
$result = mysql_db_query (string деректер базасы, string сұрау [, айнымалы байланыс]);
```

немесе

```
$result = mysql_query (string сұрау [, айнымалы қосылыстар]);
```

НАЗАР аударыңыз! Деректер қоры көрсетілмеген `mysql_query` функциясын пайдалану үшін қолданылатын деректер базасын алдын ала таңдау керек:

```
mysql_select_db(string деректер базасы);
```

Енді бізде `$result` айнымалысы бар. Бұл сұрауды орындау нәтижесіне көрсеткіш. Онда қанша болса, онда кесте жолдарының. Бұл жолдарды `mysql_fetch_row` және `mysql_fetch_array` арқылы алуға болады:

```
echo "
```

```
";
```

```
while ($row = mysql_fetch_array ($result))
```

```
echo"
```

```
";
```

```
echo " ";
```

`mysql_fetch_array` функциясы көрсетілген айнымалыға (бұл жағдайда `$row`), оның индекстері - өрістердің аттарын (егер сіз сұрау өрістері тізімінде `table` жазсаңыз) береді. `field`, ал массив индексі `field` болады). `mysql_fetch_row` массивін береді, оның индекстері - 0 бастап сандар.

Қандай функцияны пайдалану жақсы? Егер сіз жұлдызшаны сұрасаңыз, яғни кестенің барлық өрістері, ал өрістерді белгілі бір ретпен шығару қажет (мысалы, кестеде бөрік сызылған кезде), `mysql_fetch_array` пайдалану жақсы. Егер сіз бір-екі-үш өрісті сұрасаңыз, олардың тізбегін анық біле отырып, `mysql_fetch_row` - бұл бағдарлама кодының көлемін азайтады.

4. Сұрау-әрекеттер

Бұл `DELETE` және `UPDATE` командалары. Мұндай сұраулар - "құқықтарда" `SELECT` сияқты, сондықтан команданы серверге жіберу-`mysql_query` (`mysql_db_query`) тәсілімен жүзеге асырылады. Бірақ бұл жағдайда функция нәтижені қайтармайды:

```
$result = mysql_query ("SELECT * FROM sometable");
```

бірақ

```
mysql_query ("DELETE FROM sometable WHERE id=...");
```

Тиісінше, егер біз сұрауды-іріктеуді орындасақ және нәтижені өзгеріске жазбасақ, деректер еш жерде сақталмайды.

5. Сұраулар қателерін өңдеу

Соңғы қате туралы хабарды `mysql_error` функциясы арқылы алуға болады:

```

echo " деректер қорының қатесі. MySQL жазады:", mysql_error();
Егер функцияның нәтижесі айнымалыға жазылса, оны тексеруге болады:
$result = mysql_query($request);
if (!$result)
echo " деректер қорының қатесі. MySQL жазады:", mysql_error();
else {
echo "
", $row " ", $row
["field1"]," ["field2"], "
";
while ($row = mysql_fetch_array ($result))
echo"
";
echo " ";
};
Егер айнымалы жазылмаса, онда солай:
$request = " UPDATE (...)";
mysql_query($request);
if (!mysql_error())
echo " Деректерді жаңарту сәтті өтті!";
else echo " деректер қорының қатесі. MySQL жазады:", mysql_error();
Егер сұраныс автоматты түрде жасалса, сұраудың өзін де шығаруға болады
(айнымалыны құру және оны функция параметрі ретінде пайдалану пайдалы).

```

Сервер жағында бағдарламалау.

Программалардың ұсынылған кешені MS Windows 95/98/Me/NT/2000/XP басқаруымен жұмыс жасайтын PHP қолдаудағы толық бағалы Apache веб-серверін, Perl және машинадағы MySQL серверін орнатады. Көбінесе орнату кезінде ешқандай баптау қажет етілмейді.

Базалық кешеннің құрамы:

1. Apache: орындалатын файлдар, дистрибутивті және адаптелген конфигурациялық файлдар.
2. PHP: орындалатын файлдар, Apache веб-сервері үшін керекті модуль, дистрибутивті және адаптелген конфигурациялық файлдар, GD кітапханасы - әлі GIF форматынсыз.
3. MySQL: орындалатын файлдар, қателер туралы орыс пен ағылшын тіліндегі хаттамалар, mysql деректер қоры.
4. Деректер қорын басқару панелі MySQL - phpMyAdmin, сонымен қатар жаңа қолданушыны қосатын MySQL скрипті.
5. Perl: орындалатын файлдар, модульдері болмайды.
6. sendmail (/usr/sbin/sendmail) үшін ұстап тұратын «бітеуіш», ол хаттарды жібермей тұрып, /tmp!/sendmail директориясына сақтайды.
7. Виртуалды хосталардағы автоматты іздеу жүйесі және hosts жүйелік файлының жаңартылуы, сонымен қатар Apache серверінің конфигурациясы. Жаңа виртуалды хостты қосу әрекеті (немесе үшінші деңгей домені) қарапайым /home каталогын жасауға және кешенді қайта жүктеуге тұжырымдалады. Барлық өзгертулер конфигурациялық және жүйелік файлдарға автоматты түрде енгізіліп тұрады, бірақ та сіз осы процесті хосттардың шаблондары арқылы басқара аласыз.

Серверді жүктеу мен тоқтату әрекеттері кешеннің директориясында орналасқан etc\Run.exe және etc\Stop.exe файлдары көмегімен жүзеге асырылады.

- серверді жүктеу: etc\Run.exe файлы
- серверді тоқтату: etc\Stop.exe файлы
- тоқтату мен виртуалды дискті сөндіру: etc\utils\SwitchOff.exe файлы

Серверді тек Stop.exe командасымен сөндіріңіз, басқа жағдайда

MySQL қорындағы барлық мәліметтер жоғалып кетуі мүмкін.

Кешенді орнатып, бірінші жүктеме жасағаннан кейін бірден браузерді жүктеген дұрыс, ол үшін оған URL кодын енгізу керек: "http://localhost". Сіз жүйенің тестілік бетіне түсуіңіз қажет. Керекті сілтемелерді шерту арқылы сізге әрбір компонентті тестілеуіңізге болады.

HTML-құжаттары /home/<имя_хоста>/www директорияларында орналасуы керек (/home/localhost/www). Үнсіздік бойынша 4 виртуалды хост жасалған:

- http://localhost (әр түрлі утилиттер мен тестілеу скрипттері);
- http://test1.ru;
- http://test2.ru;
- http://custom-host:8648 (өзінің IP-адресі және порты бар хост);
- http://127.0.0.1 (IP-адресі бойынша ғана қолжетімді хост).

/usr/local директориясы программалық компоненттерінен тұрады – орындалатын және конфигурацияланған файлдар Apache, PHP, MySQL. Авторлар осындай орнатуды бекітілген UNIX ортасына жақындатып жасауға тырысты. Бірақ та маңызды айырмашылықтар бар - пакеттің барлық компоненттері тиісті директорияларына сәйкес келмейді. Қай жағдайды алсақ та, UNIX ЖО-мен толық сәйкестікті қамтамасыздандыруға болмайды, бірақ та орындалатын және конфигурацияланған файлдардың берілген орнатуы сервердің компоненттерінде жеңіл бағдар табуға мүмкіндік береді.

/usr/bin директориясы - ерекше болып табылады, себебі онда Perl тілінің интерпретаторлары орналасқан, ол интерпретаторлар кешеннің жақсы жұмыс жасауы үшін қажет. Бұл интерпретаторлар қолданушыға ешқандай міндеттер салмайды, себебі осы тілде жасалатын барлық жұмыс «мөлдір» түрде жасалады. Адам кейбір программалар Perl тілінде жазылатынын да білмейді.

```
<?php phpinfo()
?>
```

Броузердің серверге үндеуі кезіндегі **php-көрінісінің жүзеге асу процесін** қарастырайық. Сонымен, алдымен браузер .php кеңейтіліміндегі бетті талап етеді, одан кейін web-сервер программаны PHP машинасы арқылы өткізеді және нәтижені html-коды ретінде шығарады.

MySQL деректер қоры.

MySQL – дүние жүзінде ең көп қолданылатын, қайнары тегін және ашық, реляцияланған мәліметтер қоры жүйесі (**RDBMS**). Серверлік бағдарлама ретінде, бірнеше қолданушыларға бірнеше мәліметтер қорын қолдануды қамтамасыз етеді. MySQL сөзіндегі "My" сөзі, бағдарлама жасаушысы Майкл Видныйос-тың (Michael Widenius) қызының аты - "My" сөзінен алынған. Ал **SQL** фразасы - Құрылымдасқан Тапсырыс Тілі (Structured Query Language) дегенді білдіреді.

MySQL, проект қайнарын GNU General Public License (GPL) және әр түрлі жеке меншік лицензияларында шығарды. MySQL-ды кезінде MySQL AP атты Шведттық фирма демеу еткен, ал қазір оны толығымен Oracle корпорациясы сатып алды.

Толық функционалды мәліметтер қорын қажет ететін, қайнары тегін проекттер жиі MySQL-ды қолданады. Серверді коммерциялық проекттерде қолдану үшін, қосымша функциялармен ұсынылатын бірнеше нұсқалары бар. MySQL-ді қолданатын бағдарламаларға TYPO3, Joomla, WordPress, phpBB, MyBB, Drupal және де басқа LAMP бағдарламаларын жатқызуға болады. MySQL сонымен қатар Google, Wikipedia, Facebook және Twitter сияқты дүниежүзілік және кең ауқымды web-бағдарламаларында қолданылады.

Веб-бағдарламаларында MySQL кең қолданылады және кең тараған LAMP бағдарламаларының ішінде басты компонент рөлін атқарады. LAMP сөзі - "Linux, Apache, MySQL, Perl/PHP/Python" сөйлемінің акронимі.

MySQL - қайнары тегін бағдарламалар мен Интернеттегі танымал Flickr, Nokia.com, YouTube, Wikipedia, Google, Facebook және Twitter сайттарында қолданылады.

Платформалар және бағдарламашының қолMySQL - C және C++ тілдерінде, ал SQL парсері және sql_lex.cc деп аталатын лексикалық анализаторы уасс тілінде жасалған.

MySQL көптеген жүйелер платформаларында жұмыс істейді. Оларға кіретіндер: AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, Mac OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos және Tru64. Сонымен қатар OpenVMS жүйесіне портталған түрі бар.

Мәліметтер қорымен жұмыс істеу үшін, көптеген бағдарламалау тілдеріне кітапханалар жасалынған. Мысалы, C# және VB.NET үшін MySQL Connector/Net, ал Java үшін JDBC драйвері. Бұнымен қатар ODBC интерфейсін қолдайтын ASP және ColdFusion сияқты тілдер үшін MyODBC кітапханалары бар. Ал URL арқылы кез-келген web-клиенттің MySQL серверімен араласуға мүмкіндік беретін HTSQL технологиясы, MySQL-серверімен бірге келеді.

Windows-қа арналған MySQL Workbench бағдарламасы

MySQL көбінесе реляциялық мәліметтер қоры ретінде қолданылатындықтан ол, жүйені және ондағы мәліметтерді басқаруға арналған клиенттік бағдарламамен келмейді. Оның орнына, өзінің командалық жолмен орындалатын құралдарын немесе басқалар жасаған клиенттік бағдарламаларын жүктеп алып қолдануға болады.

Ресми бағдарламалар

Графикалық интерфейсте MySQL мәліметтер қорын басқаруға және олардың структурасын жасау үшін, MySQL AB фирмасы MySQL Workbench атты бағдарламасын шығарды. Ол бұғанға дейін болған MySQL GUI Tools бағдарламасының орнына келді. MySQL Workbench қолданушыларға келесідей мүмкіншіліктер ұсынады:

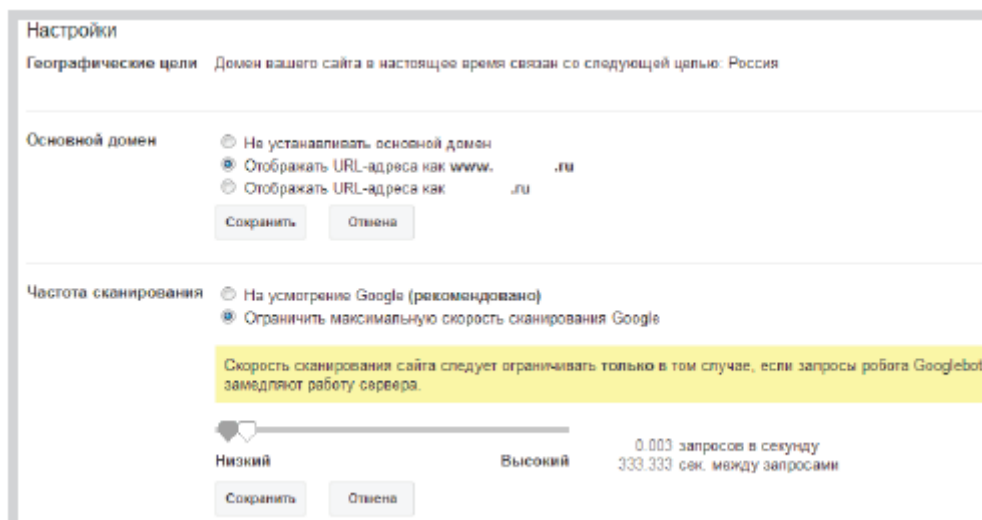
- Мәліметтер қорын дизайндау және модельдеу
- SQL бағдарламалау - MySQL Query Browser бағдарламасының орнына
- Мәліметтер қорын басқару - MySQL Administrator-дың орнына

MySQL Workbench екі нұсқада келеді, Олар, қайнары тегін қоғамдастық нұсқасы (Community Edition) және қосымша мүмкіншіліктері бар стандартты нұсқасы (Standard Edition).

Оңтайлы жұмыс істеу үшін веб-жүйені теңшеу.

"Конфигурация" → "теңшеу" бөлімінде сайттың негізгі айнасы — www-мен немесе www-сыз баптауға болады, сайттың географиялық аймаққа тиесілігін

көрсету, сондай-ақ Google роботының сайтты сканерлеу жиілігін көрсету. Соңғы орнату Google роботының сайтқа жүгінулерін шектеуге мүмкіндік береді; бұл, мысалы, баяу хостинг жағдайында қажет болады. Дегенмен, көптеген жағдайларда Google ұсынған мәнді қалдыру ұсынылады.



Google вебмастерлеріне арналған құралдардың "сайт сілтемелері" бөлімінде Яндексте жылдам сілтемелерге ұқсас сілтемелерді теңшеуге болады. Google Яндекс сияқты, қандай сілтемелерді көрсететінін өзі анықтайды, бірақ ол сізге пайдалы жылдам сілтемелерді көрсетуге көмектеседі.

"URL параметрлері" бөлімінде Google роботын Сіз көрсеткен параметрлерді URL мекенжайына қандай мақсатпен пайдаланатынын білуге болады. Бұл функция "парақтар", "сұрыптау" және т. б. жиі пайда болатын динамикалық url мекенжайларын индекстеу қажет болғанда пайдалы болады.

Егер сайтты басқа доменге ауыстыру қажет болса, "мекенжайды өзгерту" бөлімі қажет. 301 редиректі қолданыңыз, ал Google веб-сайтындағы индекстеу процесін жеделдету үшін пайдаланушылар қай сайтқа бағытталғанын қолмен көрсетуге болады.

"Пайдаланушылар" бөлімінде басқа пайдаланушылар үшін "веб-шеберлер құралдары" қызметін басқару құқығын баптауға болады.

Коммерциялық емес жоспар жүйесінің модульдерімен жұмыс істеу әдістері.

1С-Битрикс-сайтты басқару жүйесі

"1С-Битрикс" – бұл сайттың мазмұнын және оған орналастырылған материалдарды басқаруды жүзеге асыратын жүйе, ол ақылы болып табылады, бірақ өзінің ыңғайлылығы арқасында интернет дүкенін жайлылықпен орнатуға мүмкіндік береді. "1С-Битрикс" студияда сайтты құруға тапсырыс беруді шешетін және түсінікті басқарумен кеңейтілген функционалды алуды қалайтын адамдар үшін тамаша нұсқа болып табылады. Бұл жүйе фото, Бейне, құжаттама және деректер базасы сияқты әртүрлі деректер түрлерімен жұмыс істей алады.

"1С-Битрикс" ерекшеліктерінің бірі - "1С-бухгалтериямен" синхрондау мүмкіндігі бар. Үлкен дерекқорларды көшіру көп уақыт алуы мүмкін, бірақ компания 1С-Битрикс қолдауымен сайтқа тапсырыс беруді шешсе, ол осы базаларды тез және автоматтандырылған түрде тасымалдау мүмкіндігіне ие болады.

Клиент өз бизнесінің түріне және жүйенің болжамды жүктелуіне байланысты "1С-Битрикс" таңдай алады:

Bitrix Нұсқасы. Шағын Бизнес" шағын ұйымдар мен интернет-дүкендер үшін өте қолайлы. Бұл сайтта тауарларды оңай өңдеуге және дилерлік желіні қадағалауға мүмкіндік береді, бірақ бір тауарға бірнеше баға қою функциясы болмайды.

Битрикстің Бизнес нұсқасына бірнеше баға, жеке жеңілдіктер орнату мүмкіндігі, пайдаланушыға тауарды бақылау мүмкіндігі және тағы басқалар кіреді.

Битрикс 24: клиентпен жұмыс

"Битрикс 24 "" 1С-Битрикстен " бірінші кезекте ол сайттың жұмысына және клиенттермен өзара әрекеттесуге бағытталмаған, бұл сервис қызметкерлердің жұмысын тікелей компания ішінде ұйымдастыру үшін қызмет етеді. Оның негізгі міндеті өнімді барынша арттыру үшін қызметкерлерді синхрондау мүмкіндігін жасау. Бұл компанияның жеке әлеуметтік желісі.

Неге бұл пайдалы? "Битрикс 24" пайдалану компанияның барлық жұмыс процесін бақылауға және басқаруға және қызметкерлерге қажетті ақпаратты санаулы секундта жеткізуге мүмкіндік береді. Сондай-ақ, ол қарапайым және түсінікті интерфейс бар, бұл әсіресе маңызды, егер көптеген компанияларда әртүрлі жастағы адамдар жұмыс істейді. Жаңалықты игеру интуитивті және уақыт пен күш жұмсауды талап етуі керек.

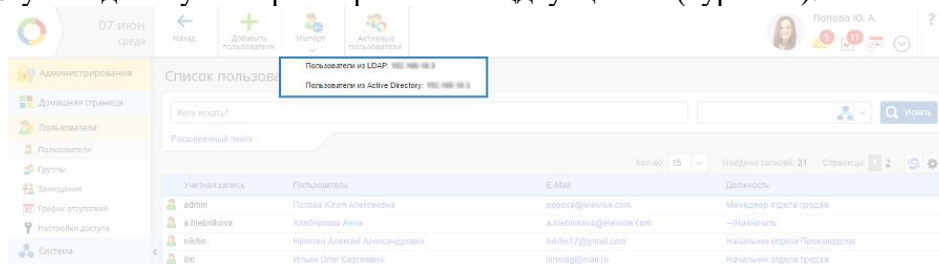
Қорытынды

Битрикс компаниясы автоматтандыруға жауапты функционалдың өндірісіне шоғырлана отырып, ақпараттық технологияларды дамытуға үлкен үлес қосты. Олардың тауарлары арзан интернет-дүкенге тапсырыс беруді шешкендерге және персонал арасында ең көп ұйым маңызды болып табылатын, түсінікті басқарумен және шағын компанияларға барынша функционалдылықты алғысы келеді.

LDAP сервері арқылы пайдаланушыларды импорттау әдістері.

Егер ELMA сервері LDAP қызметімен жұмыс істеу үшін орнатылса, осы қызмет арқылы ELMA пайдаланушылары тізімін импорттау мүмкіндігі бар. LDAP қосылымын қалай орнату керектігі туралы LDAP біріктіру бетінде толығырақ сипатталған.

Пайдаланушыларды LDAP серверінен импорттау үшін Импорттау батырмасын басу қажет және ашылмалы мәзірден LDAP пайдаланушылары немесе Active Directory пайдаланушылары тармағын таңдау қажет (сурет. 1).

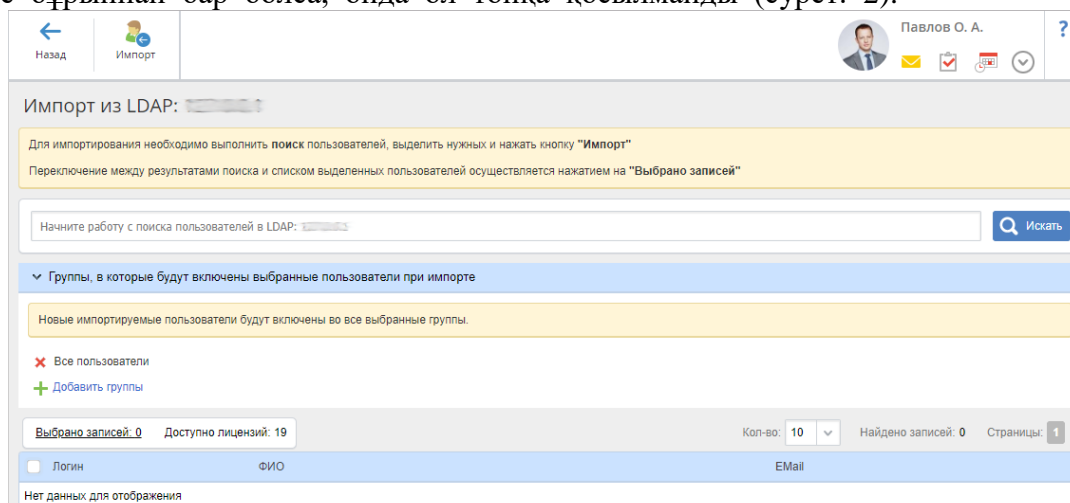


Сур. 1. LDAP пайдаланушыларды импорттау түймесі

Әрбір жаңа кестеленген импортталатын пайдаланушы ELMA жүйесіндегі "барлық пайдаланушылар" тобына қосылады.

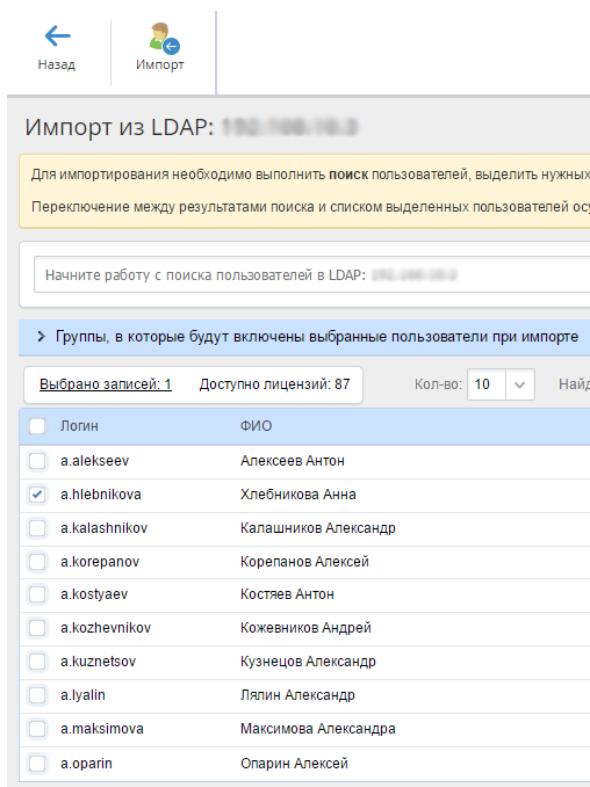
Ашылған бетте топты қосу түймесі арқылы жаңа пайдаланушылар импорттау кезінде автоматты түрде түсетін басқа топтарды қосу мүмкіндігі бар. Айта кету

керек, тек жаңа пайдаланушылар таңдалған топтарға түседі. Егер пайдаланушы жүйеде бұрыннан бар болса, онда ол топқа қосылмайды (сурет. 2).



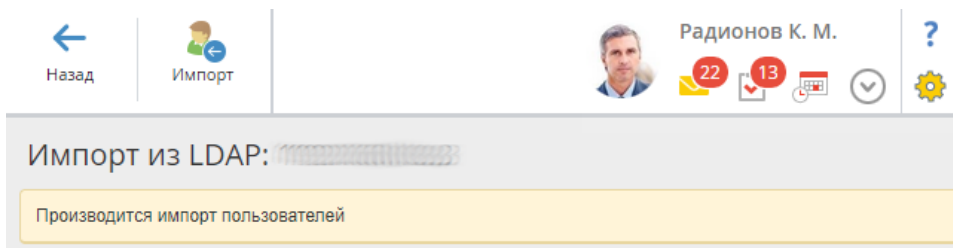
Сур. 2. LDAP пайдаланушыларды импорттаудың бірінші кезеңі

Іздеу түймесін басқаннан кейін (сурет. 3) импорт үшін қол жетімді пайдаланушылар тізімі пайда болады (сурет. 3). Тізімде қажетті пайдаланушылар құсбелгісін белгілеп, Импорттау түймесігін басу керек. Таңдалған пайдаланушыларды LDAP-дан ELMA жүйесіне импорттау процедурасы іске қосылады.

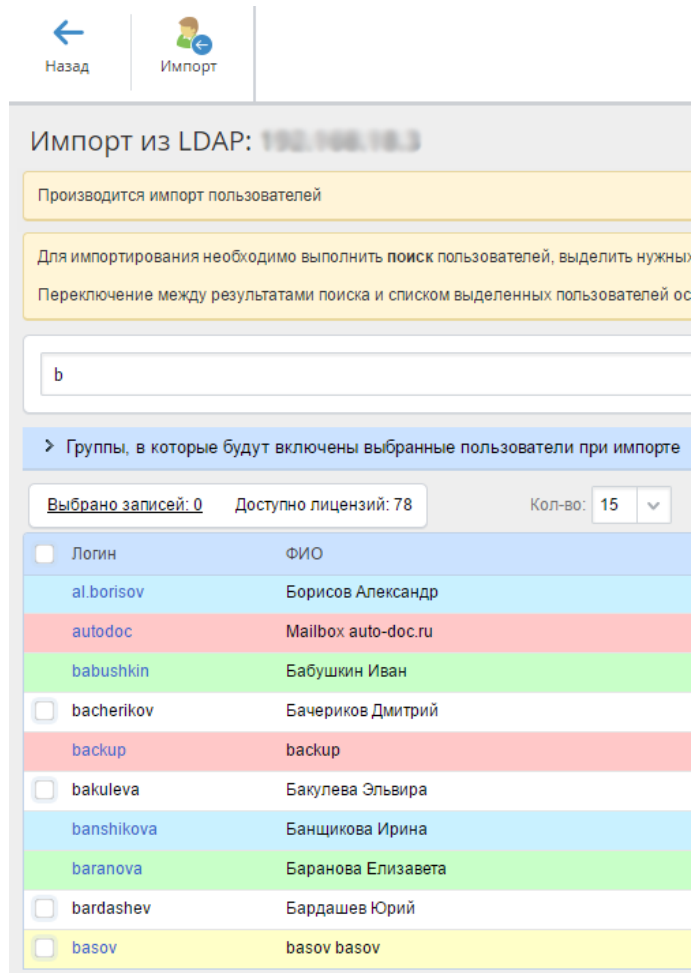


Сур. 3. LDAP-дан ELMA импорттау үшін табылған пайдаланушылар тізімі

Импорт рәсімін іске қосқаннан кейін тиісті хабарлама бейнеленеді (сурет. 4). Импорт фондық режимде жүзеге асырылады: импорттың аяқталуын күтпестен, беттегі жұмысты жалғастыруға болады.



- Сур. 4. Пайдаланушыларды импорттаудың басталғаны туралы хабарлама
-
- LDAP-дан импортталған пайдаланушылардың есептік жазбалары тікелей ELMA жүйесінде сақталмайды. Мұндай пайдаланушыларды жүйеде аутентификациялау кезінде олар енгізген ақпарат LDAP-та сақталатын деректермен салыстырылады.
- Іздеу жолында пайдаланушы логинінің бір бөлігін теріп, Іздеу батырмасын басуға болады-жүйе логиндерінде символдардың енгізілген комбинациясы бар пайдаланушылар тізімін көрсетеді –
- Егер импорт рәсімі екінші, үшінші және т.б. рет орындалса, пайдаланушылар тізімде түсі бойынша ерекшеленеді (сурет. 5):
- * жасыл-пайдаланушы ELMA-ға сәтті импортталды;
- • қызыл-пайдаланушы ELMA-ға импортталды, бірақ бұғатталған немесе белсендірілмеген, яғни жүйеде авторлана алмайды;
- * ақ-пайдаланушы әлі импортталмаған;
- * сары-пайдаланушы ELMA жүйесінде де, LDAP базасында да бар, бірақ деректер сәйкес келмейді. Импорттау кезінде жүйеде Пайдаланушының деректері (соның ішінде пароль) LDAP мәліметтеріне ауыстырылады;
- * көк-пайдаланушы ELMA жүйесінде және LDAP базасында бар, оның деректері толығымен сәйкес келеді. Пайдаланушы бұрын басқа модульден (Active Directory сияқты) импортталса, бұл түс пайда болады.
-



Сур. 5. Қайта импорттау кезіндегі пайдаланушылар тізімі

Сайтты рұқсатсыз кіруден қорғау құралдары.

Қазіргі заманғы шындық веб-қосымшаларға үнемі өсіп келе жатқан шабуылдарды көрсетеді — жүйелердің компрометациясы 80%-ға дейін веб-қосымшадан басталады.

Шабуылдың құралдары мен техникаларының саны ұлғайған кезде сайттың қол жетімділігін қамтамасыз ету, веб-қосымшаны немесе оның компоненттерін контентті бұзу мен ауыстырудан қорғау қиынға соғады. Техникалық мамандар мен әзірлеушілердің күш-жігеріне қарамастан, қорғаныс жағы веб-қосымшаның бұзылуынан кейін қорғау шараларын іске асыра отырып, дәстүрлі түрде қуып жетуші позицияға ие болады. Веб-сайттар көпшілік қол жетімділігіне, әрдайым сапалы жазылмаған кодқа, серверлік бөлікті баптауда қателердің болуына, сондай-ақ АҚ АҚ қызметі тарапынан бақылаудың жоқтығына байланысты шабуылдарға ұшырайды, осылайша зиянкестерге сыни деректерге қол жеткізуді қамтамасыз етеді

Осыған байланысты веб-қосымшаның архитектурасын ескеретін және сайттың жұмысын кідіртпейтін қорғаныс құралдарын пайдалану қажеттілігі туындайды.

Статистика көрсеткендей, көптеген веб-қосымшалар, сондай — ақ, жылдардағыдай, бұл әр түрлі инъекциялар, инклюдтар, клиент-сайд шабуылдар, сондықтан қорғаныс құралы келесі осалдықтарды пайдалануға бағытталған шабуылдарды анықтай және бұғаттай білуі тиіс:

SQL айдау-sql инъекция;

Remote Code Execution (RCE) — кодты қашықтан орындау;
Cross Site Scripting (XSS) - ақаралық скриптинг;
Cross Site Request forgery (CSRF) — ақ аралық сұраулар;
Remote File Inclusion (RFI) — қашықтағы инклюд [1];
Local File Inclusion — LFI) - жергілікті инклюд; Auth Bypass-авторландыруды айналып өту;

Insecure Direct Object reference-нысандарға қауіпсіз тікелей сілтемелер;

Bruteforce-күпия сөздерді таңдау.

Қолданбалы деңгейде қорғау

Қолданбалы деңгейдегі Протокол — хаттама жоғарғы (7-ші) деңгей OSI моделінің желілік өзара әрекетті қамтамасыз етеді желісі мен пайдаланушы. Деңгей пайдаланушының қолданбаларына деректер базасына сұрау өңдеушісі, файлдарға қол жеткізу, электрондық поштаны жіберу сияқты желілік қызметтерге қол жеткізуге рұқсат береді. Қолданбалы деңгейде қорғау ең сенімді болып табылады. Зиянкестер пайдаланатын осалдықтар көбінесе 8 пайдаланушының деректерді енгізудің күрделі сценарийлеріне сүйенеді, бұл оларды басып кіруді табудың классикалық жүйелерінің көмегімен қиын анықтайды .

Веб-қосымшаларға жаңа сценарийлерді анықтауға және шабуылдарды іске асыруға мүмкіндік беретін көздердің бірі қазіргі заманғы компаниялардың нақты инфрақұрылымын имитациялайтын "кіруге тестілеу зертханалары" болып табылады. Зертханаларға бүкіл әлем бойынша ақпараттық қауіпсіздік бойынша 9000-ға жуық маман қатысады. Зертхана объектілеріне бағытталған шабуылдарды талдау бұзушының модельдерін жасауға және шабуыл векторларын іске асыруға мүмкіндік береді.

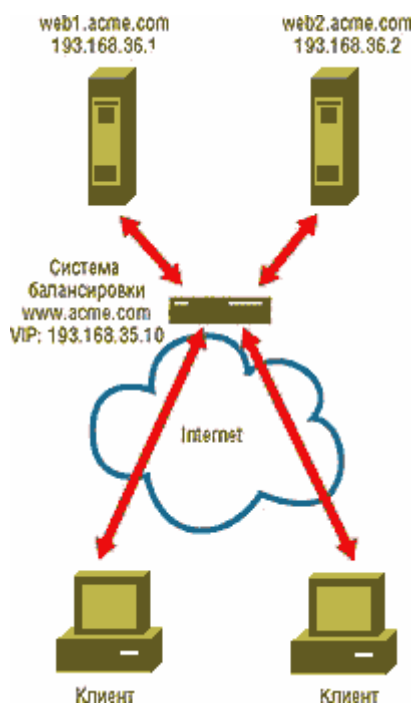
Жоғары жүктеме кезінде сайттың өміршеңдігін қамтамасыз ету құралдары.

Web-серерлердің өнімділігін теңшеуге арналған құралдар

Электрондық коммерция индустриясы дамуын жалғастыруда және бүгінгі күні барлық жаңа компаниялар өз клиенттерімен Web-тің көмегімен қарым-қатынас жасай бастайды. Материалдарды тез және сәтсіз ұсына алатын жоғары өнімді сайт жаңа клиенттерді тартуға ғана емес. Ол электрондық сауда кәсіпорындарының табысты қызметінің және олардың бәсекеге қабілеттілігін арттырудың аса маңызды алғышартына айналады. Әлеуетті клиенттер бір кездері тітіркендіргіш "тыныш" торапқа оралады, онда келуші өз сұранымына жауап береді, ал тіпті жауап бермейді. Сондықтан кез келген ұйымда Web торабының инфрақұрылымын жоспарлау кезінде оның өнімділігін арттыру жөніндегі шараларға ерекше назар аудару қажет. Сайттың жылдамдығын арттырудың бірнеше жолы бар: өткізу жолағын ұлғайтуға, жоғары өнімді желілік жабдықты орнатуға, Web үшін тиімді қосымшаларды әзірлеуге, Web-сервердің бағдарламалық және аппараттық компоненттерін оңтайландыруға және жаңартуға, сондай-ақ Web ортасында кәштеу технологиясын қолдануға болады. Web торабының өнімділігін арттырудың тағы бір жолы Web-серверлер мен сайттар санын көбейту және оларға материалдардың "айналы" көшірмелерін орналастыру болып табылады. Осылайша, жалпы жүктемені жүйенің барлық компоненттері бойынша бөлуге және сервердің клиенттік сұраныстарды өңдеудің ішкі рәсімдерін орындау кезінде ақпаратты қайтару уақытын қысқартуға болады. Сонымен қатар, қолданыстағы серверлер де сақталады, өйткені оларды пайдаланудан шығару және жаңамен ауыстыру қажет емес. Сонымен қатар, бір-бірімен байланысқан, бір-бірімен байланысқан, бір-бірімен байланысқан, бір-бірімен байланысқан, бір-бірімен байланысқан. Web-серверлер арасында жүктемені

бөлу үшін әдетте кез келген клиентке сайтты құрайтын кез келген Web-сервердің IP-мекенжайын айналмалы түрде беру мүмкіндігін қарастыратын циклдік таңдау (round-robin feature) деп аталатын DNS функциясы қолданылады; нәтижесінде барлық Web-серверлер трафикпен теңдей жүктеледі. Алайда, бұл механизм жекелеген Web-серверлердің аппараттық және бағдарламалық компоненттерінің мүмкіндіктері тең емес орталарда жеткілікті тиімді емес. Pentium III екі 450 мегагерц процессорымен және 1 Гбайт сыйымдылығы бар жадымен жабдықталған жүктемелі ортада Windows 2000 Server жүйесі бір 300 мегагерц Pentium II және 256 Мбайт жады процессорымен Windows NT Server жүйесіне қарағанда ауыр жүк көтере алады. Алайда, осы жүйелер арасында DNS қызметінің циклдік үлгісі процедурасы бойынша ешқандай айырмашылық жоқ; сонымен қатар, бұл функция қандай да бір Web-сервердің қол жетімділігі туралы "көріністері жоқ", себебі компьютердің жұмыс істеп тұрғанын немесе істен шыққанын анықтай алмайды.

Көп ұзамай Жабдықтаушылар жүктемені теңестіру жүйелерін - бағдарламалық өнімдерді шығаруды ретке келтірді, олар жүктемені бірнеше серверге бөледі. Сонымен қатар, олар Web-серверлердің бас тартуға төзімділігін арттырады: бір машина істен шыққан жағдайда деректер пакеттерін басқа серверге немесе сайтқа жібереді. Осылайша, күту уақыты азаяды, ал өңделмеген сұраулар саны азаяды. Жүктемені теңгеру жүйесін тек бір Web-сайт болған кезде де, бірнеше тораптармен жұмыс істеген кезде де пайдалануға болады. Жүктемені теңестіру жүйесі қандай және олар қалай жұмыс істейтіні туралы түсінік ала отырып, жүктемені теңестіру құралын таңдау кезінде ескеру қажет олардың ең маңызды сипаттамаларын анықтауға болады. Что такое система балансировки нагрузки



1-ЭКРАН. Бір сайт үшін теңгеру жүйесі.

Web-серверлердің жүктемесін теңгеру жүйесі - бұл ақпараттық ресурстың айналы көшірмелері сақталатын машиналар тобынан ең аз жүктелген немесе неғұрлым қолайлы Web-серверге клиенттік сұраныстарды бағыттауға арналған құрал. Клиент серверлердің тұтас тобына хабарласады деп күдіктенбейді: олардың барлығы оған бір бірыңғай виртуалды сервер түрінде ұсынылады. Мысалы, біз бір Web-сайтқа қызмет көрсетеміз және екі Web-сервер бар делік: web1.acme.com IP мекенжайы 193.168.36.1 және web2.acme.com IP мекенжайы 193.168.36.2. Бұл конфигурацияны 1-суретті көрсетеді. Internet пайдаланушыларына біздің сайтты

ұсына отырып, жүктемені теңестіру жүйесі виртуалды компьютердің атауын пайдаланады (бұл атау болсын www.acme.com), сондай-ақ виртуалды IP-мекен - жайы (VIP-мекен-жайы-айталық, 193.168.35.10). Виртуалды жүйенің атын және тиісті VIP-мекенжайды біздің екі веб-сервермен байланыстыру үшін жүйенің атын және оның VIP-мекенжайын DNS серверінде жариялауымыз керек. Жүктемелерді теңгерімдеу жүйесі үнемі жүктемені және әрбір Web-сервердің дайындық деңгейін бақылайды. Торапқа қашан www.acme.com келуші қарайды, оның сұрауы Web-серверлердің біріне емес, жүктемені теңгеру жүйесіне түседі. Бұл жүйе сұранысты қай серверге жіберу туралы шешім қабылдайды. Бұл ретте, ол әрбір қамқорлықтағы серверді жүктеу сияқты критерийлерді басшылыққа алады, сондай-ақ әкімші қалыптастырған шарттар мен ережелерді сақтайды. Содан кейін жүктемені теңгеру жүйесі клиенттің сұрауын тиісті серверге жібереді (әдетте, ол сервердің жауабын клиентке жібереді, бірақ бұл нақты іске асыруға байланысты).

Жүктемені бөлу жүйелері бірнеше Web-сайттардың жүктемелерін теңестіруді қамтамасыз ете алады. Бірнеше сайттарды қолдану серверлерді-"дублерлерді" (айналы Web-серверлерді) сайтқа келушілерге жақын орналастыруға және сайт пен клиенттер арасында ақпарат алмасу кезінде кідірістерді қысқартуға мүмкіндік береді. Сонымен қатар, бірнеше Web-сайттар Бар болған жағдайда олардың арасында жүктемені біркелкі бөлу, сондай-ақ сайт жұмысында бұзушылықтар болған жағдайда (энергиямен қамтамасыз ету жүйесіндегі іркілістер немесе есептеу орталығындағы Internet-пен байланысты жоғалтудан болған болса да) олардың дайындығының жоғары деңгейін және бас тартуға төзімділігін қамтамасыз ету мүмкіндігі пайда болады. 2-суретте көрсетілгендей бірнеше сайттар болған жағдайда, барлық сайттардағы жүктемені теңгеру жүйелерінің бір жалпы виртуалды жүйенің атауы бар, бірақ әр түрлі VIP-адрестері бар. Мәселен, Нью-Йорктегі 1 сайтында 1 теңестіру жүйесі виртуалды басты машинаның аты берілді www.acme.com VIP-мекен-жайы 193.168.35.10. Ал виртуалды негізгі машинаның атын пайдаланатын Лос-Анджелесте 2 сайтында 2 теңгерушісі басқа VIP-мекен-жайы бар-193.200.1.10. Барлық баланстау жүйелерінің жергілікті Web-серверлермен байланысы бір сайтпен болған жағдайдағыдай орнатылады. Жергілікті серверлердің жүктемелеріне мониторинг жүргізумен қатар, теңгеру жүйелері өздерінің "әріптестерімен" басқа сайттарда конфигурация және жүктеу туралы ақпаратпен алмасады; бұл ретте сайттың дайындық деңгейі де тексеріледі. Нәтижесінде барлық теңгеру жүйелері өз иелігінде жүктемелерді бөлудің және әртүрлі тораптардың жұмысқа дайындығының жалпы көрінісі болады. Бірнеше сайттар Бар болған жағдайда жүктемелерді теңестіру жүйесі жиі параллельді тағы бір тапсырманы орындайды: виртуалды жүйенің атауына қызмет көрсететін DNS серверлерінің рөлін өзіне алады. Осы атауды көрсеткен клиенттен DNS сигналын алып, теңгерімдеу жүйесі клиентке жүктеменің ағымдағы деңгейін, клиенттен алшақтық дәрежесін және басқа параметрлерді ескере отырып, ең қолайлы сайттың VIP-мекенжайын қайтарады. Содан кейін клиент автоматты түрде осы торапқа қол жеткізе алады.

Іздеуді онталайндыру және сайтқа кіруді бақылау құралдары.

Сайтты таңдау және трафикті жаһандық деңгейде басқару

Ақпараттық ресурстар бірнеше айналы тораптарда орналастырылған жағдайларда жүктемені теңгеру жүйелері (сонымен қатар жүктемені теңестірудің жаһандық жүйелері деп аталады) серверді таңдаудың сипатталған тетіктерінің көмегімен клиент үшін қолайлы торапты анықтайды. Сонымен қатар, сайтты таңдау критерийі ретінде баланстаудың ғаламдық жүйесі сайт пен клиент арасындағы

қашықтық (транзиттік учаскелер саны мен желілік кідіріс ұзақтығы көрсетілген) сияқты көрсеткішті пайдалана алады. Ең қолайлы сайтты анықтау кезінде, теңгерім жүйесі жиі клиенттің трафигін DNS зияткерлік функциясы арқылы тиісті торапқа жібереді.

Бұл үдерістің механизмін жақсы түсіну үшін, біздің мысалымызға оралайық. Аты бар виртуалды жүйе `www.ame.com` екі сайтта ұсынылған. Бұл машина үшін DNS серверлерінің функцияларын жүктемені теңгерудің екі жүйесі орындайды: біреуі - Нью-Йоркте, екіншісі - Лос-Анджелесте. FTP және электрондық пошта сияқты қызметтерге, сондай-ақ Internet-дегі басқа серверлер мен компьютерлерге арналған атауларды DNS доменінің ресми сервері ажыратады `ame.com`. Міне, әкімші поддоменнің атын бере алады `www.ame.com` Internet доменінің бөлігі болып табылады `ame.com`, екі теңгеру құралы, содан кейін екі жүктеме теңестіру жүйесі тегеурінді серверлер болады `www.ame.com`. осындай конфигурацияны алу үшін әрбір теңгерім жүйесі үшін DNS атауын анықтау қажет `www.ame.com` және оны өз Жергілікті VIP мекенжайына сәйкес қойыңыз. Екі баланс жүйесі конфигурация және жүктеме туралы ақпаратпен алмасады, екеуі де виртуалды жүйе екенін "түсінеді" `www.ame.com` екі VIP-мекен-жайы бар (яғни. екі түйінмен ұсынылған). Демек, екі сайттардың әрқайсысының жүктемесі мен қолжетімділігі туралы ақпарат бар.

AOL клиенті мекен-жай бойынша хабарласқанда `www.ame.com` 3-суретте көрсетілгендей, қоңырау рәсімі клиент жергілікті DNS серверінен компьютердің IP мекенжайын сұрайды `www.ame.com`. егер AOL қызметінің DNS жергілікті серверінің кәшінде сұралған IP мекенжайы туралы деректер болмаса, бұл сервер DNS доменінің ресми серверіне сұрау жібереді `ame.com`. Қалай есте DNS Домен сервері `ame.com` аты тағайындады `www.ame.com` екі жүктеме теңгеру жүйесі, сондықтан домен `ame.com` AOL IP мекен-жайы қызметінің DNS жергілікті серверіне екі жүктемені теңестіру жүйесінің атау сервері ретінде қайтарады `www.ame.com`. (3 - суретте мен DNS серверінің зияткерлік қызметін белгілеп, оны жеке тіктөртбұрыш деп белгіледім-кейбір жеткізушілер бұл технологияны жеке серверде іске асырады. AOL қызметінің DNS жергілікті сервері жүктемені теңестірудің екі жүйесінің бірінің атын алуға сұрау жібереді. Екі жүйе атаулардың серверлерін білдіреді, сондықтан бірінші серверден жауап алмай, AOL қызметінің DNS жергілікті сервері екінші машинаға қайта сұрау жібереді. Жүктемені теңестіру жүйесі сайт үшін берілген критерийлердің негізінде ең қолайлы серверді таңдап, сайт серверінің AOL VIP-мекенжайын AOL қызметінің DNS жергілікті серверіне қайтарады. AOL VIP қызметінің DNS жергілікті серверінен бас машинаның мекен-жайын алып `www.ame.com`. клиент өзінің HTTP трафигін таңдалған сайттың жүктемесін теңгеру жүйесіне (мысалы, нью-Йорк) жібереді және бұл жүйе клиент үшін жергілікті серверлердің бірін таңдайды. Жергілікті DNS сервері қол жетімді DNS жазбасын Time To Live (TTL) параметріне сәйкес келетін мерзімге кәштейтіндіктен, жеткізушілер әдетте клиент жаңа VIP-мекенжайды тез алуға және басқа қол жетімді сайтқа ауысуға мүмкіндік беретін TTL үлкен мәндерін белгілемеуді ұсынады.



2-ЭКРАН. Бірнеше сайттар үшін DNS сұрау қайта бағыттау.

Жүйесі жүктемені теңдестіру мүмкін таңдау, ыңғайлы сайт, сондай-ақ бағыттауға трафик және басқа әдіспен: көмегімен бағыттау құралдарымен хаттама HTTP. Бұл әдіс жүктемені теңгеру жүйесінің DNS-функцияларын пайдалануды қарастырмайды. Оның орнына (біздің мысалымызға қайтадан жүгінемін) әкімші DNS серверінде жазба анықтайды `www.acme.com` және оған сәйкес VIP-мекенжайлар. Клиент VIP-мекенжайды алған кезде `www.acme.com` және HTTP хаттамасы бойынша жүктемені теңгеру жүйесіне сұрау жібереді, соңғысы оған ең қолайлы торапты таңдайды. Егер таңдаулы сайт қашықтағы болмаса (тым алыс емес), теңгерім жүйесі клиенттің браузеріне HTTP-командасын бағыттайды және браузер көрсетілген сайтпен байланыс орнатады. Бұл әдіс жүктемені теңгеру жүйесіне торапты таңдағанға дейін клиент туралы толық ақпаратты (мысалы, оның IP-мекен-жайы) алуға мүмкіндік береді. Дегенмен, клиент DNS серверімен қайтарылған VIP-мекенжайды пайдалана алады және сұрауларға жауап бермейтін тораппен байланыс орнатуға тырысады.

Сонымен қатар, бұл жүйенің негізгі мақсаты-белгілі бір уақыт аралығында белгілі бір уақыт ішінде Мысалы, бізде Еуропада айна Web торабы бар. Бізге Еуропадан келген клиенттер бүкіл уақытта еуропалық сайтқа түсуі керек-ол істен шыққан және жүктемені теңгерімдеу жүйесі еуропалық трафикті АҚШ-та орналасқан торапқа бағыттайтын жағдайларды қоспағанда. Администратордың жүктемені теңгеру жүйесін еуропалық IP-мекен-жайы бар клиенттен кез келген сұрау алдымен Еуропадағы сайтқа бағытталатындай етіп реттеуге мүмкіндігі бар. (Тиісті баптауларды орындау үшін, бүкіл еуропалық IP блогын теңгерімдеу жүйесіне енгізу қажет.) Мұндай нұсқауды алғаннан кейін жүктемені теңгеру жүйесі бірінші кезекте еуропалық торапқа еуропалық кері мекен-жаймен кез келген сұрау жібереді және осыдан кейін ғана сайтты таңдаудың басқа да белгіленген критерийлерін есепке алады.

Артық теңгеру жүйелері

Жүктеменің теңгеру жүйелері-Web серверлеріне қол жеткізу құралы сияқты өзге нәрсе емес, сондықтан мұндай құралдың істен шығуы түйін жұмысының толық тоқтатылуына әкелуі мүмкін. Сонымен қатар, жүйенің жоғары өнімділігін қамтамасыз ете алатын кең Өткізу жолағы бар шешімдерді таңдау қажет. Бұл жүйе бір уақытта екі жүйенің жұмысын қарастырады (active - and-standby type). екінші схема екі жүйенің бір уақытта жұмыс істеуін қарастырады (active-and-active type).

Екі схема бір торапта жүктемені теңгеру жүйелерінің екі данасының болуын болжайды.

Әдісті қолданған кезде бір жүйе белсенді, ал екіншісі күту жағдайында, теңгерімдеудің резервтік жүйесі басты жүйенің күйін үнемі бақылайды. Теңгерімдеудің резервтік жүйесі негізгі функцияларды өзіне алады (яғни трафикті басқаруды бастайды). Негізгі жүйе жұмысын қайта бастаған кезде, резервтегі трафикті басқаруды оған береді және қайтадан күту режиміне өтеді.

Екі белсенді жүктемені теңгеру жүйесі бар схемада екі жүйе де трафикке қызмет көрсетеді және бір-бірін сақтандырады. Мысал үшін төрт серверден тұратын Web торабын ұсынамыз. Олардың екеуі жүктемені теңгерудің бірінші жүйесімен және екеуі - екіншісі. Бір теңгеру жүйесі істен шыққан кезде, барлық төрт серверге қамқорлық екінші жүйені алады. Бұл әдіс теңгеру құралдарының ресурстарын толық пайдаланады және түйіннің өнімділігін арттырады.

Технологиядан артық

Web ақпараттық орталықтарына қызмет көрсететін және электрондық коммерциямен айналысатын компаниялар - трафикті басқару және өзінің виртуалды шаруашылықтарында тәртіпті сақтау үшін жүктемені теңгеру жүйесін қолданатын жалғыз емес ұйымдар. Көптеген компаниялар өздерінің Web тораптарының өнімділігін және қолжетімділігін арттыру үшін теңгерімдеу құралдарын қолға алды.

Жүктемені теңестіру жүйелері жүктемелер мен серверлердің жағдайына мониторингті, клиенттің сұранысын ең жақсы түрде өңдеуге қабілетті машина серверлерінің пулынан дұрыс таңдауды, сондай-ақ тораптың ішінде де, жаһандық ауқымда да трафикті басқаруды қамтамасыз етеді. Осының арқасында олар киберкеңістіктегі өз өкілдіктерін ашқан компаниялар арасындағы бәсекелік күресте күшті қару болады.

Әр түрлі CMS мысалында контентті басқару жүйесі үшін үлгілер жасау.

Кез келген WEB-сайт беттердің жиынтығынан тұрады, ал айырмашылықтар тек олар қалай ұйымдастырылған. Web-сайтты ұйымдастырудың екі түрі бар-статикалық және динамикалық. Бірінші жағдайда сайтты құру және қолдау үшін жауапты мамандар HTML-нысанда әрбір жеке бетті, оның рәсімделуі мен мазмұнын жазады. Екінші – кез келген Web-беттің негізінде web-браузер терезесінде беттің барлық компоненттерінің орналасуын анықтайтын үлгі жатыр және нақты ақпаратты енгізу процеске қатысушыдан HTML тілін білуді талап етпейтін және Web-бетті жариялау процедурасының маман емес үшін жеткілікті күрделі стандартты құралдарды пайдалана отырып жүргізіледі.

Егер сайт көптеген беттен тұрса немесе ол жиі жаңартылуы керек болса, динамикалық ұйымның артықшылығы айқын болады. Web-сайтты әзірлеушілер ақпараттық толтыру немесе дизайн өзгерген кезде барлық бетті қайта жазудың қажеті жоқ. Беттер толығымен сақталмайды, ал оларды қолданған кезде динамикалық түрде қалыптасады.

Осылайша, дизайнды мазмұннан бөлу динамикалық сайттардың статистикалық сайттардың басты ерекшелігі болып табылады. Осы негізде сайттың құрылымын одан әрі жетілдіру мүмкін, мысалы әр түрлі пайдаланушылық функцияларды анықтау және бизнес-процестерді автоматтандыру, ең бастысы, сайтқа түсетін контентті бақылау.

Динамикалық сайт жасау үшін екі жол болуы мүмкін. Біріншіден, бұл қажетті үлгілерді жасауға жауапты және қажетті функцияларды қолдайтын өз бағдарламаларын жазу. Бұл ретте құрылған жүйе қажеттіліктерге толық жауап береді, бірақ үлкен бағдарламалық күш пен уақытты талап етуі мүмкін. Екінші

жол-Web-контентті басқару жүйелері деп аталатын қолданыстағы жүйелерді пайдалану. Бұл жолдың артықшылығы-уақыт пен күш шығындарын азайту. Оның кемшіліктеріне икемділіктің төмендеуі, мүмкіндіктердің жеткіліксіз немесе шамадан тыс жиынтығын ұсыну жатады.

Контент (мазмұнын, мазмұнын білдіретін content терминінің сөзбе – сөз аудармасы) сайттың ақпараттық толтырылуын-яғни сервердегі материалдардың барлық түрлерін: web-беттер, құжаттар, бағдарламалар, аудио-файлдар, фильмдер және т.б. түсінеді. Осылайша, мазмұнды басқару-бұл ұқсас материалдарды басқару процесі – Ол келесі элементтерді қамтиды: материалдарды серверге орналастыру, материалдарды серверден жою, қажет болмаса, материалдарды ұйымдастыру (қайта ұйымдастыру), олардың күйін бақылау мүмкіндігі.

Контентті басқару жүйелері (ағылшын тілінде қалыптасқан термин бар – Content Management Systems немесе, қысқартылған, CMS) – бұл контентті басқару процедурасын автоматтандыратын бағдарламалық кешендер.

CMS функциялары

Мазмұнды басқару жүйесінің функцияларын бірнеше негізгі санаттарға бөлуге болады:

1. Жасау-авторларға контент жасаудың ыңғайлы және үйреншікті құралдарын ұсыну.

2. Басқару-контентті бірыңғай репозиторийде сақтау. Бұл құжаттардың нұсқаларын қадағалауға, оларды кім және қашан өзгерткенін бақылауға, әрбір пайдаланушы жауапты бөлімді ғана өзгерте алатынына көз жеткізуге мүмкіндік береді. Бұдан басқа, қолданыстағы ақпараттық көздермен және ат-жүйелермен интеграциялау қамтамасыз етіледі. CMS құжаттардың жұмыс ағынын бақылауды қолдайды, яғни оларды мақұлдау процесін бақылау. Осылайша, контентті басқару нұсқаларды сақтауды, қадағалауды, кіруді бақылауды, басқа ақпараттық жүйелермен интеграциялауды және құжаттардың ағынын басқаруды қамтиды.

3. Жариялау-пайдаланушы терминалында мазмұнды автоматты түрде орналастыру. Тиісті құралдар беттің көрінісін бүкіл сайттың дизайнына автоматты түрде бейімдейді.

4. Көрініс-деректерді көрсету пішімін жақсартуға мүмкіндік беретін қосымша функциялар; мысалы, репозиторий құрылымы бойынша навигацияны құруға болады.

Мазмұнды басқару жүйелері төрт негізгі санатқа бөлінеді, олар ішінара жабылады:

1. Басқару жүйесінің бастапқы кодтарымен дәстүрлі түрде қолдайды басқармасы бастапқы кодтарымен бағдарламалар, және жиі ұсынады кейбір web-интерфейс, ол пайдаланылуы мүмкін корпоративтік желі ішіндегі, сондай-ақ одан тыс параллелді жұмыс бастапқы кодтарымен.

2. Құжаттарды басқару жүйелері көп құжаттармен жұмыс істейтін ұйымдарға арналған, мысалы, ірі компаниялардың кеңселері, редакциялар және сақтандыру компаниялары.

3. Web-контентті басқару жүйелері бағдарламалық өнімдердің жаңа индустриясын білдіреді. Бұл жүйелер әртүрлі күрделілік деңгейіндегі Web-сайттарды әзірлеу және басқару үшін арналған. Әдетте мұндай жүйелер жұмыс ағынын басқарудың кейбір түрлерін қолдайды.

4. Электрондық коммерция жүйелері-тауарлардың электрондық каталогтарын сақтауды және басқаруды қамтамасыз етеді. Шын мәнінде, бұл жүйелер бір-бірінен шамалы ерекшеленеді. Бұл жүйелердің ең басты айырмашылығы-оларды қолданатын адамдар –

CMS пайдалану келесі артықшылықтарды ұсынады:

1. Ақпаратты жедел жаңарту-ақпаратты техникалық мамандар ретінде қосымша делдалдарсыз ақпаратты білетін қызметкер жариялайды. CMS WEB-сайтта ақпаратты жариялау процесін автоматтандыруға арналған, пайдаланушыларға материалдарды WWW-да жариялауға және олардың визуалды көрінісін анықтауға мүмкіндік береді, ол үшін HTML тілін білуді талап етпейтін стандартты құралдарды және маман емес процедураларды пайдалана отырып. CMS көмегімен, кәсіби әзірлеуші бола алмай, сайттардың ақпараттық толтырылуын жасауға және өзгертуге болады.

2. Қолдау құнын төмендету-ақпаратты жаңарту дербес жүргізіледі, өз немесе сыртқы web-шебердің еңбегін төлеу қажеттілігі жоқ. Құнның төмендеуі құжаттарды іздеуге кететін уақыт шығынын азайту, қайталану мен қателіктердің жолын кесу, серіктестермен және клиенттермен байланыс жылдамдығын арттыру есебінен орын алады.

3. Пайдаланушыға қосымша Сервистерді ұсыну – сервистердің бір бөлігі-іздеу, форумдар, дауыс беру және т.б. пайдаланушымен интерактивті өзара әрекеттесуді талап етеді. Олар CMS аясында жүзеге асырылған.

4. Әзірлеу мерзімі мен құнының азаюы-ең талап етілетін функционалдылық CMS-де жүзеге асырылған және бірден пайдаланылуы мүмкін.

5. Әзірлеу сапасын арттыру-әзірлеу кезінде бірнеше рет тестілеуден өткен дайын модульдер толық немесе ішінара пайдаланылады.

6. Одан әрі модификация құнының төмендеуі-CMS деректерді және олардың көрінісін бөлуге мүмкіндік береді. Бұл статикалық веб-сайтқа қарағанда сайттың көрінісін өзгертуге мүмкіндік береді.

CMS-жүйелер арасында жиі жүйе құру үшін құралдар (content management framework, CMF) деп аталады.

Мазмұнды басқару жүйелерін әзірлеумен көптеген компаниялар, соның ішінде IBM, Microsoft, Oracle, Macromedia айналысады.

CMS стандарттау

Соңғы уақытта CMS әзірлеушілерін біріктіруге, осындай жүйелердің әлеуетті пайдаланушылары үшін бірыңғай ақпараттық органы құруға, бірыңғай стандарттарды ілгерілетуге және бекітуге тырысатын ұйымдар пайда бола бастады. Ең алдымен, бұл OSCOM қауымдастығы (Open Source Content Management) және CMSWatch.

OSCOM WebDav, RSS, ATOM және JSR-170 сияқты стандарттарды бекітті.

Өз кезегінде, CMSWatch жыл сайын CMS-жүйелер нарығының шолуын, олардың кейбірін салыстыруды, CMS-жүйелердегі мазмұнды өмірлік циклінің сипаттамасын және оны басқаруды қамтитын есепті шығарады.

CMS деректерді ұсыну модельдері

Деректерді ұсыну моделіне негізделген CMS жіктелуі бар — Объектілік, желілік немесе модульдік.

Объект моделі

Деректерді ұсынудың Объектілік моделі класс пен объект сияқты ұғымдармен жұмыс істейді. Кластар деректер құрылымын анықтайды және атрибуттар жиынтығын білдіреді (мәтіндік жол, бүтін сан, сурет және т.б.). Класс даналарының (объектілерінің) белгілі бір құрылымы бар және еркін иерархиялық құрылымды құра отырып, басқа да объектілерді қамтуы мүмкін. Нысандар ондағы объектілердің қасиеттерін, мазмұнын және мінез-құлқын мұраға алуы мүмкін. Нысандардың мысалдары-құжаттар, суреттер, папкалар және пайдаланушылардың есептік жазбалары. Мазмұн класы нақты деректерді сақтамайды — мұндай ақпаратта Нысандар (сынып даналары) бар. Бір сыныпты анықтау арқылы оның көптеген өкілдерін (Нысандар мазмұнын) жасауға болады.

CMS-жүйелерде деректер әдетте реляциялық немесе Объектілік деректер базасында сақталады. Бірінші жағдайда деректердің Объектілік моделі деректер базасының реляциялық моделіне бейнеленеді.

Әдетте, объектілі-бағытталған деректер моделіне негізделген жүйелер, ең функционалды, икемді, бірақ сонымен қатар ең күрделі.

Желілік модель

Мәліметтерді ұсынудың осы моделі графтар теориясына сүйенеді: ақпарат құрылымы олардың арасында белгіленген байланыстары бар түйіндер түрінде ұсынылады. Жүйенің негізі деректерді сипаттаудың желілік моделі көрсетілген желілік және дәстүрлі реляциялық ДҚБЖ болып табылады. Реляциялық кестелерде тораптар, олардың атрибуттары және олардың арасындағы байланыстар туралы ақпарат сақталады. Байланыс атрибуттан ерекшеленеді, онда басқа түйінге сілтеме сақталады, ал атрибутта — нақты мән. Бағытталған бағаннан деректерді алу үшін әдетте түйіндердің тізімін жасау, ата-ананың атрибуттары бойынша түйін атрибуттарын анықтау және т. б. сияқты рекурсивті өңдеу рәсімдері қолданылады.

Модульдік үлгі

Мұндай жүйелерде мазмұн мазмұн түрі бойынша бөлек модульдерге бөлінеді. Деректер құрылымы модульге байланысты және мазмұнмен барлық жұмыс модуль ішінде шоғырланған. Модульдер тәуелсіз және осы үлгідегі құжаттармен жұмыс істеуге толық жауап береді. Құжаттар белгіленген сипаттамалар жиынтығы арқылы сипатталады — құжаттардың түрлері қатаң бекітілген. Жаңа модуль қосу, бар кодты ауыстыру немесе өңдеу арқылы функционалдылықты кеңейтуге болады. Көбінесе әртүрлі модульдердің құжаттары мен бір модульдің құжаттары арасында байланыс жүйесі жоқ. Мазмұн (Модульдер) түрлерінің стандартты жиынтығы: сілтемелер, мақалалар, файлдар, жаңалықтар, бөлімдер, форум.

Деректер моделінің айқын шектеулігіне қарамастан, оның негізіндегі жүйелер қарапайымдылығының арқасында ең танымал. Модульдік CMS жүйелерінің бір жалпы кемшілігі бар-модульдің ішінде қатаң бекітілген мазмұн құрылымы. Алайда, олардың функционалдығын кеңейту үшін желіде көп Сыртқы модульдерді пайдалануға болады. Бұл жүйелердің айқын артықшылығы — қысқа уақыт ішінде порталды пайдалануға толық дайын алу мүмкіндігі.

ЗЕРТХАНАЛЫҚ-ПРАКТИКАЛЫҚ ЖҰМЫСТАР

Зертханалық жұмыс №1. Web-сайтты ұйымдастыру

Бүгінгі таңда компьютерлік дизайн, Web-дизайн, жүйелік программалаушы, администратор және тағы басқа мамандардың қажет екендігін түрлі жарнамалық газеттерден, бұқаралық ақпарат құралдарынан да көруге болады. Аталған мамандықтарды қалай дайындап, оларға қандай программалық құралдарды үйрету арқылы жетілдіруге болатыны әдістемелік жұмыстың өзекті мәселелерінің бірі болып отыр.

Кез-келген шығармашылық мамандық сізден қазіргі заманға сәйкес компьютерлік технологияларды меңгеруге талап етеді. Шығармашылық жұмыстың қортындысы әр қашан жаңа білім, жаңа ақпарат, ал осы ақпаратты таратудың ең жеңіл әдісі – Интернетте жариялау.

Интернетте материалды жариялау тиімді болу үшін ол Web-дизайн ережелеріне сәйкес болу керек. Демек, Web-дизайн негіздерін білу кез келген мамандықта қажет болады, өткені Web-бет дизайны – бұл ақпараттық дизайн.

Web-сайт бұл дүниенің кішкентай моделі. Бұрынғы кезде Web-сайтты бір адам - Web-мастер жасаған болса, қазіргі кезде Web-сайттарды бірнеше адам жасайды. Олар Web-дизайнер, программист, бизнес-кеңесші, маркетинг бойынша басқарушы, менеджер.

Өзіміз білетіндей Internet желісі түрлі сайттардан тұрады. Сайттарды пайдалана білумен қатар оны құра білу де қажет. Ол әрине көптеген жұмыстарды қажет етеді, ең бастысы ол не мақсатта құрылып жатыр? Әрбір Web-сайт текстен, суреттерден, видео ұнтаспаларынан тұруы мүмкін. Мұндай сайттар ғаламшардың кез-келген нүктесіндегі компьютерде болуы мүмкін. WEB-тің негізгі қызметі - қажетті ақпаратты шапшаң түрде кіріп көру, жинастыру және де оны экранға шығаруды ұйымдастыру. Гипермәтінді сілтеме-келесі беттермен байланысты қамтамасыз етеді. Сілтемені тышқанмен шертіп сіз келесі WEB-сайтқа өте аласыз. WEB-сайтты біз келесі бағдарламалар арқылы көру мүмкіншілігіне ие бола аламыз: Microsoft Internet Explorer [Майкрософт Интернет эксплорер], Netscape Navigator [Нетскейп навигатор], Mozilla[Мозилла], Opera[Опера]. Бұл бағдарламалардың артықшылығы сайтты сілтемелер немесе адрестер арқылы бейнелеп көрсетіп береді және де дискіге сақтай алатын мүмкіншілігі бар. Сайт даяр болғаннан кейін оны алдын-анықтап алу қажет.

Соңғы жылдары компьютерлік техниканың жедел дамуына байланысты сайт құруға арналған бірнеше программалар шықты. Атап айтар болсақ:

- HTML тілі (Hyper Text Markup Language)
- PHP1-PHP5
- Python
- Java

HTML-тілі құжаттардың құрылымын суреттеу үшін берілетін командалық қарапайым тегтерден тұрады. Жұмысы - құжат тақырыптарын белгілеу, гипермәтінге белгілер орнату. Ал PHP тілі сол HTML-тілі жасай алмайтын мүмкіндіктерді жасай алады, яғни сайт бетіне процедуралық бағдарламалау жолын ұсына алады. Алғашында PHP тілі онша таныс болмаған мен қазір оның құдыреті кез келген сайт құру тілдерінен асып түседі. PHP тілі жылда дамып отыр оның алғашқы нұсқалары 1994 жылдары PHP болса қазір оның PHP5 нұсқасы шығып үлгерді.

Сайттар және олардың адрестері

Қаладағы серверлерге шығу үшін домендерді қолдануға болады. Ал ондағы аудан, үйге қатынас жасау үшін сайттар қолданылады. Қандай да бір мекемеге немесе жеке тұлғаның және қандай да бір тақырыпқа арналған желінің логикалық түрде аяқталған элементі. Әрбір сайттың өз адресі болады, ол әріптен тұрады.

Желіні жасаушылар арнайы, домендік аттарды, (DNS) серверін құрады. Олар автоматты түрде әріптік адресстерді (URL) цифрлыққа (IP) ауыстырады. DNS өзінде сол адресстердің сәйкесіне таблицасы сақтайды.

Сайт адресстері бірнеше маңызды элементтерден тұрады. http:// - гипертекстік құжаттарды теру протоколын білдіретін префикс.

WWW – ресурстың WWW жүйесіне жататындығын білдіретін “сигналдық жалауша”. Бұдан кейін адрес атының ең төмендегі элементі – сайттың өз аты орналасады.

Беттер – біртұтас «ағзаны» -сайтты құрайтын жеке гипертекстік құжаттар.

Домендік зоналар

Интернеттің компьютерлерінің барлығының өз адресстері (IP -адресі) болады. Сервер ыңғайлы болғандықтан домендік зоналар деп аталатын логикалық топтарға біріктірілген. Бұл зоналар географиялық , тематикалық болуы мүмкін.

Географиялық домендік зона (бірінші деңгейдегі зона) желіге өз компьютерлері арқылы қосылған әрбір мемлекетке беріледі. Ол ереже бойынша екі әріппен белгіленеді:

Ch –Қытай

Fr –Франция

Ge –Германия

Jp –Жапония

Ru –Ресей

Tw –Тайвань

Uc –Украина

Ur –Ұлыбритания

Тематикалық домендік зонаның географиялықтан қарағанда белгілі бір аймаққа қатысы жоқ. Ол әр елдегі ғана емес, әр контингенттегі компьютерлерді біріктіреді. Мұнда компьютерлер оларды басқаратын мекемелердің түрі бойынша топтастырылған. Ал домендік индекс үш және одан көп әріптермен белгіленеді:

Gov - өкіметтік мекемелер;

Com - кез келген коммерциялық ұйымдар;

Net - желілік қызметке қатысты ұйымдар;

Nur - әскери мекемелер;

Int – халықаралық мекемелер;

Edu - білім беру мекемелері;

Shop - желілік магазин;

Museum – мұражай;

Biz – кез-келген бизнес -жоба;

Name - жеке беттер.

PHP дегеніміз не?

PHP-бұл Web –серверге жіберілетін скриптердің көмегімен Web-беттерінің динамикалық генерациясына қажетті бағдарламалау тілі. Сіз бетті PHP және HTML-дің көмегімен ашасыз. Сайтты пайдаланушы бетті ашқанда, сервер html-код қосылған PHP операторларын орындайды және нәтижені пайдаланушының браузеріне жібереді. Бұл әрекет дәл осылай ASP және Cold Fusion-ның көмегімен жасалады. Дегенмен ASP және Cold Fusionға қарағанда, PHP ашық бастапқы кодты өнім болып табылады және платформалы тәуелсіз. PHP Windows NT және Unix-тің көптеген версияларында жұмыс жасайды. Ол Apache –дегі модуль ретінде жіберіле

береді. Егер жіберілу Apache модулі түрінде болса PHP оңай және жылдам жұмыс жасайды. Бұл кезде процессті жасауға байланысты туындайтын қосымша шығындар болмайды. Сондықтан нәтижесі тез шығады және сервердегі сақтаудағы шығынды азайтатын `mod_perl`-ді орнатудың қажеттілігі болмайды.

Сіздің құжат бетіңіздегі әртүрлі операциялардан басқа сіз PHP көмегімен HTTP-тақырыптарды қалыптастырып HTTP орната аласыз. Сонымен қатар аутентификацияны басқарып пайдаланушыны басқа бетке бағыттай аласыз. PHP берілгендер қорына еруге үлкен мүмкіндіктер ашады. Бұл сізге PDF құжаттарды инерациялаудан, XML-дегі грамматикалық талдауды жасауға мүмкіндік береді.

PHP операторлары сіздің бетіңізге Web-беттерді қойып береді, сондықтан арнайы ортада жүзеге асырудың қажеттілігі туындамайды. Сіз PHP-кодының блогын тегімен аяқтайсыз. Бұл тегтердің арасындағылардың барлығы PHP код ретінде интерпритацияланады.

PHP тілінің синтаксисі Си және Perl синтаксисіне ұқсас келеді сіз ауспалыларды оларды пайдаланудан бұрын жарияламауыңыз керек. Массивтермен Хэмтер оңай жүзеге асырылады.

Дегенмен PHP Apache құрылған жағдайда бәрінен жылдам жұмыс жасайды. PHP Web – сайтында оны Microsoft IIS және Netscape Enterprise Server орналастырудың реті бар. Егер сізде PHP ді орнатуға бағдарламалық қамтамасыз етудің көшірмесі жоқ болса, оны сізге ресми Web –сайттан ала аласыз. Ол жерден сіз PHP дің барлық ерекшелігі мен қызмет бейнеленген түсіндірілген жетекшілікті таба аласыз.

Зертханалық жұмыс №2. Гипермәтіндік құжаттарды құру технологиялары.

Сайтта ақпаратты ұйымдастыру келесі түрде болуы керек:

- Сызықты;
- Иерархиялық;
- Контексті-тәуелді;
- Қиыстырылған.

Сызықты құрылым – деректерді ұйымдастырудың ең қарапайым әдісі. Бірінен кейін бірі келетін веб-беттердің терімін құрайды. Сызықты құрылымға операцияның қадамды орындалуы жатады.

Иерархиялық құрылым – ақпаратты орналастырудың ең таралған нұсқасы. Бөлек беттерде орналасқан сайттар бөліміне сілтемелер мәзірі орналасатын негізгі бетті құруды қарастырады. Бөлімдердің өзі ішкібөлімдерден және басқа да бөлшекті ақпараттан тұруы мүмкін. Контексті-тәуелді құрылым – бұл жерде басқа сайттарға сілтемелер қолданушының арнайы бір қызметтеріне байланысты құрылады. Бұндай құрылымның мысалдарына интернет-магазин элементтері, сайтта кілттік сөзі бойынша ақпаратты іздеу жатады. Қиыстырылған құрылым – алдыңғы бірнеше құрылымдардың бірігуі. Мысалы, сайтта арнайы жерлерде қадамдық қызметтерден (сызықтық құрылым) тұратын иерархиялық құрылым болуы мүмкін.

Мысал

Ақпаратты ұйымдастырудың қиыстырылған құрылымының мысалы ретінде "Эйдос" қашықтықтан білім беру орталығының сайты – <http://www.eidos.ru> қарастырамыз.

Сайттың негізгі бетіндегі мәзір (сайт бөлімдеріне сілтеме иерархиялық құрылымға сәйкес берілген):

Орталық жайлы

Дистанционды курстар

Эвристикалық олимпиадалар
Оқушыларға арналған жобалар
«Эйдос» Интернет-журналы
Орталықпен ынтымақтастық
Жылдың дистанционды мұғалімі
Педагогикалық конференциялар
Электронды дүкен

«Эвристикалық олимпиадалар» бөлімінің бетіндегі мәзір (сайт бөлімдеріне сілтеме иерархиялық құрылымға сәйкес берілген):

Олимпиадалар			жайлы
Олимпиадалар			тізімі
Олимпиадалар			кестесі
Тапсырмалар	мен	жұмыстар	мысалы
Қалай		тіркелу	керек
Қатысушылардың			ойы
Сұрақтар және жауаптар			

«Олимпиадалар жайлы» ішкі бөлімінің бетіндегі мәзір (сайт бөлімдеріне сілтеме иерархиялық құрылымға сәйкес берілген):

ДЭО деген не?

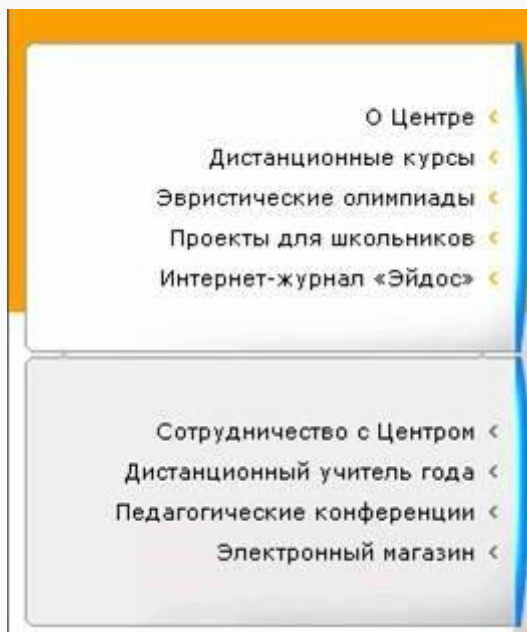
Бәрі қалай болып жатыр. 1 Бөлім

Бәрі қалай болып жатыр. 2 Бөлім

Бәрі қалай болып жатыр. 3 Бөлім

Сайттың қиындығы

Ақпаратты орналастырудың қолайлылығынан бөлек оның көлемін және қиындығын бақылау керек. Адаидар жеті түрлі бөлімдерден тұратын ақпаратты қабылдай алады. Сол себепті сайтқа келушілер өздеріне керекті материалдарды олар төрттен жетіге дейінгі элементтерден тұратын топтарға бөлінген болса оңай табады. Сол себепті жоғарыда көрсетілген мысалдағы "Эйдос" қашықтықтан білім беру орталығының сайтының негізгі бетінің сегіз негізгі бөлімдері әрқайсысында төрт бөлім бар екі блокқа бөлінген (Сурет 1).



Сурет 1

Сайтты құрудың тағы бір артықшылығы онда көрсетілген ақпаратты экранға айналдыруды қажет етпей шығаратын бөліктерге бөлу тиімдірек. Бұл жағдайда бөлек веб-беттер арасындағы байланысты ұйымдастыру гипермәтіндік сілтемелердің көмегімен іске асырылады, мысалы сызықты құрылымға сүйене отырып.

Гипермәтіндік сілтемелер

Гипермәтін құжаттың бір үзіндісінен осы құжатта немесе сұраныс кезінде қол жетімді басқа кез келген жерде болу мүмкін үзіндіге тез өтуге мүмкіндік береді. Сілтеме құру үшін браузерге нақты қайсының сілтеме екендігін көрсету керек (сөз, мәтін, белгі, сурет, фото және т.б.), сонымен қатар сілтеме апаратын құжаттың адресін көрсету керек. Екі әрекет те HREF параметріне ие A тэгінің көмегімен орындалады. Бұл параметрдің мәні ретінде сілтеме алып келетін құжат адресі қолданылады (URL).

Мәтіндік сілтемелер

1. Мәтін түріндегі гиперсілтеме құрыңыз тапсырмасында сіз өзіңіздің визиткаңызды құрдыңыз - vizitka-family.htm типіндегі файл. Өзіңіздің визиткаңызда блакноттың көмегімен жаңа жолдан келесі кодты теріңіз:

```
<A HREF="http://www.eidos.ru">Это интересно!</A>
```

Браузерде не болғанын көріңіз. Тэгтың ішінде орналасқан мәтін көк және асты сызылған болды. Оған тышқан мәзірін апарған кезде қолдың суреті шығады. егер сіз Интернет желісінде саяхаттаған болсаңыз онда мұның гиперсілтеме екендігін білесіз. Оны басу сізді "Эйдос" қашықтықтан білім беру орталығының сайтына алып келеді. Ол үшін сіздің компьютеріңіз интернет желісіне қосылған болуы керек

Интернеттегі негізгі қызмет – ақпараттық теңізде гиперсілтемелер көмегімен саяхаттау оңай беріледі.

Зертханалық жұмыс №3. Динамикалық бағдарламалау тілдері.

Java бағдарламалау тілі бұл Internet желісінде жұмыс істейтін объектілі-бағытталған, платформалы – тәуелсіз, желі ішінде жұмыс істейтін тармақталған қосымшалардың өңдеуіне қолданылатын программалау тілі. Java жобасы 1995 жылы Sun Microsystems бірлестігінің арқасында көрсетілген болатын. Java бағдарламалау жүйесі кішкене интерактивті қолданбалы программалар - апплеттерді таратуға World Wide Web (WWW) жүйесін қолдануға мүмкіндік береді. Олар Internet серверлерінде орналасып, таратылады, клиентке желі арқылы тасымалданады, автоматты түрде орнатылады және WWW құжатының бөлігі ретінде орнында орындалуға жіберіледі. Апплетке клиент компьютері қорларына қолжетімділігіне шек қойылған, сондықтан да ол жанама мультимедиялық интерфейссті бере алады және дискіде сақталған мәліметтерге ешқандай ақау келтірместен күрделі есептеулерді жүргізе алады. Бағдарламалардың басқа түрі ретінде Java қосымшаларын алуға болады. Олар кез келген компьютерде, тіпті оның архитектурасына қарамастан орындалатын тасымалдана алатын кодтарды көрсетеді. Осы кезде генерацияланатын виртуалды код виртуалды Java - машинадағы (JVM – Java Virtual Machine) виртуалды код интерпретаторында орындалатын нұсқаулардың жиынтығы болып табылады. Клиенттерге сервердегі қосымшаларға және деректер базасына қолжетімділікке мүмкіндік беретін JSP (Java Server Pages) және сервлеттер өте кең ауқымда таралуға мүмкіндік алды.–Java тілі .

Java тілі C++ тілінің синтаксисын қолданады, бірақ объектілік үлгі Smalltalk тілінен алынған. Осыдан Java тілінің C++ тілімен ұқсастықтары тек қана сыртқы түрде екенін көруге болады. Басқа программалау тілдерімен салыстырып қарағандағы негізгі айырмашылығы - программалар мөлшерінің азаюын қажет етуі

мен желіде жұмыс істейтін тасымалданатын қосымшалардың қауіпсіздігі шарттарының ұлғаюы. Java көрсеткіштерді (C++, Pascal және тағы да басқа тілдердің ең қауіпті құралы) қолдамайды, себебі, жадтың жанама адрестерімен типі көрсетілмеген көрсеткіштер арқылы жұмыс істеу мүмкіндігі жадтың қорғанышын елемеуге рұқсат береді. Java тілінде айнымалы арифметикамен есептеудің тәсілдері өзгерген, сондықтан да тіл түрлерінің арасында аралық код шыдамдылығын қамтамасыздандыру үшін strictfp кілттік сөзі енгізілді. Ол компиляторға айнымалы үтірі бар сандар үшін арифметикалық әрекеттерді алдыңғы түрдегі есептеулерге сәйкес орындау керек екендігін көрсетіп, жаздырады.

Тіл кластарының жүйелік кітапханасы кластар және пакеттерден тұрады, олар тілдің әртүрлі базалық мүмкіншіліктерін жүзеге асырады. Бұл кітапханаларға қосылған кластардың әдістері JVM-нан Java – бағдарламаның интерпретациясы кезінде шақырылады. Java-да бағдарламаның барлық объектілері динамикалық жадта орналасқан (heap) және стектерде сақталынатын объекті сілтемелер арқылы қолжетімді. Бұл шешім жадқа тікелей қолжетімсіздікке мүмкіндік берді, бірақ массив элементтерімен жұмыс істеуді қиындатып жіберді. Java тіліндегі объектілі сілтемелер өздері бағытталып тұрған объектілердің класы туралы хабардан тұрады. Сондықтан да объектілі сілтемелер дегеніміз, көрсеткіштер емес, олар объектілердің дескрипторлары. Дескрипторлардың болуы JVM-ге код интерпретациясы фазасында типтердің сәйкес келуін тексеруге мүмкіндік береді. Java-да жадты динамикалық бөлу концепциясы да қайта қарастырылған: динамикалық бөлінген жадты босату тәсілдері жоқ болады. Оның орнына new (қоқыс жинаушысы) операторының көмегі арқылы көрсетілген жадты автоматты түрде босату жүйесі іске асырылған. Java - бағдарламаларда класс спецификациясы мен оның жүзеге асырылуы әрқашан да тек қана бір файлда болады. Java тілі операторларды қайта жүктеуді және typedef, белгісіз бүтіндерді (егер ол ретінде char- ды есептемесек) қолдамайды. Java-да көптік мұрагерлік жоқ, тек құрастырушылар бар, бірақ деструкторлар жоқ (қоқысты автоматты түрде жинастыру қолданылады), тілдің кейінге сақталған сөздері бола тұрса да, goto операторы және const сөзі қолданылмайды.

Java тілінде пайда болған маңызды мүмкіндіктер интерфейстер мен кең ағымдылық (бағдарлама бөлімдерінің бір уақытта орындалу мүмкіншілігі).

1 – бағдарлама. Қарапайым программа.

```
public class Hello {
    public static void main(String[] args) {
        System.out.print("Java әлеміне қош келдіңіз!");
    }
}
```

Аталған бағдарлама хабарламаны консольге (немесе бұйрық қатарына) шығарады. (01) қатарындағы Hello атаулы кластың сипаттамасы келтіріледі, онда тек бір ғана main()(02) әдісі бар. Бағдарлама іске қосылған кезде басқару осы әдіске беріледі және хабарлама жылдам консольге шығарылады.

Ең басында процестің қалай жинақталғанын түсінуге тырысып қажеті жоқ, тек мынаны есте ұстаған жөн: түйін print әдісін қолданатын System.out тұрақты шығару объектісі деп аталатын құрылыммен орындалады. Println әдісі қатар шығарылуынан соң курсордың жаңа бір класқа өтуін орындауға мүмкіндік береді.

2 – бағдарлама. Мәтінді бірнеше қатарлармен шығару.

```
public class Hello2 {
    // Java қосымшасы орындалуда, main әдісінен басталады
    public static void main(String[] args) {
```



```
System.out.println(" Java \пәлеміне \пқош \пкелдіңіз!");  
} // main әдісінің аяқталуы  
} // Hello2 класының аяқталуы
```

Зертханалық жұмыс №4. Сервер тұжырымдамасы.

Жергілікті желілердегі компьютерлердің бір-бірімен қарым-қатынас жасау сипатын олардың функциональдық қызметімен байланыстыру қалыптасқан. Берілгендерді өңдеу процесі клиент серверге күрделі процестердің орындалуына, файлды оқуға, берілгендер базасынан информация іздеуге және т.б сұраныс жібере алады. Сұраныс орындалып, нәтижесін сервер клиентке жөнелтеді. Клиент алынған берілгендерді өңдеп, нәтижені қолданушыға ыңғайлы түрде береді. Осындай байланысқан жүйе клиент сервер деп аталады. Жергілікті желі аймағында да компьютердерді тікелей байланыстыру кезіндегі сияқты клиент және сервер ұғымдары қолданылады.

Клиент-сервер технологиясы – жергілікті желідегі компьютерлерді бір-бірімен байланыстырудың ерекше тәсілі, мұнда бір компьютер (сервер) өз құрылғыларын басқаларға – клиенттерге пайдалануға бере алады. Осыған орай жергілікті желілер бір деңгейдегі желілер және серверлік желілер болып екіге бөлінеді.

Бір деңгейдегі желілерде айрықша бөлінген сервер болмайды, мұнда әрбір жұмыс станциясы әрі сервер, әрі клиент болып қызмет атқара береді. Мұның ерекшелігі – әрбір жұмыс станциясы өз ресурстарының белгілі бір бөлігін барлық жұмыс станцияларының ортақ пайдалануы үшін бөліп береді.

Көбінесе бір деңгейдегі желілер мүмкіндіктері бірдей компьютерлерді біріктірген кезде құралады. Бір деңгейдегі желілер жұмыс істеуге ыңғайлы, олардың қызметі де қарапайым түрде жүзеге асырылады.

Егер компьютерлер саны аз болып, олардың негізгі функциясы бір-бірімен мәлімет алмасу болған жағдайда, ең тиімді шешім – оларды бір деңгейлі желілерге біріктіру болып табылады. Мұндай желі кең тараған Windows 98 операциялық жүйесінің басқаруымен жұмыс істей береді.

Мәліметтердің компьютерлерге таралып орналасуы және әр жұмыс станциясының ресурстарын ортақтастыруды өзгерту мүмкіндігі бір деңгейдегі желілердегі мәліметтерді заңсыз пайдаланудан сақтауды қиындатады. Осыған орай мамандар мұндай желілердегі ақпаратты сақтау мәселелеріне өте көп көңіл бөлуге мәжбүр болып отыр.

Бір деңгейдегі желілердің тағы бір кемшілігі – олардың жұмыс өнімділігінің төмендігі. Ол ресурстардың әр жұмыс станцияларына бөлініп, жұмыс кезінде олардың әрқайсысының әрі клиент, әрі сервер қызметін атқаруына байланысты болып отыр.

Серверлік желілерде компьютерлердің функциялары алдын ала айқындалған – оның кейбіреуі сервер болып қызмет атқарса, қалғандары тұрақты түрде клиент болып табылады. Компьютерлік желілердің атқаратын қызметтерінің көп түрлілігіне байланысты серверлердің бірнеше типтері бар, олар: желілік сервер, файлдық сервер, баспа сервері, пошталық сервер, т.б.

Желілік сервер желіні жалпы басқару функциясын және есептеу жұмыстарының негізгі бөлігін атқаратын арнайы компьютер болып табылады. Бұл серверде желіні толық басқаратын операциялық жүйенің негізгі ядросы сақталады. Желілік сервердің жады көлемі үлкен және оның жұмыс жылдамдығы да жоғары дейгейде болуы тиіс. Мұндай желідегі жұмыс станцияларының функциясына мәліметтерді енгізу мен шығару және сервермен ақпарат алмасу жатады.

Файлдық сервер термині негізгі функциясы мәліметтер файлын сақтау, басқару және тасымалдау істерінен тұратын арнайы компьютерге байланысты шыққан. Ол өзінде сақталатын және тасымалданатын файлдарды өңдемейді және өзгертпейді. Сервер ол файлдардың мәтіннен, немесе графикалық бейнелерден, немесе электрондық кестеден тұратынын білуге тиісті емес.

Сервердің файл сервер түрі жиі қолданыста. Өйткені онда желіні қолданудың барлығынан түскен мәліметтер сақталады.

Сондай-ақ осы берілгендерге доступ қатамассыз етеді. Бұл компьютер үлкен жылдамдықты оперативті жаппен, қатты дискілермен, қосымша дискіжинағыштармен қамтылған болуы шарт. Файл сервер арнайы операциялық жүйелермен жұмыс жасайды. Ол операциялық жүйе бір мезгілде қолдануға доступ ашуы керек. Файл сервер файлдары:

- 1) берілгендерді сақтау;
- 2) архивтеу;
- 3) жөнелту;
- 4) әртүрлі қолданыстағы берілгендерді өзгертуді синхрондау;

Кейде файл сервер жеткіліксіз болсағ онда желіге бірнеше серверлерді қосуға болады. Әдетте орталықтандырылған ЭЕМ хост машина деп аталады. Берілгендерді өзгертуді орталықтандыр жүйелерінде қолданылады.

Жалпы файлдық серверде монитор мен пернелік тақта болмаса да болады. Файлдық мәліметтердегі барлық өзгертулер клиенттік жұмыс станцияларында жүргізіледі. Сондықтан клиенттер серверден файлдарды көшіріп алады да, керекті өзгерістерді жасап, нәтижесін қайтадан серверге қайтарады. Мұндай желі жұмысы – көптеген тұтынушылардың бір үлкен ортақ базаны біріге отырып пайдалануы кезінде өте тиімді болып табылады. Үлкен желілерде бір мезетте бірнеше файлдық серверлер пайдаланылуы мүмкін.

Баспа сервері (принтер-сервер) желі адаптері көмегімен мәлімет тасымалдау ортасына қосылған баспа құрылғысы болып табылады. Мұндай желілік баспа құрылғысы басқа желі құрылғыларынан тәуелсіз күйде өзі жеке жұмыс жасайды. Баспа сервері желідегі барлық серверлерден және жұмыс станцияларынан тапсырма ала отырып, олардың мәліметтерін басып шығарады. Баспа сервері ретінде қуатты, жұмыс өнімділігі жоғары принтерлер пайдаланылады.

Желілік басып шығарғыш – бұл жекелеген пайдаланушылар ортақ қатынаса алатын басып шығарғыш. Басқа пайдаланушылар сіздің басып шығарғышыңызға қатынауын ашу үшін мына әрекеттер орындалады:

1. Бастау – Басып шығарғыш пен факстар әмірін орындаңыз, Сіздің компьютеріңізге қосылған таңбашалары бар барлық басып шығарғыштармен факстарды сұқбат терезесі ашылады.

2. Сіз қатынас құруды қалаған сол басып шығарғыштың таңбашасын табыңыз да, онда тінтуірдің оң жақ батырмасын шертіңіз және ашылған мәзірден Ортақ қатынас құру мәзірін таңдаңыз.

3. Ашылған терезеде қатынас құру қосымшасына өтіңіз де, ауыстырып қосқышты берілген компьютерге ортақ қатынас құру жағдайларына орнатыңыз. Белсенді етілген мәтіндік мәтіндік өрісте желіні пайдаланушылардың бәрі көре алатын берілген басып шығарғыштың атауын енгізе аласыз.

4. Жарайды батырмасын басамыз.

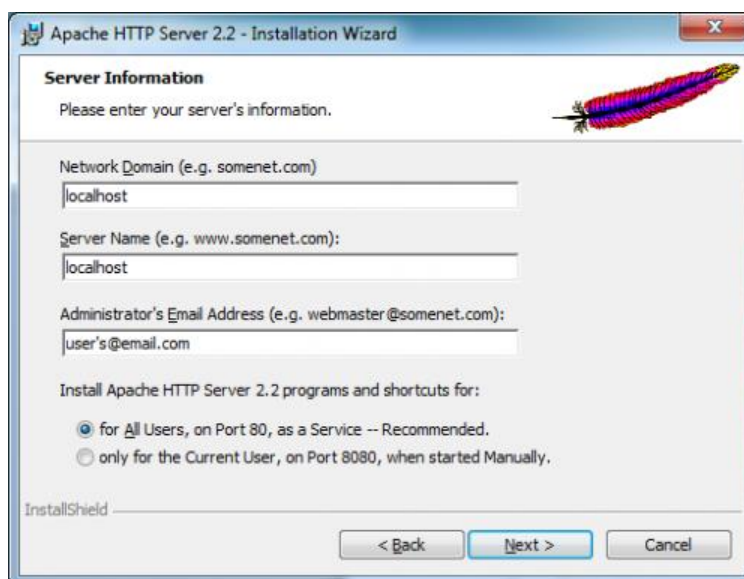
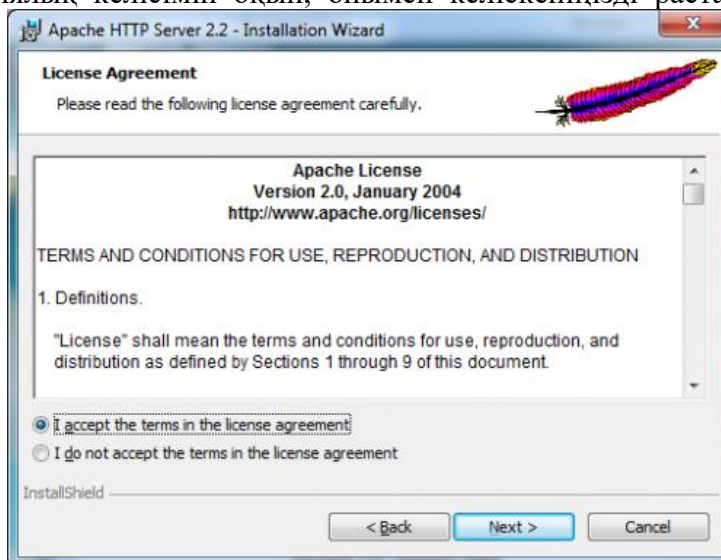
Жергілікті желі ауқымды желімен өте көп мәлімет алмасатын болған жағдайда, пошталық серверлер пайдаланылады. Олар электрондық пошта мәліметтерін өңдеу үшін қолданылады. Интернет желісімен тиімді қатынас құру үшін Web-серверлер пайдаланылуы мүмкін.

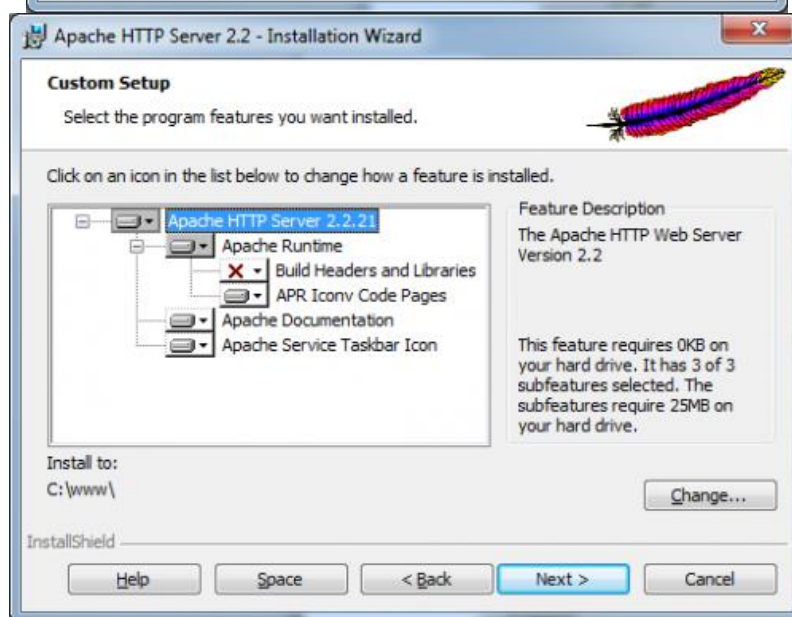
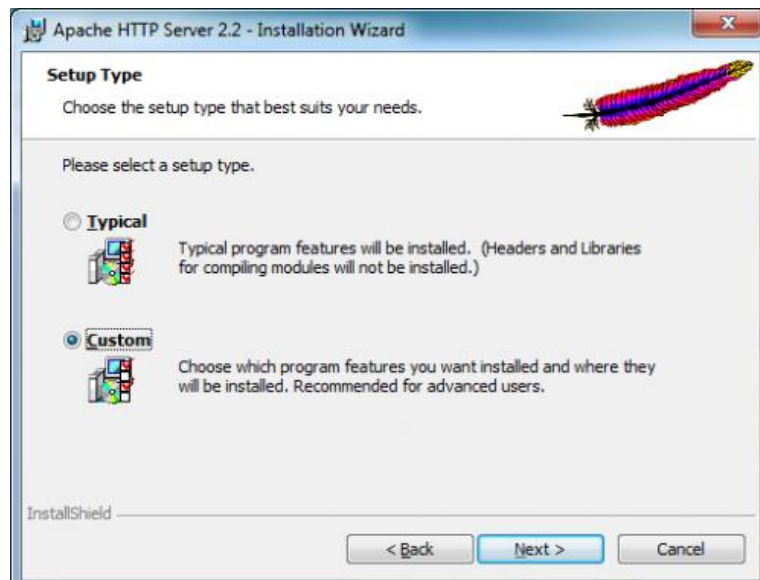
Клиент сервер архитектурасы бір рангілі локальді есептеу желісін де, ерекшеленген серверлі желіні де қолдана алады.

Зертханалық жұмыс №5. Серверді орнату.

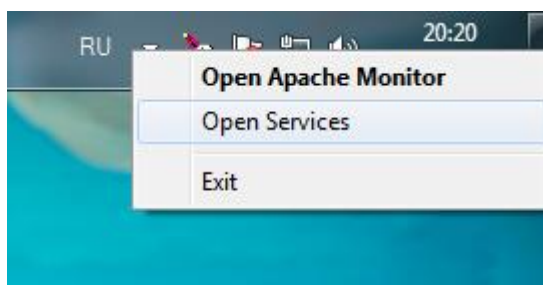
Apache орнату

Біріншіден, SSL қолдауынсыз Apache дистрибуциясының соңғы нұсқасын жүктеп, орнатуды бастау керек. Сәлемдесу терезесінде «Келесі» түймесін басып, әзірлеушінің лицензиялық келісімін оқып, онымен келіскеніңізді растаңыз.





Егер бәрі ойдағыдай болса, Apache веб-серверін басқару белгішесі науада пайда болуы керек. Тінтуірдің сол жақ батырмасымен басу арқылы оны тоқтатып, қайта бастауға болады.





It works!

Негізгі веб-серверді күйге келтіру

Apache өте ыңғайлы және өте сенімді болғанымен, әркім оны бірнеше себептер бойынша жергілікті сервер ретінде пайдалануға дайын емес, ал бастысы - Microsoft Windows жүйесінде де кез-келген графикалық конфигурацияшының жоқтығы, бұл көптеген қолданушылар үшін әдеттен тыс.

Сервер конфигурациясы httpd.conf конфигурация файлымен қолмен өңдеу арқылы жүзеге асырылады. Алайда, бұл процестің түсініксіздігі мен күрделілігінің идеясына қарама-қайшы, бұл операцияда екі себеп бойынша күрделі ештеңе жоқ: біріншіден, қолайлы және ыңғайлы жұмыс істеу үшін конфигурацияланған Apache серверін жасау үшін конфигурация файлы өзгерту керек біршама деректер, екіншіден, httpd.conf түсініктемелері конфигурацияны түсіну үшін қажет көптеген пайдалы ақпаратты қамтиды.

- Windows 7 платформасында Apache іске қосыңыз,
- Пайдаланушыға ыңғайлы каталогта болашақ сайттың файлдарын сақтаңыз (мысалы, C: www),
- Кодтау, әсіресе кирилл алфавитін көрсету кезінде қиындықтар туындамаса,
- Бір уақытта бірнеше сайтпен жұмыс істеу мүмкіндігі.

Осы мәселелерді сәтті шешу үшін Apache орнатылған қалтаға өтіп, conf ішкі қалтасынан httpd.conf файлы тауып ашыңыз. Назар аударыңыз, «фунттан» басталатын жолдар - бұл мәтіндік түсініктеме, ал веб-сервердің параметрлері басында фунт белгісі жоқ жолдарды пайдаланады.

Алдымен сервер веб-сайт папкасы ретінде пайдаланатын каталогты таңдау керек. Мәтіндегі DocumentRoot-тен басталатын жолды табыңыз. Бұл директива қай қалта түбір болатынын анықтайды.

Оны DocumentRoot бағдарламасында «C: / www» түзетіңіз. Бұл файлдағы қиғаш сызықтар Windows пайдаланушылары әдеттегідей солға емес, оңға және солға қарай қисайғанын нақтылау керек. Жоғарыда аталған манипуляциялардан кейін сіздің сайтыңыздың орналасуы туралы мәселе шешілді.

Әрі қарай, каталог үшін қауіпсіздік ережелерін орнату керек. Apache осы мақсатта Каталог директивасын қолданады:

Пайдаланылған параметрлер тізімі:

Опциялар Келесі опциялар мүмкін:

- Қамтиды - SSI рұқсат етілген
- NOEXEC қамтиды - SSI шектеулермен рұқсат етілген (# және #exec кірмейді),
- Индекстер - индекстелген файлдарды қолдануға рұқсат етіледі, егер URL мекен-жайы индексті файл жоқ сайттың каталогына сілтесе (мысалы, www.domain.com/dir/), осы каталогтың мазмұны көрсетіледі, ал егер мұндай мүмкіндік болмаса, хабарлама жіберіледі бұл қол жетімді емес
- ExecCGI - CGI сценарийін орындауға рұқсат етілген,
- FollowSymLinks - сервер каталогтың символдық сілтемелерін қадағалайды (Unix жүйелерінде қолданылады),
- SymLinksIfOwnerMatch - егер мақсатты файлда сілтеме бар иесі болса ғана, сервер каталогтың қол жетімді символдық сілтемелерін қадағалайды,

- Барлығы - жоғарыда айтылғандардың бәріне бірге рұқсат етіледі,
- Ешқайсысы - жоғарыда айтылғандардың барлығына тыйым салынады,
- MultiViews - белгілі бір мазмұнды таңдап, оны браузердің қалауына қарай көрсету мүмкіндігі (бәрі қосылған болса да (Опциялар Барлығы). Бөлек көрсетілген).
- AuthConfig - авторизация үшін директиваларды қолдануға мүмкіндік береді,
- FileInfo - әр түрлі құжаттармен жұмыс істеуге арналған директиваларды қолдануға мүмкіндік береді,
- Индекстер - индексті файлдармен жұмыс істеуге арналған директиваларды қолдануға мүмкіндік береді,
- Шектеу - хостқа қол жетімділікті анықтау үшін директиваларды қолдануға мүмкіндік береді,
- Опциялар - директиваларды анықтамалықтардың белгілі бір функцияларымен жұмыс істеуге мүмкіндік береді,
- Барлық - жоғарыда айтылғандардың барлығы бірге,
- Ешқайсысы - жоғарыда көрсетілгендердің ешқайсысы бірге болмайды.
- Растау, Рұқсат беру - Рұқсат беру директивасына дейін анықталған, кіруден бас тарту жолдан кейін көрсетілген хосттардан басқа, әдепкі бойынша рұқсат етілген,
- Рұқсат ету, Қабылдамау - Рұқсат беруден бас тарту директивасына дейін анықталған, рұқсат етілмегеннен кейін жолда тізімделген хосттардан басқа, кіру әдепкі бойынша қабылданбайды
- Өзара сәтсіздік - рұқсат беруде жоқ және рұқсат берілмеген хосттарға ғана рұқсат етіледі.

Жоғарыда айтылғандарға сүйене отырып, серверіңізді орнатып көріңіз. Httpd.conf файлында, әдетте, Директория директивасы бірден екі жағдайда болады - және. Бірінші нұсқаға қол тигізбеу керек, сондықтан екінші параметрде келесідей параметрлер орнатылады:

Басқаша айтқанда, C: / www қалтасына және оның барлық ішкі қалталарына келесі параметрлер таңдалды:

- Қолданыстағы сервердің мүмкін болатын функцияларының ішінен каталогтардағы индекстерге және символдық сілтемелерге ауысуға рұқсат етіледі,
- .htaccess файлдарын пайдалану арқылы параметрлерді алдын-ала ауыстыру мүмкіндігі мүлдем жоқ, алайда серверге толық қол жетімділік болған жағдайда, бұл маңызды емес - барлығын httpd.conf арқылы конфигурациялауға болады,
- Веб-серверге барлық хосттардан рұқсат етілген.

Енді httpd.conf файлын сақтаңыз және Apache Monitor немесе пәрмен жолында apache -k қайта қосу пәрменін пайдаланып Apache қайта іске қосыңыз. Сайттың түбірлік қалтасын конфигурациялау аяқталды.

Сіз бәрін дұрыс жасағаныңызды тексеруіңіз керек. C: www қалтасында қарапайым веб-парақ құрыңыз, шолғышты ашып, http://127.0.0.1/your_created_page енгізіңіз. Беттер ашылуы керек. Әйтпесе, httpd.conf файлындағы барлық өзгерістердің дұрыстығын мұқият тексеру қажет.

Егер бет сәтті ашылса, кириллица таңбаларының орнына оқылмайтын таңбаларды көруге мүмкіндік бар. Оларды екі себеп бойынша көрсетуге болады. Біріншіден, веб-сервер парақты сұраған шолғышты әдепкі кодтамамен қамтамасыз етеді. Екіншіден, таңқаларлық жай, бұл кодтау кириллица емес.

Егер әйгілі Internet Explorer мұндай жағдайларда кодтауды парақтың өзінен анықтаса, онда, мысалы, Mozilla Firefox және Opera мұндай әрекеттерге мүлдем

бейім емес және браузерде кодтауды қолмен орнатудың әдісі ыңғайлы деп айту қиын. Сондықтан, сіз Apache-ны әдепкі кодтауды қамтамасыз ету үшін конфигурациялауыңыз керек.

AddDefaultCharset-тен басталатын сызықты httpd.conf файлынан табыңыз. Сірә, онда кириллица таңбалары жоқ ISO-8859-1 шифрланған. ISO-8859-1-ді windows-1251 етіп өзгертіңіз, файлды сақтап, Apache қайта іске қосыңыз. Енді сіздің сайтыңыздағы орыс тілін дұрыс көрсету кез-келген шолғышта жұмыс істейді.

Apache серверінде бірнеше сайттармен жұмысты конфигурациялау жеткілікті. Олар үшін қолдануға болатын мекен-жайлар: 127.0.0.2, 127.0.0.3, т.б. Сонымен бірге, желіде әдепкі сайт (127.0.0.1) ғана көрінеді, алайда бұл жергілікті жұмыс үшін маңызды емес. Бұл үшін қажетті барлық нәрсе конфигурацияланған httpd.conf файлының бөлімі ең соңында орналасқан және VirtualHosts деп аталады.

Қосымша сайт қосу үшін оның түбірі ретінде пайдаланылатын каталог жасаңыз, мысалы, C: www2. Сайт 127.0.0.2 адресіне жауап береді деп болжап, сайтқа 911 атау беріңіз және VirtualHosts бөлімінің соңында келесі жолдарды қосыңыз:

Apache серверін қайта іске қосыңыз. Шолғыштың мекен-жай жолына 127.0.0.1 теру арқылы сіз өзіңіздің бірінші жергілікті сайтыңызға, ал 127.0.0.2 - екінші жергілікті сайтқа кіресіз. Әр VirtualHosts контейнерінде барлық Apache веб-серверінің директивасын қолдануға болатындығын есте ұстаған жөн, сондықтан сіз әр сайтты жақсы түрде теңшей аласыз.

Орнату және конфигурациялау аяқталды, енді сіз тәжірибеде Apache серверінің жұмысын толығымен зерттей бастай аласыз.

Егер сіз веб-сайтты басқарумен тығыз айналысқыңыз келсе, сіздің келесі қадамыңыз Apache PHP MySQL топтамасын зерделеу болып табылады, өйткені дерекқорды басқару жүйесінсіз, сонымен қатар веб-бағдарламалаудың ең кең таралған тілдерінің бірін және MySQL жүйесін басқаруға арналған веб-құралды қолдамай, бүкіләлемдік ғаламтордың кеңдігінде сервердің құны жоқ. «Оқу қиын, бірақ ұрыста оңай» дегендей.

Зертханалық жұмыс №6. HTML –мен танысу.

HTML файлының анықтамасы

Кез келген HTML-құжатында орналасатын ең бірінші тэг – бұл `<HTML>...` `</HTML>`. Бұл тэг арқылы біз құжатта HTML- мәтіні бар екендігін білеміз. Барлық жазылып отырған құжаттағы мәліметтер осы тэгтардың ішінде орналасуы қажет.

`< HTML >`

...

Құжат мәтіні

...

`</HTML >`

Сіздің құжатта шынымен HTML-командалары бар екендігін білдірдіңіз. Құжат негізі екі бөліктен: "тақырыбынан" және "құжат денесінен" тұрады.

Бұл 3 тэг стандарт бойынша қатысты қарастырылғанымен, кейбір браузерлер құжатты олардың көмегінсіз-ақ бейнелей алады. Бірақ бұл тэгтарды сілтеп кету керек, себебі келешекте тэгтар қажет болған жағдайда құжатты редакциялау қажет болады.

1-жаттығу. Қарапайым HTML файлдарын құрастыру.

1. Құрастырылған Web-парақты сақтау үшін KURS-бумасын ашыңыз.
2. Блокнот (Notepad) стандарт программасын жүктеңіз.
3. Қарапайым HTML-файлының мәтінін редактор терезесінде теріңіз.

```
<HTML >
<HEAD>
<TITLE> HTML оқу файлы </TITLE>
</HEAD>
<BODY>
Сейсенбіге арналған сабақ кестесі
</BODY>
</HTML >
```

4. RASP.HTM атымен файлды сақтаңыз.

5. Құрастырылған Web-парақты көру үшін Microsoft Internet Explorer браузерін жүктеңіз.

6. Файл (File), Открыть (Open), Просмотр (Обзор, Browse) командаларын орындап KURS бумасының ішінен RASP.HTM файлын тауып, оны жүктеңіз. Web-парақтың (“HTML оқу файлы”) атауы браузердің жоғарғы статусы жолында бейнеленгенін көріп тексеріңіз.

HTML-құжатын бейнелеген кезде браузерлер автоматты түрде мәтінді экранда орналастырады. Ол файлда кездесетін символдар мен бос жол адымдарына мән бермейді.

Жолдың тасымалдауын орындайтын және жаңа абзацтың басталуын беретін арнайы командалар бар. Одан қалса браузер программасына форматтауды тыйым салатын және экранға дәлме-дәл мәтін файлының үзіндісін бейнелейтін командалар бар.

**
** - жолды тасымалдаушы тэг жолды келесі мәтіннен немесе графикадан бөліп тұрады. **<P>** тәгі де жолды бөледі, сонымен қатар абзацты айқындап тұратын бос жол қалдырады. Бұл 2 тэг бірэлементті болып табылады.

2-жаттығу. Жол-тасымалдау мен абзац тэгтарын қолдану.

1. HTML файлының мәтініне өзгеріс еңгізіңіз.

```
<HTML>
<HEAD>
<TITLE> HTML оқу файлы </TITLE>
</HEAD>
<BODY>
Сейсенбіге арналған <P>
сабақ <BR>
кестесі
</BODY>
</HTML >
```

2. Өзгертілетін мәтінді RASP.HTM файлында сақтаңыз.

3. Шыққан Web-парақты Microsoft Internet Explorer браузері көмегімен қарап шығыңыз. Экрандағы мәтін көрінісі қалай өзгерді?

Мәтін үзінділерін ерекшелеу

Мәтін үзінділерін ерекшелеу тэгтері жеке символдар мен сөздердің бейнелеуін басқаруға мүмкіндік береді. Осы мақсатта үш түрлі тэгтар қолданылады:

... - жартылай қалың қарамен ерекшелеу (полу жирный)

<I>...</I> - курсивпен ерекшелеу
<U>...</U>-астын сызу арқылы ерекшелеу

Құрама ерекшелеу қолданылуы мүмкін. Мысалы, <I> Сейсенбіге арналған </I> <I> сабақ </I> <U> кестесі </U> деп өзгерту енгізіңіз.

Бірақ бұл жағдайда құрама тэгтардың жазылу ережесін есте сақтау керек.

<1-тэг> <2-тэг>...</ 2-тэг> </ 1-тэг> - дұрыс жазу

<1-тэг> <2-тэг>...</1-тэг> </2-тэг> - қате жазу.

Символ өлшемдерінің берілуі

Браузер арқылы бейнеленген мәтінді басқарудың 2 тәсілі бар.

- Тақырып стилін қолдану;
- Негізгі құжат шрифтінің өлшемін беру немесе ағымдағы шрифтінің өлшемін беру;

Тақырып стилін қолдану. Мұнда б-тақырып тәгі қолданылады (<H1>-ден <H6>-ға дейінгі). Әрбір тэгке браузер параметрлерінің берілген нақты стили сәйкес келеді.

3-жаттығу. Тақырып стилін қолдану.

1. RASP.HTM файлына өзгеріс еңгізіңіз.

```
< HTML >
```

```
<HEAD>
```

```
<TITLE> HTML оқу файлы </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<H1> сейсенбіге арналған </H1> <I> сабақ </I> <U> кестесі </U>
```

```
</BODY>
```

```
</ HTML >
```

2. Шыққан Web-парақты қарап шығыңыз.

Ағымдағы шрифтінің өлшемін беру. шрифтік тәгі 1-ден 7-ге дейінгі диапазонда мәтіннің кез келген жерінде ағымдағы шрифтінің өлшемін беруге мүмкіндік береді.

4-жаттығу. Ағымдағы шрифтінің өлшемін беру.

1. RASP.HTM файлына келесі өзгерістер енгізіңіз.

```
< HTML >
```

```
<HEAD>
```

```
<TITLE> HTML оқу файлы </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<FONT SIZE = "7"> сейсенбіге арналған </FONT>
```

```
сабақ кестесі
```

```
</BODY>
```

```
</HTML >
```

2. Өз бетіңізбен “сейсенбіге арналған сабақ кестесі” мәтінінің шрифт өлшемін тәгі арқылы өзгертіңіз.

3. HTML-құжатының хаттамасын фрагментті ерекшелеу және жолды тасымалдау мен абзац тэгтары арқылы өзгертіңіз.

Гарнитура мен шрифт түсі

 тәгі гарнитура пішіні мен мәтін түсін өзгерту мүмкіндіктерін береді. Шрифт гарнитурасын өзгерту қарапайым тәгіне FACE атрибутын қосу арқылы жүзеге асады. Мысалы, мәтінді Arial шрифтімен жазу үшін деп жазу керек.

Шрифт түсін өзгерту үшін тәгінде COLOR = "X" атрибутын қолдануға болады. X-тің орнына түстің атын (тырнақшада) қоямыз немесе 16-лық мәндегі сандар жиынтығын жазамыз. Түсті 16-лық сандармен беру үшін, бұл түсті әрқайсысы 0-дан FF-қа дейінгі мәнге ие болатын, жіктелген үш құрамға бөліп көрстеуге болады: қызыл (R-Red), жасыл (G-Green), көк (B-Blue). Бұл жағдайда біз RGB деп аталатын форматпен жұмыс істейтін боламыз.

5-жаттығу. Шрифтінің түсі мен гарнитурасын беру.

1. RASP.HTM-құжатына келесі өзгерістер енгізіңіз.

```
< HTML>
<HEAD>
<TITLE> HTML оқу файлы </TITLE>
</HEAD>
<BODY>
<U><I><B><FONT COLOR = "#FF0000" FACE="ARIAL" SIZE = "7">
Сейсенбіге арналған сабақ кестесі </U></I></B>
</BODY>
</ HTML>
```

2. Өз бетіңізбен құжат мәтінінің өлшемін, түсін, гарнитурасын, стилін өзгертіңіз.

Мәтінді туралау

HTML-тәгтарының көмегімен мәтіннің туралауды басқаруға болады. Егер туралау тәсілі қорсетілмесе, құжаттағы барлық элементтер сол шеттен тураланып, оң жағы ретсіз орналасады.

Қазіргі браузерлер ALIGN = : мәтінді туралау атрибутын қолданады.

ALIGN = CENTER – центрі бойынша туралау

ALIGN = RIGHT- оң шеті бойынша туралау

ALIGN = LEFT – сол шеті бойынша туралау.

6-жаттығу. Мәтінді туралау.

1. RASP.HTM файлына келесі өзгерістер енгізіңіз.

```
< HTML>
<HEAD>
<TITLE> HTML оқу файлы </TITLE>
</HEAD>
<BODY>
<P ALIGN = CENTER>
<FONT COLOR = "#008080" SIZE = "7"> <B>
Сейсенбіге арналған </B> </FONT> <BR>
<FONT SIZE = "6"> <I> сабақ кестесі </I> </FONT>
</P>
</BODY>
</ HTML>
```

Зертханалық жұмыс №7.HTML: Жалпы құжат құрылымы

Негізгі түсініктер:

 тэгінде көптеген атрибуттар бар, оларға төменде көрсетілгендей қосымша жұмыстар жүктеуге болады. Олар IMG кодынан кейін тэктің кез-келген жерінде орын алады.

SRC – ең міндетті параметр. Суреті бар файлдың мекен-жайын (URL) көрсетеді.

HEIGHT және WIDTH – сәйкесінше суреттің биіктігі мен ұзындығын анықтайды. Егер берілген өлшемдер нақты өлшемнен өзгеше болса, онда сурет масштабталады (кейде көрнекті сапасын жоғалтады).

HSPACE және VSPACE – бұлар суреттің құжат объектілерінен көлденең және тігінен қанша шегіністе орналасқанын анықтайды.

ALIGN – міндетті параметр. Құжаттағы суреттің туралануын береді. Келесідегідей мәндерге ие болады:

left – суретті құжаттың сол шетінен туралайды. Оған қатысты мәтін оң жағынан көрінеді.

right – суреттегі құжаттың оң шетінен туралайды. Оған қатысты мәтін сол жағынан көрінеді.

top және textop – суреттің жоғарғы жиегіндегі мәтін жолының ағымдағы жоғарғы сызығын туралайды.

middle – қазіргі ағымдағы мәтін жолының базалық сызығы сурет центрмен тураланады.

absmiddle –ағымдағы мәтін жолының центрін сурет центрмен туралайды.

bottom және baseline –ағымдағы мәтін жолының базалық сызығы мен суреттің төменгі жиегін туралайды.

absbottom – суреттің төменгі жиегін ағымдағы мәтін жолының төменгі жиегімен туралайды.

NAME – Осы құжатқа арналған суреттің атын анықтайды. Сіз кез келген атты бос жол қалдырмай латын символдары мен цифрлары арқылы бере аласыз. Егер сіз суретке кіру жолын, мысалы, Java-Script-сценарийлері жолдарынан жүзеге асырсаңыз атау керек болады.

ALT – егер браузер керекті суреті бар файлды таба алмаса немесе мәтіндік ережеге қосылмаса, суретті сол жерде браузер бейнеленген мәтінді анықтайды.

BORDER – суреттің айналасындағы рамканың ұзындығын пикселдерде анықтайды. Егер сурет гипермәнді сілтеу болса ғана рамка шыға келеді. Бұндай жағдайда BORDER мәні нольге тең.

LOW SRC – бұл SRC параметріндегі берілген суреттен сапасы да, көлемі де төмен суреті бар файлдың мекен-жайын (URL) береді. Бұл параметрді қолдаушы браузерлер бірінші LOW SRC-ден сурет жүктейді, сосын оны SRC-тегі суретпен алмастырады.

USEMAP – MAP элементімен берілген навигациялық картаны (image map) суретке қолданады. Мағына ретінде картаның алдында торшасы бар атымен беріледі. Мысалы, карта аты – “map1” болса, онда оған қатысты ссылақ мынадай болады: “#map1”. Бұл жерде назар аударатын нәрсе жазу және басылым әріптері бұл параметрде браузерлер арқылы әр түрлі деп беріледі.

ISMAP – Сервер өңдейтін навигациялық карта (image map) ретінде суретті анықтайды. Сурет гиперссылақ болған кезде қолданылады. Тышқанымен шақыру кезінде серверге басылған x, y – координаталары жіберіледі. Алынған координаталарға байланысты сервер (егер сәйкес программалық қамтамасыз етілсе)

сізге кез келген құжатты көрсете алады. Бұл параметр мағынаға ие болмай-ақ жалау ретінде болады.

Мысал1.

```
<IMG src="img/picture.gif" WIDTH="45" HEIGHT="53" ALT="Рысь"
HSPACE="10" ALIGN="left">
```

Бұл мәтін суреттің оң жағынан көрінеді және суреттен 10 пиксель қашықтықта орналасады.

Мысал2.

Суретті гиперссылка түрінде қолдану.

```
<A HREF="carlson.html">
<IMG src="img/button.jpg" WIDTH="70" HEIGHT="30" ALIGN="right"
BORDER="0" ALT="Досье Карлсона">
</A>
```

Досьені қарап шығу үшін оң жақ батырмасын басыңыз.

Мысал3. ISMAP-ты қолдану.

```
<A HREF="http://www.igf.ru/bin/imagemaps/map1">
<IMG SRC="map.gif" ISMAP></A>
```

Мысал4 .USEMAP-ты қолдану:

```
<IMG src="img/buttons.jpg" WIDTH="170" HEIGHT="120" ALIGN="middle"
BORDER="0" USEMAP="#ButtonsMap">
```

1-жаттығу. Бейне атрибуттарын орналастыру

RASP.HTM файлына өзгеріс енгізіңіз: ALT, BORDER, HEIGHT, WIDTH графикалық атрибуттарын қолданып көріңіздер.

Ескерту. Әрқашан графикалық файлдың пішініне (көлемі байтпен) назар аударыңыз, себебі ол Web-парақтың жүктелу уақытына әсер етеді.

Кестелер дайындау

Кестелер HTML құжатының маңызды бөлігі болып табылады. Ол тік бағандар мен көлденең жолдардан тұратын тікбұрышты торлар. Жолдар мен бағандардың қиылысуы ұяшық деп аталады. Ұяшықта мәтін, графика немесе басқа кесте орналасуы мүмкін.

Кесте негізгі үш бөліктен тұрады:

- кесте атауынан
- бағандар тақырыбынан
- ұяшықтардан

Кесте жолдар бойынша толтырылады (солдан оңға жол бойынша және келесі жолға көшу). Әрбір ұяшық толтырылуы тиіс.

Кестені дайындау тэгтері және олардың атрибуттары 3 -5 кестелерде берілген

Кесте безендіру тэгтері

Тэг Форматы Сипаттамасы

TABLE <TABLE> мәтін </TABLE> Кестені жариялау

TR <TR> мәтін </TR> Жолды көрсететін тэг

TD <TD> мәтін </TD> Бағанды көрсететін тэг

<TABLE> тәгінің атрибуттары

Атрибут Форматы Сипаттамасы

BORDER <TABLE BORDER =1> Кесте айналасына қоршау орнатады

WIDHT <TABLE WIDHT=60%> Кесте енін парақ еніне байланысты процентпен немесе пикселмен береді

BGCOLOR <TABLE BGCOLOR ="#RRGGBB"> Кесте фонының түсін береді

<TD> және <TR> тэгтарының атрибуттары
Атрибут Форматы Сипаттамасы
ALIGN <TR ALIGN=RIGHT> Көлденеңінен туралауды тағайындайды (RIGHT, LEFT, CENTER)
VALIGN <TR VALIGN=TOP> Тігінен туралауды тағайындайды (TOP, MIDDLE, BOTTOM, BASELINE)
BGCOLOR <TR BGCOLOR ="#CCCCFF"> Фон түсін береді

3-жаттығу. Кесте құру.

1.Блокнот программасын жүктеңіз

2.Редактор терезесінде мына мәтінді теріңіз:

```
<HEAD>
<TITLE>5-сынып сабақ кестесі</TITLE>
</HEAD>
<BODY COLOR="FFFFFF">
<P ALIGN=CENTER>
<FONT COLOR="RED"SIZE="6"FACE="ARIAL">
<B>5 КЛАСС</B></FONT><BR></P>
<FONT            COLOR="BLUE"            SIZE="4"            FACE="OOURIER">
<B>Дүйсенбі</B></FONT><BR>
<TABLE BORDER="1" WIDTH=100% BGCOLOR="99CCCC">
<TR BGCOLOR="CCCCFF" ALIGN=CENTER>
<TD>Сабақ</TD><TD>5 А</TD><TD>5 В</TD><TD>5 В</TD>
</TR>
<TR>
<TD>1</TD><TD>Орыс тілі</TD><TD>Адебиет</TD><TD>Тарих</TD>
</TR>
<TR>
<TD>2</TD><TD>Алгебра</TD><TD>Ақпараттану</TD><TD>Ағылшын
тілі</TD>
</TR>
<TR>
<TD>3</TD><TD>Тарих</TD><TD>Ақпараттану</TD><TD>Алгебра</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

3. Құжатты 5.HTM атымен сақтаңыз

4. Web-парақты көру үшін Microsoft Internet Explorer браузерін жүктеңіз

4-жаттығу. Кесте құру.

Шыққан Web-парақты келесі күндергі сабақ кестесімен толықтырыңыз.

Бір құжат аймағындағы сілтеме

Бұл сілтемелер үшін 2 бөліктің бар болуы шарт: таңба және сілтеменің өзі. Сілтеме таңба атын қолданады. Сілтемелер ерекше түспен немесе астын сызу

арқылы ерекшеленеді. Бұл браузердің орналасуына байланысты болады. Сілтеме түсін өзгерту үшін <BODY> тәгінің LINK және VLINK атрибуттары қолданылады.

Сілтеме сипаттамасы.

```
<A HREF = "#DC"> Дүйсенбі </A>
```

Таңба (DC) алдына қайда көшуін бағыттайтын # символы қойылады. ">" және "<" символдары арасында мәтін жазылады. Бұл мәтінде сілтеме бойынша көшу шертпесі жүзеге асады.

Таңбаны анықтайық.

```
<A NAME = "DC"> Дүйсенбі </A>
```

5-жаттығу. Бір құжат аумағында сілтеме құру.

1. 5.HTM файлын Web-парақтың басына апта күндерінің аттарын жазып толықтырындар:

```
<HTML>
<TABLE WIDTH=100%>
<TR>
<TD>Дүйсенбі</TD>
<TD>Сейсенбі</TD>
<TD>Сарсенбі</TD>
<TD>Бейсенбі</TD>
<TD>Жума</TD>
<TD>Сенбі</TD>
<BODY BGCOLOR="FFFFFF">
</TR>
</TABLE>
<BR>
<HEAD>
...
</HTML>
```

3. Файлды 5.HTM атымен сақтаңыз.

4. Файлды қарап шығу үшін Microsoft Internet Explorer браузерін жүктеңіз.

Гипермәтіндік байланыстар құру

HTML-тілінің бір құжаттағы сілтемелерде екінші бір құжатқа көшіру мүмкіндігі маңызды қасиеті болып саналады. Сілтеме мүмкіндіктері:

1. HTML файлына дүйсенбіге сілтейтін таңба қойыңыз:

```
...
<FONT COLOR="BLUE" SIZE="4" FACE="OURIER"><B><A NAME="DC">
Дүйсенбі </A></B></FONT><BR>
```

...
3. Таңдалынған таңбаға апта күндерінің атауына сілтеме жасаңыз.

```
...
<TABLE WIDTH=100%>
<TR>
<TD><A HREF ="DC">Дүйсенбі </A></TD>
<TD>Сейсенбі</TD>
<TD>Сарсенбі</TD>
```

...
4. Файлды сақтаңыз.
5. Шыққан Web-парақты қарап шығыңыз.

Басқа HTML-файлына сілтеме жасау
Сілтемелер ерекшеленген сөзде немесе сөйлемде батырманы басу арқылы басқа файлға өтуге мүмкіндік береді.

Сілтемені сипаттайық:

```
<A HREF = "5.HTM"> 5сынып </A>
```

Файл атынан (5. HTM) кейін ">" және "<" символдары арасында мәтін орналасады. Бұл сөзге шерткен кезде басқа файлға өтуге болады.

6-жаттығу. Басқа HTML-құжатына сілтеме жасау.

1. Браузер RASP.HTM файлын жүктеңіз.

2. Файлға келесі өзгерістер енгізіңіз.

```
<HTML>
<HEAD>
<TITLE> HTML оқу файлы </TITLE>
</HEAD>
<BODY BGCOLOR = "#FFFFFF" TEXT ="#330066">
<P ALIGN = CENTER>
<FONT COLOR = "#008080" SIZE = "7"> <B>Сейсенбіге арналған
</B></FONT> <BR>
<FONT SIZE = "6"> <I> сабақ кестесі </I> </FONT><BR>
<IMG SRC = "PH02750G.GIF">
</P>
<CENTER>
<TABLE WIDTH = 60%>
<TR> <TD> <A HREF="5.HTM"> 5сынып </A> </TD> <TD> 6сынып </TD>
</TR>
<TR> <TD> 7сынып </TD> <TD> 8сынып </TD> </TR>
<TR> <TD> 9 сынып </TD> <TD> 10 сынып </TD> </TR>
<TR> <TD> 11сынып </TD> <TD> </TD> </TR>
</TABLE>
</CENTER>
</BODY>
</HTML >
```

3. Файлды сақтаңыз.

4. Шыққан Web-парақты қарап шығыңыз.

Курсорды 5 сынып сілтемесіне әкеліп, сол сөзде шертіңіз, сонда басқа файлға көшесіз.

7-жаттығу. Басқа HTML-құжатына графикалық түрдегі сілтеме құру.

1. 5.HTM файлына өзгеріс еңгізіңіз, яғни ссылақ 5 сынып сабақ кестесі деген беттің басына сілтеме жасаңыз.

Сілтеме ретінде келесі графикалық файлды қолданыңыз.

```
...
</TR>
</TABLE> <BR>
<CENTER>
<A HREF = "RASP.HTM"> <IMG SRC = "HOME.GIF" BORDER = "0">
</A>
</CENTER>
</BODY>
</HTML>
```

2. Шыққан Web-парақты қарап шығыңыз. Сілтеме ретінде HOME.GIF файлындағы сурет шығып тұрады (бағыттаушы сызық, оңнан солға бағытталған).

Тапсырма нұсқалары

Нұсқа №	Тақырыбы
1.	Менің болашақ мамандығым
2.	Жол қозғалысы ережелері
3.	Қазақстанның даму тарихы
4.	Менің халқымның салт-дәстүрі
5.	Футбол клубтары және футболшылар
6.	Спорт түрлері
7.	Қазіргі замандағы музыка.
8.	Әлемнің атақты боксерлары
9.	Қазақстанның хайуанаттар әлемі
10.	Компьютерлік техниканың даму тарихы
11.	Қазақ киносының актерлері
12.	Біздің қаланың ірі өндіріс орындары
13.	Біздің қаланың кітапханалары
14.	Біздің қаланың жоғарғы оқу орындары
15.	Біздің ескі және сүйікті кинофильмдеріміз
16.	Қазақстан таулары.
17.	Қазақстанның курорттық аймақтары
18.	Менің сүйікті қалам
19.	Автомобиль маркалары
20.	Ұлттық тағамдар
21.	Қазақ эстрадасының жұлдыздары.
22.	Салалатты өмір салтын қолдаймыз.
23.	Дербес компьютердің құрылымы

Зертханалық жұмыс №8. HTML: Тізімдер

Негізгі түсініктер:

Тізімдер

HTML-де тізімдердің келесі түрлері бар:

- Таңбаланған
- Нөмірленген
- Көп деңгейлі тізімдер

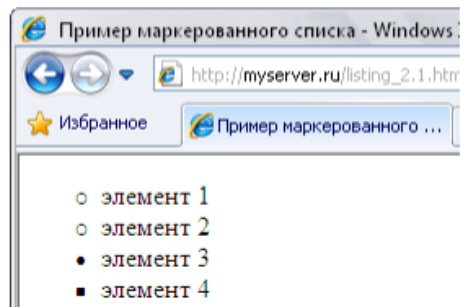
Таңбаланған тізімдер қою үшін `` және `` тегтері пайдаланады. `` тегі бүкіл тізімнің басы мен соңын белгілейді. `` тегі жеке тізім элементінің басы мен соңын белгілейді.

Әдепкі бойынша тізім элементтері қара шеңбермен белгіленеді. *Type* атрибуты көмегімен таңбаның төлсипатын өзгертуге болады.


```

1 <html>
2 <head>
3   <title>Пример маркерowanego списка</title>
4 </head>
5
6 <body>
7 <ul type="circle">
8   <li>элемент 1</li>
9   <li>элемент 2</li>
10  <li type="disc">элемент 3</li>
11  <li type="square">элемент 4</li>
12 </ul>
13 </body>
14 </html>
15
16

```

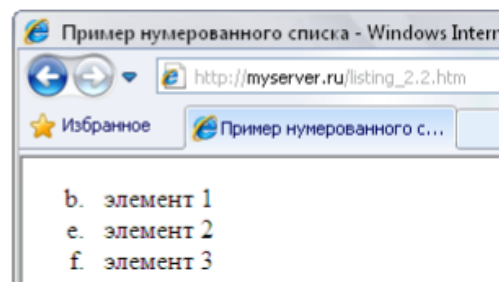


Таңбаланған тізімдерге ұқсас, және тегтерін пайдаланып нөмірленген тізімдер жасай аласыз. Нөмірлеу стилін таңдау үшін *type* атрибутын қолданамыз және тізім нөмірлеуін қай таңбадан бастауды көресту үшін *start* қолданылады. тегі үшін *value* атрибуты ағымдағы тізім элементінің санын орнатуға мүмкіндік береді.

```

1 <html>
2 <head>
3   <title>Пример нумерованного списка</title>
4 </head>
5
6 <body>
7 <ol start="2" type="a">
8   <li>элемент 1</li>
9   <li value="5">элемент 2</li>
10  <li>элемент 3</li>
11 </ol>
12 </body>
13 </html>

```



Анықтау тізімі - екі байланыс жиынтықтардан тұратын мәтіннен тұрады: терминдер тізімі мен терминдердің анықтамалары.

Анықтау тізімінің құрылымы

Термин 1

Термин анықтамасы 1

Термин 2

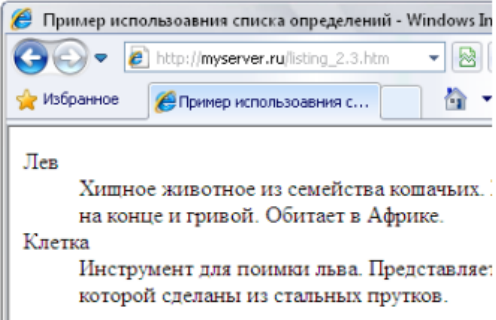
Термин анықтамасы 2

Тізім <dl> тегі арқылы беріледі, термин - <dt>, термин анықтамасы <dd>тегі арқылы беріледі.

```

1 <html>
2 <head>
3 <title>Пример использования списка определений</title>
4 </head>
5
6 <body>
7 <dl>
8 <dt>Лев</dt>
9 <dd>Хищное животное из семейства
10 Обитает в Африке.</dd>
11
12 <dt>Клетка</dt>
13 <dd>Инструмент для поимки льва.
14 из стальных прутков.</dd>
15 </dl>
16
17 </body>
18 </html>

```



Сілтемелер

HTML тілінде мәтіннің бір фрагментінен екіншісіне өту үшін <a> (anchor-якорь) тегі қолданылады.

Атрибут	Анықтамасы
href	Гиперсілтеме арқылы өту жасалатын ресурстың атын көрсетеді. <a> тегінің негізгі атрибуты болып саналады. Салыстырмалы және абсолютті болып келеді.
name	Құжаттың сілтемесінің орналасуын анықтау үшін якорь атауын белгілейді.
target	Құжат жүктелген терезенің немесе фреймнің атын тағайындайды. Сілтеменің солтерезеде ашылуы үшін <code>_self</code> , ал басқа терезеде ашу үшін <code>_blank</code> пайдалынады.
title	Көмек немесе анықтама ретінде пайдалынады

```

1 <html>
2 <head>
3 <title>Тег А, атрибут name</title>
4 </head>
5
6 <body>
7 <p><a name="top"></a>
8 В эту точку будет осуществлен переход при щелчке на нижнюю ссылку</p>
9
10 <a href="http://google.com" title="" target="_blank"
11 title="ссылка откроется в новом окне">Поисковая система google</a>
12
13 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit,
14 tincidunt ut lacreet dolore magna aliquam erat volutpat.</p>
15
16 <p><a href="#top">Наверх</a></p>
17 </body>
18 </html>

```

Практикалық бөлімі.

1. Виртуалды машинада Денвер веб-серверін қосыңыз
2. Сіздің доменіңізде file lab3.htm файлын жасаңыз.
3. Құжатқа мұрағаттан сурет қойыңыз. Суреттің енін, ұзындығын, туралауын, кенестерін дұрыс туралап қойыңыз.
4. Суретке сәйкес анықтамалар тізімін жасаңыз. Ол үшін мына тегтерді пайдаланыңыз <dl>, <dt>, <dd>
5. тегін пайдаланып таңбаланған тізім жасаңыз. Тізім элементі қара квадрат болсын.

6. Нөмірленген тізім құрастырыңыз. Нөмірлеуді рим цифрмен болсын, нөмірлеуді XV деген саннан бастаңыз.
7. Көпдеңгейлі нөмірлеу тізімін жасаңыз.
8. Бүгінгі зертханалық жұмысының әрбір бөлігінің тақырыбына ``, `` атты белгілеңіз.
9. Беттің басында барлық барлық бөлімдерге сілтемелер тізімін қосыңыз. Атау анкерге сілтеме жасаған кезде анкер алдына # қоюды ұмытпаймыз.
10. Беттің төменгі жағына Денвердің орнату файлын жүктеу үшін сілтемені қойыңыз.
11. Беттің төменгі жағына өзінің вконтате бетіне сілтеме қойыңыз. Ол жаңа терезеде ашылсын.
12. Беттің екінші суретіне Google іздеу жүйесіне сілтеме жасаңыз.
13. my.html атты жаңа құжатты жасаңыз. Оған: сізге белгілі бағдарламалау тілдерінің нөмірленген тізімін жазыңыз, сіз білетін түстердің түрлерін таңбаланған тізімге жазыңыз.
14. Орындаған жұмысты мұғалімге көрсетіңіз.

Зертханалық жұмыс №9. HTML: кестелер, фреймдер.

Негізгі түсініктер:

Жылжымалы фреймдер

Жылжымалы фреймдер тек мәтінді мәтінге орналастыру үшін ғана емес, сонымен бірге басқа құжаттарға сілтемелер ұйымдастыру үшін де қолданады.

Жылжымалы фреймнің айналасында мәтін немесе басқа графикалық кескіндер орналасу мүмкін.

Өкінішке орай Netscape Navigator және NSCA Mosaic навигаторлары жылжымалы фреймдермен жұмыс істемейді. Жылжымалы фрейм HTML құжатына `<IFRAME>` және `</IFRAME>` операторлары арқылы орналасады. Бұл оператор үшін мәтінді туралайтын `ALIGN` параметрін, фрейм мөлшерлерін беретін `WIDTH` және `HEIGHT` параметрлерін, `` операторында қолданылатын `HSPECE` және `VSPACE` параметрлерін беруге болады.

Тағы да бір `FRAME BORDER` параметрі жылжымалы фреймді сипаттағанда қолданылады. Ол 1 немесе 0 мәнін қабылдайды. Егер бұл параметрдің мәні бірге тең болса, онда жылжымалы фрейм үш өлшемді жақтауға алынады, ал егер ол нөлге тең болса, онда ол жақтауға алынбайды. `SCROLLING` параметрі “yes”, “no” немесе “auto” мәндерін қабылдай алады.

Егер “yes” мәні берілсе, жылжымалы фреймнің терезесінде қарау жолақтары болады.

Егер “no” болса қарау жолақтары болмайды. Ал егер `SCROLLING` параметрлері үшін “auto” мәні берілсе, онда қарау жолақтары құжаттың мазмұны фрейм терезесіне сыймаған кезде ғана құралады.

Жылжымалы фреймі бар, HTML құжатының бастапқы мәтінін келтірейік.

Тапсырма – 1. Жылжымалы фреймдер

`<HTML>`

`<HEAD>`

`<TITLE> HTML құжатындағы жылжымалы фреймдер</TITLE>`

`</HEAD>`

`<BODY BGCOLOR=#FFFFFF>`

`<H2> HTML құжатындағы жылжымалы фреймдер </H2>`

Жылжымалы фреймі бар HTML құжаты:

```

<P><IFRAME SRC="1-1.html" ALIGN=LEFT>
</IFRAME>Мына берілген тік төрт бұрышты алаңда келесі бір HTML файлдағы
мәліметтері берілген
<BR CLEAR=LEFT>Бұл өте қызықты!!!
</BODY>
</HTML>

```

1-1.html файлындағы HTML құжатына жүктелетін жылжымалы фрейм бұл құжатта анықталған.

Тапсырма – 1.1 Жылжымалы фреймдер

```

<HTML>
  <HEAD>
<TITLE> HTML құжатындағы жылжымалы фреймдер </TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<P>АҚПАРАТТЫҚ ТЕХНОЛОГИЯЛАР ФАКУЛЬТЕТІ
<P>ЕСЕПТЕУШІ ТЕХНИКАСЫ ЖӘНЕ БАҒДАРЛАМАМЕН ҚАМТАМАСЫЗ ЕТУ
МАМАНДЫҒЫ
<P>ЕСЕПТЕУШІ ТЕХНИКАСЫ ЖӘНЕ БАҒДАРЛАМАМЕН ҚАМТАМАСЫЗ ЕТУ
МАМАНДЫҒЫ РЕСПУБЛИКА БОЙЫНША 3 ОРЫНДА.
СОНЫМЕН ҚАТАР БІЗДІҢ МАМАНДЫҚ СТУДЕНТТЕРІ ЖЫЛ САЙЫН
ОЛИМПИАДАҒА ҚАТЫСЫП БЕРЕКЕЛІ ОРЫН АЛУДА
</BODY>
</HTML>

```

Бұл жерде назар аударатын бір жер фреймнан кейін жаңа жолға мәжбүрлі өту үшін біз
 операторының CLEAR = LEFT параметрін қолдандық.

Көп терезе Веб-бетін әзірлеу

Мысалда үш терезелі Веб-бетті құру күрсетілген.

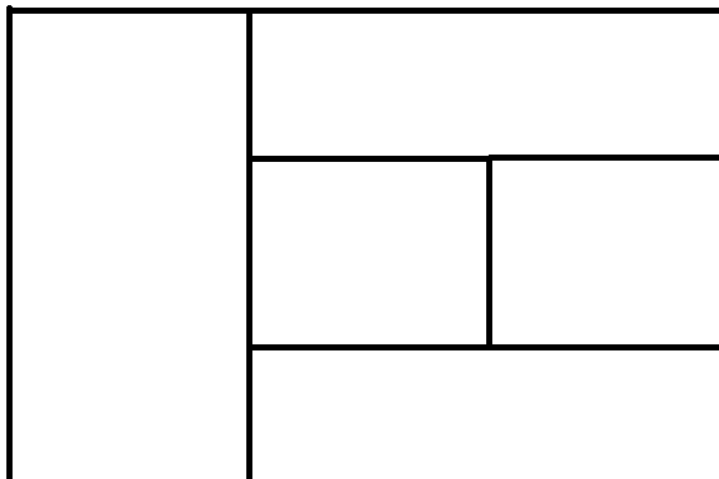
```

<HTML>
<HEAD>
<TITLE> Фреймы в HTML на Web-странице </TITLE>
</HEAD>
<FRAMESET ROWS="60,*" FRAMESPACING=0>
<FRAME NAME="title" SRC="title.htm" NORESIZE SCROLLING="no"
MARGINHEIGHT=0 MARGINTEIGHT=0
STYLE="border: red 6 dashed" >
<FRAMESET COLS="30%,*" FRAMESPACING=2 >
<FRAME NAME="left" SRC="left.htm" FRAMEBORDER=0 >
<FRAME NAME="right" SRC="prim1-1.htm" FRAMEBORDER=1
BORDERCOLOR=#0000E0 >
</FRAMESET>
</FRAMESET>
</HTML>

```

Практикалық тапсырма

1 суретте күрсетілгендей HTML бетін құрастыру. Веб-бетте басқа құжаттарға да сілтемелер болуы керек. Бір терезеде фондық сурет болуы қажет.



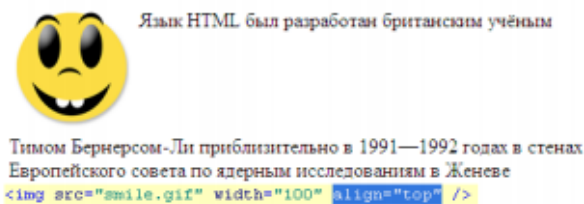
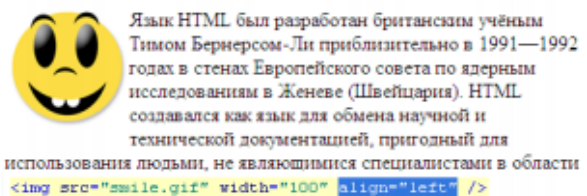
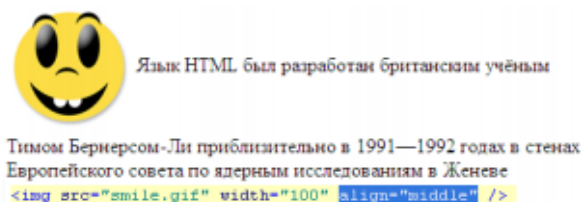
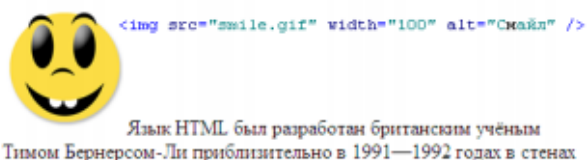
Сурет 1. Фреймдік құрылымы

Суреттер

Суреттерді HTML құжатына `...` тегтер арқылы орнатамыз. Тегті қолмен жазып, содан кейін автотолтыру код арқылы немесе Inspectro панель арқылы оның атрибуттарын көрсетуге болады.

` ... ` тегінде келесі тәлсіпаттар бар

Атрибут	Анықтамасы
src	Графикалық құжатының жолына көрсетеді. Жолдары салыстырмалы <code>photo.jpg</code> , <code>../img/photo2.gif</code> немесе абсолютті <code>http://mysite.com/pic/photo3.jpg</code> болады
lowsrc	Ең нашар сападағы графикалық құжатқа жол (және кішігірім өлшемі болатын), ол толық өлшемді сурет жүктелгенге дейін жүктеледі
alt и title	Сурет үшін баламалы мәтін, көмек ретінде немесе іздеу үшін пайдаланады
border	Сурет айналасындағы жақтаудың қалыңдығы. Жақтаудың түсі ағымдағы мәтін түсімен анықталады.
hspace	Суреттен мәтінге дейінгі көлденең шегініс
vspace	Суреттен мәтінге дейінгі тік шегініс
height	Сурет биіктігі. Егер көрсетілмесе онда суреттің негізгі өлшемі көрсетіледі
width	Суреттің ені. Егер суреттің ұзындығы мен ені сәйкес келмесе, онда сурет бұрмалайды
align	Суреттің қалай тураланатынын және айналасын мәтін қалай оратынын анықтайды



Фондық графика

Фондық бейне – бұл кішкене тікбұрышты плашкасы бар графикалық файл. Браузермен қарау барысында бұл плашка бірнеше рет қайталанып, терезені толтырады және ол терезенің пішініне тәуелсіз. Фондық графика <BODY> тәгінде беріледі.

2-жаттығу. Web-параққа фондық бейне беру.

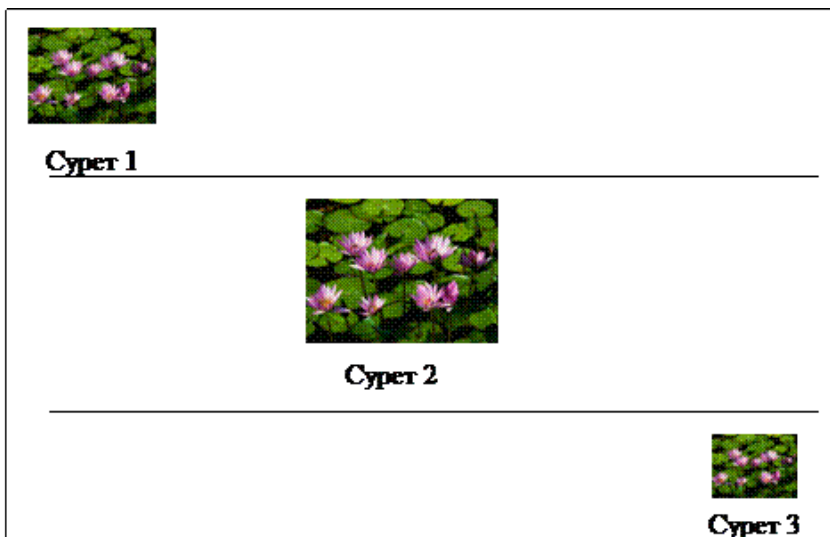
RASP.HTM файлына өзгерістер енгізіңіз:

```
<HTML>
<HEAD>
<TITLE>HTML оқу файлы</TITLE>
</HEAD>
<BODY BACKGROUND="WAGON.GIF" TEXT="#330066">
<P ALIGN=CENTER>
<FONT COLOR="#008080" SIZE="7"> <B>Сейсенбіге арналған
</B></FONT><BR>
<FONT SIZE="6"><I>сабақ кестесі</I></FONT><BR><BR>
</P>
</BODY>
</HTML>
```

WAGON.GIF файлы фондық бейнені қояды.

Тапсырма 1

Суреттер орналасқан Web-бет жаса



Тапсырма 2

Кесте жасау. Бланкта реквизиттердің орналасу сызбанұсқасы

1	2 Эмблема	3	4 Код
	5 Министерство	6 Адресация	
	7		

Тапсырма 3

Көлемі 3 беттен кем болмайтын ішкі гиперсілтемеден Web-бетті жасау қажет. Мәтіндерде түрлі түзетулерді қолдану (центрге қарай, беттің соңына және басына қарай).

Зертханалық жұмыс №10. HTML: формалар

Негізгі түсініктер:

HTML формалары сайттың ең басты маңызды элементтерінің бірі болып табылады. Формаларды пайдаланушылар енгізген деректерді серверге өңдеуге жіберу үшін пайдаланады.

Формаларға мысал ретінде сайттарға тіркелу, пікірлер жазып жіберу, кері байланыс, сұрақтарға жауап беру болмаса дауыс беру, файлдарды жүктеу сияқтыларды келтіруге болады.

Сонымен, форма HTML дің жұптасқан тегтері `<form>` `</form>` арқылы құрылады. Яғни, форманың басы `<form>`мен басталып, аяғы `</form>`мен аяқталады. Осы екі тегтің ортасына форманың қалған элементтері жайғасады.

Форма параметрлері.

Бұл <form> тегінің бірнеше параметрлері бар, солардың ішіндегі ең бастылары мыналар:

action - пайдаланушы енгізген деректерді өңдеуге жіберетін файлдың адресі.

Егер ол көрсетілмесе, осы парақша коды сол мәндермен қайта жүктеледі.

method - деректерді серверге жіберу әдістерін сипаттайды. Ол 2 әдіс - GET және POST.

GET арқылы деректер браузердің адрестік тармағында беріледі. POST арқылы деректер браузердің өз сұранысымен бірге серверге жөнелтіледі және GET ке карағанда мөлшері көп деректер жібере алады. Сондықтан мұны форумдарда, почта байланысында, дерекқор деректерін өңдеу үшін пайдаланады.

target - өңделген форманың нәтижесін көрсететін web-парақша беті браузердің осы окносына немесе жаңа окноға жүктеле ме, соны белгілейді. Оның мәндерінің ішіндегі көп кездесетіні төмендегідей:

_blank - парақшаны браузердің жаңа окносына жүктейді.

_self - парақша осы окноға жүктеледі.

4 Форма элементтері.

Бұдан әрі қарай баяндалатын форма элементтері дегеніміз деректерді енгізетін өрістер болып саналады. Олардың көбін <input> тегі арқылы сипаттайды. Ол тегтің негізгі атрибуттары мынадай болып келеді:

name - Форма элементінің аты, өріске жазылған деректі өңдейтін сервердегі скрипте анықталады.

type - Формадағы мәтін жазылған өрістің типін білдіреді.

1. Бір тармақтан құралған мәтінге арналған өріс.

Бір тармақтан тұратын өріс тек бір қатарға мәтін сөздерін енгізуге мүмкіндік береді.

Бұл өрісті жасау үшін мынадай синтаксис қолданылады:

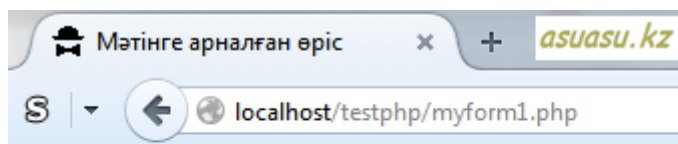
```
<input type = "text" параметрлер>
```

Оның параметрлері мына кестеде берілген:

Параметр	Сипаттамасы
size	Өрістің мөлшерін білдіреді. Ол мәтін әріптерінің санымен өлшенеді. Өріске енгізілетін таңбалардың ең үлкен мөлшерін белгілейді. Одан асып
maxlength	мәтінді әрі қарай жазу тоқтайды.
value	Өріске берілетін алғашқы мән.

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Мәтінге арналған өріс</title>
</head>
<body>
  <form action="myscript.php">
    <br>
    Аты-жөніңіз:<br>
    <input type="text" name="firstname" maxlength="35" size="40">
  </form>
</body>
</html>
```


Бұл HTML кодының нәтижесі мынадай болады:



Аты-жөніңіз:

2. Көп тармақты мәтінге арналған өріс.

Бұл форма элементі `<textarea>` `</textarea>` жұптасқан тегтерімен сипатталады.

Мұндай өрісті жасаудың синтаксисі мынадай:

```
<textarea параметрлер көп тармақты мәтін /textarea>
```

Мұндай өріске қалаған мөлшерде мәтін орналастыруға болады.

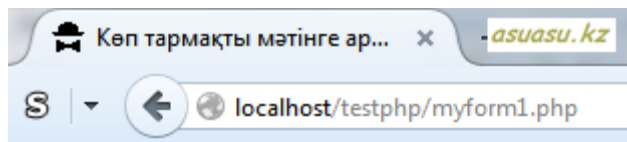
Келесі кестеде ол тегтің параметрлері берілген:

Параметр	Сипаттамасы
cols	Өрістің мөлшерін белгілейді.
rows	Өрістің қатарларының саны.
wrap	Өрістегі мәтін симағанда келесі қатарға көшіреді.
readonly	Өрістің мәтіні тек оқылу үшін арналған.
disabled	Өрісте өзгеріске ешқандай рұқсат жоқ.

Келесі HTML кодын алдыңғыдай Notepad++ редакторына жазып, php файлы ретінде сақтап болған соң алдыңғы кодтардың нәтижесін қарағандай оны да Денверден шақырайық:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Көп тармақты мәтінге арналған өріс</title>
</head>
<body>
<form action="myscript.php">
<br>
Пікір қалдыру:<br>
<textarea rows="5" cols="30" name="comments"></textarea>
</form>
</body>
</html>
```

Оның нәтижесі былай болып шықты:



Пікір қалдыру:

Осы жоғарыда сипатталған форманың 2 элементі де менің мақалаларымның пікірлер жазатын бөлігінде жасалынған.

Формалардың негізгі элементтері

Парольге арналған өріс.

Бұл кәдімгі мәтінге арналған өріс.Тек одан бір айырмашылығы - жазылған мәтін таңбалары жұлдызша арқылы белгіленеді.Яғни бұлай көрсету ол парольді ешкім көріп қалмас үшін жасалған.

Бұл өрістің жазылу синтаксисі мынадай:

`<input type="password" параметрлер>`

Оның параметрлері келесі кестеде сипатталған:

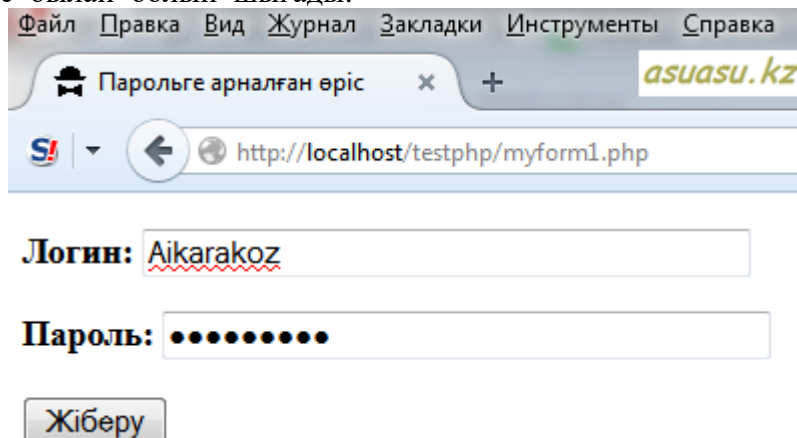
Параметр	Сипаттамасы
size	Мәтінге арналған өрістің ені,ол өріске енгізілген жұлдызшалар саны анықталады.
maxlength	Өріске енгізілетін таңбалардың ең үлкен мөлшерін белгілейді.Одан асып мәтінді әрі қарай жазу тоқтайды.
value	Өріске берілетін алғашқы мән.

Енді осы парольге арналған өріс үшін HTML кодын жазып шығайық:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Парольге арналған өріс</title>
</head>
<body>
<form action="myscript.php">
<p><b>Логин:</b> <input type="text" maxlength="30" size="40"
name="text"></p>
<p><b>Пароль:</b> <input type="password" maxlength="20" size="40"
name="pass"></p>
<p><input type="submit" value="Жіберу"></p>
</form>
```

```
</body>
</html>
```

Енді осы код жазылған файлды Денверде орындап, өрістерді толтырсақ нәтиже былай болып шығады:



Жоғарыда байқаған шығарсыздар, серверге деректерді жіберуге арналған ЖІБЕРУ деп жазылған түйме-бастырма-кнопка көрсетіліп тұр. Келесі тақырып осы туралы болады.

4. түйме-батырма-кнопка-button

Мен бұл элементтің қазақша атын түйме деп атап жүргенмін, жақында батырма деген атауды кездестіріп енді осылай атауды жөн көрдім. Себебі менің ойымша осы қазақша атау орынды секілді.

Web интерфейсінде форма жасалғанда осы элемент жиі кездеседі. Түріне карап-ақ оны үстінен шерту керек екенін сол бойда түсінуге болады.

Бұл батырма екі қызмет атқарады- деректерді серверге жіберу және экран бетіндегі парақша өрістеріндегі таңбаларды өшіру немесе тазалау.

Оларды парақша бетінде де екі әдіспен жасауға болады - `<input>` және `<button>` тегтері арқылы. Осылай сипатталған бұл батырмалар өздігімен еш әрекет жасай алмайды, сондықтан олардың әрекеті өзіміз жазатын форма өңдегіштері арқылы жүзеге асырылады.

1. Батырманы `<input>` арқылы жасаудың синтаксисі мынадай:
`<input type="button" параметрлер>`

Оның параметрлері төменгідей болып келеді:

Параметр	Сипаттамасы
name	Батырманың аты, өріске жазылған деректі өңдейтін сервердегі скрипте анықталады.
value	Батырманың атының мәні және батырманың бетінде жазылатын мәтін.

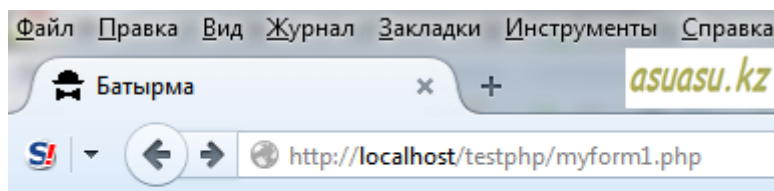
Енді осы батырмаға арналған HTML кодын жазайық және сосын оның нәтижесін браузерде көрейік.

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Батырма</title>
</head>
<body>
```

```

<form action="myscript.php">
  <p><input type="button" name="justpress" value="Мені түртіп жібер"></p>
</form>
</body>
</html>

```



Мені түртіп жібер

2. Енді батырманы `<button>` тегі арқылы жасап көрейік.

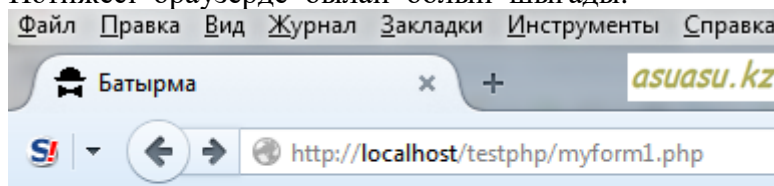
Бұл да алдыңғы `<input>` тегінің әрекетіне ұқсайды, бірақ мүмкіншіліктері кеңірек. Яғни бұл батырмада кез келген HTML тегін орналастыруға болады. Мысалы төмендегідей HTML кодын жазып, суретті мәтінмен қоса батырма ретінде пайдалануға болады.

```

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Батырма</title>
</head>
<body>
  <form action="myscript.php">
    <button> Батырма
сурет</button>
  </form>
</body>
</html>

```

Нәтижесі браузерде былай болып шығады:



Батырма сурет

5. submit батырмасы.

Бұл батырма формадағы өрістерге енгізілген деректерді `<form>` тегіндегі `action` параметрінде көрсетілген серверде орналасқан программаға жібереді.

Submit батырмасының синтаксисі `<input>` және `<button>` тегтеріне байланысты болады.

```
<input type="submit" параметрлер>
```

```
<button type="submit">Батырма бетіндегі жазу</button>
```

Батырманың осы түріне мысал жоғарыда парольге арналған өрісте көрсетілген болатын.

6. reset батырмасы.

Бұл батырма өрістерге алдыңғы енгізілген алғашқы мәндерін қайта енгізу не болмаса енгізілген деректерден тазарту үшін қолданылады.

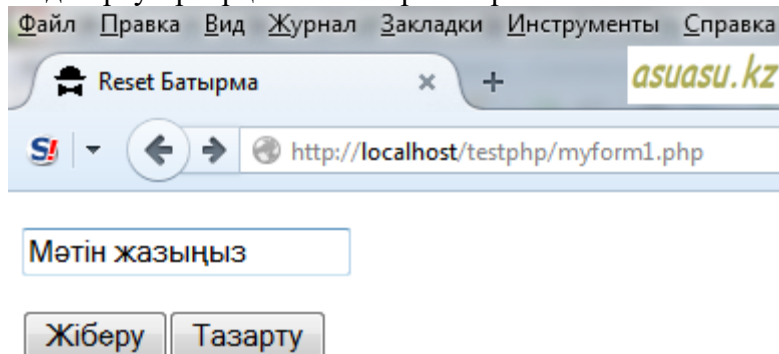
Синтаксисі жоғарыда сипатталған батырмалардағыдай:

```
<input type="reset"                                параметрлер>
<button type="reset">Батырма бетіндегі жазу</button>
```

Төмендегі HTML коды арқылы жазылған скрипте формадағы бір мәтінді өріс алғашқы мәнімен жазылған. Егер сол мәнді өз қалауыңызша өзгертіп болған соң, reset батырмасын түртсеңіз ол өріс өзінің алғашқы мәнін қайта иемденеді.

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Reset Батырма</title>
</head>
<body>
  <form action="myscript.php">
    <p><input type="text" value="Мәтін жазыңыз"></p>
    <p><input type="submit" value="Жіберу"><input type="reset"
value="Тазарту"></p>
  </form>
</body>
</html>
```

Енді браузер арқылы тексеріп көрейік.



7. ауыстырғыштар.

Формада ұсынылған бірнеше мәндердің ішінен тек біреуін таңдау қажет болған жағдайда осы ауыстырғыштарды қолданады.

Оның синтаксисі былай беріледі:

```
<input type="radio" name="аты" параметрлер >
```

Ал параметрлер төмендегі кестеде беріліп тұр:

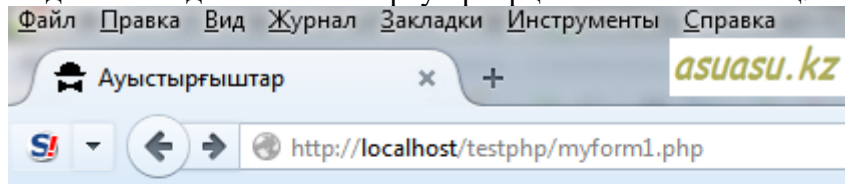
Параметр	Сипаттамасы
name	Өрісті айқындайтын ауыстырғыштар жиынтығының аты, демек жиынтық элементтерінің бәріне тек қана бір атау беріледі.
value	Элементтердің әрқайсысының аты қайталанбайтын бірегей болу керек. Бұл мән сервердегі форманы өңдегіш скриптке жіберіледі.

Алдын ала ауыстырғышты осы checked арқылы белгілеп өрістің мәні checked ретінде ұсыну.

Енді осы ауыстырғыштардың қалай жұмыс істейтінін көру үшін HTML кодын жазып шығайық.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Ауыстырғыштар</title>
</head>
<body>
<form action="myscript.php">
<p><b>Сізді қоғамда болып жатқан өзгерістер мазалай ма?</b></p>
<p><input name="socty" type="radio" value="ysocty" checked="checked">
Ия,мазалайды<br>
<input name="socty" type="radio" value="nsocty"> Жоқ,мазаламайды<br>
<input name="socty" type="radio" value="isocty"> Маған бәрі-бір</p>
<p><input type="submit" value="ОК"></p>
</form>
</body>
</html>
```

Енді осы код нәтижесін браузер арқылы тамашалайық.



Сізді қоғамда болып жатқан өзгерістер мазалай ма?

- Ия,мазалайды
- Жоқ,мазаламайды
- Маған бәрі-бір

OK

8. жалаушалар.

Формада ұсынылған бірнеше мәндердің ішінен тек біреу емес, керісінше бірнешеуін таңдау қажет болған жағдайда жалаушалар қолданылады.

Оның синтаксисі былай беріледі:

```
<input type="checkbox" name="аты" параметрлер >
```

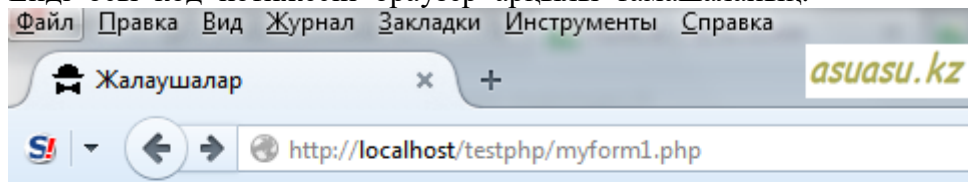
Ал параметрлер төмендегі кестеде беріліп тұр:

Параметр	Сипаттамасы
name	Жалаушалардың атаулары, олар қайталанбайтын бірегей болуы керек.
value	Бұл мән сервердегі форманы өңдегіш скриптке жіберіледі.
checked	Алдын ала жалаушаны осы checked арқылы белгілеп өрістің мәні ретінде ұсыну.

Енді осы жалаушалар үшін HTML кодын жазып шығайық.

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Жалаушалар</title>
  </head>
  <body>
    <form action="myscript.php">
      <p><b>Транспорттың қай түрлерімен саяхат жасағанды ұнатасыз?</b></p>
      <p><input type="checkbox" name="opt1" value="o1" checked>Поезд<br>
      <input type="checkbox" name="opt2" value="o2"> Самолет<br>
      <input type="checkbox" name="opt3" value="o3"> Автомобиль<br>
      <input type="checkbox" name="opt4" value="o4"> Параход</p>
      <p><input type="submit" value="OK"></p>
    </form>
  </body>
</html>
```

Енді осы код нәтижесін браузер арқылы тамашалайық.



Транспорттың қай түрлерімен саяхат жасағанды ұнатасыз?

- Поезд
- Самолет
- Автомобиль
- Параход

OK

9. Жайылып ашылатын және жазылмалы тізімдерге арналған өріс.

Форма бетінде тізімдер әртүрлі болып келеді. Олардың жазылу синтаксисі мынадай:

```
<select параметрлер >
  <option параметрлер>Мән1</option>
  <option>Мән2</option>
</select>
```

Солардың бірі осы жайылма немесе жайылып ашылатын тізім. Бұл аталған тізім тек бір қатармен көрсетіледі. Сондықтан тізімді толықтай көру үшін оны үстінен түрту керек болады және сол жайылып ашылған тізімнен тек бір ғана мән таңдалады.

Тег	Параметр	Сипаттамасы
select	name	Элементтің қайталанбайтын бірегей аты, форманы өңдейтін

сервердегі скрипте пайдаланылады

Сервердегі скриптке жіберілетін осы таңдалған тармақтағы

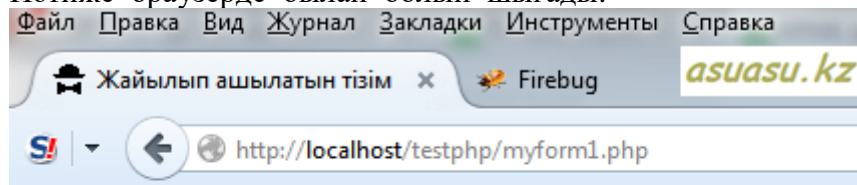
option value мән.

Бұл тізім параметрлерінің сипаттамасы мынадай:

Енді осы тізімнің түрін HTML кодымен өрнектеп шығайық:

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Жайылып ашылатын тізім</title>
</head>
<body>
  <form action="myscript.php">
<p><b>Қазақстанмен шекаралас мемлекеттер саны қанша?</b></p>
  <select name="shekara">
    <option value="shek1">3</option>
    <option value="shek2">4</option>
    <option value="shek3">5</option>
    <option value="shek4">6</option>
  </select>
  <p><input type="submit" value="Жіберу"></p>
</form>
</body>
</html>
```

Нәтиже браузерде былай болып шығады:



Қазақстанмен шекаралас мемлекеттер саны қанша?

5 ▾

Жіберу

Ал енді **жазылмалы тізімге** келетін болсақ, оның параметрлері сәл көбірек.

Тег	Параметр	Сипаттамасы
select	multiple	Бұл параметр тізімде бірден көп,яғни бірнеше мәндер таңдауға болатынын көрсетеді.Ал сол бірнеше мәндерді белгілеу үшін CTRL және SHIFT клавиштерін курсормен бірге қолдану керек.
	name	Элементтің қайталанбайтын бірегей аты,форманы өңдейтін сервердегі скрипте қолданылады
	size	Бұл тізімнің биіктігін көрсетеді.Егер бұл size мәні тізімдегі мәндердің санынан кіші болса, онда ол тізімді айналдыруға тура

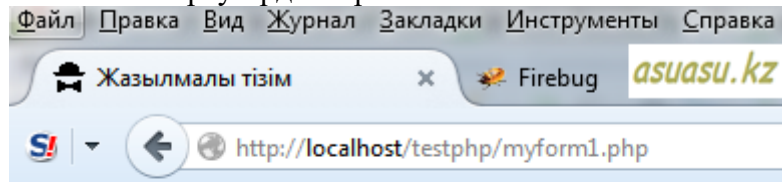
келеді

option selected сіз оны өзгерте аласыз.
value Сервердегі скриптке жіберілетін таңдалған пунктің мәні.

Енді осы аталған тізім үшін HTML кодын жазып шығайық:

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Жазылмалы тізім</title>
  </head>
  <body>
    <form action="myscript.php">
      <p><b>Қазақстанда экологиясы таза қалалар қайсы?</b></p>
      <select name="city" size="3" multiple>
        <option value="city1">Алматы</option>
        <option selected value="city2">Астана</option>
        <option value="city3">Атырау</option>
        <option value="city4">Ақтөбе</option>
        <option value="city4">Ақтау</option>
      </select>
      <p><input type="submit" value="Жіберу"></p>
    </form>
  </body>
</html>
```

Нәтижесін браузерде көрейік:



Қазақстанда экологиясы таза қалалар қайсы?

Алматы ▲
Астана ■
Атырау ▼

Жіберу

Зертханалық жұмыс №11. CSS-пен танысу(стильдердің каскадты кестелері)

CSS селекторлары бірнеше түрге бөлінеді, олардың санының нақты қанша екенін айтып қажеті де жоқ. Өйткені CSS әр жаңа нұсқасымен бірге жаңа түрлері пайда бола береді, сондықтан мен тек қазіргі қолданыста жүргендерін және көп қолданылатын түрлерін ғана айтып өтпекпін.

Осы жерде естеріңізге сала кетейін, бұл селекторларды біз бұрын айтып кеткенімдей бөлек CSS файлда сипаттап жазамыз да, HTML кодының <head></head> бөлігінде link арқылы жүктеу үшін жазып кетеміз.

Селекторлардың мынадай түрлері бар:

HTML тегтері селекторлар ретінде қолданылады.

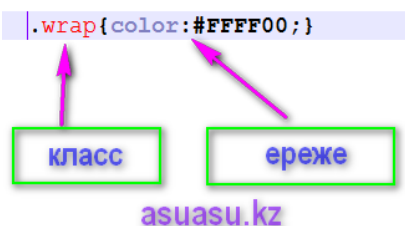
Мен осының алдындағы сабағымда соның бір түрі "P" тегінің селектор ретінде абзац фонын әр түрлі түске бояғанын практика жүзінде көрсеттім.



asuasu.kz

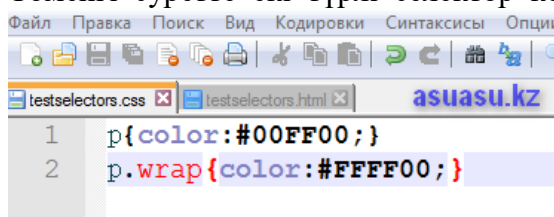
Яғни кез келген HTML тегін селектор ретінде қолдана аламыз. Бірақ, мұның кемшілігі - ереже тек бір тегке байланып қалады. Мысалы, мен басқа бір абзацтың әріптерін қызылдан басқа түске бояғым келуі мүмкін ғой. Егер сондай қажеттілік туындай қалса көмекке келесі "класс" деп аталатын селекторлар келе алады.

5 2. Класс селекторлары былай жазылады:



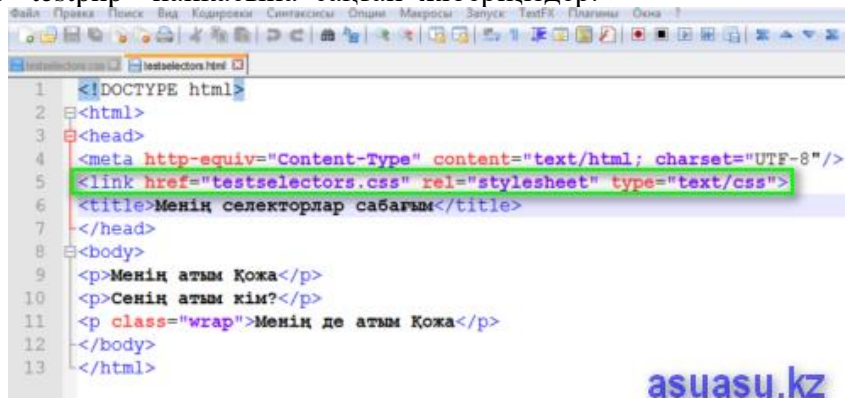
asuasu.kz

Төменгі суретте екі түрлі селектор көсетіліп тұр.



Көріп отырғаныңыздай P тегі екі түрдегі ережемен беріліп тұр. Бұлар не деп тұр екен, талдау жасайық.

Бірінші ереже, яғни бұл алдында айтып өткен тег селекторы ережесі бойынша, барлық абзацтарымыз жасыл түспен жазылуы керек. Бірақ, олардың ішінен қалаған абзацтарды өзгерте аламыз. Қалай дейсіздер ғой, жауабын табу үшін жоғарыдағы CSS стильдік таблицаны "testselectors.css" деп, ал төмендегі HTML кодын "testselectors.html" деп өзіміз жақсы білетін Notepad редакторына көшіріп жазып "testphp" папкасына сақтап жіберіңіздер.

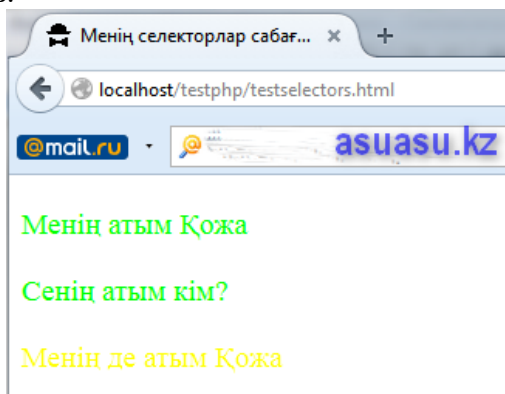


asuasu.kz

Енді осыны локальдық сервер Денверде ашып қарайық. Ол үшін әр уақытта оған бірінші кірерде ярлық "Start Denwer"ді түртеміз. Сосын кез келген браузердің

адрес теретін жолында "localhost/testphp/testselectors.html" деп шақырамыз.Нәтиже төмендегідей болды.

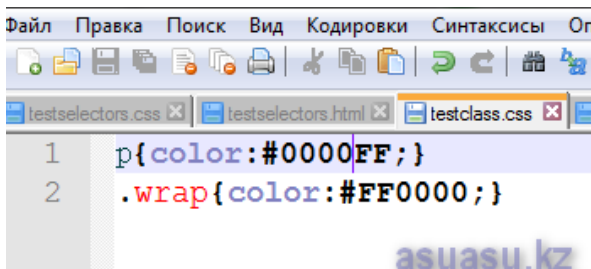
Ол html файлды Денвер арқылы емес, жай түртіп жіберіп те нәтижесін көруге болады.Бірақ, сіз егер PHP скрипт тілін үйренгіңіз келсе, онда Денверде жұмыс істеу сізге дағдыға айналуы керек.Сондықтан, тест файлдарды осы локальдық серверде шақырғанымыз дұрыс деп санаймын.Егер Денверді пайдаланғыңыз келмесе, онда ол файлды папкасында екі рет түртіп жіберіп қарай беріңіз.



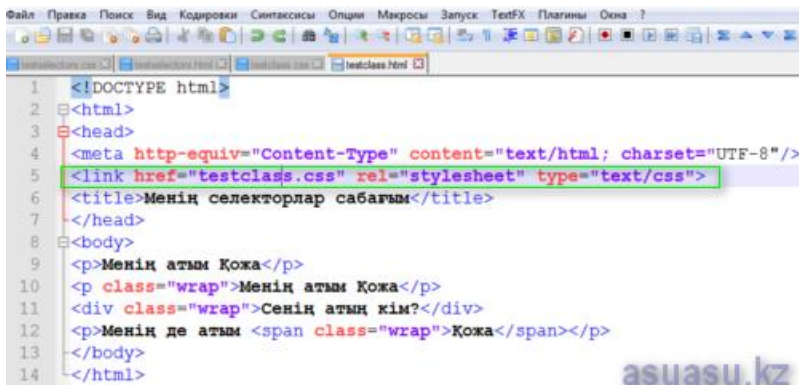
Яғни бұл жерде өзіңіз көріп отырғандай бірінші ереже бойынша барлық абзацтар жасылға боялды, бірақ сол абзацтардың кез келгенін белгілі бір өзгеше ережемен сипаттасақ, ол "p"тегімен байланысын үзіп басқа ережеге бағынады.Яғни біздің мысалда ол барлық берілген жасыл түстен басқа сары түске ие болды.Яғни "p.wrap"-класс селекторының дәрежесі "p" - тег селекторы дәрежесінен жоғары болып тұр.

Ал егер бір ережемен бірнеше тегтерді бағындыру керек болса, онда жоғарыда көрсетілген екінші ереже сықылды бірақ HTML тегі көрсетілмеген класс селекторын пайдалану керек.

Олай болса Notepadке төмендегідей қылып CSS файлын жазайық:



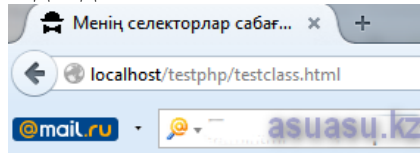
Сосын осы каскадты стильдік кестелер ережесі бекітілетін HTML кодын жазайық:



Бұл файлдың атын "testclass.html" деп атап "testphp" папкасына салып жіберелік.

Өздеріңіз көріп отырғандай, бірінші "p" абзацы көк түс кодымен #0000FF , ал екінші "p" абзацы, "div" блог тегі және "span" тегімен қоршалған Қожа сөзі қызыл түс кодымен #FF0000 ережеленген.

Енді осы жазған HTML коды жазылған файлымызды браузерде localhost/testphp/testclass.html деп шақырайық. Менде нәтижесі төмендегідей болды және сізде де солай болғанына сенімдімін.



Менің атым Қожа

Менің атым Қожа

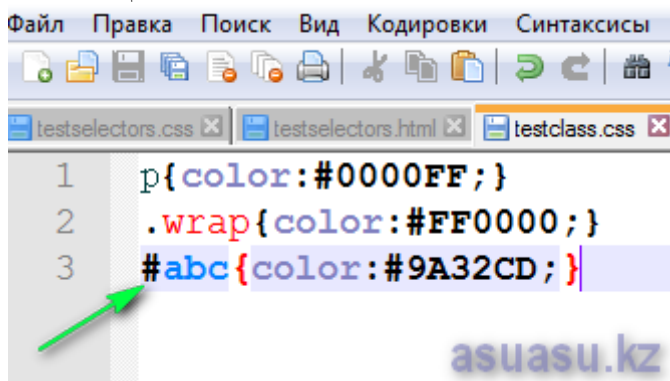
Сенің атың кім?

Менің де атым Қожа

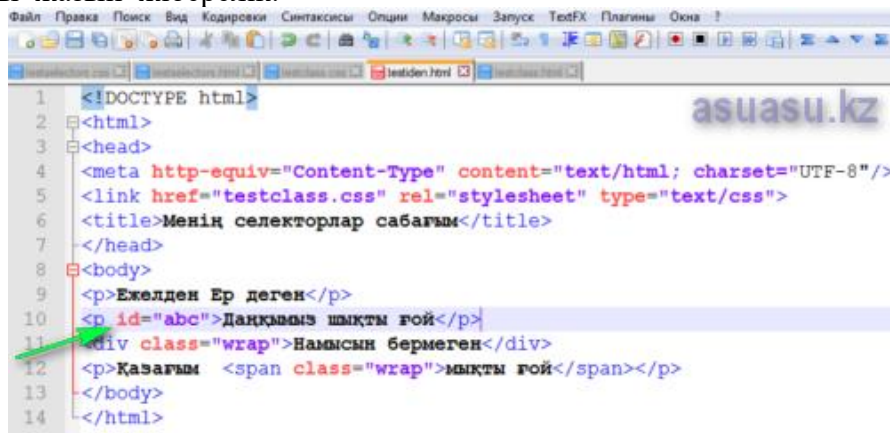
Үшінші селекторлар түрі - бірегей (бірден-бір) идентификатор деп аталады.

Олардың класс селекторларынан айырмашылығы нүктенің орныны # (хеш) қолданылады және бірегей идентификатор парақша бетінде тек бір тегпен бірге бір-ақ рет жазылады.

Тағы да Notepad те алдыңғы жазып кеткен "testclass.css" файлын ашып ішіне #abc деген бірегей идентификаторын күлгін түс коды #9A32CD ережесімен қосып жазайық.



Енді осы стильдік кестелер файлын төмендегі көрсетілген HTML кодына link арқылы жазып жіберелік.



Зертханалық жұмыс №12.CSS: тізімдермен жұмыс істеу

HTML сабақтарында мен таңбаланған және нөмірленген тізімдер туралы баяндап өткенмін.

Бірақ, олардың кейбірінің орналасуы код жазу барысында тізімдерді өзіміз қалаған түрге келтіруге қиындық жасайды.Мысалы, мен меню жасағанда ді қолданамын, бірақ мен менюде ешқандай таңбаның да, нөмірдің де болғанын қаламадым.Ал HTML дегі тізім үнсіздік бойынша әр меню тақырыбының алдына дөңгелек немесе араб сандарын жазады.Одан қалай құтылуға болады.Міне, осы кезде CSS көмекке келеді.

Не болмаса, мысалы таңбаның не нөмірлердің орнына сурет болғанын қалайтындар болса ше? Міне, осындай жағдай да CSS тің көмегімен шешіледі. |

Тізім стилі оның негізгі қасиеті list-style арқылы сипатталады, ал ол өз алдына 3 қасиеттен құралады:

list-style:list-style-type || list-style-position || list-style-image
list-style-type таңбаның түрін сипаттайды
list-style-position таңбаның орнын белгілейді
list-style-image таңба ретінде қолданылатын суретті сипаттайды

Тізім стилінің бұл 3 қасиеттері бірдей қолданылуы міндетті емес, тек қолданылатын қасиеттер қанша болса да осы көрсетілген ретте жазылуы тиіс және аралары пробел арқылы бөлінуі керек.

Төменгі кестеде осы стильдің 3 қасиеттері сипатталып беріліп тұр:

Қасиеті:Мәні	Сипаттамалары
list-style-type:disc	Дөңгелек(үнсіздікбойынша)
list-style-type:circle	шеңбер
list-style-type:square	квадрат
list-style-type:decimal	араб саны
list-style-type:upper-roman	жоғарғы регистрдегі рим саны
list-style-type:lower-roman	төменгі регистрдегі рим саны
list-style-type:upper-alpha	латынның бас әріптері
list-style-type:lower-alpha	латынның кіші әріптері
list-style-type:none	ешқандай таңба жазылмайды
list-style-position:outside	таңба тармақтың сыртында орналасады (үнсіздік бойынша)
list-style-position:inside	таңба тармақтың ішінде орналасады
list-style-image:none	ешқандай сурет болмайды
list-style-image: url	сурет қолданылады

Енді мен HTML кодында CSS ті белгілейтін тәсілдің екіншісін пайдаланып, HTML кодын екі түрде төмендегідей жазып шығайын.

Біріншісінде, тізім стилі ондағы таңбаның қасиеттерін бір ғана әдіспен сипаттауға мүмкіндік береді, соны қолданайын.

Ал екінші код сол стильдің қасиеттерін бөлек-бөлек сипаттайды.

Екінші қасиет бойынша жазылған "таңба тармақтың сыртында не ішінде орналасады" дегенді түсіну үшін, тармақтың шекарасын CSS арқылы сипаттап тізім стилінің жанына жазып қоялық:

1.Таңба тармақтың ішінде орналасады

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

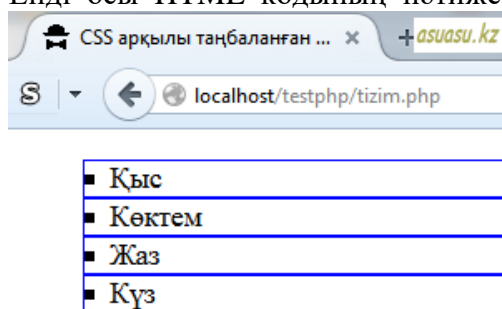
```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```

<title>CSS арқылы таңбаланған тізім</title>
<style type="text/css">
  ul {
    list-style: square inside; /* Таңба квадрат түрінде тармақтың ішінде
орналасады*/
  }
  li { border: 1px solid blue; /* Тармақтың шекарасын ені 1px көк түспен
белгілейік*/ }
</style>
</head>
<body>
  <ul>
    <li>Қыс</li>
    <li>Көктем</li>
    <li>Жаз</li>
    <li>Күз</li>
  </ul>
</body>
</html>

```

Енді осы HTML кодының нәтижесін Денверде қаралық:



2.Таңба тармақтың сыртында орналасады

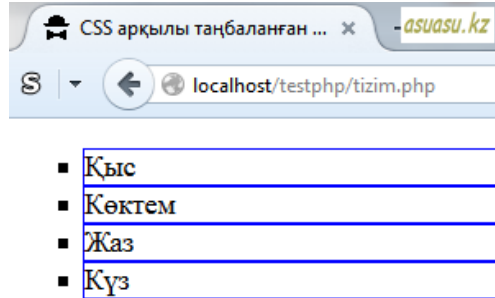
```

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>CSS арқылы таңбаланған тізім</title>
  <style type="text/css">
    ul {
      list-style: square; /* Таңба квадрат түрінде белгіленеді */
      list-style-position: outside; /* Осы таңба тармақтың сыртына орналасады */
    }
    li { border: 1px solid blue; /* Тармақтың шекарасын ені 1px көк түспен
белгілейік*/}
  </style>
</head>
<body>
  <ul>
    <li>Қыс</li>
    <li>Көктем</li>
    <li>Жаз</li>
    <li>Күз</li>
  </ul>

```

```
</ul>
</body>
</html>
```

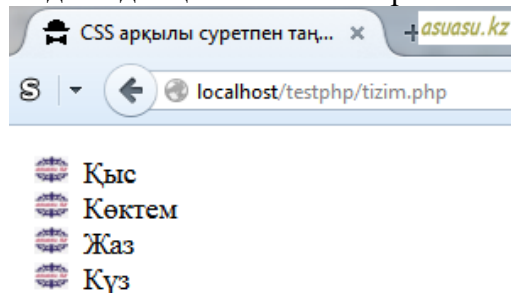
Енді осы HTML кодымыздың нәтижесін Денвер арқылы тамашалайық.



Енді CSS тізім стилінің үшінші қасиетін талдайық. Ол үшін HTML кодын жазып, таңба орнына өзіміз қалаған суретті орнаталық. Демек, бізге өте кішкене сурет керек. Мен ол үшін өз сайтымдағы қазақша өрнектелген суретті алып, оның мөлшерін 20x16 ге кішірейтіп алдым. Ол суретті сіздер де қаласаңыздар өздеріңізге көшіріп алыңыздар.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>CSS арқылы суретпен таңбаланған тізім</title>
<style type="text/css">
  UL {
    list-style-image: url(images/cssornament.png); /* Таңба сурет түрінде
белгіленеді */
  }
</style>
</head>
<body>
<ul>
  <li>Қыс</li>
  <li>Көктем</li>
  <li>Жаз</li>
  <li>Күз</li>
</ul>
</body>
</html>
```

Енді кодтың нәтижесін көрелік:



Зертханалық жұмыс №13.CSS: фонмен жұмыс істеу

Шын мәнінде әдемі, жарқын және бірегей сайттарды жасау үшін, Сіз бір немесе бірнеше фондық суреттерді көрсетуге мүмкіндік беретін background-image сияқты қасиеттерімен және артқы фонмен жұмыс істеу әдістерін оқып, танысуыңыз қажет.

Элемент фоны – ішкі шегіністердің (padding) және шекаралардың (border) мәндерін қоса алғанда, бірақ сыртқы шегіністердің мәнін қоса алғанда емес, элементтің жалпы өлшемі-margin қасиеті.

Қазіргі уақытта браузерлер үш графикалық форматпен жұмыс істейді:

- GIF (ағыл. Graphics Interchange Format-сурет алмасу пішімі).
- * JPEG (ағылш. • Joint Photographic Experts Group-әзірлеуші ұйымның атауы).
- PNG (ағылш. Portable network graphics-графикалық ақпаратты сақтаудың растрлық форматы).

Суретті артқы Фон ретінде орнату үшін background-image қасиетін пайдаланып, салыстырмалы және абсолюттік болуы мүмкін сурет файлына жолды көрсету керек:

```
селектор {  
background-image: url(images/main_bg.png);  
}
```

Сіздің назарыңызды фондық кескінді қолданғанда фондық кескінді пайдалану болжанатын беттің HTML емес, мәнерлер кестесі файлының мекенжайына қатысты көрсету қажет екеніне назар аударамын.

<Body>элементі үшін 100 пиксель өлшемі 100 түрлі-түсті үш квадратты құрайтын артқы фонды орнататын мысал қарастырайық:

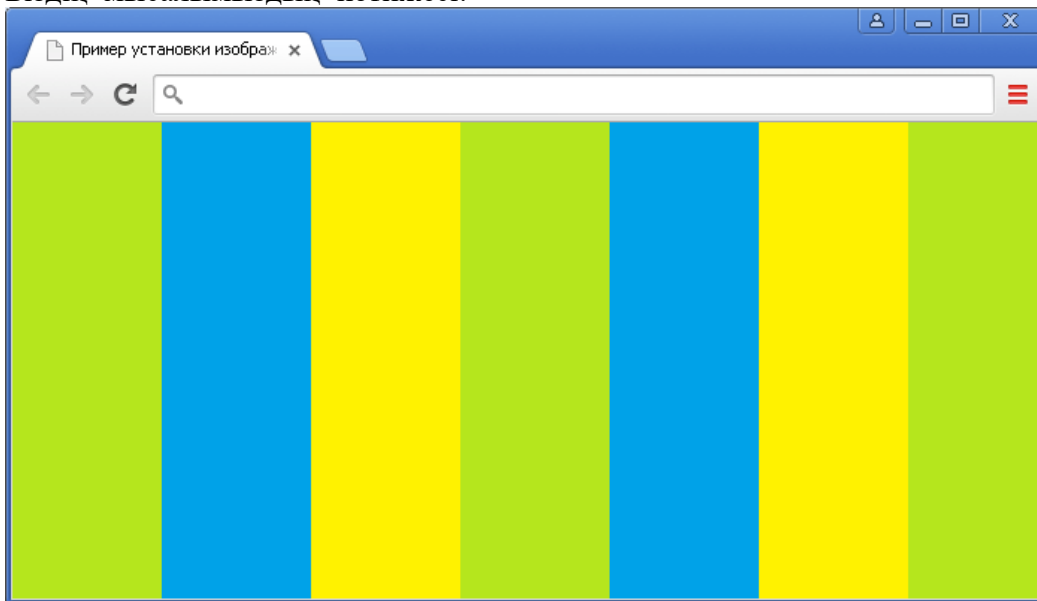


```
<!DOCTYPE html>  
<html>  
<head>  
<title>Пример установки изображения в качестве фона</title>  
<style>  
body {  
background-image: url("main_bg.png"); /* указываем путь к файлу  
изображения, которое будет использоваться как задний фон */  
background-color: white; /* задаем задний фон для элемента */  
}  
</style>  
</head>  
<body>  
</body>  
</html>
```

Әдепкі бойынша, фондық сурет элементтің жоғарғы сол жақ бұрышында орналасады және тігінен де, көлденеңінен де қайталанады, біздің мысалда бұл элемент <body> фондық бейнемен толығымен толтырылғанына әкелді.

Егер сурет қандай да бір себептермен жүктелмесе, онда сіз белгілеген түс қолданылады.

Біздің мысалымыздың нәтижесі:



Сур. Суретті Фон ретінде орнату үлгісі.

Фондық кескінді қайталауды басқару

Мысал ретінде, фондық сурет элементтің жоғарғы сол жақ бұрышында әдепкі бойынша орналастырылады және тігінен және көлденеңінен қайталанады. Бұл алдын ала анықталған мәндерді Өзгертуді үйренейік және сурет қайталауын қалай басқаруға болатынын қарастырайық, ал бұл CSS background-repeat қасиетіне көмектеседі.

Бұл сипат келесі қолжетімді мәнге ие:

Значение	Описание
repeat	Фоновое изображение будет повторяться как по вертикали, так и по горизонтали. Это значение по умолчанию.
repeat-x	Фоновое изображение будет повторяться по горизонтали (по оси x).
repeat-y	Фоновое изображение будет повторяться по вертикали (по оси y).
no-repeat	Фоновое изображение не будет повторяться.

Келесі мысал үшін 10 пиксельдегі екі түрлі түсті квадрат бар артқы фонды қолданамыз.

```
<!DOCTYPE html>
<html>
<head>
<title>Пример управления повтором фонового изображения</title>
<style>
body {
background-image: url("main_bg.png"); /* указываем путь к файлу
изображения, которое будет использоваться как задний фон */
}
div {
display: inline-block; /* устанавливаем, что элементы становятся блочно-
строчными (чтобы выстроились в линейку) */
background-image: url("small_bg.png"); /* указываем путь к файлу
изображения, которое будет использоваться как задний фон */
width: 200px; /* устанавливаем ширину элемента */
height: 200px; /* устанавливаем высоту элемента */
border: 1px solid; /* устанавливаем сплошную границу размером 1 пиксель */
margin-right: 10px; /* устанавливаем внешние отступы справа */
text-align: center; /* выравниваем текст по центру */
line-height: 200px; /* задаем высоту строки */
background-color: azure; /* указываем цвет заднего фона */
}
.noRepeat {
background-repeat: no-repeat; /* фоновое изображение не будет повторяться. */
}
.repeatX {
background-repeat: repeat-x; /* фоновое изображение будет повторяться по
горизонтали */
}
.repeatY {
background-repeat: repeat-y; /* фоновое изображение будет повторяться по
вертикали */
}
</style>
</head>
<body>
<h1>Значение repeat для body (по умолчанию)</h1>
<div class = "noRepeat">no-repeat</div>
<div class = "repeatX">repeat-x</div>
<div class = "repeatY">repeat-y</div>
</body>
</html>
```

алдыңғы мысалмен <body> үшін біз көлденең және тігінен қайталанатын артқы фон орнатдық. Сонымен қатар, біз background-repeat қасиеттері үшін әр түрлі мәндерді пайдалана отырып, фондық кескінді қалай қайталауға болатынын анықтайтын үш класты блоктарды құрдық:

* Бірінші блок үшін сурет қайталанбайды (no-repeat мәні), бұл мән көбінесе жұмыста қолданылады.

* Екінші блок repeat-x мәнін алды, ол фондық сурет көлденеңінен қайталанатынын анықтайды •

* Ал, үшінші блокта фон тігінен қайталанатын (repeat-y мәні).

Біздің мысалымыздың нәтижесі:

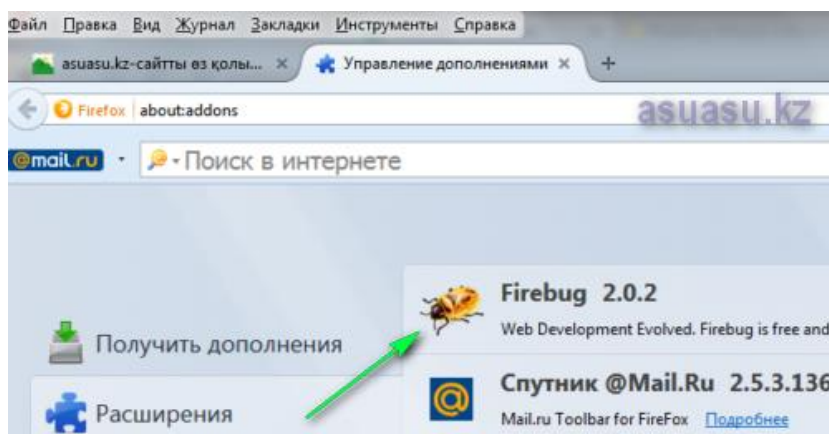


Сур. Фондық кескінді қайталауды басқару үлгісі.

Зертханалық жұмыс №14. CSS: элементтердің орналасуы және көрінуі

Бүгінгі сабақ "padding", "margin" және "border" деген CSS ережелеріне арналады.

Ал бұл сабақты түсіндіру үшін мен сіздерге көптен бері айтқым келіп жүрген Mozilla Firefox браузерінің қосымша кіріспесі Firebug туралы қысқаша айтып кеткім келіп отыр, себебі оны пайдаланбайтын web-сайт жасаушылар жоқ сияқты. Онсыз сайт жасау мүмкін емес, себебі ол сайт жасаудағы ең басты және ең керекті құрал. Оны сіз Мозилланың жоғарыдағы менюінің ішінен Инструменты->Дополнения деген парақша бетіне төменде көрсеткенімдей кіріп, қоңыздың суретімен белгіленген Firebug расширениясын өз компьютеріңізге тегін жүктей аласыз.



Мен өзім 3 браузерді пайдаланамын: негізінде Опера және кейде Google Chrome браузерлерін ғаламторға шығу үшін, ал Мозилла Firefox браузерін сайт парақшаларын жасау үшін.

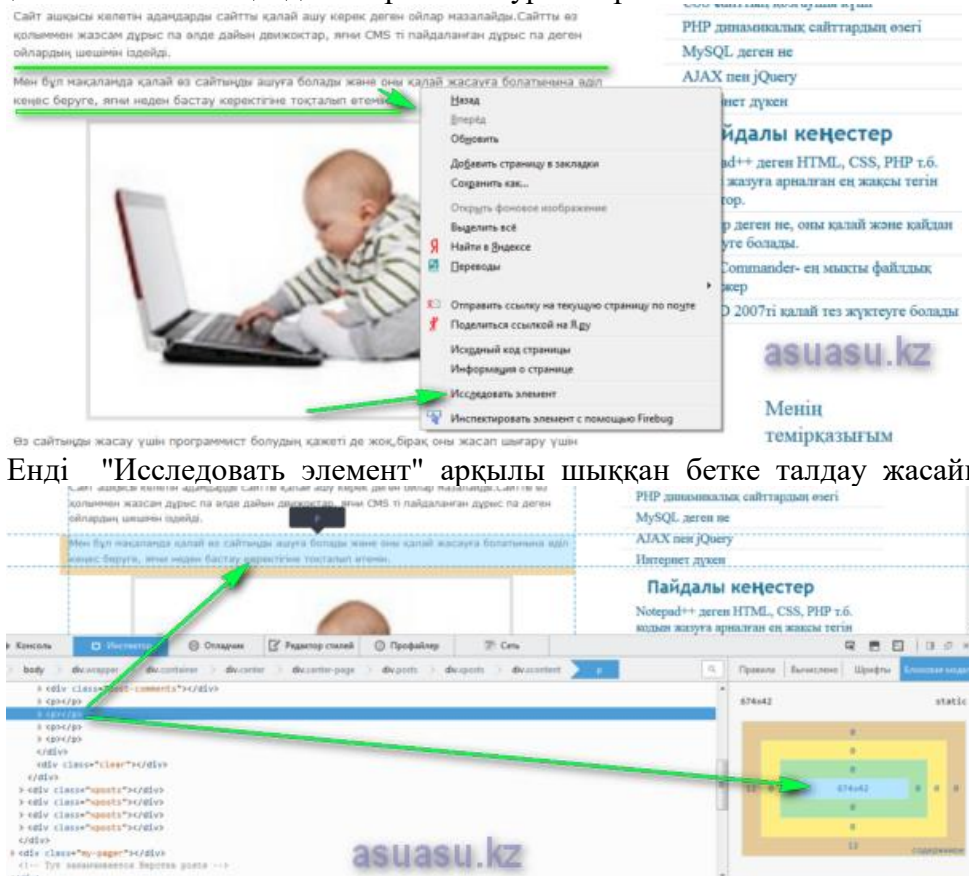
Егер қаласаңыз, оларды өз атаулары бойынша көрсетілген сілтемелер арқылы компьютеріңізге тегін жүктей аласыз.

Әзірге осы Firebug-тің тек бір қасиетін назарларыңызға ұсынбақпын, себебі ол бүгінгі тақырыпты түсінуге көп көмек бере алады.

Яғни, кез келген сайт бетін ашып, оның кодын болмаса белгілі бір тегінің сипаттамасын көргіңіз келсе, онда сол жерде мышканың оң жақ кнопкасын басыңыз керек. Осылай сіз сайттың көпшілікке көрінетін кодын тамашалай аласыз.

Оның тек осы қасиетін ғана басқа да браузерлер көрсете алады, бірақ программаны дұрыстауды (отладка) олар Firebug сияқты жақсы атқара алмайды.

Мен назарларыңызға өз парақшамдағы екінші абзацтың үстінен мышканың оң жақ кнопкасын басқандағы түсілген суретті ұсынамын:



Енді "Исследовать элемент" арқылы шыққан бетке талдау жасайық:

Бұл бетте алдарыңыздағы кодтың бір бөлігі, оның ішінде көк жолмен боялып тұрған үстінен мышкамен түртілген p абзацы. Жасыл стрелкалармен көрсетіп отырғанымның жоғарыға бағытталғаны сол абзацты көрсетіп, ал төмен қараған стрелка тіктөртбұрыштың ішіндегі сол абзацтың парақша бетіндегі орнын көрсетіп тұр.

Міне, бүгінгі сабақ осы тіктөртбұрышты талдау арқылы бүгінгі тақырып иелері "padding", "margin" және "border" сипаттамаларын түсіну болмақшы.

Яғни парақша бетіндегі кез келген HTML тегінің (тармақ тегтері және блок тегтері) осындай сипаттағы тіктөртбұрышы, демек ол тегтің "padding", "margin" және "border" сипаттамалары болады деген тұжырым туады.

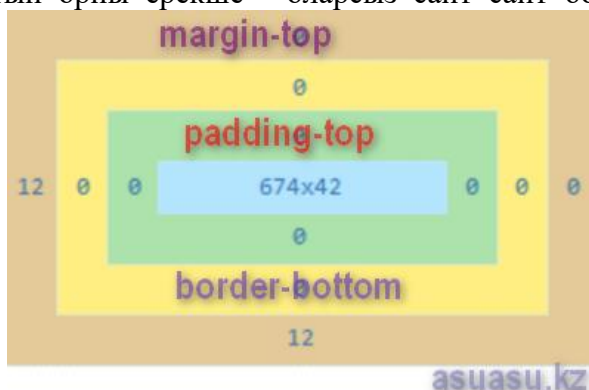
Сонымен, суреттегі тіктөртбұрыштың ең ортасындағы көк түсті бөлігі біздің p абзацымыз, ал оның сыртындағы жасыл түспен көмкерілген аймақ- "padding" (ішкі бос кеңістік) деп аталады.

Ал "padding" сыртын айнала жағалап орналасқан сары түсті аймақ - "border" (шекара) деп аталады.

Сыртқы қоңыр түсті "border"-ді қоршаған аймақ "margin" (сыртқы бос кеңістік) деп аталады.

Мысалы, парақша бетіне мәтін жазғанда болмаса сурет орналастырғанда, оны "div" не "p" контейнерлеріне салғанда ол мәтін не суреттің шекарасы-"border" пайда болады. Оны сіз қалауыңызша оған белгілі бір мән беру арқылы көрсете аласыз, ал егер оның мәнін көрсетпесеңіз ол шекара мүлде көрінбейді.

Ал "padding" мен "margin" сипаттамаларының анықтамасы басқаша, олар ана бір фильмдегі көрінбейтін адам сияқты ешқашан парақша бетінде көрінбейді. Бірақ, парақшаны беттеуде алатын орны ерекше - оларсыз сайт сайт бола алмайды.



Яғни, бұл суреттен өздеріңіз көріп отырғандай "padding-top"-HTML тегінен (мысалы, мәтіннен не суреттен) шекараға (border) дейінгі аралықты, "margin-top"-шекарадан жоғарғы келесі HTML тегінің "margin-іне" дейінгі аралықты, "border-top" - сол шекараның жоғарыдағы енін көрсететін мөлшемдер.

Ал "padding-right"- оң жақтағы "border-right"-қа дейінгі аралықты, "margin-right" - "border-right" тан келесі көрші HTML тегінің "margin"-іне дейінгі аралықты, "border-right" - оң жақтағы шекараның енін көрсетіп тұр.

Қалғандары осылар тектес сипатталады.

Енді W3C стандарттары бекітілген сайттан олардың CSS ережелерін көшіріп алайық:

1. "padding": "padding-top"- "padding-right" "padding-bottom" "padding-left"

'padding-top' 'padding-right' 'padding-bottom' 'padding-left'	<padding-width> inherit	0
--	---------------------------	---

The properties defined in this section refer to the <padding-width>
asuasus.kz

<length> **сандық өлшем**

Specifies a fixed width.

<percentage> **пайыздық өлшем**

2. "margin":

'margin-right' 'margin-left'	<margin-width> inherit	0
'margin-top' 'margin-bottom'	<margin-width> inherit	0

The properties defined in this section refer to the `<margin-width>` v
asuasu.kz

`<length>` сандық өлшем

Specifies a fixed width.

`<percentage>` пайыздық өлшем

The percentage is calculated with respect to the *width* of the `parent` element. If the containing block's width depends on the `width` of the `parent` element, the percentage is calculated with respect to the `width` of the `parent` element *bottom*' as well. If the containing block's width depends on the `width` of the `parent` element, the percentage is calculated with respect to the `width` of the `parent` element.

`auto` браузердің еркінде

See the section on [calculating widths and margins](#) for behavior

3."border":

<code>'border-top'</code>	[<code><border-width></code> <code><border-style></code> <code>'border-top-color'</code>] inherit
<code>'border-right'</code>	
<code>'border-bottom'</code>	
<code>'border-left'</code>	

The border width properties specify the width of the [border area](#).
 take one of the following values:
asuasu.kz

`thin` жіңішке

A thin border.

`medium` орташа

A medium border.

`thick` жуан

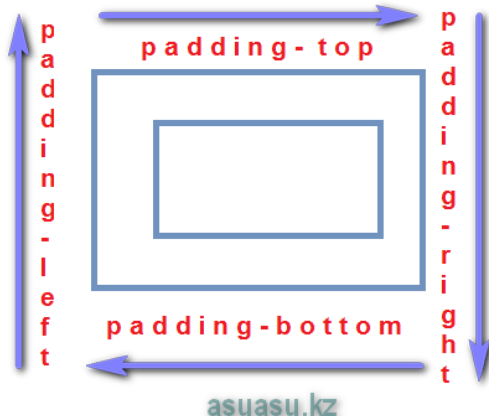
A thick border.

`<length>` сандық өлшем

The border's thickness has an explicit value. Explicit border

6 "Padding", "margin" мен "border" CSS ережелерінің жазылуы.

1. Осы аталған ережелер жазылғанда белгілі бір тәртіпке бағынады, яғни төмендегі суретте көрсетілген стрелка-нұсқағыштар көрсеткен бағыт бойынша алдымен top-жоғарғысы, сосын right-оң жақ, одан кейін bottom-төменгісі, соңынан left-сол жақтағысы жазылуы керек.



2. Осы аталған CSS ережелерінің 4 түрлі жазылуы бар. Олар төмендегі суретте нақтылап көрсетілген, әр ереженің жанында стрелкамен оның баламасы бейнеленген. Егер сипаттамалардың арасында мәндері ұқсасы кездессе, олар ережелер баламасымен өрнектеле алады. Яғни, бұл балама ережелер сайт парақшаларының кодын жазғанда көп жеңілдік береді.

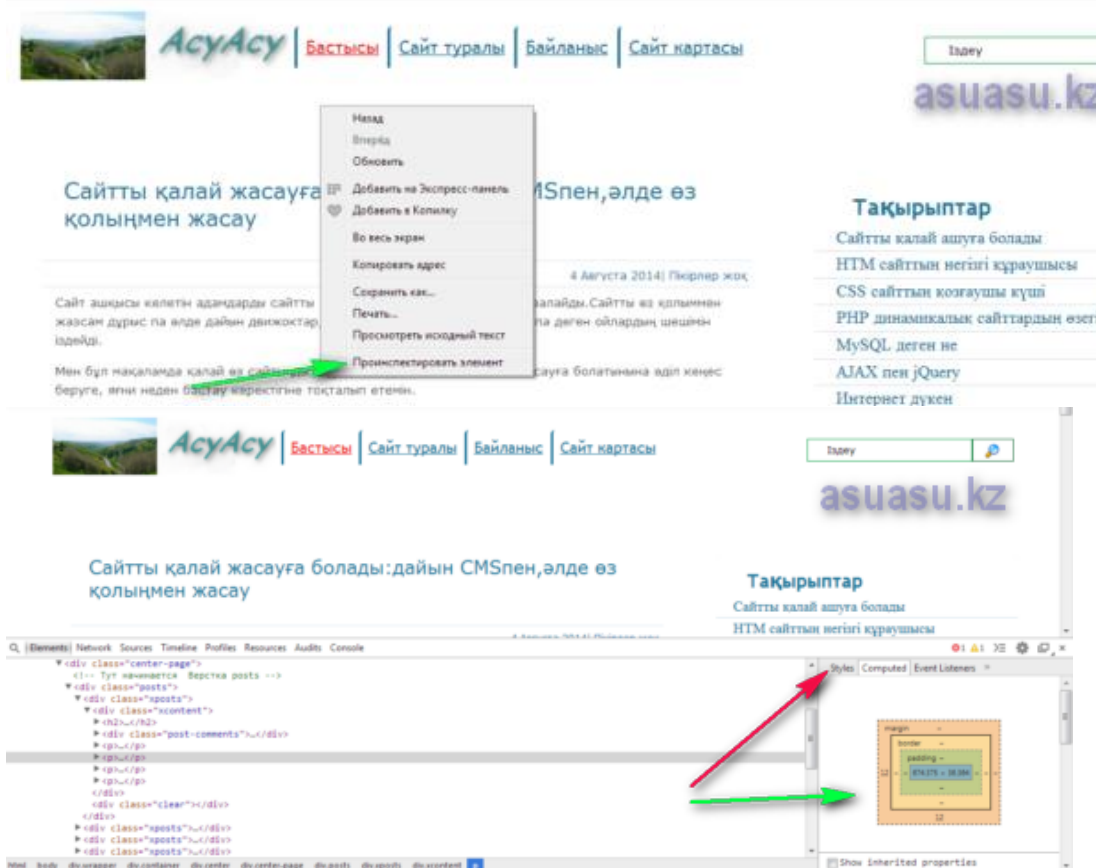
padding: padding-top, padding-right, padding-bottom, padding-left

- 1. padding-top:10px;
padding-right:20px;
padding-bottom:30px;
padding-left:40px; padding:10px 20px 30px 40px;
- 2. padding:10px 20px 30px 20px; padding:10px 20px 30px;
- 3. padding:10px 20px 10px 20px; padding:10px 20px;
- 4. padding:10px 10px 10px 10px; padding:10px;

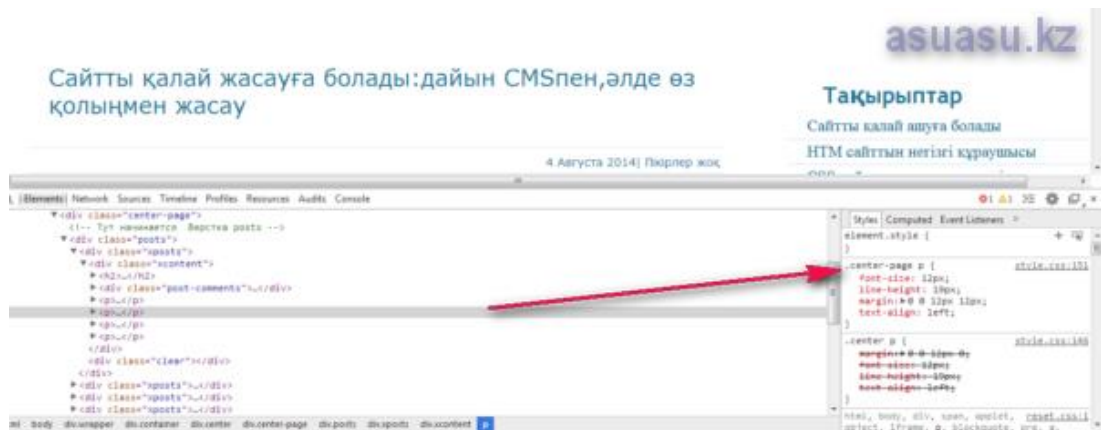
7 Кез келген сайт парақшасының кодын көру мүмкіндігі

Жоғарыда айтып өткенімдей, сізде Firebug болмаса да қалаған браузеріңізде кез келген сайттың кез келген жерінде тұрып мышканың оң жақ кнопкасын басып HTML тегінің жоғарыда талдап өткен блоктық макетін ғана емес, сол сияқты сайттың кодын да және оның CSS ережелерін де көре аласыз.

Мысалы, Опера браузеріндегі менің парақшамды алайық:



Енді осы окнодағы мен қызыл стрелкамен көрсеткен "Styles" деген вкладканы бассаңыз, стильдік кестелер, яғни осы белгілі бір HTML тегіне қатысты жазылған CSS ережелерін көре аласыз.



Қаласаңыз, сіз бұл стильдік кестелер, яғни CSS мәндерін өзгертіп өз браузеріңізде парақшаның өзгерген сипатын көре аласыз. Бірақ, бұл тек сіздің браузерде ғана орын алады, әрине негізгі кодты тікелей өзгерте алмайсыз. Мәнін өзгерту үшін мышканың нұсқағышын сол көрсетілген мәнің үстінен екі рет басып, пайда болған блокқа қалаған мәніңізді қоясыз, парақша сол бойда өзгеріске кшырайды.

Зертханалық жұмыс №15. Браузер деңгейінде CSS-ті қолдау

Тапсырма: HTML мен CSS қолданып портфолио құрастыру

Портфолио деген – белгілі бір маманның орындаған жұмыстарының сәнді де мәнді етіп жасалған тізбесі.

Мысалы,

копирайтердің (жалға мәтін жазғыш) портфолиосы – оның өзі жазған, тақырыптар немесе мәтін түрлері бойынша (коммерциялық ұсыныс, сайт контенті, слогандар) сұрыпталған мәтіндерінен тұрады.

Модельдің портфолиосы – модельдің түрлі образдардағы бейнесін көрсететін фотосуреттерінен құралады.

Веб-дизайнердің портфолиосы – ол жасаған сайттардың, баннерлердің немесе дизайнерлік өнімдерінің суреттерінен жасалады.

Фотографтың портфолиосы – оның түрлі бағыттарда (тойлар, портреттер, репортаждар) түсірген фотосуреттерінің жиынтығынан жасалады.

Бағдарламашының портфолиосы – өзі жазған CMS, скрипттер, программалардың интерфейс суреттері мен олардың жұмысын көруге болатын сілтемелерден тұрады.

Портфолиодағы жұмыстардың сапасы мен деңгейіне қарап клиенттер оны орындаушы туралы пікір қалыптастырады – мұнымен жұмыс істеуге бола ма, жоқ па. Сондықтан портфолионы жасауға өте үлкен мән беру керек.

Портфолио қалай жасалады?

5. Портфолионың бағыттарын немесе бөлімдерін белгілеп алыңыз. Мысалы, копирайтерде бұл: коммерциялық ұсыныстар, сайттарға арналған мәтіндер, нейминг, слогандар. Фотографта: репортаждық түсірілімдер, портреттер, той түсірілімдері және т.б.,

6. Әр бағыт бойынша ең жақсы жұмыстарыңыз бен орташа сапалы жобаларыңызды таңдап алыңыз. Портфолиоға әр бағыт бойынша 10-20 жұмыс қосу оптималды болады. Осының өзі сіздің кәсіби деңгейіңізді клиентке жеткілікті дәрежеде көрсетеді.

7. Портфолионы жақсылап сәндеуді ойластырыңыз. Жұмыстарыңызды ыңғайлы ретпен және көрнекі түрде орналастырыңыз.

8. Портфолионы Интернеттегі жеке сайтыңызға орналастырыңыз. Қосымша компьютеріңіз бен флешкаға да сақтап қойған жөн.

Портфолиоға қандай жұмыстарды қосу керек?

Портфолио жасағанда алғашқы туындайтын сұрақ – қандай жұмыстарымды көрсету керек? Егер Сіз кәсібіңізді енді-енді бастаған болсаңыз, онда барлық тындырған істеріңізді көрсеткен дұрыс. Сізге сұраныс беруші клиенттеріңіз еңбекқор, ізденгіш, тындырымды екеніңізді және жеткілікті тәжірибе жинақтағаныңызды көреді. Ал егер өз кәсібіңізбен ежелден айналысып келе жатқан, өте тәжірибелі шебер болсаңыз, онда портфолиода ең сапалы да қызықты жұмыстарыңызды іріктеп көрсеткен жөн. Сіздің үлкен шеберлігіңізді сол жоғары сапалы жұмыстарыңыз бірден-ақ көрсетеді.

Зертханалық жұмыс №16. JavaScript кіріспе

Алғаш скрипттер

JavaScript – HTML-тілінің динамикасына қажетті құралдардың бірі болып табылады. JavaScript көмегімен гипермәтіндер HTMLды жаңа мүмкіндіктерімен жүргізе алатын кіріктірілген бағдарламаға ие болады. Бұл браузерге түсінікті бағдарламалау тілі. Демек браузер тілдің командаларын орындай алады.

JavaScript бағдарламасын (оны *скрипт немесе* сценарий деп атайды) HTML-программасының ішіне орналастыруға немесе жеке файл етіп ұстауға болады. Бұл файлды браузер арнайы команда арқылы HTML- кодының интерпретациясы кезінде оқиды.

JavaScript кодтары HTML кодтарын толықтырады да, олармен бірге жұмыс істейді. Олар жеке файлда орналасса да.

Скрипт `<SCRIPT>. . .</SCRIPT>` екі жұп тегінің арасында орналасады.

Әдетте скрипт жазуы мысалда көрсетілгендей болады (4.1 листинг).

3.1 листинг

```
<HTML>
<HEAD>
<TITLE>браузерді тексеру</TITLE>
</HEAD>
<BODY bgcolor=white text=black link=blue alink=red vlink=purple>
<H1>браузерді тексеру</H1>
<HR>
<SCRIPT language=JavaScript>
<!--
alert("Сіздің браузеріңіз JavaScript тілін қолдайды!");
//-->
</SCRIPT>
<NOSCRIPT>
<H2>Ескерту</H2>
<P>
Сіздің браузерыңыз JavaScript тілін қолдамайды. Бұл құжат дұрыс емес
көрсетіледі, қате жайында хабарлар шығады.
<P> Скрипттің құжаттың версиясы орналасқан
<A href=doc1.htm>Мүнде</A>.
```

</NOSCRIPT>

<P>

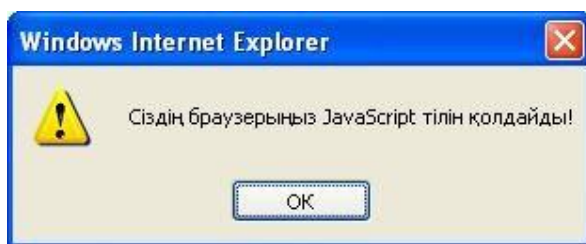
Негізгі мәтінге қайтып келу үшін браузердың саймандар панеліндегі

Артқа басқышын басыңыз.

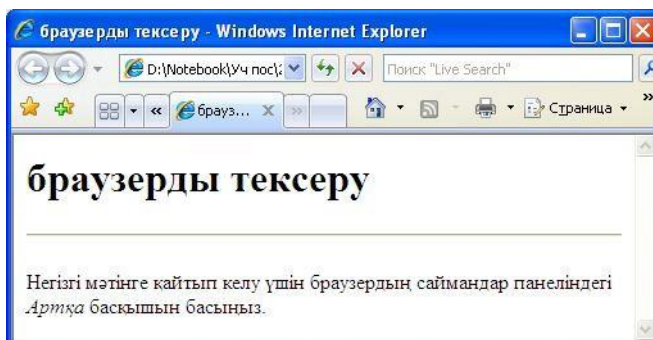
</BODY>

</HTML>

Жоғарыда көрсетілген мысалда скрипт JavaScript тіліндегі жалғыз командаға ие болады – alert функциясын шақыру. Alert функциясы ішінде жалғыз ОК басқышы бар ақпараттық терезені құрады (4.1 сурет). Қолданушы өабарламаны оқиды, басқышты басады да, терезе экраннан кетеді. Енді браузер HTML-кодтағы скриптте орналасқан командаларды интерпретациялауға кіріседі – 4.2 суретте көрсетілген мәтінді терезеге шығарады.



4.1- сурет. Alert ақпараттық терезесі.

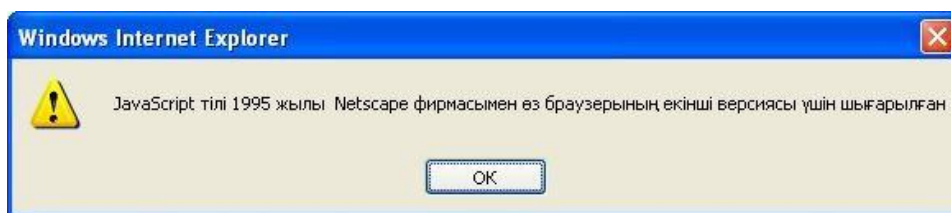


4.2- сурет. Скрипттерді қолдайтын браузердегі беттің түрі.

Егер браузер JavaScript тілін қолдамаса немесе бұл қолдау баптауларда өшірілген болса, онда хабарлама құжат терезесіне шығарылады.

Егер alert функциясының мәтін аргументі тым ұзын болса, онда оны «+» белгісін қолдана отырып (қосылу операциясы), бірнеше жолда жазуға болады:

```
alert("JavaScript тілі 1995 жылы "+ " Netscape фирмасымен өз браузерының екінші версиясы үшін шығарылған");
```



4.3- сурет. Бұл жолды жазу үшін қосу операциясы қолданылған.

Қосылу операциясы экрандағы жолдың ажырауын білдірмейді (4.4 сурет). Жолдарды аудару үшін «\n» символдарын қолдану керек:
alert ("Алғаш жол.\nЕкінші жол.");



4.4- сурет. Мәтінді екі жолға шығару

8 Тапсырмалар

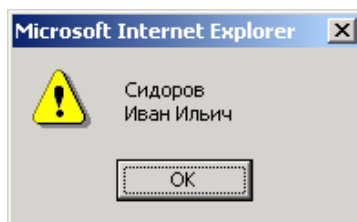
alert функциясымен тәжірибе жүргізіңіз.

Басқа мәтіндерді, жолдарды аудару символарын, бір функцияны бірнеше жолға жазуды, бірнеше функцияға жазуды тексеріп көріңіз. Тәжірибелер үшін келесіні қолданыңыз (4.2 листинг, 4.5 сурет).

9 4.2 листинг

```
<HTML>
<HEAD>
<TITLE>< alertпен тәжірибе </TITLE>
</HEAD>
<BODY bgcolor=white text=black>
<H2> alert функциясымен тәжірибе </H2>
<SCRIPT language=JavaScript>
<! --
alert ("! ! !") ;
//-->
</SCRIPT>
</BODY></HTML>
```

alert функциясының көмегімен бірнеше ақпараттық терезелерді алу керек:



4.5 - сурет

Зертханалық жұмыс №17. HTML құжатына JavaScript енгізу

Жұмыстың мақсаты-HTML-құжатқа өзгерістер енгізу үшін нысандардың қасиеттері мен әдістеріне қатынаудың түрлі тәсілдерін пайдалануды үйрену.

HTML құжаттағы барлық өзгерістер құжаттың Объектілік моделінің құрамына кіретін нысандардың қасиеттері, әдістері мен оқиғалары арқылы

жасалады. (1.1-мысал) сценарийді қарастырайық, онда объектілердің қасиеттері мен әдістеріне қатынаудың типтік тәсілдері қолданылады.

Мысал 1.

```
<HTML ID='DOCUM'>
<CENTER><H1 ID='zag'> ВАСЬКА</h1>
Вариант, совместимый с Mozilla<P>
<FORM NAME='f1'>
<INPUT TYPE=BUTTON name="bot" onclick=bm() value='Увеличить'>
</FORM >
<P><IMG SRC="VaskaM.jpg" ID='Vas' onclick=bm()>
</center>
<!--В скрипте закомментирован вывод окна со списком объектов -->
<SCRIPT>
/*
var msg="
for(i=0; i<document.all.length;i++)
msg+=i + ' ' + document.all[i].tagName + ' ID=' + document.all[i].id+'\n'
alert(msg)
*/
flag=true
function bm()
{ if(flag)
{ document.images[0].src='VaskaB.jpg' flag=false;
document.forms[0].bot.value="Уменьшить"
document.forms[0].bot.style.background='red'
document.all.bot.style.color='black';
}
else
{ document.getElementById("Vas").src='VaskaM.jpg';
//обращение к кнопке по индексу
document.forms[0].bot.value="Увеличить"
document.forms[0].bot.style.background='green'
document.all.bot.style.color='white'; flag= true
}
}
</script>
</HTML>
```

1мысалында сценарий фотосуретті ауыстыру және батырманың түсі мен жазбасын өзгерту үшін қолданылады. Бір-бірін киімнің кішкентай және үлкен суреттерін ауыстырады.

Суретті ауыстыру және батырманың түрін өзгерту мысалында батырманы немесе фотосуреттің өзін басыңыз. <INPUT> және тегтерінде onclick=bm () атрибуты бар, оның көмегімен тінтуірге жауап ретінде BM () функциясы пайда болады. Мұндай атрибут оқиғаны өңдеуші деп аталады. Фотосуреті бар Файл атауы IMG нысанының SRC қасиеті болып табылады. Фотосуретті ауыстыру сценарийде тең екі жолмен жасалады:

```
document.images[0].src='VaskaB.jpg'
document.getElementById("Vas").src='VaskaM.jpg'
```

Бірінші нұсқада фотосуретке кіру images топтамасы арқылы жүзеге асырылады, онда фотосурет бетінде орналасқан барлық суреттер (мысалы, бір фотосурет) нөмірленген. Нөмірлеу нөлден басталады. Екінші нұсқада document нысанының getElementById() әдісі қолданылады. getElementById() әдісінің параметрі IMG объектісінің ID (ID) болып табылады. Барлық оператор былай оқылады: "ID=Vas бар және document нысанының құрамына кіретін src объектісіне VaskaM мәнін беру.jpg ". Document нысанының өзі window объектісіне қатысты еншілес болып табылады. Сондықтан src қасиетіне толық қарау осылай жазу керек еді:

```
window.document.images[0].src='VaskaB.jpg'
window.document.getElementById("Vas").src='VaskaM.jpg'
```

Әдетте, window объектісінің қасиеттерін немесе әдістерін пайдалану қажет болған жағдайларды қоспағанда, бұл сөз төмен түседі.

Түйме сипаттарына қараған кезде нысандардың күрделі иерархиясы байқалады:

```
document.forms[0].bot.value="Уменьшить"  
document.forms.f1.bot.style.background='red'
```

Бірінші оператор кнопкадағы жазбаны өзгерту үшін қызмет етеді. Батырмаға оның аты бойынша қатынайды (name='bot'). Формаға кіру forms коллекциясы және коллекциядағы форманың нөмірі арқылы жүзеге асырылады.

Екінші оператормен түйменің түсі өзгереді. Формаға кіру forms топтамасы және f1 нысанының аты арқылы жүзеге асырылады. Стильмен берілетін қасиеттерге қол жеткізудің өзіндік ерекшелігі-style еншілес нысан ретінде формальды түрде әрекет етеді. Осы мысалда style-бұл bot аты бар нысан, яғни, батырма.

Объектілерге бару үшін All document нысанының коллекциясы ыңғайлы. All коллекциясына HTML-құжаттың барлық нысандары кіреді, сондықтан нысанды оның ID немесе аты бойынша таңдау қажет. All топтамасы кнопкадағы жазудың түсін өзгерту үшін қолданылады:

```
document.all.bot.style.color='black'
```

Тапсырма-1. Браузер терезесінде келесі үш жол түрінде көрсетілетін HTML құжатын жазыңыз:

- ҚАСИЕТТЕРІ МЕН ӘДІСТЕРІНЕ ҚОЛ ЖЕТКІЗУ
- All топтамасы
- GetElementById әдісі()

Бірінші жолды контейнерге <H2>қойыңыз...< / h2>, екіншісі – контейнерге <P> ...</p>, үшіншісі-контейнерге <DIV> ... </div>. Фон түстерін және жазулардың әріптерін өзгерту үшін скрипт жазыңыз, осы жолдарды басыңыз. Бірінші жолда шерткен кезде әріптердің түсі қара түстен ақ түске немесе ақ түстен қара түске, ал фоны сары түстен көкке немесе көк түстен сары түске өзгеруі тиіс. Сондай-ақ үшінші жолдың түстері өзгеруі керек.

Екінші жолда сілтеген кезде әріптердің түсі қызылдан аққа және керісінше, ал фонның түсі ақ түстен жасыл түске және керісінше өзгертілуі тиіс.

Бірінші және үшінші жолдарды өзгерту үшін getElementById () әдісін, ал екінші жол үшін all жиынтығын қолданыңыз.

HTML құжатында сценарийлерді орналастыру

HTML құжаттағы сценарийді үш жолмен орналастыруға болады: ашу тегінде оқиға төлсипатының мәні ретінде (2.1-суретті қараңыз)); тегімен шектелген контейнерде <SCRIPT> ...< / script>; жеке файлда.

Сценарийлері бар контейнерлер HTML-құжаттың кез келген жерінде орналастырылуы мүмкін. Бір HTML құжаттағы сценарийлер саны шектелмейді.

Сценарийлерді орындау HTML-құжатты жүктегенде және оқиға болған кезде жүзеге асырылады. HTML құжатын жүктеу кезінде браузер кездесетін сценарийлерді қарап, оларды орындауға тырысады. Егер опция қоңырауы болса, браузер ағымдағы және барлық бұрын жүктелген сценарийлерде оның сипаттамасын іздейді. Сондықтан, функцияның сипаттамасы қоңыраумен бір сценарийде немесе қоңыраудан жоғары орналасқан функцияның сценарийінде орналастырылуы керек.

Сценарийді бөлек файлда сақтауға болады. HTML құжат сценарийін енгізу үшін SRC тегінің атрибуты <SCRIPT>болып табылады. Жеке файлда орналасқан

сценарийді сайттың көптеген беттерінде пайдалануға болады. 1.2 мысалында дүкендер желісі сайтының бір беті қарастырылады.

Мысал-2.

```
html<<!-- СКРИПТ загружается из файла primJs.js--> <HEAD>
<TITLE>Сеть</title>
</head>
<body>
<SCRIPT language="JavaScript" src="PrimJs.js">
</script>
<H2 align=center style="color:green">Магазин "ПОДАРКИ"</h2>
Адрес: Лесная ул., д.2<P>
Транспорт: трамвай 7, 23, автобусы 56, 93
</body>
</html>
// Файл primJs.js
a="background-color:#00ffff; color:#ff00ff;"
a+="font-size:24pt; font-family:'Times New Roman'"
naim='Сеть магазинов "ВСЁ ДЛЯ ДОМА"'
var da=new Date()
d=da.getDate()+"."+da.getMonth()+1+"."+da.getFullYear()
document.write('<P align=center style= "'+a+'">'+
naim+'</p><P>Сегодня '+d+'</p>')
```

Барлық желі дүкендерінің беттеріне ортақ тақырып және бүгінгі күн жеке primJs файлында орналасқан скриптпен қалыптасады.js . Write(HTML коды) әдісі Белгіленген мәтін бетіне (HTML коды) қою үшін қолданылады. Бұл әдіс тек бет жүктелгенде ғана қолданылады, өйткені жүктеу аяқталғаннан кейін оны қолдану беттің ескі мазмұнын өшіруге әкеледі.

Тапсырма-2. 2 мысалына үш бет қосыңыз. Екі бет ыдыс-аяқ және жиһаз дүкендері туралы ақпаратты көрсету керек. Үшінші бет-басты сайтта дүкендер желісінің барлығы үйге арналған. Онда желіге кіретін дүкендердің беттеріне сілтеме болуы керек. Дүкен беттерінде primJs файлынан скрипт қолданылуы керек.js.

Зертханалық жұмыс №18. Мәліметтер типі, айнымалылар, өрнектер

Жұмыстың мақсаты-бетке өзгерістер енгізу үшін құжат үлгісінде бар оқиғаларды пайдалануды үйрену.

Ең жиі сценарийлерде onclick жоғарыда қарастырылған оқиға пайдаланылады. Белгілі бір HTML құжат элементіне пайдаланушының назарын аудару үшін, оған тышқан жүгіргі түскен кезде осы элементтің сипаттарын өзгертуге, ал жүгіргіні алып тастағанда бұрынғы сипаттар мәндерін қалпына келтіруге болады. Мысалы, элементтің түсін немесе өлшемін өзгертуге болады. Тінтуір меңзерінің элементке түсуі onMouseOver оқиғасы тіркеледі. Ол үшін жұптық onMouseOut оқиғасы тінтуір меңзерін элементтен алып тастағанда орын алады.

OnMouseDown және OnMouseUp оқиғаларының басқа жұбы тінтуірдің сол жақ батырмасын басқанда және босатқанда орын алады. Бұл екі оқиғаны элементтердің сипаттарын өзгерту немесе тінтуір түймесін басу кезінде элементтерді ауыстыру үшін қолдануға ыңғайлы.

Оқиғаға жеке элементте реакция

Объектілік модельде объектілер бір-біріне салынуы мүмкін болғандықтан, еншілес объектіде болып жатқан оқиға бір мезгілде ата-аналық объектіде де орын

алады. JavaScript оқиғалар иерархиясына әсерін локализациялаудың әр түрлі тәсілдерін ұсынады. Локализациялаудың қарапайым тәсілі (3 мысал) оқиға әсер етуі тиіс тегте сценарийді орналастыру болып табылады.

Мысал 3.

```
<HTML>
<BODY>
<P align=right ID='alfa'
onMouseOver="document.all.alfa.align='center'
"onMouseOut="this.align='left'">
События onMouseOver и onMouseOut </p>
</body>
</html>
```

3 мысалындағы бет <P> ...</p>контейнерге салынған бір жолдан тұрады. Объектілік бет үлгісінде P объектімен болған оқиға BODY ата-аналық объектімен де орын алады. Оқиға реакциясын тек жолдан тыс, яғни объект P, оқиға сценарийі <P >тегіне орналастырылады.

Сценарийді орындау нәтижесінде жолдағы мәтіннің жағдайы өзгереді. Бастапқыда жол терезенің оң жақ шетіне қысылады. Оған курсор түскен кезде ол ортасына тураланады, ал курсорды алғаннан кейін терезенің сол жақ жиегіне қысылады. Объектіге кіру үшін Internet Explorer 6.0 және Mozilla Firefox 2.0 браузерлерімен дұрыс қабылданатын all топтамасы қолданылады. This кілт сөзі ағымдағы нысанның сілтемесін білдіреді.

Егер оқиға орын алған кезде көп әрекет жасау керек болса, онда сценарийді функция түрінде жазу және оны элементтен бөлек сценарийлерге арналған <SCRIPT> ...</script>контейнерге орналастыру ыңғайлы. 4 мысалында onMouseOver және onMouseOut оқиғаларының әрқайсысы екі әрекетті тудырады: жолда мәтін түсін Туралау және өзгерту.

Мысал 4.

```
<HTML>
<P align=right ID='alfa' onMouseOver="M_Over()"
onMouseOut="M_Out()">
Событие onMouseOver</p>
<SCRIPT>
function M_Over()
{ document.all.alfa.align='center'
  document.all.alfa.style.color='FF00FF'
}
function M_Out()
{ document.all.alfa.align='left'
  document.all.alfa.style.color='0000FF'
}
</script>
</html>
```

Тапсырма 3. Келесі төрт жол түрінде браузер терезесінде көрсетілетін HTML құжатын жазыңыз:

- Тышқанмен бес оқиға
- Маған тінтуір басыңыз
- Осы мәтінде тінтуірдің сол жақ батырмасын басыңыз, ұстаңыз және босатыңыз

Осы жазу бойынша тышқан меңзерімен баяу жүріңіз

Бірінші жол-беттің тақырыбы. Екінші жол тышқанды басу кезінде келесідей өзгереді:

- қаріп 48pt дейін артады;
- қаріп түсі ақ өзгереді;
- фон түсі көкке өзгереді.

Екінші жолды бастапқы түрге қайтарады.

Тінтуірдің сол жақ батырмасы басылса немесе босатылса, үшінші жолдың фоны өзгереді. Басқан кезде фон жасыл, ал босатқан кезде – сары болады.

Тышқан меңзері төртінші жолға түскен кезде оның фоны қызыл болады, ал алу кезінде – көгілдір болады.

Басты элементте оқиғаны бекіту

Егер қандай да бір оқиғаға реакция беттегі бірнеше элементтерді талап етсе, осы оқиғаны тек ата-аналық элементте өңдеу мүмкіндігін шақыруға болады. Функцияда оқиғаның қай элементте орын алғаны және тиісті әрекеттер орындалатыны анықталады. Мұндай тәсілдің ыңғайлылығы түрлендірудің барлық алгоритмі бір жерде, ал кемшілігі – функцияның өз күрделілігінде. Браузер терезесіндегі кез келген үш нысанның қасиеттерін өзгерту үшін бір функция қызмет ететін сценарийді қарастырайық (5-мысал).

Мысал 5.

```
<HTML><HEAD><TITLE>Реакция на событие</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<STYLE>H1 {color:FF00FF}
        #k1{position:absolute; left:50;top:200;width:300; height:100;background-color:blue}
        #k2{position:relative; left:50;top:25; width:200; height:50;background-color:yellow}
</STYLE>
</HEAD>
<BODY ID="B" onclick="rodEl(event)">
  <H1 ID="HH" >ЦВЕТ </H1>
  <DIV ID="k1" >
    <DIV ID="k2" ></div>
  </div>
<SCRIPT>
/*      Функция запускается при щелчке мышкой по любой точке документа */
flag=0;
function rodEl(evt)
{ var e = evt || window.event; //e -это объект event
  var elem = e.target || e.srcElement; //elem - элемент (объект),
                                     // на котором произошло событие

  id1=elem.id
  z1=document.getElementById(id1)
  switch(id1)
  { case "k1": //Изменение цвета внешнего прямоугольника
    z=z1.style.backgroundColor
    if(z!="red")z="red"
    else z="green"
    z1.style.backgroundColor=z
    break
  case "k2": //Изменение цвета внутреннего прямоугольника
    z=z1.style.backgroundColor
    if(z!='rgb(0, 255, 255)'){z='rgb(0, 255, 255)'}
    else{z='rgb(0, 255, 0)'}
    z1.style.backgroundColor=z
    break
  case "B": //Изменение цвета заднего плана документа
    z=z1.style.backgroundColor
    if(z!='rgb(190, 190, 190)'){z='rgb(190, 190, 190)'}
    else {z='rgb(0, 190, 190)'}
    z1.style.backgroundColor=z
    break
  case "HH": //Изменение цвета слова "Цвет"
    if(flag==0)
    { document.getElementById(id1).style.color='rgb(170,0,170)';
      flag=1;
    }
    else
    { document.getElementById(id1).style.color='rgb(0,255,170)';
      flag=0;
    }
  }
}
```



```
}  
</SCRIPT>  
</BODY>  
</HTML>
```

5 мысалында H1 элементіне және DIV екі элементіне қатысты ата-аналық BODY элементі болып табылады. Сондықтан <BODY> Тегінде OnClick оқиғасын өңдеуге арналған rodEl () функциясы пайда болады.

Оқиға басталған кезде ол туралы барлық ақпарат event объектісінде есте сақталады. Бұл нысан әртүрлі W3C стандарты мен Internet Explorer браузерінде сипатталған. Internet Explorer жаңа нұсқалары W3C стандартын қолдайды.

W3C стандартында event объектісі параметр ретінде функцияға беріледі, ал оқиға болған объектіге бару үшін event қасиеті қызмет етеді.target.

Internet Explorer window объектісі.event-жаһандық және сондықтан оны параметр түрінде функцияға жіберу қажет емес. Оқиға болған объектіге хабарласу үшін Internet Explorer-те window қасиеті болады.event.srcElement

5 мысалында, дененің бірінші екі жолы rodEl()функциясы оқиға болған объектіге кроссплатформалық (кез келген браузерде) қатынас үшін қызмет етеді. Event объектісін кроссплатфорлы пайдаланудың басқа тәсілі 6.2 зертханалық жұмыс мысалында сипатталады (brows (evt))

Келесі жолдарда rodEl() функциясы алдымен пайдаланушы тінтуірмен басқан элементтің Id анықталады, содан кейін Switch операторының көмегімен көрсетілген элементтің қасиеттерінің өзгеруіне өту жасалады.

Тапсырма 4 . Оның астында бейнеленген және қол қойылған бетті жасаңыз. Қолтаңбаны басқанда, ол өз түсін өзгертуі керек. Суретті нұқыңыз сурет пен қолтаңбаны ауыстыруы керек. Оқиғаны өңдеу функциясы кескінге және нысанның қол қоюына қатысты ата-анасынан шақырылуы тиіс.

Зертханалық жұмыс №19. Функция және әдістер ұғымы.

Функцияны шақыру

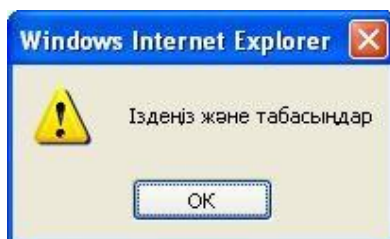
Бұған дейін тек **alert** функциясы көрсетілген болатын:

```
alert(мән);
```

Бұл функцияның аргументі мәтіндік те, сандық та мәнге ие бола алады.

```
var ex1 = "Іздеңіз";  
var ex2 = "табасындар";  
alert(ex1+" және " +  
ex2);
```

скриптті орындау нәтижесі 4.6. суретінде көрсетілген



4.6- сурет

```
var x = 48;
var ex1 = "тотықұстар";
alert(x + " " + ex1);
```

скриптті орындау нәтижесі 4.7. суретінде көрсетілген



4.7 сурет. Alert ақпараттық терезесі.

```
Var x =
  2; var
  y = 3;
alert(«Нәтиже: « + x * y);
```

скриптті орындау нәтижесі 4.8 суретінде көрсетілген



- 4.8сурет. Alert ақпараттық терезесі.

Меншіктеу

«айнымалы=мән» түріндегі әдеттегі меншіктеумен қатар келесі жазуларды қолдануға болады (4.2 кесте).

++ x және x ++ (-- x және x --) командаларының арасындағы айырмашылық осы командалар басқа командалардың негізгі бөлігі болып қолдаылған кезде білінеді. Бірінші жағдайда операция айнымалыны қолданғанша (атына дейінгі белгілер), ал екіншісінде – одан кейін.

4.2 кесте

Жалпы түрі	Мысал	Білдіруі
айнымалы ++	x ++	x = x + 1
++ айнымалы	++ x	x = x + 1
айнымалы --	x --	x = x - 1
-- айнымалы	-- x	x = x - 1
айнымалы += мән	x += "кот"	x = x + "кот"

айнымалы -= мән	x-=2	x = x - 2
айнымалы *= мән	x *= 10	x = x * 10
айнымалы /= мән	x /= y + z	x = x / (y+z)
айнымалы %= мән	x%=9	x = x%9

x = 5; — Бұл командаларды орындағаннан кейін: x пен y 6 ға тең болды.

y = ++ x;

x = 5; — Бұл командаларды орындағаннан кейін: x 6 ға, ал y 5 ке тең болды.

y = x ++;

Тапсырмалар

1. Келесі командаларды орындау нәтижесінде alert функциясының терезесіне не шығады: 10 саны, 35 саны немесе қате жайында хабарлама ма?

а) // var x
= 10; var x =
35;

// var x = 8;
alert(x);

б) /* var x
= 10; var x
= 35;
/* var x
= 8; alert
(x) ;

в) /* var x = 10;

/var x = 35; /
var x = 8;

*/alert(x);

г) /* var x = 10

var x = 35; var x = 8;

*/alert(x);

2. Келесі командаларды орындау нәтижесінде alert функциясының терезесіне не шығады?

а) var x =
"60"; var y
= 20; alert
(x + y)

б) var x =
"60"; var y = 20;
alert (x - y);

в) var x =
"60"; var y
= 20; alert
(x / y);

```

г) var x = "с";
    var y =
    "ка";
alert(x + "оба" + y);
д) var x = 13;
var y = 3;
y = ++ x / -- y;
    alert(y);
е) var x = 13;
    var y = 3;
y = -- x / ++ y;
    alert (y);
ж) var x =
13; var y = 3;
y = x ++ % y ++;
    alert(y);
з) var x
= ""; x +=
"обед";
alert("п" + x + "а");
и) var x = "м";
x += "а";
x = "рам" + x;
    var y = "рог";
    alert("п" + y
    + x);
к) var str1 = "Барлығымыз аздап атпыз,\n";
    var str2 = "Әр қайсымыз өз бетінше ат.";
    alert(str1 + str2);
л) var str1 = "Есептеулер нәтижесі :\n";
    var num1 = 35;
var num2 = 8;
alert(str1 + num1 / num2);
м) var str1 = " Есептеулер нәтижесі:\n";
var num1 = 35;
    var num2 = 8;
alert(str1 + num1 % num2);
н) var str1 = " Есептеулер нәтижесі:\n";
    var num1 = 35;
var num2 = 8;
alert(str1 + (num1 ++)% (num2 ++));
о) var str1 = " Есептеулер
нәтижесі:\n"; var num1 = 35;
var num2 = 8;
alert(str1 + (num1 ++ ) % (num2 ++));
п) var str1 = " Есептеулер нәтижесі:\n";
    var num1 = 12;
var num2 = 8;
alert(str1 + num1 + "+" + num2 + "=" + num1 + num2);

```

```

p) var str1 = " Есептеулер
нәтижесі:\n"; var num1 = 12;
var num2 = 8;
alert(str1 + num1 + "+" + num2 + "=" + (num1 + num2));
с) var str1 = " Есептеулер нәтижесі:\n";
var num1 = 12;
var num2 = 8;
alert(str1 + num1 + "*" + num2 + "=" + num1 *
num2); т) var str1 = " Есептеулер нәтижесі:\n"; з
var str2 = 4;
var num1 =
12; var
num2 = 8;
alert(str1+num1+"*" +num2+"+"str2+"="+num1*num2+str2); y)var
str1 = " Есептеулер нәтижесі:\n";
var str2 = 4; var num1 = 12; var num2 = 8;
alert(str1+num1+"*"4-num2+"+"str2+"="+num1*num2+str2)); ф)
var str1 = "10";
var str2 = "5";
alert(str1 /
str2);
х) var str1 = "Этажерка";
var str2 = 2; alert(str1 /
str2);

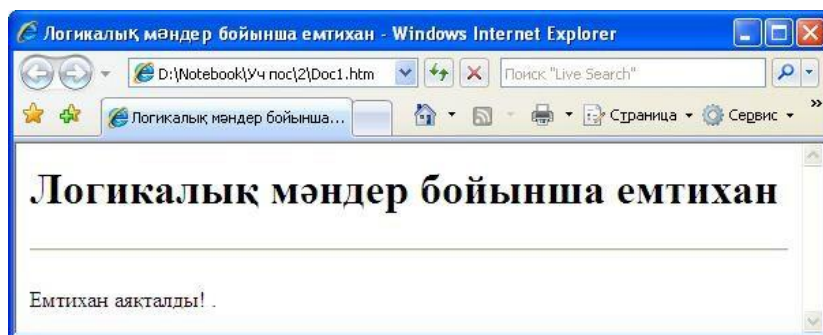
```

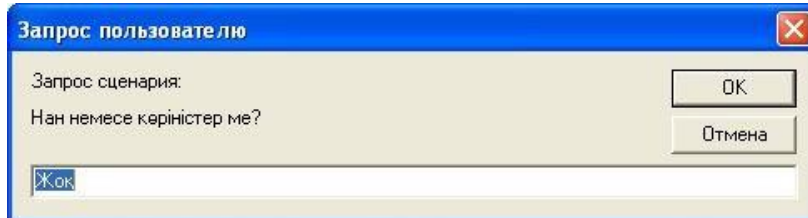
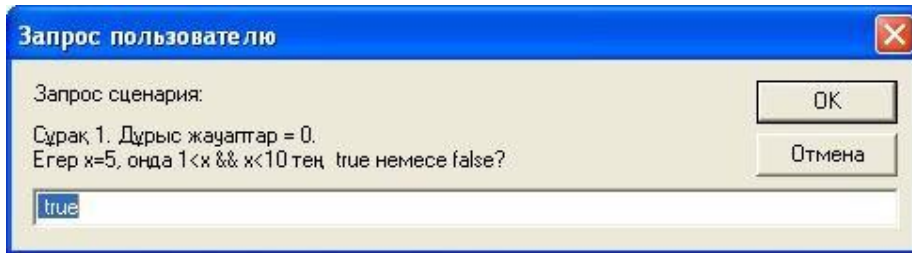
Зертханалық жұмыс №20. Объектілер және қасиеттер

Арифметикалық және жолдық мәндер шарт ретінде

if командасында шарт ретінде тек қана логикалық емес, сонымен қатар арифметикалық та, жолдық та мәндерді қолдануға болады. Мұндай жағдайда мән жалған деп есептеледі, егер оның мәні 0ге немесе "" бос жолға тең болса, кері жағдайда ақиқат болады. (4.14 кесте).

4.13- сурет. Логикалық мәндер бойынша емтихан





4.15– сурет

4.14 Кесте

Мысал	Түсініктеме
<pre>var x = 20; if(20 - x) x ++;</pre>	Шарт жалған, x мәні өзгермейді
<pre>var x = 20; if(2 * x) x ++;</pre>	Шарт ақиқат, x мәні 1ге артады
<pre>var x = ""; if(x) x += "киса";</pre>	Шарт жалған, x мәні өзгермейді
<pre>var x = ""; if(x + "?") x += "киса";</pre>	Шарт ақиқат, x мәні «киса» мәніне ие болады
<pre>var x = ""; if(!x) x += "киса";</pre>	Шарт ақиқат, x мәні «киса» мәніне ие болады

Логикалық типтің мәндері

JavaScript тілінде айнымалы логикалық типтің мәнін қабылдай алады. Бұл тип true және false деп белгіленетін екі тұрақтыға ие (4.10 кесте).

Альтернативаларды енгізу

Қолданушы сұраққа екі жауаптың бірін таңдауды жасау керек болса, онда prompt функциясын қолдануға болады (4.15 сурет).

4.5 кесте

Мысал	Түсініктеме
<pre>var x = true;</pre>	Командаларды орындағаннан кейін у мәні 3ке тең

<pre> var y = 2; if (!x) y ++; var x = true; var y = 2; if (!x) y ++; var x = true; var y = 2; if (x !x) y ++; var x = true; var y = 2; if (x && !x) y ++; </pre>	<p>Командаларды орындағаннан кейін у мәні 2ге тең</p> <p>Командаларды орындағаннан кейін у мәні 3ке тең</p> <p>Командаларды орындағаннан кейін у мәні 2ге тең</p>
---	---

Дегенмен қолданушының тілегін білудің одан да жақсы әдісі бар — confirm функциясы:

```
confirm("жазу");
```

Бұл функция былай жұмыс істейді. Экранға екі басқышы бар терезе шығарылады (4.16 сурет):



4.16 – сурет. Confirm терезесінің түрі

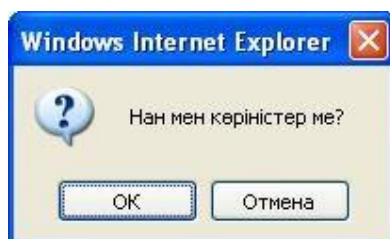
Егер қолданушы ОК басқышын басса, confirm функциясы true мәнін, ал егер **Қайтару** (Cancel) — false мәнге ие болады. <Enter> және <Esc> эквивалентті пернелерін қолдануға болады.

Мысал

```

if(confirm("Нан мен көріністер ме?")) alert("Қазір қарап ал, кейін тамақ іш!");
else alert ("Қысқасы ол қарбыз жегісі келді,\n" + "Ал анаусы тұздалған қиярды.");

```



Бұл кодты қосу кезінде экранға 4.17 суретте көрсетілгендей терезе шығады.

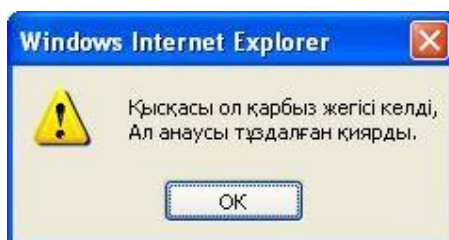
4.17 - сурет. Нан мен көріністер ме?

OK басқышын басса, скрипт өз жұмысын 4.18 суретінде көрсетілгендей жалғасырады.



4.18 - сурет. OK таңдағаннан кейін

Қайтару басқышын басса, онда 4.19 суреттегідей болады:



4.19 - сурет. Қайтару таңдағаннан кейін

Зертханалық жұмыс №21. JavaScript-тегі құжаттың объектілік моделі

Math математикалық класы — математик-бағдарламалаушыларға тригонометриялық және логарифмдік функциялар жиынын береді. Яғни ол тұрақтылар мен әдістерді қамтитын класс. Олардың шақырылу түрі:

Math.константа Math.функция(i..)

Кесте 13. Math класының тұрақтылары

Тұрақтылар	Түсініктемесі	Жазылуы
E	E саны (натурал логарифмдерге негізделген) (2,718281828)	alert(Math.E) var x = Math.log(Math.E); // 1
LN10	10 санының натурал логарифмі (ln10)	alert(Math.LN10)
LN2	2 санының натурал логарифмі (ln2)	alert(Math.LN2)
PI	P тұрақтысы ("пи" саны)	alert(Math.PI) var z=Math.tan(Math.PI/4); // 0.9999999999999999
SQRT1_2	½ санының квадрат түбірін	alert(Math.SQRT1_2)

	есептейді	
SQRT2	2-нің квадрат түбірін есептейді	alert(Math.SQRT2)
LOG2E	e санының логарифмі 1.4426950408889634	alert(Math.log2e)
LOG10E	e санының логарифмі (log10e) 0.4342944819032518	alert(Math.log10e)

Кесте 14. Math класының әдістері (стандартты функциялары)

Әдісі	Түсініктемесі	Жазылуы
abs(сан)	Санның модулін береді.	Var x=-5; var y; y=Math.abs(x) ; alert(y);
sqrt(сан)	Санның квадрат түбірін береді.	y=Math. sqrt (x)
pow(негізі, дәрежесі)	Санның негізінің көрсетілген дәрежесін шығарады.	Math.pow(5.3) 5 ³ =125.
random()	0 мен 1 диапазонындағы кездейсоқ санды шығарады.	random() alert(Math. random());
ceil(сан)	Жоғарғы жағына қарай дөңгелектейді, яғни ең кіші бүтін санды қайтарады, ол аргументтен үлкен немесе тең.	y=Math. ceil (-5. 3) y=-5 var x = Math.ceil(9.2); // 10
floor(сан)	Төменгі жағына қарай дөңгелектейді, яғни ең үлкен бүтін санды қайтарады, ол аргументтен кіші немесе тең.	y=Math. floor (5. 3) y= 5 var y = Math.floor(-5.9); // - 6
round(сан)	Көрсетілген аргументті бүтін мәнге айналдырады.	var n = Math.round(-5.5); // -5 var m = Math.round(-5.6); // -6
cos(сан)	Санның косинусын есептейді.	y=Math.cos(x)
sin(сан)	Санның синусын есептейді.	var x = Math.sin(90); // 0.8939966636005579
tan(сан)	Санның тангенсін есептейді.	y=Math.tan(x)
asin(сан) acos(сан) atan(сан)	Кері тригонометриялық функциялар	var y = Math.acos(1); // 1
exp(сан)	Экспонентаны есептейді.	y=Math.exp (x)
log(сан)	Санның натурал логарифмін есептейді.	var z = Math.log(10); // 2.302585092994046

$\max(\text{arg1}, \text{arg2})$ $\min(\text{arg1}, \text{arg2})$	Екі аргументтің үлкенін немесе кішісін табу	<pre> Var x=2; var y=Math.max(x,10) ; alert(y); var max = Math.max(1, 2, 3, -9, 46, -23); // 46 </pre>
--	---	--

JavaScript операторында әдісті іске қосу үшін (әдетте, оны әдісті шақыру деп атайды), оған сілтеме жасау керек.

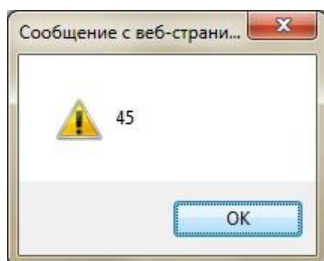
Мысал-1. 5-тің 10 дәрежесін табыңыз. Нәтижесін *st* айнымалысына жазу керек.

```

<script>
var st = Math.pow(2, 10); alert(st);
</script>

```

Мысал-2. Осы скрипте Math объектісіне *floor* әдісі қолданды. Әдістің параметрі ретінде d1 айнымалысының мәні қолданылған. Бұл үшін Math.floor(d1) конструкциясы пайдаланылды. Бұл әдіс аргументтің бөлшек бөлігін алып тастайды. Бұл мән d2 айнымалысына беріледі және alert(d2) әдісінің көмегімен терезеге шығарылады.



```

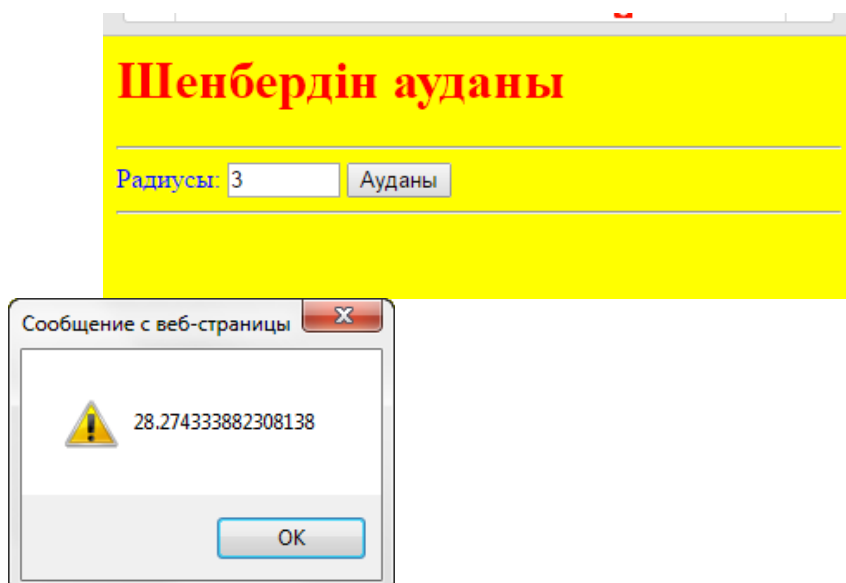
<script> d1=45.6;
d2=Math.floor(d1); alert(d2);</script>

```

Сурет 34. Нәтиже

Өзіндік тапсырмалар:

- 215 санының квадрат түбірін табыңыз.
- 587 санының квадрат түбірін табыңыз. Оны жоғары және төмен жағына қарай дөңгелектеңіз ('**floor**' және '**ceil**' қолданыңыз).
- $\sin(x^2) + \cos(x^2)$ есептеңіз. Нәтижесін «round» арқылы бүтін мәнге айналдырыңыз.
- a және b айнымалылары берілген. Олардың айырымдарының модулін табыңыз.
- Мына сандар берілген: 4, -2, 5, 19, -130, 0, 10. Осылардың ішінен минималды и максималды санды табыңыз.
- Шаршының ауданын есептейтін бағдарламаны жазыңыз.
- Шеңбердің радиусын есептейтін бағдарламаны жазыңыз. Есептеу радиусын форманың мәтіндік өрісіне енгізу керек. Ауданы батырманы басқан кезде есептелуі керек.



Сурет 35. Шеңбердің радиусын есептеу

2 бөлім. XML негіздері.

Зертханалық жұмыс №22. XML шолу

1.Төменде ұсынылған XML құжатын жасау және стильдердің каскадтық кестелері арқылы оны көрсету мысалын жасаңыз.

2.Notepad мәтіндік редакторында жаңа файлды жасап, XML-құжат мәтінін кеңейте отырып енгізіңіз.xml

```
<?xml version="1.0" encoding="windows-1251"?>
<!--Имя файла:fale_1.xml-->
<FILE_1>
  <STUDENT>
    <FAMILY>Иванов</FAMILY>
    <NAME>Сергей </NAME>
    <YEAR>1993</YEAR>
    <GROUP>ИФ 87</GROUP>
  </STUDENT>
  <STUDENT>
    <FAMILY>Петрова</FAMILY>
    <NAME>Галина </NAME>
    <YEAR>1992</YEAR>
    <GROUP>ИФ 87</GROUP>
  </STUDENT>
  <STUDENT>
    <FAMILY>Семенов</FAMILY>
    <NAME>Валерий </NAME>
    <YEAR>1993</YEAR>
    <GROUP>ИФ 88</GROUP>
  </STUDENT>
  <STUDENT>
    <FAMILY>Павлова</FAMILY>
    <NAME>Ирина </NAME>
    <YEAR>1994</YEAR>
    <GROUP>ИФ 89</GROUP>
  </STUDENT>
</FILE_1>
```

Бұл құжат екі негізгі бөліктен тұрады: Пролог және түбірлік құжат (сондай-ақ құжат элементі деп аталады). Құжаттың элементі мұнда FILE_1 деп аталады, оның бастапқы тег - <FILE_1>, ал соңғы - </FILE_1>, ал мазмұны - 4 ішкі элемент student. Өз кезегінде STUDENT әр элементі бірқатар салынған элементтерден тұрады.

3.Internet Explorer браузері арқылы құжатты ашыңыз. Синтаксис тексерілгеннен кейін, құжат экранда көрсетіледі. Қате болған жағдайда, құжаттың орнына бетті көрсету мүмкін еместігі туралы хабар беріледі.

4. Құжат элементтерінің көрсетілуінің егжей-тегжейлі дәрежесін өзгертіп көріңіз. Элементті бұру үшін бастапқы тегтің сол жағындағы минус (-) белгісінің таңбасын немесе қосу (+) таңбасында бүктелген элементтің жанындағы оны ашу үшін басыңыз. Мысалы, минус (-) белгісінде FILE_1 элементінің жанында басу арқылы сіз суретте көрсетілген сияқты аласыз:

```
<?xml version="1.0" encoding="windows-1251" ?>
<!-- Имя файла:fale_1.xml -->
+ <FILE_1>
```

5. File_2 файлын жасаңыз.CSS стильдер Каскад кестесі:

```
STUDENT
  {display:block;
  margin-top: 12pt;
  font-size: 10 pt}
FAMILY
  {font-style:italic}
NAME
  {font-weight:bold}
```

6. Мәтіді құрылған бірінші тармағында тапсырмалар, және екінші жолмен құжаттың келесі нұсқаулық бойынша өңдеу:

```

<?xml version="1.0" encoding="windows-1251"?>
<?xml-stylesheet type="text/css" href="file_2.css"?>
<!--имя файла:fale_10.xml-->
<FILE_1>
  <STUDENT>

```

7. Пән аймағының белгілі бір нұсқасы бойынша ақпаратты ұсынатын XML құжатын жасаңыз. Жасалған құжат келесі талаптарға сәйкес болуы тиіс:

- құжаттар кемінде төрт элементтің салыну тереңдігі болуы тиіс;
- салынған құжат элементтерінің саны кемінде бес болуы тиіс;
- құжат элементтерінде мазмұн туралы түсініктеме болуы тиіс;
- құжатта символдық деректер мен қосалқы элементтер бар элементтер

болуы тиіс;

8.Құрылған XML құжатын пішімдейтін каскадты стильдер кестесін жасаңыз. Құрылған CSS-кесте келесі ережелерге сәйкес болуы тиіс:

- CSS-кесте контекстуалды да, рулық селекторларды қамтуы тиіс;
- еншілес элементтер CSS-ата-аналық элемент пішімін мұраға алуы керек;
- құрылған CSS кестесі басқа мәнерлер кестесін импорттауы керек;
- стиль кестесі STYLE атрибутын пайдалануды қамтуы тиіс;

жасалған стиль кестесі болуы керек:

- қаріп қасиеттері;
- фон қасиеті;
- мәтінді бөлу және туралау қасиеттері;
- мәтін аймағының қасиеттері;
- өрістер мен шекаралардың қасиеттері;
- жалған элементтер

9. Есепте XML-құжат кодын, css және құжатты кестелік ұсынудың скриншоты кестелерінің кодын ұсыну.

Жасалатын XML-құжаттардың пәндік аймақтарының нұсқалары:

1 нұсқа:

әдебиеттер тізімінің библиографиялық сипаттамасы

2 нұсқа:

видеотека фильмдерінің сипаттамасы

3 нұсқасы:

ұйым қызметкерлерінің тізімі

4-нұсқа:

ұялы телефон модельдерінің тізімі

5 нұсқасы:

факультет студенттерінің тізімі

6-нұсқа:

оқытылатын пәндер тізімі

Зертханалық жұмыс №23. XML синтаксисі

1. HTML-Table элементін XMLfile_1 құжатының деректерімен байланыстырыңыз.зертханалық жұмыста жасалған xml 1:

1. Notepad мәтіндік редакторында жаңа lab2 файлын жасаңыз.html:

```
<!--имя файла: lab2.html -->
<HTML>
<HEAD>
<TITLE>лабораторная работа 2</TITLE>
</HEAD>
<BODY>
<XML ID="dsostudent" SRC="file_1.xml"></XML>
<H2> Список студентов </H2>
<TABLE DATASRC="#dsostudent" BORDER="6" CELLPADDING="5" width="100%">
  <THEAD style="background-color: maroon">
    <TH> фамилия</TH>
    <TH> имя</TH>
    <TH> Год рождения</TH>
    <TH> группа</TH>
  </THEAD>
  <TR ALIGN="center">
    <TD><SPAN DATAFLD="FAMILY" STYLE="font-style:italic"> </SPAN> </TD>
    <TD><SPAN DATAFLD="NAME" > </SPAN> </TD>
    <TD><SPAN DATAFLD="YEAR"> </SPAN> </TD>
    <TD><SPAN DATAFLD="GROUP"> </SPAN> </TD>
  </TR>
</TABLE>
</BODY>
</HTML>
```

1. файлды file_1 орналасқан қалтаға сақтаңыз.xml және Internet Explorer 5 шолғышында ашыңыз;

2. әр элемент пен атрибуттың мәнін түсіндіріңіз.

2. Жаңа lab2_str файлын жасаңыз.HTML, онда file_1 құжат жазбаларының беттік шығуын жүзеге асыру.xml:

1. TABLE элементіне ID төлсипатын қосып, оған бірегей мәнді және DATAPAGESIZE төлсипатын қосыңыз:

2. беттің жоғарғы жағына төрт элементті қосыңыз ONCLICK атрибуты бар, оның мәні әр түйме үшін элемент әдістерінің бірімен анықталады:

3. жасалған құжат Internet Explorer 5-де келесідей көрінуі керек.

Список студентов

Первая страница	Последняя страница	Следующая страница	Предыдущая страница
Фамилия	Имя	Год рождения	Группа
Иванов	Сергей	1993	ИФ 87
Петрова	Галина	1992	ИФ 87

4. Түймесін басыңыз <Келесі бет> келесі екі жазбаны көрсету керек. Тышқанды басқа үш түйме бойынша басқанда қол қою әрекеті сол сияқты сәйкес келуі керек.

5. 3. Пәндік аймақ опцияларын пайдалана отырып, кемінде 15 жазбадан тұратын XML құжатын жасаңыз. TABLE HTML элементінің XML құжат деректерімен ілінісуін пайдаланып, браузердегі жазбаларды (бір бетте бес жазбадан) және парақтан өту түймешіктерін шекаралық шығуды жүзеге асырыңыз.

6. 4. Жасалатын XML-құжаттардың пәндік салаларының нұсқалары:

1 нұсқа: әдебиеттер тізімінің библиографиялық сипаттамасы

2 нұсқа: видеотека фильмдерінің сипаттамасы

- 3: нұсқа ұйым қызметкерлерінің тізімі
- 4 нұсқа: ұялы телефон модельдерінің тізімі
- 5 нұсқа: факультет студенттерінің тізімі
- 6 нұсқа:

Зертханалық жұмыс №24. DTD сызбалары – құжат түрін анықтау

XML құжатқа қосылатын мәндерді сипаттау. Бұдан әрі, осы дәрістің материалын баяндау үшін біз ақпараттық технологиялар бойынша кітаптар кітапханасын сипаттайтын төменде берілген XML құжатты пайдаланатын боламыз:

```
<?xml version="1.0" encoding="Windows-1251"?>
  <library>
    <books>
      <book          id="N1"          isbn="978-5-8459-1118-6"    year="2008"
issue="translation">
        <udk value="004.541.47"/>
        <authors>
          <author country="US">С. Граннеман</author>
        </authors>
        <title>Linux. Карманный справочник</title>
        <cover border="1" align="center" width="350" height="200"
          href="www.williamspublishing.com/covers/linux.gif"/>
        <publishing>Вильямс</publishing>
        <annotation>Область          применения          <scope>операционные
системы</scope></annotation>
      </book>
      <book          id="N2"          isbn="5-314-00189-6"    year="2000"
issue="translation">
        <udk value="681.3" />
        <authors>
          <author country="US">К. Уилмер</author>
          <author country="US">А. Хоумер</author>
        </authors>
        <title>Dynamic HTML. Справочник</title>
        <cover border="1" align="center" width="350" height="200"
          href="www.piter-press.ru/covers/dhtml.gif"/>
        <publishing>Питер</publishing>
        <annotation>Область          применения
<scope>Интернет</scope></annotation>
      </book>
      <book          id="N3"          isbn="978-5-9775-0104-0"    year="2007"
issue="original">
        <udk value="681.3.06"/>
        <authors>
          <author country="RU">Д.Н. Колисниченко</author>
        </authors>
        <title>Ubuntu Linux</title>
        <cover border="1" align="center" width="350" height="200">
```

```

        href="www.bhv.ru/covers/ubuntu.gif"/>
        <publishing>БХВ-Петербург</publishing>
        <annotation>Область применения ОС Linux</scope></annotation>
    </book>
</books>
</library>

```

1 XML құжатында DTD хабарландыру

DTD құру ережелерін зерттеуді бастамас бұрын, XML құжат DTD жариялайды деп қарастырайық. Бұл DTD зерттеу кезінде компьютерде жасалған мысалдарды сынап көруге мүмкіндік береді.

XML құжатқа қатысты DTD екі түрі болуы мүмкін: сыртқы және ішкі. Сондай-ақ, құжатта екі DTD түрін де бір уақытта пайдалануға болады.

XML құжатында DTD жариялау үшін құжаттың XML түбірлік элементінің алдында орналасуы тиіс DOCTYPE ұсынысы пайдаланылады, ал онда көрсетілген DTD атауы түбірлік элементтің атымен сәйкес келуі тиіс. DOCTYPE ұсынысы көрініс бар:

```
<!DOCTYPE _dtd орналасқан жері _dtd >
```

мұнда: имя_dtd-құжаттың XML түбірі элементінің атымен сәйкес келетін DTD сызбасының атауы;

DTD – түрі DTD түріне байланысты болатын DTD схемасының орналасу орнының сипаттамасы.

1.1 Сыртқы DTD

Сыртқы DTD кеңейтімі болуы тиіс бөлек файлда.dtd, мысалы, library.dtd. DOCTYPE ұсынысының синтаксисі сыртқы байланыс жағдайында келесі түрді қабылдайды:

```
<!DOCTYPE атым_dtd >
```

мұнда: URI-DTD схемасының орналасқан жерін көрсету.

SYSTEM және PUBLIC-келесі алгоритмде қолданылатын альтернативті қолданылатын кілт сөздер.

SYSTEM көрсеткен кезде, анализатор нақты көрсетілген URI бойынша DTD табу керек, мысалы:

```
<!DOCTYPE library SYSTEM "http://www.library.org/dtd/library.dtd">
```

PUBLIC көрсеткенде құжаттың XML өңдеушісіне "жақсы таныс" жария DTD қолданылады деп саналады. Мысалы, кітапхана жүйелері үшін Интернетке белгілі бір сайтта орналасқан DTD жария схемасы бар. Егер кітапханалық XML құжаттарымен жұмыс істеу үшін стандартты емес, мамандандырылған талдағыштар пайдаланылса, онда ол өзіне қажетті DTD сұлбасы қайда екенін өзі білуі мүмкін. Бұл жағдайда DTD схемасы бар файлдың дәл URI орналасуын көрсету міндетті емес, мысалы:

```
<!DOCTYPE library PUBLIC "world/library.dtd">
```

PUBLIC сипаттамасын файлдың нақты орналасуын DTD сызбасымен қоса беруге болады. Бұл жағдайда, егер талдағыш жария схеманы таппаса, ол нақты мекен-жай бойынша көрсетілген схеманы пайдаланатын болады, мысалы:

```
<!DOCTYPE library PUBLIC "world/library.dtd"
"http://www.library.org/dtd/library.dtd">
```

Зертханалық жұмыс №25. XML - XSD сызбасы

W3C консорциумы схемаларды сипаттаудың ең танымал тілдерін бір стандартқа біріктіре отырып, XML (XSD) сұлбаларын анықтау тілін ұсынды. Бұл

ретте көздеген негізгі мақсат — стандартты алу, оны кеңінен жүзеге асыруға болады және ол платформа-тәуелсіз.

Тіл XML Schema Definition Language, ол сондай-ақ XML Schema Language деп аталады, көп жағдайда сіз бұрын танысқан XDR тіліне ұқсас. XSD схемалары келесі міндеттерді шеше алады:

XML құжатында элементтерді аудару және құжатта тек жарияланған элементтердің болуын тексеру.

Құжат элементтерін түрлендіретін атрибуттарды жариялау және анықтау.

Элементтердің арасындағы бас-еншілес қатынастарды анықтау.

Элементтер мен атрибуттар үшін мазмұнның жай-күйі мен үлгілерін анықтау.

Деректер типтерін анықтау.

Әдепкі мәндерді орнату.

Кеңейту мүмкіндігі.

Атау кеңістігін пайдалануды қолдау.

XML схемасындағы түбірлік элемент схема құжаттарындағы барлық қалған элементтерді қамтитын Schema элементі болып табылады. Xmlns атрибутымен XSD схемасының түбірлік элементі аясында XSD схемасының элементтері мен атрибуттары бар XMLSchema атауларының кеңістігі анықталады.

```
<xsd: schema xmlns: xsd="http://www.w3.org/2001/XMLSchema"> >
```

Барлық XSD элементтері xsd префиксінен басталады:, ол схема данасының түбірлік элементінде жарияланған XSD есімдерінің кеңістігі үшін көрсетіледі.

XML-схема арқылы тексерілетін құжат, сондай-ақ атау кеңістігін жариялау болуы тиіс. Атау кеңістігі әрқашан құжат данасының түбірлік элементінде xmlns атрибуты арқылы көрсетіледі:

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

Бұл атау кеңістігі XML құжатына қосуға болатын xmlschema элементтері мен атрибуттарын қамтиды. Жалпы келісім бойынша xsi префиксі осы аттар кеңістігі үшін қолданылады және олардың қос нүктемен бөле отырып, аттар кеңістігіне тиесілі барлық элементтер мен атрибуттар атауларының басында қосылады.

Нақты схемаға сілтеме атрибутта келтіріледі

```
xsi:schemaLocation="http://kit.znu.edu.ua/scemes/имя\_файла.xsd"
```

XSD элементін және төлсипатын жариялау

Схеманы құру процесі екі қадамды қамтиды — элементтердің типтерін немесе атрибуттардың типтерін анықтау және жариялау. XML-құжаттың элементтері мен атрибуттары xsd:element және xsd:attribute схемаларының элементтерімен жарияланады. XML-құжат құрылымы xsd:simpleType және xsd:complexType схемаларының элементтерімен анықталады.

Элементтің негізгі хабарландыруы аты мен деректер түрінен тұрады

```
<xsd: element name=" имя_элемента " type="xsd:тип_данных"/>
```

XML құжаттарында пайдаланылатын XSD дескрипторлар екі санатқа бөлінеді- күрделі типтері мен қарапайым типтері. Күрделі типті элементтердің басқа элементтері болуы мүмкін, сондай-ақ белгілі бір атрибуттары бар; қарапайым типті элементтердің мұндай мүмкіндіктері жоқ.

Атрибут-басқа элементтерді қамтымайтын қарапайым түрдегі хабарландыру. Атрибутты хабарландыру элементті хабарландыру сияқты:

```
<xsd:attribute name=" имя_атрибута "type=" xsd: үлгі_данных" / >
```

Қарапайым деректер түрлері

Қарапайым үлгілердің екі негізгі санаты бар:

кірістірілген түрлері;

пайдаланушы анықтаған қарапайым түрлері.

XSD тілінің көп саны енгізілген қарапайым деректер типтері бар. Кірістірілген түрлері примитивті типтерді және туындыларды қамтиды. Қарапайым деректер түрлері басқа деректер түрлерінен алынбады. Мысалы, өзгермелі үтір бар сандар-басқа деректер түрлерінен алынбаған математикалық ұғым. Деректер типтері бар деректер типтерінің терминдерінде анықталған. Мысалы, бүтін сан - деректердің ондық түрінен алынған жеке жағдай.

Келесі кесте XML-сызбалар, деректер түріне және деректер түрін сипаттауға қолданылуы мүмкін аспектілер мәліметтерінің қарапайым түрлерінің тізімін ұсынады.

Деректер түрі	Аспектiлер	Сипаттамасы
string	length, pattern, maxLength, minLength, enumeration, whiteSpace	Символдық жолды бiлдiредi.
Boolean	pattern, whiteSpace	True немесе false болуы мүмкiн логикалық мәндi ұсынады.
decimal	enumeration, pattern, totalDigits, fractionDigits, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Еркiн санды бiлдiредi.
float	pattern, enumeration, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	32-биттiк Сан бiр дәлдiктегi құбылмалы үтiрмен берiледi.
double	pattern, enumeration, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	64-биттiк Сан екi есе дәлдiктегi құбылмалы үтiрмен берiледi.
duration	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Уақыт ұзақтығын ұсынады. Duration үшiн үлгi келесi- PnYnMnDtnHnMnS, мұнда nY жылдар санын бiлдiредi; nM- ай; nD-күн; T-күн және уақыт бөлгiшi; nH - сағат саны; nM - минут; nS-секунд.
dateTime	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Нақты уақытты ұсынады. DateTime үшiн үлгi келесi-CCY-mm-DDThh:mm:ss, мұнда CC Жүзжылдықты ұсынады; YY - жыл; MM - ай; DD - күн; T - күн мен уақытты бөлгiш; hh - сағат саны; mm - минут; SS - секунд. Қажет болса, секунд үлесiн көрсетуге болады. Мысалы, үлгiдегi жүздеген үлес: ss.ss
time	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Күннiң нақты уақытын ұсынады. Уақыт үлгiсi келесi- hh:mm:ss.SSS (секундтың үлестiк бөлiгi мiндеттi емес).
date	enumeration, pattern,	Күнтiзбелiк күндi ұсынады. Мұндай date үлгiсi - CCY-MM-

	minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	DD (мұнда уақытты білдіретін бөлік міндетті емес).
gYearMonth	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Нақты жылдың нақты айы (CCY-MM).
gYear	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Күнтізбелік жыл (CCYY) ұсынады.
gMonthDay	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Нақты айдың нақты күнін ұсынады (--MM-DD).
gDay	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Күнтізбелік күнді ұсынады (- - - DD).
gMonth	enumeration, pattern, minInclusive, minExclusive, maxInclusive, maxExclusive, whiteSpace	Күнтізбелік ай (--mm -) ұсынады.
hexBinary	length, pattern, maxLength, minLength, enumeration, whiteSpace	Еркін он алтылық-кодталған екілік ақпаратты ұсынады. HexBinary-төрт жұп он алтылық таңбадан тұратын белгіленген ұзындықтағы екілік октеттер жиынтығы. Мысалы, 0-9a-fA-F.
base64Binary	length, pattern, maxLength, minLength, enumeration, whiteSpace	Еркін Base64-кодталған екілік ақпаратты ұсынады. Base64Binary-белгіленген ұзындықтағы екілік октеттер жиынтығы.
anyURI	length, pattern, maxLength, minLength, enumeration, whiteSpace	RFC представляет URI, как определено в 2396. Значение AnyURI может быть абсолютно или сравнительно и может быть необязательным идентификатором фрагмента.
QName	length, enumeration, pattern, maxLength, minLength, whiteSpace	Құрамдас атауды ұсынады. Аты қос нүктемен бөлінген префикстен және жергілікті атаудан тұрады. Префикс және жергілікті атаулар NCNAME болуы керек. Префикс ат кеңістігін хабарландыру арқылы namespace URI сілтемесімен байланысты болуы тиіс.
NOTATION	length, enumeration, pattern, maxLength, minLength, whiteSpace	Белгілер жүйесі төлсипатының түрін ұсынады. QNamef жиынтығы.

Зертханалық жұмыс №26. XML атаулар кеңістігі

Аттар кеңістігі XML элементтердің атаулар қақтығысын болдырмайды.

Атаулардың қақтығыстары

XML элементтердің атаулары әзірлеуші анықтайды. Жиі бұл әртүрлі XML қосымшаларынан бірнеше XML құжаттарын бір мезгілде пайдалану кезінде атаулар қақтығысына себеп болады.

Келесі XML коды HTML кесте туралы ақпаратты қамтиды:

```
<table>
  <tr>
    <td>Яблоки</td>
    <td>Бананы</td>
  </tr>
</table>
```

Келесі XML коды үстел туралы ақпаратты қамтиды (жиһаз пәні), ол ағылшын. сондай-ақ table:

```
<table>
  <name>Африканский кофейный столик</name>
  <width>80</width>
  <length>120</length>
</table>
```

XML кодының осы екі бөлігі біріктірілсе, атаулар қақтығысы пайда болады. Екі құжат да әртүрлі мазмұн мен мәні бар болса да, <table> элементін қамтиды.

Пайдаланушы немесе XML қолданбасы осы айырмашылықтарды қалай өңдеу керектігін білмейді.

Префикс көмегімен атаулар қақтығысын шешу

XML-да атау қақтығысын элемент атының префиксі арқылы болдырмауға болады.

Келесі XML коды HTML кестесі және үстел туралы ақпаратты қамтиды:

```
<h:table>
  <h:tr>
    <h:td>Яблоки</h:td>
    <h:td>Бананы</h:td>
  </h:tr>
</h:table>

<f:table>
  <f:name>Африканский кофейный столик</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

Бұл мысалда атаулар қақтығысы болмайды, өйткені екі элемент <table> түрлі аттарға ие.

XML атрибуты xmlns

Сонымен қатар, бұл жағдайда, бір-бірімен байланысқан, бір-бірімен байланысқан, бір-бірімен байланысқан, бір-бірімен байланысқан, бір-бірімен байланысқан, бір-бірімен байланысқан.

Атау кеңістігі элементтің бастапқы тегіндегі xmlns атрибуты арқылы анықталады.

Атау кеңістігі декларациясының келесі синтаксисі бар-xmlns: префиксі="URI".

```
<root>
  <h:table xmlns:h="http://www.w3.org/TR/html4/">
    <h:tr>
      <h:td>Яблоки</h:td>
      <h:td>Бананы</h:td>
    </h:tr>
  </h:table>

  <f:table xmlns:f="http://www.w3schools.com/furniture">
    <f:name>Африканский кофейный столик</f:name>
    <f:width>80</f:width>
    <f:length>120</f:length>
  </f:table>
</root>
```

Бұл мысалда <table> тегіндегі xmlns төлсипаты h: және f префикстерін анықтайды: атаулардың біліктілік кеңістігі.

Аттардың кеңістігі белгілі бір элемент үшін анықталған кезде, оның барлық еншілес элементтері бірдей префикспен оның аттардың кеңістігімен байланыстырылады.

Атаулардың кеңістігі тікелей элементтің өзінде немесе XML құжатының түбірінде декларациялануы мүмкін:

```
<root xmlns:h="http://www.w3.org/TR/html4/"
xmlns:f="http://www.w3schools.com/furniture">
  <h:table>
    <h:tr>
      <h:td>Яблоки</h:td>
      <h:td>Бананы</h:td>
    </h:tr>
  </h:table>

  <f:table>
    <f:name>Африканский кофейный столик</f:name>
    <f:width>80</f:width>
    <f:length>120</f:length>
  </f:table>
</root>
```

Ескерту: атау кеңістігі URI кез келген ақпарат алу үшін парсермен пайдаланылмайды. Мұның барлық мақсаты-аттардың кеңістігіне бірегей атау беру.

Дегенмен, компаниялар жиі осы атау кеңістігі туралы ақпарат бар веб-бетке көрсеткіш ретінде атау кеңістігін пайдаланады.

Ресурстың сәйкестендірілген идентификаторы (URI)

Ресурстың сәйкестендірілген идентификаторы (URI)-интернет ресурсын анықтайтын символдық жол.

URI-дің ең жалпы түрінде интернетте домендік мекенжайды анықтайтын жалғыз ресурс көрсеткіші (URL) болып табылады. Басқа, URI — дің жеке түрі-ресурстың бірыңғай атауы (URN).

Біздің мысалдарда біз тек URL мекенжайын пайдаланамыз.

Әдетті атау кеңістігі

Әдепкі атау кеңістігін анықтау барлық еншілес элементтерде префикстерді пайдаланбауға мүмкіндік береді. Мұндай анықтамада келесі синтаксис бар:

```
xmlns="URI пространства имен"
```

Келесі XML коды HTML кестесі туралы ақпаратты қамтиды:

```
<table xmlns="http://www.w3.org/TR/html4/">
  <tr>
    <td>Яблоки</td>
    <td>Бананы</td>
  </tr>
</table>
```

Келесі XML коды жиһаз үстел туралы ақпаратты қамтиды:

```
<table xmlns="http://www.w3schools.com/furniture">
  <name>Африканский кофейный столик</name>
  <width>80</width>
  <length>120</length>
</table>
```

Аттар кеңістігін нақты пайдалану

XSLT-бұл XML құжаттарын HTML сияқты басқа пішімдерге түрлендіру үшін қолдануға болатын XML тілі.

Келесі XSLT құжатында элементтердің көпшілігі HTML тегтері екенін көре аласыз. HTML тегтері емес элементтерде xmlns атауларының кеңістігімен анықталған xsl префиксі бар: xsl="http://www.w3.org/1999/XSL/Transform":

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <body>
        <h2>Моя коллекция CD</h2>
        <table border="1">
          <tr>
            <th style="text-align:left">Title</th>
```

```

        <th style="text-align:left">Artist</th>
    </tr>
    <xsl:for-each select="catalog/cd">
        <tr>
            <td><xsl:value-of select="title"/></td>
            <td><xsl:value-of select="artist"/></td>
        </tr>
    </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Зертханалық жұмыс №27. XML сілтемелері – Xpath тілі

XPath дегеніміз не?

- XPath-құжаттың XML бөліктерін анықтау үшін арнайы тіл
- XPath қызметтері XML құжаттарын шарлау үшін маршруттық өрнектерді пайдаланады
- XPath қызметтері стандартты кітапхананы қамтиды
- XPath-XSLT-дегі басты элемент
- XPath XQuery, XPointer және XLink-да қолданылады
- XPath маршруттық өрнектер

XPath XML құжатындағы түйіндерді немесе тораптарды таңдау үшін маршрут өрнектерін пайдаланады. Бұл маршруттық өрнектер дәстүрлі файлдық жүйелермен жұмыс істеу кезінде көруге болатын өрнектерге ұқсас.

Қазіргі уақытта XPath өрнектерін JavaScript, Java, XML схемалары, PHP, Python, C және C++, сондай-ақ басқа да бағдарламалау тілдерінде қолдануға болады. XPath XSLT-да қолданылады

XPath XSLT стандартының негізгі құрамдас бөлігі болып табылады. XPath білімсіз XSLT құжаттарын жасау мүмкін емес.

Мысал XPath

XPath демонстрациясы үшін келесі XML құжатын қолданамыз:

```

<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
</bookstore>

```

```

</book>
<book category="WEB">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author>
  <author>Per Bothner</author>
  <author>Kurt Cagle</author>
  <author>James Linn</author>
  <author>Vaidyanathan Nagarajan</author>
  <year>2003</year>
  <price>49.99</price>
</book>
<book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>

```

Төмендегі кестеде кейбір XPath өрнектері және олардың жұмыс нәтижесі көрсетілген:

XPath кіріспе	Нәтижесі
/bookstore/book[1]	Бірінші book элементін таңдайды, ол bookstore элементінің ұрпағы болып табылады
/bookstore/book[last()]	Bookstore элементінің ұрпағы болып табылатын соңғы book элементін таңдайды
/bookstore/book[last()-1]	Bookstore элементінің ұрпағы болып табылатын соңғы book элементін таңдайды
/bookstore/book[position()<3]	Bookstore элементінің ұрпақтары болып табылатын алғашқы екі кітап элементін таңдайды
//title[@lang]	Lang атрибуты бар барлық title элементтерін таңдайды
//title[@lang='en']	"En" мәні бар lang атрибуты бар барлық title элементтерін таңдайды
/bookstore/book[price>35.00]	Book элементінің ұрпағы болып табылатын және 35.00-ден артық баға элементін қамтитын барлық book элементтерін таңдайды
/bookstore/book[price>35.00]/title	35.00-ден асатын price элементі бар bookstore элементінің барлық title элементтерін таңдайды

Зертханалық жұмыс №28. XML — DOM құжаттың объектілік моделі

XML файлын жүктеу

Келесі мысалда XML құжаты XML DOM нысанына сәйкес келеді, одан кейін Javascript көмегімен кейбір ақпарат алынады:

```
<html>
<body>
  <h1>Заметка</h1>
  <div>
    <b>Кому:</b> <span id="to"></span><br />
    <b>От:</b> <span id="from"></span><br />
    <b>Сообщение:</b> <span id="message"></span>
  </div>

  <script>
    if (window.XMLHttpRequest)
    { // для IE7+, Firefox, Chrome, Opera, Safari
      xmlhttp = new XMLHttpRequest();
    }
    else
    { // для IE6, IE5
      xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }

    xmlhttp.open("GET","note.xml",false);
    xmlhttp.send();
    xmlDoc = xmlhttp.responseXML;

    document.getElementById("to").innerHTML =
xmlDoc.getElementsByTagName("to")[0].childNodes[0].nodeValue;
    document.getElementById("from").innerHTML =
xmlDoc.getElementsByTagName("from")[0].childNodes[0].nodeValue;
    document.getElementById("message").innerHTML =
xmlDoc.getElementsByTagName("body")[0].childNodes[0].nodeValue;
  </script>
</body>
</html>
```

Маңызды ескерту!

"Tove" мәтінін <to> файл XML элементінен шығару үшін келесі өрнек қолданылады:

```
getElementsByTagName("to")[0].childNodes[0].nodeValue
```

XML файлында тек бір элемент болса да, массив индексін пайдалану керек [0]. Бұл getElementByTagName() әдісі массивті қайтарады.

11 XML жолы жүктеу

Келесі мысалда XML жолы XML DOM нысанына кіреді, одан кейін Javascript көмегімен кейбір ақпарат алынады:

```

<html>
<body>
  <h1>Заметка</h1>
  <div>
    <b>Кому:</b> <span id="to"></span><br />
    <b>От:</b> <span id="from"></span><br />
    <b>Сообщение:</b> <span id="message"></span>
  </div>

  <script>
    txt = "<note>";
    txt = txt + "<to>Тове</to>";
    txt = txt + "<from>Jani</from>";
    txt = txt + "<heading>Напоминание</heading>";
    txt = txt + "<body>Не забудь обо мне в эти выходные!</body>";
    txt = txt + "</note>";

    if (window.DOMParser)
    {
      parser = new DOMParser();
      xmlDoc = parser.parseFromString(txt,"text/xml");
    }
    else // Internet Explorer
    {
      xmlDoc = new ActiveXObject("Microsoft.XMLDOM");
      xmlDoc.async = false;
      xmlDoc.loadXML(txt);
    }

    document.getElementById("to").innerHTML =
xmlDoc.getElementsByTagName("to")[0].childNodes[0].nodeValue;
    document.getElementById("from").innerHTML =
xmlDoc.getElementsByTagName("from")[0].childNodes[0].nodeValue;
    document.getElementById("message").innerHTML =
xmlDoc.getElementsByTagName("body")[0].childNodes[0].nodeValue;
  </script>
</body>
</html>

```

HTML үлгісі XML

Келесі мысалда біз XML файлын ашамыз, содан кейін циклде CD әрбір элементін айналып, HTML кестесіндегі ARTIST және TITLE элементтерінің мәнін көрсетеміз:

```

<html>
<body>
  <script>
    if (window.XMLHttpRequest)
    { // для IE7+, Firefox, Chrome, Opera, Safari
      xmlhttp = new XMLHttpRequest();

```

```

    }
    else
    { // для IE6, IE5
      xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }

    xmlhttp.open("GET","cd_catalog.xml",false);
    xmlhttp.send();
    xmlDoc = xmlhttp.responseXML;

    document.write("<table border='1'>");
    var x = xmlDoc.getElementsByTagName("CD");
    for (i = 0; i < x.length; i++)
    {
      document.write("<tr><td>");

document.write(x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue);
      document.write("</td><td>");

document.write(x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue);
      document.write("</td></tr>");
    }
    document.write("</table>");
  </script>
</body>
</html>

```

Зертханалық жұмыс №29. XML үшін Simple API.

SimpleContent элементі күрделі мәтін түріне немесе элементтері жоқ қарапайым түрге арналған кеңейтімдерді немесе шектеулерді анықтайды.

- Ата-ана элементтері: complexType
- Элемент синтаксисі

```

<simpleContent
  id=идентификатор
  любые атрибуты
  >

  (annotation?, (restriction | extension))

</simpleContent>

```

(Белгі ? элемент simpleContent элементінің ішінде нөлдік немесе бір рет пайда болуы мүмкін екенін көрсетеді.)

- Атрибуты элемента

Атрибут	Сипаттамасы
id	Міндетті емес. Элемент үшін бірегей идентификаторды

Атрибут	Сипаттамасы
	анықтайды
Кез келген атрибут	Міндетті емес. Схема атауларының кеңістігінен тыс кез келген басқа атрибуттар

Қолдану мысалы

Төменде тек мәтінді қамтитын элементтің XML (<shoe size>) мысалы көрсетілген:

```
<shoesize country="france">35</shoesize>
```

Келесі мысалда мазмұн ретінде бүтін мәндерді қабылдай алатын, сондай-ақ "country" атрибутын анықтай алатын "shoesize" элементі көрсетіледі:

```
<xs:element name="shoesize">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:integer">
        <xs:attribute name="country" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

SimpleType қарапайым түрін мәлімдейді және атрибуттар немесе мәтіндік элементтер мәндері туралы шектеулер мен ақпаратты анықтайды.

- Ата-аналық элементтер: attribute, element, list, restriction, schema, union
- Элемент синтаксисі

```
<simpleType
  id=идентификатор
  name=NCName
  любые атрибуты
  >
  (annotation?, (restriction | list | union))
</simpleType>
```

(Белгі ? элемент simpletype элементінің ішінде нөлдік немесе бір рет пайда болуы мүмкін екенін көрсетеді.)

Элемент атрибуттары

Атрибут	Сипаттамасы
id	Міндетті емес. Элемент үшін бірегей идентификаторды анықтайды

Атрибут	Сипаттамасы
name	Элемент атауын анықтайды. SimpleType элементі schema элементіне қатысты еншілес болса, бұл атрибут міндетті болып табылады, керісінше оны пайдалану мүмкін емес
Кез келген атрибуты	Міндетті емес. Схема атауларының кеңістігінен тыс кез келген басқа атрибуттар

Қолдану мысалы

Келесі мысалда қарапайым түрдегі "age" элементі анықталады. Бұл элементтің мәні 0 немесе 100-ден кем болмауы мүмкін:

```
<xs:element name="age">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Зертханалық жұмыс №30. XML деректерін HTML элементтерімен байланыстыру

Деректерді бір жазба арқылы байланыстыруды пайдалану

Деректерді бір жазбадан байланыстыру кесте болып табылмайтын және байланыстырылған кестеге енгізілмеген HTML элементтері үшін қолданылады. HTML-элемент-мысалы, SPAN, BUTTON немесе LABEL - жеке XML өрісімен байланысады. Осыдан кейін HTML элементі автоматты түрде XML өрісінің мазмұнын көрсетеді, ол байланысқан (байланысқан). Мысалы, келесі HTML-SPAN элементі dsoBoofc деректер фрагменті арқылы қатынайтын XML құжатының TITLE өрісіне байланысты:

```
<SPAN DATASRC="#dsoBook" DATAFLD="TITLE"></SPAN>
```

HTML элементінің кесте сияқты бірнеше бөліктері жоқ болғандықтан, ол тек бір рет жазу үшін өрістің мәнін көрсете алады. Бір жазба бойынша деректерді байланыстыруды пайдалану үшін, XML-до-кумент қарапайым жазба жиынтығы ретінде ұйымдастырылуы керек. (Осы тараудың басында "жазбалардың қарапайым жиынтығын көрсету үшін бір HTML кестесін пайдалану" бөлімін қараңыз.)

8-8 Листингте SPAN бөлек элементін қарастырылатын құжаттың әрбір өрісімен (TITLE, AUTHOR, BINDING, PAGES, PRICE) байланыстыратын HTML-бет ұсынылған.

Листинг 8-8

Book.htm

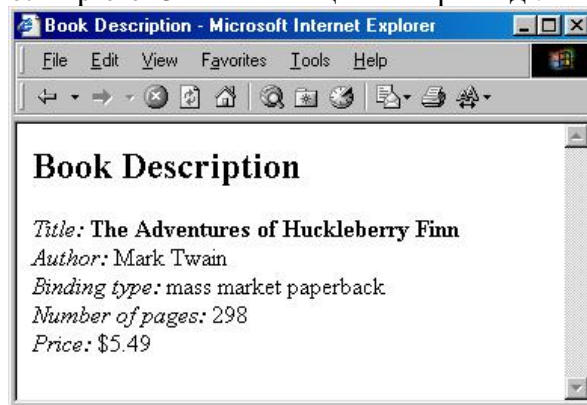
```
<!-- Имя файла: Book.htm -->
<HTML>
<HEAD>
<TITLE>Book Description</TITLE>
</HEAD>
<BODY>
<XML ID="dsoBook" SRC="Book.xml"></XML>
```

```

<H2>Book Description</H2>
<SPAN STYLE="font-style:italic">Title:</SPAN>
<SPAN STYLE="font-weight:bold" DATASRC="#dsoBook"
DATAFLD="TITIE"></SPAN>
<BR>
<SPAN STYLE="font-style:italic">Author: </SPAN>
<SPAN DATASRC="#dsoBook" DATAFLD="AUTHOR"></SPAN>
<BR>
<SPAN STYLE="font-style:italic">Binding type:</SPAN>
<SPAN DATASRC="#dsoBook" DATAFLD="BINDING"></SPAN>
<BR>
<SPAN STYLE="font-style:italic">Number of pages:</SPAN>
<SPAN DATASRC="#dsoBook" DATAFLD="PAGES"></SPAN>
<BR>
<SPAN STYLE="font-style:italic">Price:</SPAN>
<SPAN DATASRC="#dsoBook" DATAFLD="PRICE"></SPAN>
</BODY>
</HTML>

```

Келесі суретте Internet Explorer 5 осы бетті қалай көрсетеді.



Жазбалар арасында жылжыту

Егер XML-құжатта біреуден артық жазба болса (әдеттегідей), деректерді жазбалар бойынша байланыстыру біршама күрделі болады, себебі HTML-элемент тек бір жазбаны көрсете алады. Қазіргі кезде көрсетілетін жазба ағымдағы жазба деп аталады. (Деректерді бір жазба бойынша байланыстыру кейде ағымдағы жазба бойынша байланыстыру деп аталады.) Бастапқыда ағымдағы құжатта бірінші жазба болып табылады.

Мысалы, төменде келтірілген код файлдың басына жеткен жағдайда батырманы басқанда, Бірінші жазбаны тез көрсету керек:

```

<BUTTONONCLICK="dsolInventory.recordset.movePrevious();
if (dsolInventory.recordset.BOF)
dsolInventory.recordset.moveNext()">
Back
</BUTTON>

```

Следующий код проверяет достижение конца файла:

```

<BUTTONONCLICK="dsolInventory.recordset.moveNext();
if (dsolInventory.recordset.EOF)
dsolInventory.recordset.movePrevious()">
Forward
</BUTTON>

```

ONCLICK атрибутын (немесе ONMOUSEOVER сияқты оқиғаларға қатысты басқа атрибуттарды) сценарий кодының тұтас блогын тағайындауыңызға болады. Бұл

мысалдарда код Microsoft JScript тілінде жазылған. Бұдан әрі осы тарауда сіз нұсқаулардың көп санын қамтуға мүмкіндік беретін сценарий кодтарының тәуелсіз үзінділерін қалай жазуға болатынын білесіз.

Келесі жаттығуда сіз бір рет 8-3 листингінен XML құжатын көрсететін HTML бетін жасайсыз. Бет бірінші, алдыңғы, кейінгі және соңғы жазбаларға өту үшін түймелерді қамтиды.

Құжатты жеке жазбалар бойынша көрсетіңіз

1. Мәтіндік редакторда жаңа, бос мәтіндік файлды ашыңыз және 8-9 листингінде ұсынылған HTML бетінің мазмұнын енгізіңіз. (Осы листинг көшірмесі Inventory Single файлында CD-ROM кітабына қоса берілген.htm.) Назар аударыңыз, бет Inventory Big құжатымен байланысты деректердің үзіндісін қамтиды.16 жазбадан тұратын xml. (8-3 листингі мен CD-ROM файлына қараңыз.)

2. Бетті қатты дискідегі Inventorysingle файл атауын беру үшін мәтіндік редактордың сақтау (Сақтау) пәрменін пайдаланыңыз.htm.

Листинг 8-9

Inventory Single.htm

```
<!-- Имя файла: Inventory Single.htm -->
<HTML>
<HEAD>
<TITLE>Book Inventory</TITLE>
</HEAD>
<BODY>
<XML ID="dsolInventory" SRC="Inventory Big.xml"></XML>
<H2>Book Description</H2>
<SPAN STYLE="font-style:italic">Title:</SPAN>
<SPAN DATASRC="#dsolInventory" DATAFLD="TITLE"
STYLE="font-weight:bold"></SPAN>
<BR>
<SPAN STYLE="font-style:italic">Author:</SPAN>
<SPAN DATASRC="#dsolInventory" DATAFLD="AUTHOR"></SPAN>
<BR>
<SPAN STYLE="font-style:italic">Binding type:</SPAN>
<SPAN DATASRC="#dsolInventory" DATAFLD="BINDING"></SPAN>
<BR>
<SPAN STYLE="font-style:italic">Number of pages:</SPAN>
<SPAN DATASRC="#dsolInventory" DATAFLD="PAGES"></SPAN >
<BR>
<SPAN STYLE="font-style:italic">Price: </SPAN>
<SPAN DATASRC="#dsolInventory" DATAFLD="PRICE"></SPAN>
<HR>
<BUTTON ONCLICK="dsolInventory.recordset.moveFirst()">
|&lt; First
</BUTTON>
<BUTTON ONCLICK="dsolInventory.recordset.movePrevious();"
if (dsolInventory.recordsqf.EOF)
dsolInventory.recordset.moveNext()">
&lt; Back
</BUTTON>
<BUTTON ONCLICK="dsolInventory.recordset.moveNext();"
if (dsolInventory.recordset.EOF)
dsolInventory.recordset.movePrevious()">
Forward &gt;
```

```
</BUTTON>  
<BUTTONONCLICK="dsInventory.recordset.moveLast()">  
Last &gt;|  
</BUTTON>  
</BODY>  
</HTML>
```


3 бөлім. PHP бағдарламалау.

Зертханалық жұмыс №31. PHP-ге кіріспе

PHP-өз атауын бұрыннан өзгерткен сценарийлерді жасау тілі. Себебі PHP-бұл Personal Home Page сөздерінен аббревиатурасы. PHP-тің бірінші нұсқасы 1994 жылы Расмус Лордорфпен құрылды және Web-бетке келушілерді қадағалауға арналған құралдар жиынтығы болды. Уақыт өте келе PHP құралдар жиынтығынан толық бағдарламалау тіліне айналды, ал оның атауы PHP HyperText Preprocessor (PHP гипермәтінінің прпроцессоры) рекурсивті білім беру ретінде өзгертілді.

PHP-сценарийлердің серверлік тілі. HTML мәтініне енгізілген PHP конструкциялары әрбір бетке кірген кезде сервермен орындалады. Оларды өңдеу нәтижесі қарапайым HTML мәтінімен бірге браузерге беріледі.

Қазіргі уақытта PHP-дің негізгі нұсқасы жетінші болып табылады.

PHP екі негізгі бәсекелесі бар: active Server Pages (ASP) Microsoft және ColdFusion Allaire компаниясының. Олармен салыстырғанда PHP бірқатар артықшылықтарға ие, оның ішінде:

- Жоғары өнімділік. PHP бағдарламалары ASP қарағанда жылдам жұмыс істейді.
- * Функционалдық. PHP-бағдарламаны әзірлеуді Web-бетті әзірлеуден бөлуге болады, бұл программист пен дизайнер өмірін жеңілдетеді.
- * Баға. PHP мүлдем тегін.
- Қолдану қарапайымдылығы. Кең таралған тілдерде бағдарламалау тәжірибесі бар PHP синтаксисін жақсы таныс табады.
- * Төзімділік. Бір PHP кодын NT ортасында да, UNIX платформаларында да қолдануға болады.

* PHP бағдарламасын құрудың жалпы ережелері

PHP-бағдарламалар қарапайым мәтіннен тұрады, сондықтан оларды кез келген мәтіндік редакторда теруге болады. Танымал HTML редакторлары PHP бағдарламаларын өңдеу үшін кіріктірілген қолдау бар.

PHP 4-дегі әдепкі PHP-бағдарламалардың файлдарын кеңейту.php. Бұл кеңейтімнің негізінде сервер файлы PHP-бағдарлама ретінде таниды және интерпретаторды іске қосады.

PHP-бағдарлама әдеттегі HTML мәтінінен бөлінуі керек. PHP кодының төрт стилі бар:

Стиль	Ашу тегі	Жабу тег
Қысқартылған	<?	?>
XML (стандартты)	<?php	?>
ASP	<%	%>
SCRIPT (программды)	<SCRIPT LANGUAGE="php">	</SCRIPT>

Аталған тегтерден Тек Стандартты және бағдарламалық жасақтама PHP конфигурациясында жұмыс істейді.

HTML мәтінінен және PHP кодынан тұратын қарапайым аралас құжат:

Мысал 1

```
<html>
<head>
<title>Документ, состоящий из HTML-текста и PHP-кода</title>
```

```
</head>
<body>
<?php
echo "А вот и PHP!<br>";
?>
Всё очень просто!
</body>
</html>
```

Мұнда бір PHP операторы echo бар. Бұл оператор " бірақ PHP!
 " сервер жасайтын HTML бетіне. Бұл ретте,
 тег өз ісін жасайды, яғни жаңа жолға көшуді жүзеге асырады.

1 мысалдың нәтижесі:

Бірақ PHP!
Барлығы өте оңай!
Операторларды бөлу үшін (Си-ға ұқсас) үтірлі нүкте қолданылады.
PHP бағдарламасында пікірлер үш стиль болуы мүмкін:

```
/* Көпжолды түсініктеме
классикалық Си стилінде */
// C++стиліндегі бір жолдық түсініктеме
# Perl стиліндегі бір жолдық түсініктеме
```

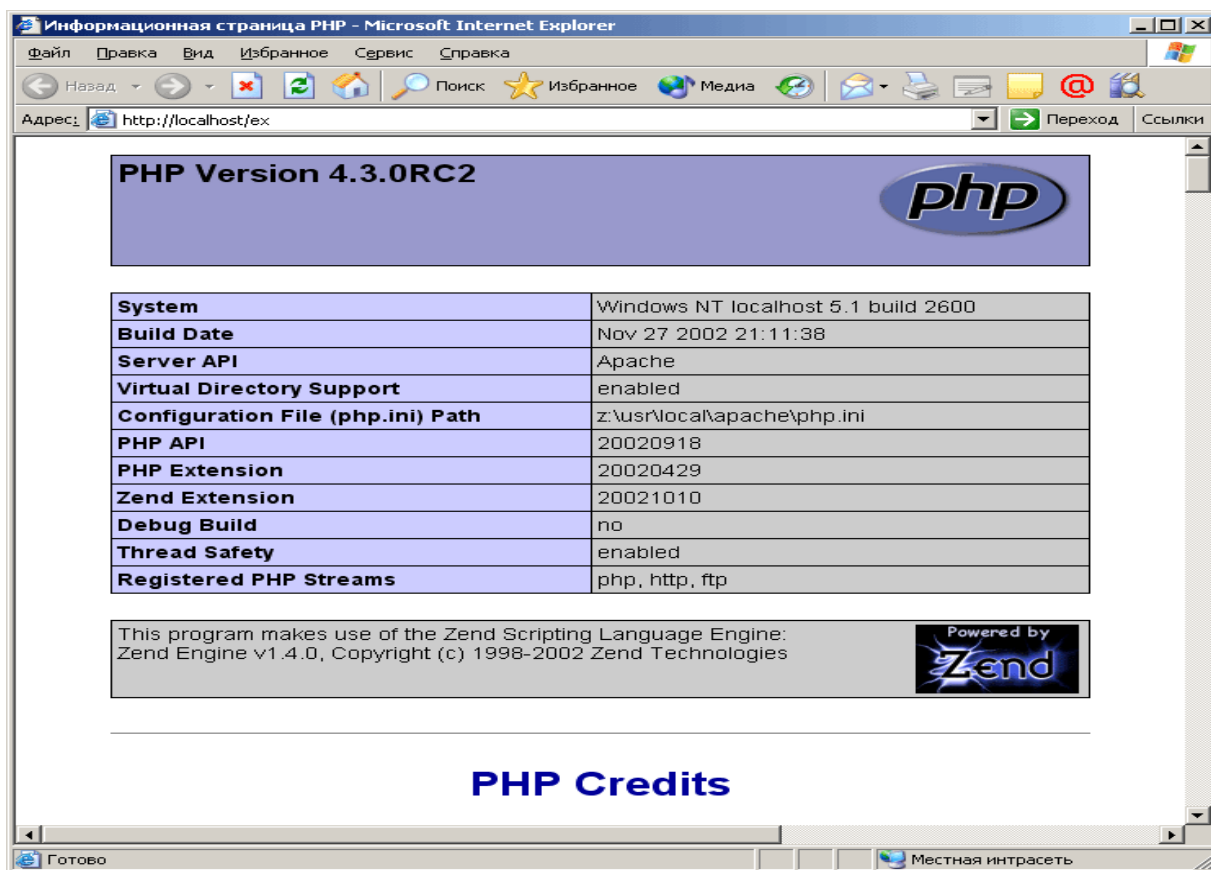
Бір құжатта HTML-мәтінді және PHP-операторлардың блоктарын кез келген уақытта ауыстыруға болады. Бұл жағдайда бірінші блокта анықталған барлық айнымалылар, функциялар мен сыныптар келесі блоктарда қол жетімді болады.

1 мысалында PHP-ортасының параметрлерінің тізімін беретін кірістірілген phpinfo () функциясына Echo функциясын шақыруды ауыстырамыз.

Мысал 2

```
<html>
<head>
<title>Информационная страница PHP</title>
</head>
<body>
<?php
    phpinfo();
?>
</body>
</html>
```

2 мысалдың нәтижесі:



Берілген анықтама PHP ортасы туралы көптеген ақпаратты қамтиды (оң жақта жүгірткіге назар аударыңыз). `Phpinfo ()` функциясы құрылған PHP-бағдарламалардың мінез-құлқын диагностикалауда өте пайдалы.

Зертханалық жұмыс №32. Негізгі синтаксис. Деректер типі. Айнымалылар. Өрнектер.

Тұрақтылар

Тұрақты PHP-бағдарламасында `define ()` функциясы арқылы анықталады. Мысалы:

```
define("PI", 3.1415927);
```

Константты анықтағаннан кейін өзгертуге болмайды. Константа атымен әдетте тек бас әріптер қолданылады.

PHP бірқатар алдын ала анықталған тұрақтарға ие. Мысалы:

- `_FILE_` қазіргі уақытта PHP оқитын толық жолды қоса алғанда, файл атауын қамтиды
- `_LINE_` осы файлдың жол нөмірін қамтиды
- `__DIR_` файлдың жолын ғана ұсынады
- `__CLASS_` ағымдағы сынып атауын ұсынады
- `_FUNCTION_` ағымдағы функция атауын көрсетеді
- * `_METHOD_` ағымдағы әдіс атауын ұсынады
- `__NAMESPACE_` ағымдағы атау кеңістігі атауын ұсынады

Айнымалылар

Имя любой переменной в PHP начинается со знака `$`. Имена айнымалы таңбалар регистріне сезімтал.

Айнымалы түрі арнайы сұраудың қажеті жоқ. Айнымалының нақты түрі бағдарламаны орындау барысында белгіленеді және өзгереді.

PHP сегіз деректер түрін қолдайды:

логикалық (true немесе false мәндерін қабылдайды);

* бүтін сан;

* жылжымалы нүктелі заттық Сан;

* жол;

* нысан;

* массив;

* ресурс (арнайы түрі);

• null (арнайы түрі).

Айнымалы түрін `gettype ()` көмегімен тексеруге болады.

Мысал 1

```
<html>
<head>
  <title>Автоматическое изменение и проверка типа переменной</title>
</head>
<body>
<?php
$t = "0";
echo "t = $t -- ".gettype($t)."<br>";
$t += 2;
echo "t = $t -- ".gettype($t)."<br>";
$t = $t + 3.5;
echo "t = $t -- ".gettype($t)."<br>";
$t = 5 + "5 поросят";
echo "t = $t -- ".gettype($t)."<br>";
$t = 5.0 + "5 поросят";
echo "t = $t -- ".gettype($t);
?>
</body>
</html>
```

1 мысалдың нәтижесі:

t = 0 -- string

t = 2 -- integer

t = 5.5 -- double

t = 10 -- integer

t = 10 -- double

Айнымалының түрін `settype ()` функциясы арқылы өзгертуге болады.

Мысал 2

```
<html>
<head>
  <title>Изменение типа переменной</title>
</head>
<body>
<?php
$t = 3.14;
echo "t = $t -- ".gettype($t)."<br>";
```

```

settype($t,"string");
echo "t = $t -- ".gettype($t)."<br>";
settype($t,"integer");
echo "t = $t -- ".gettype($t)."<br>";
settype($t,"double");
echo "t = $t -- ".gettype($t)."<br>";
settype($t,"boolean");
echo "t = $t -- ".gettype($t);
?>
</body>
</html>

```

2 мысалдың нәтижесі:

```

t = 3.14 -- double
t = 3.14 -- string
t = 3 -- integer
t = 3 -- double
t = 1 -- boolean

```

Айнымалы түрін келтіру операторлар арқылы жүзеге асырылады:

- (bool) - логикалық түрге;
- * (int) - мақсатты санға;
- (double) - заттық Сан;
- * (string) - жолға;
- (array) - массивке;
- (object) - объектіге.

Settype() функциясы арқылы түрдің өзгеруінен айырмашылығы-келтіру операторы айнымалыны өзгеріссіз қалдырып, жаңа түрдің уақытша көшірмесін жасайды.

Мысал 3

```

<html>
<head>
<title>Приведение типа переменной</title>
</head>
<body>
<?php
$t = 3.14;
$tmp = (string) $t;
echo "tmp = $tmp -- ".gettype($tmp)."<br>";
$tmp = (int) $t;
echo "tmp = $tmp -- ".gettype($tmp)."<br>";
$tmp = (double) $t;
echo "tmp = $tmp -- ".gettype($tmp)."<br>";
$tmp = (bool) $t;
echo "tmp = $tmp -- ".gettype($tmp)."<br>";
echo "t = $t -- ".gettype($t);
?>
</body>
</html>

```

3 мысалдың нәтижесі:

```
tmp = 3.14 -- string
tmp = 3 -- integer
tmp = 3.14 -- double
tmp = 1 -- boolean
t = 3.14 -- double
```

Кейде бағдарламаның логикасын жеңілдету үшін айнымалы айнымалылардың атауларын пайдалану ыңғайлы. PHP динамикалық айнымалылар түрінде мұндай мүмкіндікті ұсынады. Динамикалық деп аталады айнымалы, оның аты онда сақталады.

Мысал 4

```
<html>
<head>
<title>Использование динамической переменной</title>
</head>
<body>
<?php
$t = "Всеm"; // переменной t присваиваем значение "Всеm"
$$t = "привет!"; // Переменной "Всеm" присваиваем значение "привет!"
echo "$$t ".$$t."<br>";
echo "$t ${t}<br>";
echo "$t $Всеm";
?>
</body>
</html>
```

4 мысалдың нәтижесі:

```
$Всеm привет!
Всеm привет!
Всеm привет!
```

PHP-де әр түрлі атауларды пайдалана отырып, бір айнымалыға жүгінуге болады. Мұны іске асыру үшін сілтемелер қолданылады. Сілтемелер екі немесе одан да көп айнымалыларды бірдей жад аймағына сілтеме жасауға мүмкіндік береді.

Мысал 5

```
<html>
<head>
<title>Создание ссылки на переменную</title>
</head>
<body>
<?php
$t1 = 96;
$t2 = &$t1;
echo "t1 = $t1; t2 = $t2<br>";
$t1 = 315;
echo "t1 = $t1; t2 = $t2";
?>
</body>
```

</html>

5 мысалдың нәтижесі:

$t1 = 96; t2 = 96$
 $t1 = 315; t2 = 315$

Операторлар

PHP операторлары Си тілінің белгілі операторларына ұқсайды.

Әмбебап операторлар

-	Белгінің қарама-қарсы
!	Толықтыру. Логикалық айнымалы мәндерді реверсиялау үшін қолданылады
++	Айнымалы мәnniң ұлғаюы. Префикс және жұрнақ ретінде қолданылуы мүмкін
--	Айнымалы мәnniң азаюы. Префикс және жұрнақ ретінде қолданылуы мүмкін

Арифметикалық операторлар

-	Алу
+	Қосу
*	Көбейту
/	Бөлу
%	Бөлінуден қалған қалдық

Конкатенация операторы

Оператор конкатенации "." оң жолды солға қосады.

Мысал 6

```
<html>
<head>
<title>Конкатенация строк</title>
</head>
<body>
<?php
$a = "Всеm";
$b = $a." привет!";
echo $b;
?>
</body>
</html>
```

6 мысалдың нәтижесі:

Всем привет!

Конкатенация операторы кез келген түрдегі операндаларды жол ретінде өңдейді. Оның орындалу нәтижесі әрқашан жол.

Зертханалық жұмыс №33. Негізгі синтаксис. Деректер типі. Айнымалылар. Өрнектер.

Тағайындау операторлары

=	Біліктіліг
+=	Қосу (\$n += 777; ұқсас \$n = \$n + 777;)
-=	Азайту (\$n -= 777; ұқсас \$n = \$n-777;)
*=	Көбейту
/=	Бөлу
%=	Бөлінуден қалған қалдық
.=	Конкатенация (\$n .= "777"; ұқсас \$n = \$n. " 777";)

Бит операторлары

Бит операторлары бүтін сандардың жеке биттерін өзгертуге мүмкіндік береді.

&	И
	ИЛИ
^	Исключаящее ИЛИ
~	Инверсия
>>	Сдвиг вправо
<<	Сдвиг влево

Салыстыру операторлары

>	Көбірек (екіншіден гөрі бірінші мән Көп ма?)
>=	Көп немесе тең (Бірінші мән екіншісінен кем емес пе?)
<	Аз (екінші қарағанда бірінші мән аз ба?)
<=	Аз немесе тең (Бірінші мән екіншісінен артық емес пе?)
==	Тең (екі айнымалының мәндері тең бе?)
===	Бірдей (екі айнымалының мәндері мен түрлері бірдей ме?)
!= , <>	Тең емес (екі айнымалының мәндері тең емес пе?)
!==	Бірдей емес (екі айнымалы деректердің мәндері немесе түрлері бірдей емес пе?)

Логикалық операторлар

Логикалық операторлар сандық емес, логикалық мәндермен жұмыс істейтін биттен ерекшеленеді: TRUE және FALSE.

and	И
or	ИЛИ
xor	Исключающее ИЛИ
!	Инверсия
>>	Сдвиг вправо
<<	Сдвиг влево
&&	И
	ИЛИ

Логикалық операторлар "және" және "немесе" екі форматқа ие. Бұл синонимдер емес. Себебі or операторы || қарағанда төмен, ал and - &қарағанда төмен басымдыққа ие. Осылайша, күрделі шартты өрнектерді құру кезінде жақшасыз жасай аласыз. Алайда, мұндай тәсілі нұсқаулар тәртібін жеңілдету және запутаться.

10.1.1. Айнымалы мазмұнды тексеру

Кейде айнымалының бар-жоғын немесе қандай маңызды екенін тексеру керек. Төменде осындай әрекеттерді орындауға мүмкіндік беретін функциялар бар.

`isset (ісе ісе імі_ жүкті) #ақиқат, егер айнымалы мән берілместен жарияланса.`

`empty (...айнымалы) #ақиқат, егер айнымалы мәні нөлге немесе бос жолға тең болса, не айнымалы жарияланбаса.`

PHP сондай-ақ айнымалы түрін тексеруге мүмкіндік береді. Мысалы, айнымалы бүтін екенін тексеру үшін функцияны пайдалану керек

`is_int($number)`

`$Number` айнымалы `integer` түрі болса, бұл функцияның нәтижесі `TRUE` болып табылады. Мұндай функцияларды қарастырайық.

`is_array ($var2)` айнымалының `$var2` массив екенін тексереді.

`is_float ($number)` айнымалы нүкте `$number` Сан екенін тексереді.

`is_null ($var)` айнымалының мәні `$var1` нөлге тең екенін тексереді

`is_numeric ($string)` айнымалы `$ string` сандық жол екенін тексереді.

`is_string ($string)` айнымалы `$string` жол екенін тексереді.

Кері шарттарды тексеру үшін леп белгісін пайдалану керек (!). Мысалы, келесі өрнекті өндегенде, айнымалы жарияланбаса, `TRUE` мәні алынады:

`! isset (...жүкті)`

Басқару құрылымдары

Шартты операторлар

PHP тілінде екі Шартты оператор: `if` және `?`.

`If` операторының үш түрі бар. Бірінші түрі-базалық шартты оператор. Мысалы:

```
if ($a > $b)
```

```
{  
echo "А больше Б";  
}
```

If операторының баламалы синтаксисі:

```
if ($a > $b):  
    echo "А больше Б";  
endif;
```

Екінші түрі-if-else шартты операторы . Мысалы:

```
if ($a > $b)  
{  
    echo "А больше Б";  
}  
else  
{  
    echo "А не больше Б";  
}
```

If-else операторының баламалы синтаксисі:

```
if ($a > $b):  
    echo "А больше Б";  
else:  
    echo "А не больше Б";  
endif;
```

Үшінші түрі-if-elseif шартты операторы . Мысалы:

```
if ($a > $b)  
{  
    echo "А больше Б";  
}  
elseif ($a == $b)  
{  
    echo "А равно Б";  
}  
else  
{  
    echo "А меньше Б";  
}
```

If-elseif операторының баламалы синтаксисі:

```
if ($a > $b):  
    echo "А больше Б";  
elseif ($a == $b):  
    echo "А равно Б";  
else:  
    echo "А меньше Б";
```

```
endif;
```

Шартты оператор ? қос нүктемен бөлінген екі мәннің бірін қайтарады. Операторды пайдалану ? бағдарламаның ықшам мәтінін жасай алады. Мысалы:

```
$text = ($a == $b) ? "А равно Б" : "А не равно Б";  
echo $text;
```

Таңдау операторы

Коммутатор таңдау бір өрнекті бағалайды және оның мәніне байланысты бірнеше бағдарлама блоктарының бірін орындайды. Switch операторындағы өрнек көбінесе қарапайым айнымалы болып табылады. Мысалы:

```
switch ( $a )  
{  
  case 1:  
    echo "А равно 1";  
    break;  
  case 2:  
    echo "А равно 2";  
    break;  
  case 3:  
    echo "А равно 3";  
    break;  
  default:  
    echo "А не равно ни 1, ни 2, ни 3";  
}
```

Switch операторының баламалы синтаксисі:

```
switch ( $a ):  
  case 1:  
    echo "А равно 1";  
    break;  
  case 2:  
    echo "А равно 2";  
    break;  
  case 3:  
    echo "А равно 3";  
    break;  
  default:  
    echo "А не равно ни 1, ни 2, ни 3";  
endswitch;
```

Параметрі бар Цикл

For параметрі бар Цикл циклдің ең ескі және лайықты түріне жатады.

Мысал 7

```
<html>  
<head>  
<title>Цикл for</title>
```

```
</head>
<body>
<?php
for ($a = 11; $a <= 19; $a++)
{
    echo "квадрат $a равен ".$a*$a."<br>";
}
?>
</body>
</html>
```

Альтернативный синтаксис оператора **for**:

```
for ($a = 11; $a <= 19; $a++):
    echo "квадрат $a равен ".$a*$a."<br>";
endfor;
```

Шартты циклдар

PHP тілінде екі цикл түрі бар:

- * while - цикл;
- do .. while-кейінгі цикл.

While операторы шарт мәнін бағалайды және егер ол шын болса, фигуралық жақшаларда (цикл денесі) әрекеттерді орындайды. Шарттың мәні жалған болғаннан кейін циклді орындау тоқтатылады.

Мысал 8

```
<html>
<head>
<title>Цикл while</title>
</head>
<body>
<?php
$a = 11;
while ( $a <= 19 )
{
    echo "квадрат $a равен ".$a*$a."<br>";
    $a++;
}
?>
</body>
</html>
```

Замечание для продвинутых: сокращенная запись не работает как ожидается:

```
while ( $a <= 19 )
{
    echo "квадрат $a равен ".$a*($a++)."<br>"; // так не работает!!!
}
```

Альтернативный синтаксис оператора **while**:

```
while ( $a <= 19 ):
    echo "квадрат $a равен ".$a*$a."<br>";
```

```
$a++;  
endwhile;
```

Цикл `do .. while` циклінен ерекшеленеді, бұл жағдайдың шынайылығы дейін емес, цикл денесін орындағаннан кейін тексеріледі. **Мысал 9**

```
<html>  
<head>  
<title>Цикл do .. while</title>  
</head>  
<body>  
<?php  
$a = 11;  
do {  
    echo "квадрат $a равен " . ($a*$a) . "<br>";  
    $a++;  
}  
while ( $a <= 19 );  
?>  
</body>  
</html>
```

7-9 мысалдардың нәтижелері:

*квадрат 11 равен 121
квадрат 12 равен 144
квадрат 13 равен 169
квадрат 14 равен 196
квадрат 15 равен 225
квадрат 16 равен 256
квадрат 17 равен 289
квадрат 18 равен 324
квадрат 19 равен 361*

Циклды ұйымдастыруға тағы екі оператор қатыса алады: `break` (циклдан шығу) және `continue` (келесі қадамға өту).

Зертханалық жұмыс №34. Операциялар.

PHP-де функциялардың екі негізгі түрі бар: кірістірілген және пайдаланушылық. Пайдаланушы функциясын анықтау үшін `function` кілт сөз қолданылады.

Мысал 1

```
<html>  
<head>  
<title>Определение и вызов функции</title>  
</head>  
<body>  
<?php  
function заголовок ($str, $n)  
{  
    echo "<H$n>$str</H$n>";  
}  
заголовок ("Часть", 1);
```

```
заголовок ("Глава", 2);
заголовок ("Раздел", 3);
?>
</body>
</html>
```

PHP 4.0 нұсқасынан бастап, әдепкі аргументтер мәндерін пайдалануға болады.

Мысал 2

```
<html>
<head>
<title>Функция со значением аргумента по умолчанию</title>
</head>
<body>
<?php
function заголовок ($str, $n = 2)
{
    echo "<H$n>$str</H$n>";
}
заголовок ("Часть", 1);
заголовок ("Глава");
заголовок ("Раздел", 3);
?>
</body>
</html>
```

1-2 мысалдырдың нәтижелері:

Часть

10 Глава

11 Раздел

Ескерту: төменде көрсетілген код сіз күткен сияқты жұмыс істемейді. Барлық жағдайларда \$full айнымалысы true болады.

```
fun($ar, false);
function fun($row, $full = true){ ... }
```

Аргументті сілтеме арқылы жіберу

Дәлелдерді сілтемеге жіберуге болады. Сонда функцияның денесінде дәлелмен жасалған барлық өзгерістер айнымалыға әсер етеді.

Мысал 3

```
<html>
<head>
<title>Передача аргумента по ссылке</title>
</head>
<body>
<?php
function add ( &$a, $b )
{
    $a += $b;
```

```

    $b += $b;
}
$a = 10;
$b = 6;
echo "af = $a; bf = $b";
add ( $a, $b );
echo "<br>После вызова функции:<br>";
echo "af = $a; bf = $b";
?>
</body>
</html>

```

3 мысалдың нәтижесі:

```

af = 10; bf = 6
После вызова функции:
af = 16; bf = 6
12 Возврат значений

```

Мәндерді функциядан қайтару үшін return операторы қолданылады. List () тізімін қолдансаңыз, бірнеше мәндерді қайтаруға болады.

Мысал 4

```

<html>
<head>
<title>Функция, возвращающая список значений</title>
</head>
<body>
<?php
function ФИО()
{
    return array ("Иванов", "Иван", "Иванович");
}
list ($фамилия, $имя, $отчество) = ФИО();
echo "<table>";
echo "<tr><td>фамилия: <td>$фамилия";
echo "<tr><td>имя: <td>$имя";
echo "<tr><td>отчество: <td>$отчество</table>";
?>
</body>
</html>

```

4 мысалдың нәтижесі:

фамилия:	Иванов
имя:	Иван
отчество:	Иванович

Динамикалық функциялар

PHP динамикалық функцияларды пайдалануға болады. Бұл дегеніміз, егер кейбір айнымалы функцияның атауы берілсе, онда бұл айнымалы функциямен бірдей болуы мүмкін.

Мысал 5

```
<html>
<head>
<title>Переменные функции</title>
</head>
<body>
<?php
function funct1 ()
{
    echo "Мы внутри функции funct1(<br>";
}
function funct2 ()
{
    echo "Мы внутри функции funct2(<br>";
}
$myfunct = "funct1";
$myfunct();
$myfunct = "funct2";
$myfunct();
?>
</body>
</html>
```

5 мысалдың нәтижесі:

Мы внутри функции funct1();

Мы внутри функции funct2();

Зертханалық жұмыс №35. PHP-де жолдар мен массивтермен жұмыс. Функциялар.

PHP-дегі массивтердің негізгі мақсаты-байланысқан мәндер топтарын ұйымдастыру . Массивтің әрбір элементі индексі (кілті) және мәні бар. Массив элементінің индексі массив атауынан кейін шаршы жақшада көрсетіледі. \$Array 1 массивінің бесінші элементіне жүгіну үшін жазу керек:

```
$array1[5]
```

Әдепкі массив бірінші элементтен емес, нөлден басталады.

Индекс санмен де, мәтіндік жолмен де болуы мүмкін. Жол индекстері бар Массив ассоциативті деп аталады, ал индекстердің өздері элементтердің аттарымен аталады. Мысалы, тауарлар бағасы индексі тауарлардың атауы болып табылатын \$prices ассоциативтік массивінде сақталады. Нақты тауар бағасының мәнін алу үшін жазу керек:

```
$prices["яйца 10 шт. 1 сорт"]
```


Массив элементінің мәні кез келген түрге ие болуы мүмкін. Көп өлшемді массивтер ұйымдастырылуы мүмкін, өйткені массив элементі өз кезегінде массив болуы мүмкін. Көп өлшемді массивтер элементтеріне жүгінудің мысалдары:

```
$array3[0][4][1]
$цены["овощи"]["помидор"]
$список[1]["фамилия"]
```

Массивті параметрлері массивті құрайтын агау () функциясы арқылы жасауға болады. Опцияларды "кілт=>мән" жұптары қоюы мүмкін. Егер массивті жасау кезінде кілт көрсетілмейтін болса, онда индекс массивтегі элементтің орналасуымен анықталады (0 бастап). Мысалы:

```
$рост = array (174, 181, 166); //Массив с индексацией, начинающейся с нуля
$цена = array ("помидоры" => 15, "огурцы" => 12); //Ассоциативный массив
$данные = array (
    "Иванов" => array ("рост" => 174, "вес" => 68),
    "Петров" => array ("рост" => 181, "вес" => 90),
    "Сидоров" => array ("рост" => 166, "вес" => 73)); //Двухмерный массив
```

Массивтерді басқа жолмен жасауға болады-тікелей. Мысалы:

```
$фрукты[] = "яблоко";
$фрукты[] = "груша";
$фрукты[] = "слива";
$цена["помидор"] = 15;
$цена["огурец"] = 12;
```

Массив емес массив элементтерінің индекстерін көрсетпеуге болады. PHP оларды автоматты түрде есептейді. Егер индекстерді осылай көрсетсе:

```
$фрукты[0] = "груша";
$фрукты[5] = "яблоко";
```

онда массивте екі элемент болады, ал соңғы 5 индексі бар. 1-4 элементтері инициалданбайды.

Array () функциясымен массив жасай аласыз, содан кейін оған жаңа элемент қосуға болады:

```
$фрукты = array("яблоко","груша","слива");
...
$фрукты[] = "персик";
```

Массивтегі элементтердің санын count() немесе sizeof () көмегімен анықтауға болады.

Мысал

```
<html>
<head>
<title>Размер массива</title>
</head>
<body>
```

```

<?php
$фрукты = array("яблоко", "груша", "слива", "персик");
echo "Размер массива \$фрукты равен ".count($фрукты)."<br>";
echo "Последний элемент массива \$фрукты - ".$фрукты[count($фрукты)-1]."<br>";
?>
</body>
</html>

```

1 мысалдың нәтижесі:

Массив мөлшері ...жемістер 4-ке тең

Жеміс - шабалды массивінің соңғы элементі

Соңғы элементке қол жеткізу үшін массивтің өлшемінен 1-ді алып тастау керек, себебі массивтің индекстеуі нөлден басталады. Сақталған " ... " таңбасын шығару үшін доллар белгісінің алдында "\"кері қиғаш сызығының белгісі тұр.

Элементтердің массивке кіру жиілігін array_count_values () функциясы арқылы анықтауға болады. Бұл функция кілттері зерттелетін массивтің элементтері болып табылатын массивті қайтарады, ал мәндері - олардың зерттелетін массивке кіру жиілігі.

Мысал 2

```

<html>
<head>
<title>Размер массива</title>
</head>
<body>
<?php
$фрукты = array("яблоко", "груша", "слива", "персик", "груша");
print_r (array_count_values($фрукты));
?>
</body>
</html>

```

2 мысалдың нәтижесі:

Array ([яблоко] => 1 [груша] => 2 [слива] => 1 [персик] => 1)

Print_r () функциясы аргументте көрсетілген кілттер мен массив мәндерін көрсетеді.

Массивтің мазмұнын итерациялық қарау үшін foreach функциясы қызмет етеді. Оның көмегімен қарапайым (сандар арқылы индекстелген) массивті және ассоциативті және көп өлшемді көруге болады.

Мысал 3

```

<html>
<head>
<title>Просмотр массива</title>
</head>

```

```

<body>
<?php
$фрукты = array("яблоко", "груша", "слива", "персик", "груша");
foreach ($фрукты as $фрукт)
{
    echo "$фрукт<br>";
}
?>
</body>
</html>

```

3 мысалдың нәтижесі:

яблоко
груша
слива
персик
груша

Мысал 4

```

<html>
<head>
<title>Просмотр ассоциативного массива</title>
</head>
<body>
<?php
$цена = array ("помидоры" => 15, "огурцы" => 12);
foreach ($цена as $овощи => $руб)
{
    echo "$овощи стоят $руб руб.<br>";
}
?>
</body>
</html>

```

4 мысалдың нәтижесі:

помидоры стоят 15 руб.
огурцы стоят 12 руб.

Зертханалық жұмыс №36. Кластар мен объектілер.

PHP-де және оның жеке блоктарында бағдарламаны жасау кезінде функциялар ұсынатын функционалдылық жеткілікті болуы мүмкін. Алайда, PHP объектілі-бағытталған бағдарламалау болып табылатын бағдарламаларды құруға басқа да мүмкіндіктері бар. Кейбір жағдайларда ООП пайдаланатын бағдарламалар түсіну оңай, оларды қолдау және өзгерту оңай.

ООП парадигмасының негізгі ұғымдары "класс" және "объект" ұғымдары болып табылады. Объектінің сипаттамасы класс болып табылады, ал объект осы сыныптың данасын ұсынады. Келесі ұқсастықты жүргізуге болады: адам туралы кейбір түсінік бар -

екі қол, екі аяқ, бас, ас қорыту, жүйке жүйесі, ми және т.б. кейбір үлгі бар - бұл үлгі сынып деп атауға болады. Ал шын мәнінде бар адам (осы сыныптың нақты данасы) осы сыныптың объектісі болып табылады.

PHP-да класс жасау үшін `class` кілт сөзі қолданылады. Мысалы, пайдаланушыны білдіретін жаңа сынып:

```
class User
{ }
```

Чтобы создать объект класса `User`, применяется ключевое слово **new**:

```
<?php
class User
{ }

$user = new User;
print_r($user);
?>
```

Бұл жағдайда `$user` айнымалы `User` класының нысаны болып табылады. `Print_r ()` функциясымен, массивтер сияқты нысанның мазмұнын көрсетуге болады.

Қасиеттері мен әдістері

Класс объектінің қандай да бір белгілерін сипаттайтын қасиеттерді және оның мінез-құлқын анықтайтын әдістерді қамтуы мүмкін. `User` класына бірнеше қасиеттер мен әдістерді қосамыз:

```
<?php
class User
{
    public $name, $age;

    function getInfo()
    {
        echo "Имя: $this->name ; Возраст: $this->age <br>";
    }
}

$user = new User;
$user->name="Том"; // установка свойства $name
$user->age=30; // установка свойства $age
$user->getInfo(); // вызов метода getInfo()
print_r($user);
?>
```

Мұнда `User` класы екі сипатты қамтиды: `$name` және `$age`. Сипаттары кіру модификаторы тұрған әдеттегі айнымалы ретінде жарияланады - бұл жағдайда `public` модификаторы.

Әдістер белгілі бір әрекеттерді орындайтын қарапайым функцияларды білдіреді. Мұнда `getInfo ()` функциясы бұрын анықталған айнымалылардың мазмұнын көрсетеді.

Осы кластағы ағымдағы нысанды қарау үшін, ағымдағы нысанды білдіреді. Нысанның қасиеттері мен әдістеріне жүгіну үшін қол жеткізу операторы- `>` қолданылады. Мысалы, "name" сипатының мәнін алу үшін `"this->name"` өрнегін пайдалану керек. Оның үстіне қасиеттерге жүгіну кезінде ҰБТ белгісі қолданылмайды.

`User` класындағы нысанды пайдаланған кезде, сипаттар мәнін алу немесе орнату үшін, сондай-ақ әдістерді шақыру үшін кіру операторы да қолданылады.

Конструкторлар және деструкторлар

Конструкторлар объектіні құру кезінде орындалатын және оның қасиеттерін бастапқы инициализациялау үшін қызмет ететін арнайы әдістерді ұсынады. Конструктор құру үшін `__construct` (алдында екі сызықпен) атымен функцияны жариялау керек):

```
<?php
class User
{
    public $name, $age;
    function __construct($name, $age)
    {
        $this->name = $name;
        $this->age = $age;
    }

    function getInfo()
    {
        echo "Имя: $this->name ; Возраст: $this->age <br>";
    }
}

$user2 = new User("Джон", 33);
$user2->getInfo();
?>
```

Бұл жағдайда конструктор функциясы екі параметрді қабылдайды. Олардың мәні сынып қасиеттеріне беріледі. Енді нысан жасау үшін тиісті параметрлер үшін мәндерді беру керек: `$user2 = new User ("Джон", 33);`

Әдетті опциялар

Конструкторды икемді ету үшін, біз бір немесе бірнеше параметрлерді міндетті емес деп белгілей аламыз. Сонда нысанды жасаған кезде барлық параметрлерді көрсету міндетті емес. Мысалы, келесідей дизайнерді өзгертіңіз:

```
function __construct($name="Том", $age=33)
{
    $this->name = $name;
    $this->age = $age;
}
```

Осылайша, параметрлер орнатылмаса, олардың орнына "Том" және 33 мәндері қолданылады. Енді біз `User` нысанын бірнеше жолмен жасай аламыз:

```
$user1 = new User("Джон", 25);
$user1->getInfo();
$user2 = new User("Джек");
$user2->getInfo();
$user3 = new User();
$user3->getInfo();
```

Деструкторлар

Сонымен қатар, бұл жүйенің негізгі мақсаты-деректер қорын құру ережелер қорын басқару, деректер қорын басқару

Деструктор `__destruct` функциясы арқылы анықталады (алда екі астың сызу):

```
class User
{
    public $name, $age;
    function __construct($name, $age)
```

```

    {
        $this->name = $name;
        $this->age = $age;
    }
    function getInfo()
    {
        echo "Имя: $this->name ; Возраст: $this->age <br>";
    }

    function __destruct()
    {
        echo "Вызов деструктора";
    }
}

```

Функция деструктора белгіленеді параметрлерін, және қашан объектіге қалмайды сілтемелер бағдарламасы, ол жойылған кезде болады шақырылуы деструктор.

Зертханалық жұмыс №37. Кластар мен объектілер.

Мұрагерлік-объектілі-бағытталған бағдарламалаудың негізгі аспектілерінің бірі. Мұрагер-сыныптарда бар сыныптардың функционалын қайта анықтауға мүмкіндік береді. Егер бізде бірнеше функция жоқ сынып бар болса, онда бар сыныпты қайта анықтау әлдеқайда оңай, нөлден жаңасын жасағаннан гөрі, код байламын қайта жазу.

Бір класты екіншісінен мұраға алу үшін extends Нұсқаулығын қолдану керек. Мысалы, User класынан Customer класын иеленеміз:

```

<?php
class User
{
    public $name, $age;

    function __construct($name, $age)
    {
        $this->name = $name;
        $this->age = $age;
    }
    function getClassInfo()
    {
        echo "Класс User описывает пользователей<br />";
    }
    function getInfo()
    {
        echo "Имя: $this->name ; Возраст: $this->age <br />";
    }
}
class Customer extends User
{
    public $account, $sum;

    function __construct($name, $age, $acc)
    {
        // вызов конструктора базового класс

```

```

        parent::__construct($name, $age);
        $this->account=$acc;
        $this->sum=20;
    }

    // скрывает родительский метод getClassInfo()
    function getClassInfo()
    {
        echo "Класс Customer описывает клиентов <br />";
    }

    function getInfo()
    {
        parent::getInfo();
        echo "Номер счета $this->account ; Сумма: $this->sum <br />";
    }
}

$client = new Customer("Джон", 25, "1123400895");
$client->getInfo();
$client->getClassInfo();
?>

```

Customer класы user-ата-анасы бүкіл функционалдан мұраға, сондай-ақ \$account және \$sum екі қосымша қасиеттерін қосады. \$Name сипаты User класында анықталғанына қарамастан, оны Customer класынан алуға болады: \$ this - > name

Егер мұрагер-сыныпта ата-ана сыныбындағы сияқты әдіс айқындалса, Мұрагер-сынып әдісі ата-ана-сынып әдісін алмастырады. Біздің жағдайда екі сыныпта да getClassInfo () әдісі анықталған, сондықтан Customer өзінің getClassInfo () әдісіне жүгінетін болады.

Егер біз ата-ана сыныбы әдісіне жүгінуіміз керек болса, онда біз parent нұсқаулығын пайдалана аламыз:

```
1 parent::getInfo();
```

Және Getinfo() әдісін Customer класынан шақырғанда, алдымен Getinfo() ата-аналық класы User әдісі, содан кейін басқа нұсқаулар пайда болады:

```

function getInfo()
{
    parent::getInfo();
    echo "Номер счета $this->account ; Сумма: $this->sum <br>";
}

```

Класс Customer также определяет свой конструктор:

```

function __construct($name, $age, $acc)
{
    // вызов конструктора базового класс
    parent::__construct($name, $age);
    $this->account=$acc;
    $this->sum=20;
}

```

Сондай-ақ, алдымен ата-ана класс конструкторына \$name және \$age қасиеттерін орнатуды өзі орындау үшін үндеу жасалады.

Мұраға тыйым салу және final нұсқаулығы

Жоғарыда мысалда getInfo() және getClassInfo() әдістері класс-мұрагер болып қайта анықталады. Алайда, кейде әдістерді қайта анықтауға тыйым салу қажет болған жағдайлар бар. Ол үшін ата-ана сыныбында final модификаторы бар әдістерді көрсету керек:

```
class User
{
    //.....

    final function getInfo()
    {
        echo "Имя: $this->name ; Возраст: $this->age <br />";
    }
}
```

Бұл жағдайда User сыныбынан барлық Мұрагер-сыныптарда біз сол атаумен әдісті анықтай алмаймыз.

Зертханалық жұмыс №38. PHP деректер базасымен жұмыс.

Әдетте, деректер қоймасы ретінде деректер базасы пайдаланылады. PHP түрлі дерекқорларды басқару жүйесін пайдалануға мүмкіндік береді, бірақ бүгінгі күні PHP-мен байланыстыруда ең танымал MySQL болып табылады. MySQL SQL командасы арқылы деректер базасымен өзара әрекеттесуге мүмкіндік беретін тегін бағдарламалық жасақтаманы ұсынады. Бұрын MySQL орнату және баптау процесі қаралды.

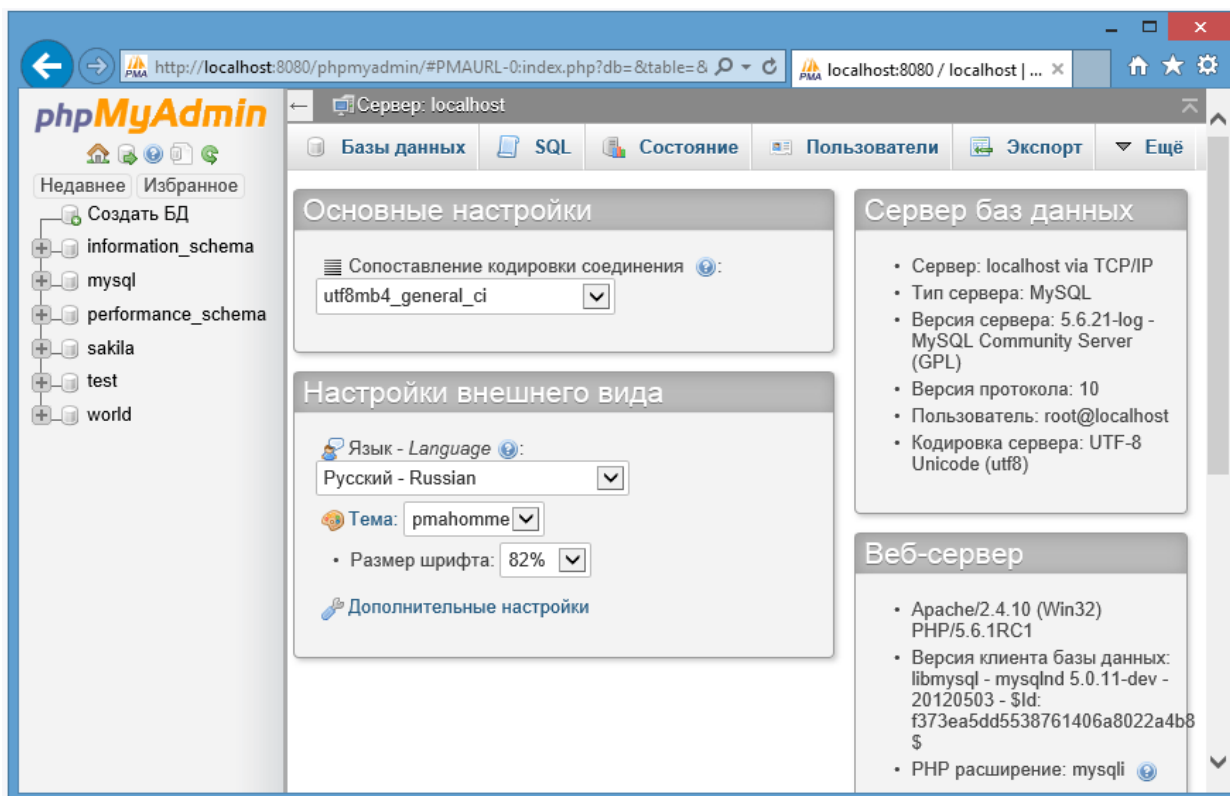
MySQL деректер базасымен жұмысты жеңілдету үшін phpMyAdmin сценарийлерінің арнайы жиынтығын орнатыңыз. phpMyAdmin MySQL деректер қорын басқару үшін интуитивті веб-интерфейсін ұсынады. Бұл құралды пайдалана отырып, MySQL-ді консоль арқылы басқарудан гөрі, деректер қорымен жұмыс істеу оңайырақ.

PhpMyAdmin орнату үшін архивті жүктеп, қалтаға тарқатамыз C:\localhost, php үшін құжаттар қайда. Тарқатылған қалтаны phpMyAdmin-ге қысқа етіп қайта атаймыз.

Rpmyadmin каталогында config файлын жасаңыз.inc.php келесі мазмұнмен:

```
<?php
$i = 0;
$i++; // для первого сервера MySQL
$cfg['Servers'][$i]['host'] = 'localhost';
$cfg['Servers'][$i]['extension'] = 'mysqli';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['compress'] = false;
$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = 'пароль от mysql'; // Пароль пользователя root
?>
```

Және бәрі дұрыс орнатылғанына көз жеткізу үшін браузерде phpMyAdmin-ге барамыз, мысалы <http://localhost:8080/phpmyadmin>:



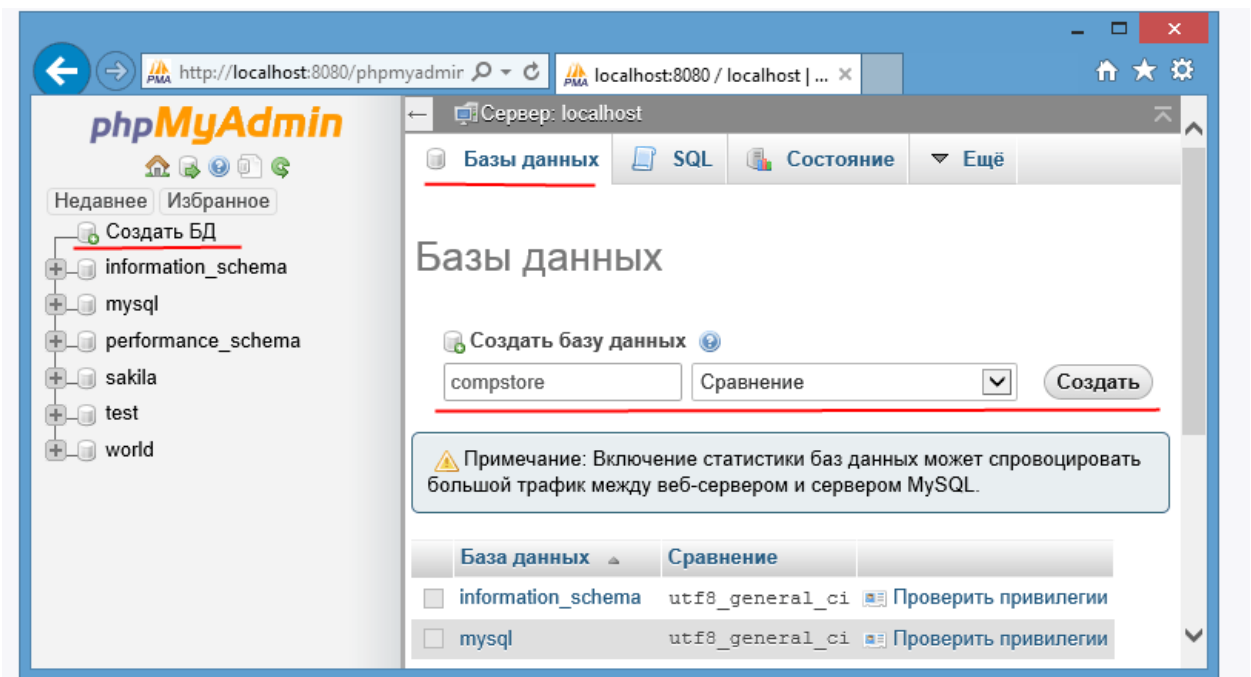
Сол жақ бағанда MySQL серверінде бар барлық деректер базасын көруге болады. Егер сіз ешқандай дерекқор жасамасаңыз да, серверде әдепкі дерекқор жиынтығы бар.

PhpMyAdmin интерфейсінің оң жағында деректер базасын басқарудың негізгі құралдары, сондай-ақ әртүрлі конфигурациялық ақпарат бар.

PhpMyAdmin-де MySQL деректер базасын құру

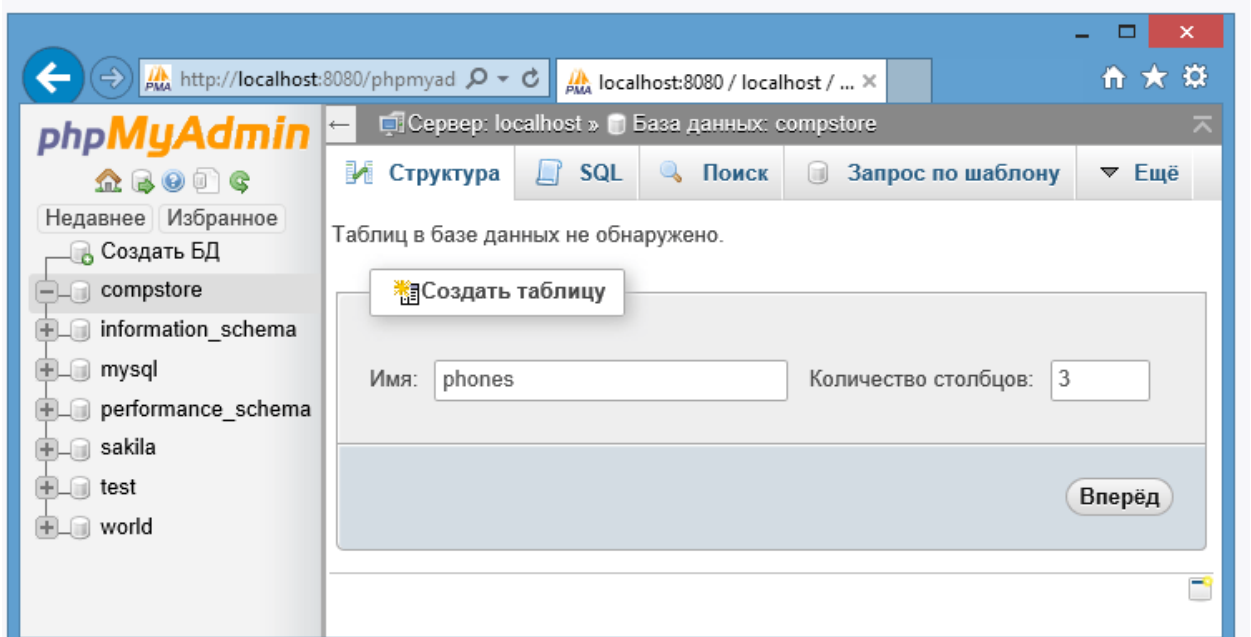
MySQL серверімен деректерді алмасу үшін (сақтау, өзгерту, жою, деректерді алу) бізге табиғи дерекқор қажет. MySQL консолінен, сондай-ақ phpMyAdmin визуалды интерфейсінен деректер базасын жасай аламыз.

Біз phpMyAdmin интерфейсін ашамыз. Дерекқор қойындысына өтіңіз. Деректер базасын құру белгісі астында жаңа ДБ үшін қандай да бір атауды енгіземіз, мысалы, compstore және "құру"батырмасын басамыз.

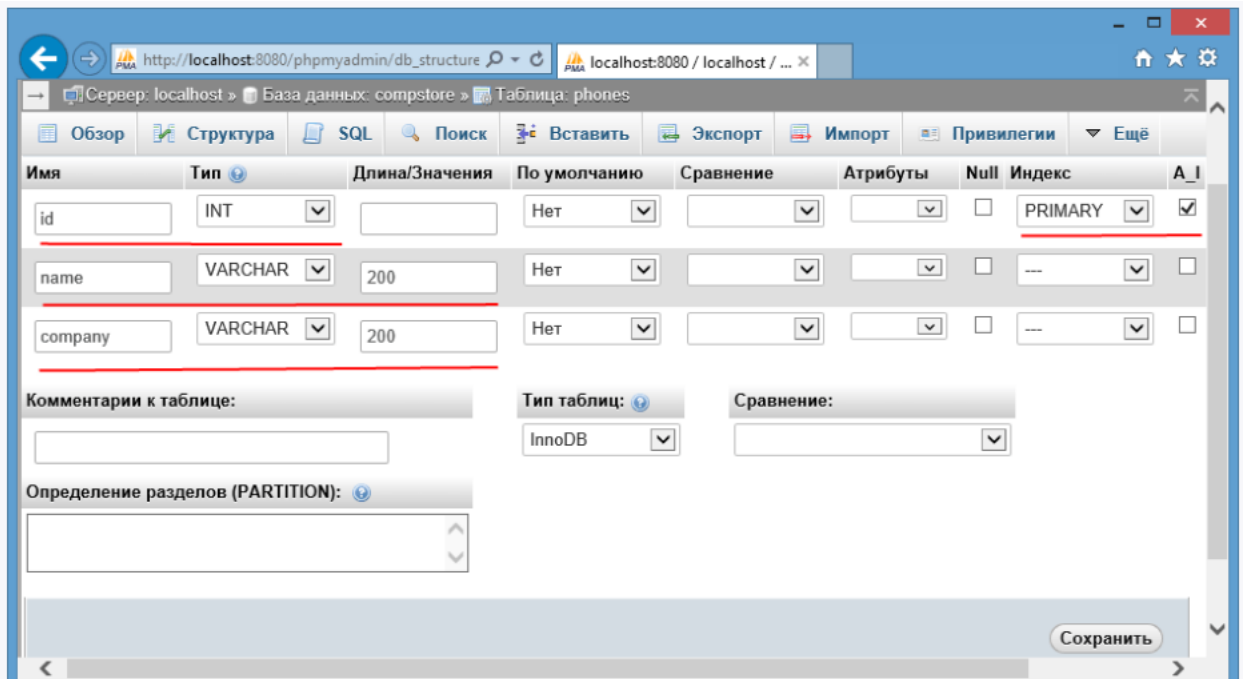


Содан кейін біз жаңа ДБ-ны сәтті құру туралы хабар аламыз және ол деректер базасының тізіміне қосылады.

Жаңа дерекқор әлі бос және ештеңе жоқ. Деректерді сақтайтын кестені қосыңыз. Ол үшін дерекқор атауына басамыз және Біз жаңа кесте опциялары ұсынылатын" құрылым "қойындысына түсеміз. "Аты" өрісіне жаңа кестенің атауын енгіземіз. Кестелер смартфондар модельдері туралы деректерді сақтасын, сондықтан "phones" атауын енгіземіз, ал бағандар саны ретінде 3 санын енгіземіз:

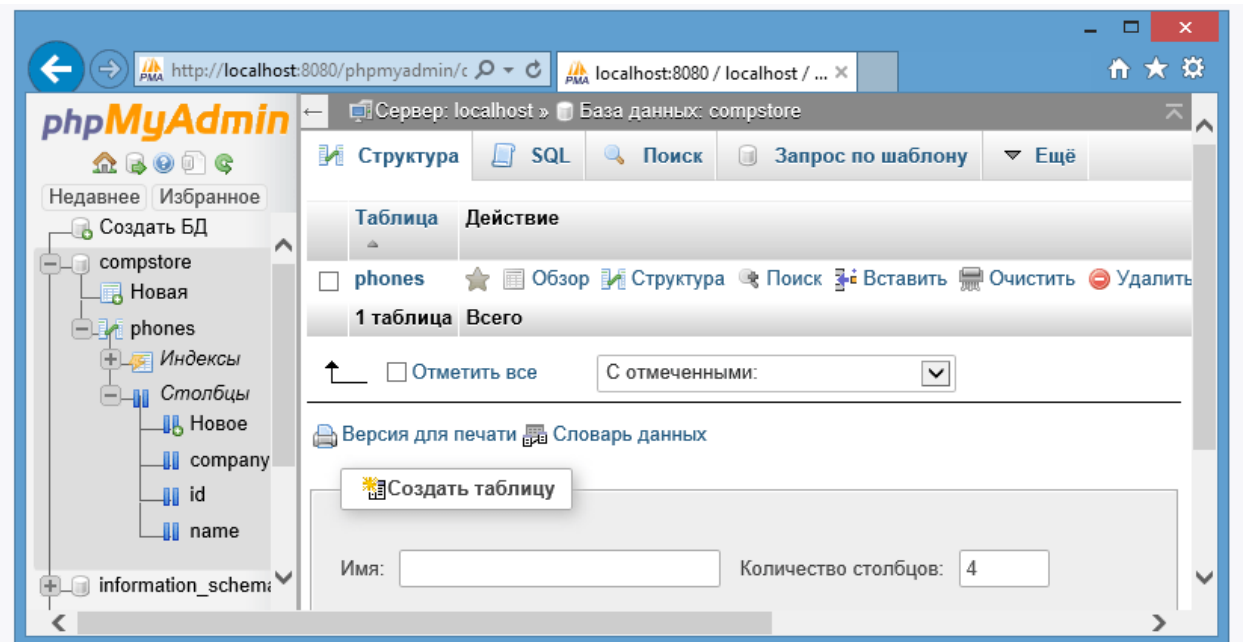


Кестені жасау үшін "алға" батырмасын басамыз. Содан кейін баған параметрлерін орнату үшін ұяшықтар жиынтығы пайда болады. Бағандардың аттары үшін дәйектілікпен көрсетеміз: id, name, company. Id типті Int бағандары үшін үлгі түрі ретінде, ал name және company бағандары үшін - VARCHAR түрі. "Ұзындығы/мәні" өрісінде name және company бағандары үшін 200 санын көрсетеміз-ол таңбалардағы жолдың ең жоғарғы ұзындығын көрсетеді. Сондай-ақ, id бағаны үшін "Индекс" алабындағы PRIMARY а "a_i" (Auto Increment) алабындағы белгі қоямыз:



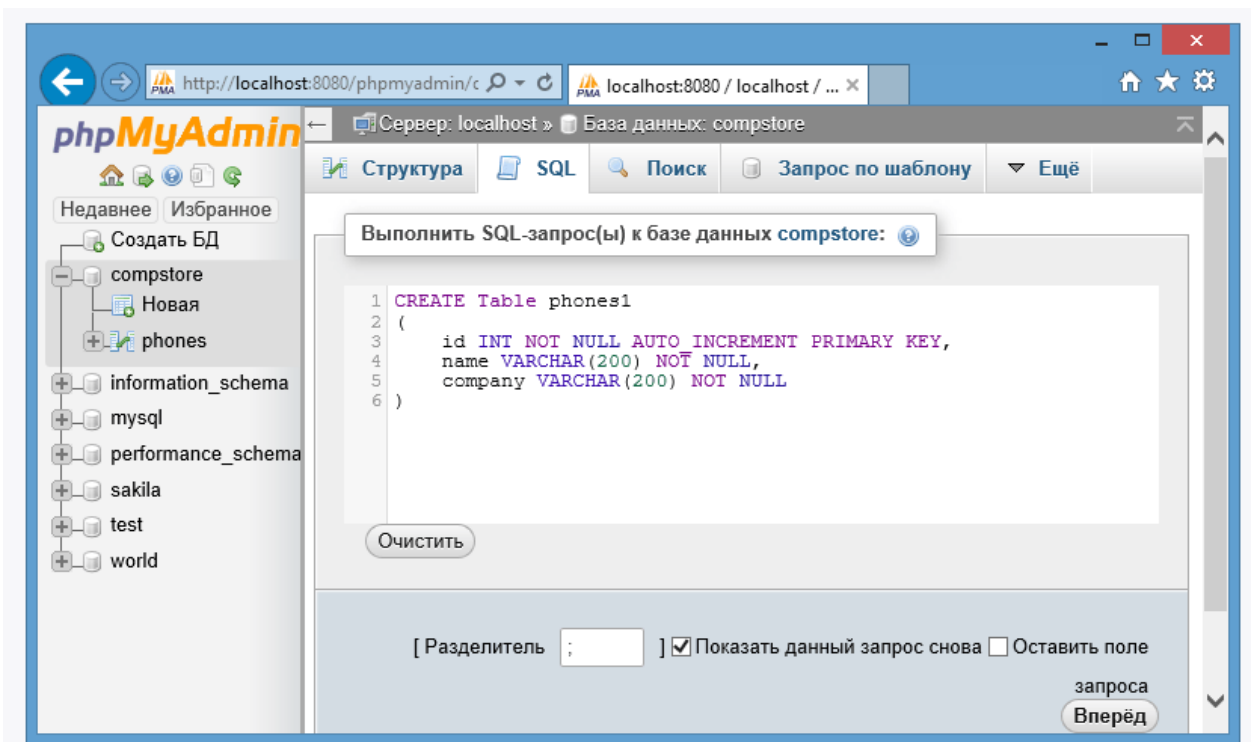
Осылайша, кесте бірегей идентификатор, телефон атауы және өндірушінің атауы үшін бағандарға ие болады. Содан кейін "Сақтау" батырмасын басыңыз.

Кестені жасағаннан кейін кестені және оның бағандарын дерекқор бағанынан көре аламыз:

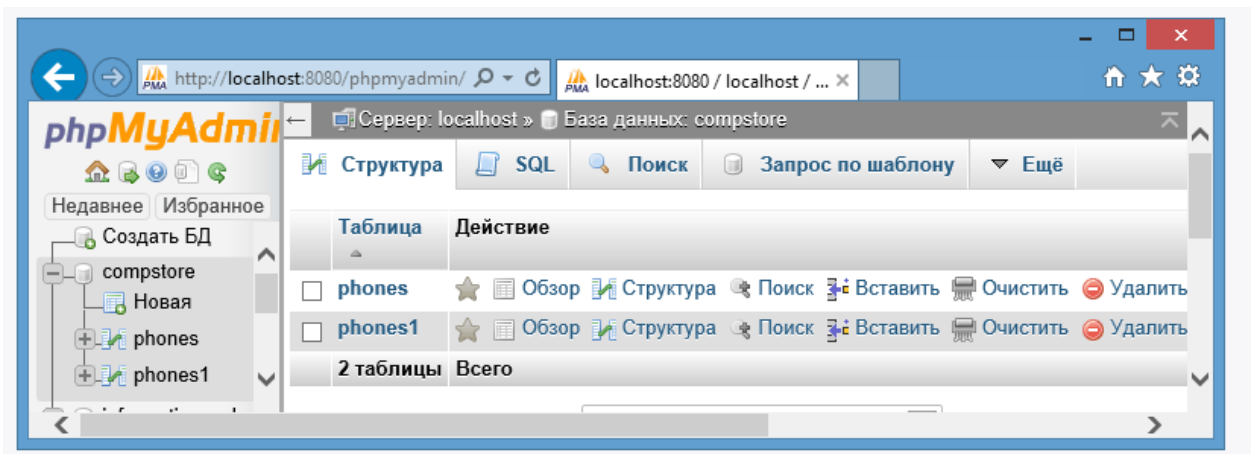


Бұл phpMyAdmin-де кестелерді жасаудың жалғыз жолы емес, өйткені мұнда SQL сұраулары арқылы деректер базасын басқара аламыз. Мысалы, дерекқор тізімінде біздің базаны ерекшелейміз және "SQL" қойындысына көшеміз. Ол SQL сұрау тілінде пәрменді енгізу өрісін көрсетеді. Оған келесі команданы енгіземіз:

```
CREATE Table phones1
(
  id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(200) NOT NULL,
  company VARCHAR(200) NOT NULL
)
```



Бұл стандартты SQL кестені жасау командасы. Негізгі сөздерден кейін CREATE Table-дің атауы жасалады, содан кейін жақшаларда бағандарды анықтау үтір арқылы жүреді. Әрбір баған анықтамасы бағанның атауын, оның түрін және қосымша мәндерді қамтиды. Мысалы, id бағанын анықтауда (id INT null AUTO_INCREMENT PRIMARY KEY) бағанның аты, түрі - INT, сондай - ақ бағанның міндетті түрде мәні болуы тиіс - NOT NULL, оның мәні автоматты түрде жаңа объект - AUTO_INCREMENT қосу арқылы бірлікке ұлғаяды және ол бастапқы кілт-PRIMARY KEY рөлін атқарады. Жалпы, осылайша жасалатын кесте бұрын жасалған кестеге тең болады. "Алға" түймесін басып, екінші phones 1 кестесі құрылады.



Зертханалық жұмыс №39. PHP деректер базасымен жұмыс.

MySQL-ді PHP-дан қосу үшін бізге қосылу баптауларын көрсету керек: сервер мекен-жайы, логин, пароль, деректер базасының атауы және т. б., өйткені, әдетте, ДБ-ға қосылу сценарийлері көп қолданылады, онда жиі қосылу баптаулары жеке файлға шығарылады, соның арқасында оларды жылдам өзгерту оңай.

Сонымен, connection файлын жасаңыз.php және оған келесі жолдарды қосыңыз:

```
<?php
$host = 'localhost'; // адрес сервера
$dbname = 'compstore'; // имя базы данных
$user = 'root'; // имя пользователя
$password = '1234567'; // пароль
?>
```

Для того, чтобы добавить MySQL с PHP, нам необходимо указать настройки подключения: адрес сервера, логин, пароль, название базы данных и т. д., Так как обычно используется много сценариев подключения к БД, в которых часто настройки подключения выводятся в отдельный файл, благодаря чему их легко изменить.

Итак, создайте файл connection.php и добавьте к нему следующие строки:

```
<?php
require_once 'connection.php'; // подключаем скрипт

// подключаемся к серверу
$link = mysqli_connect($host, $user, $password, $database)
    or die("Ошибка " . mysqli_error($link));

// выполняем операции с базой данных

// закрываем подключение
mysqli_close($link);
?>
```

Ең алдымен, сценарийді require_once нұсқауларымен теңшеу арқылы қосыңыз.

Байланыс ашу үшін mysqli_connect () функциясын қолданамыз. Ол барлық конфигурациялық параметрлерді қабылдайды және серверге қосылады. Қосылу қатесі болған жағдайда die () операторы іске қосылады, ол қате туралы хабарды шығарады және сценарий жұмысын аяқтайды. Ал сәтті қосылғанда mysqli_connect() функциясы қосылым нысанын \$link айнымалы түрінде қайтарады.

Жұмыс аяқталғаннан кейін қосылымды жабу керек. Ол үшін mysqli_close () функциясы қолданылады, ол параметр ретінде қабылдайды объект қосылу.

Mysqli_query () функциясын пайдалану керек, ол екі параметрді қабылдайды: қосылу объектісі және SQL сұрау жолы. Мысалы:

```
<?php
require_once 'connection.php'; // подключаем скрипт

// подключаемся к серверу
$link = mysqli_connect($host, $user, $password, $database)
    or die("Ошибка " . mysqli_error($link));

// выполняем операции с базой данных
$query = "SELECT * FROM phones";
$result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($link));
```

```

if($result)
{
    echo "Выполнение запроса прошло успешно";
}

// закрываем подключение
mysqli_close($link);
?>

```

Mysqli_query() функциясы сұрау нәтижесі бар \$result нысанын қайтарады. Сәтсіз болған жағдайда, бұл нысан false мәнін қамтиды.

Зертханалық жұмыс №40. Сервер жағында бағдарламалау.

Upload files дегеніміз не немесе неге жұмыс істемейді
copy ("c:\images\sample.jpg", "http://mysite.ru/uploads/sample.jpg ")

Егер сізде сервер мен жұмыс станциясы қосылған бір ғана компьютер болса да, php клиент/сервер технологиясын пайдаланатынын ұмытпаңыз. Біз жүктеп алғысы келетін файл әдетте клиенттің машинасында, яғни пайдаланушы, сайттың қарапайым келушісі. Тағайындалған орны-сервер. Файлды жіберу процесін жасау үшін келесі пішін қажет:

```

<form enctype="multipart/form-data" action="/upload.php" method="post">
  <input type="hidden" name="MAX_FILE_SIZE" value="30000">
  Send this file: <input name="userfile" type="file">
  <input type="submit" value="Send File">
</form>

```

Бұл жағдайда action өрісінде сіздің php сценарийіңіздің URL мекен-жайы көрсетілуі керек, ол одан әрі жүктелетін файлдарды өңдейді. MAX_FILE_SIZE жасырын өрісі файлды таңдаудың алдында болуы керек және байттағы файлдың барынша рұқсат етілген өлшемінен тұруы керек. Оның мақсаты - файлдың көлемін серверге жіберу сәтіне дейін тексеру. Бұл пайдаланушыны серверге ұзақ және нәтижесіз файл жүктеуден және артық трафик жасаудан құтқаруы керек, бірақ бұл шектеуді ерекше түрде ойлаудың қажеті жоқ, себебі оны айналып өту оңай.

Пайдаланушы дискідегі файлды таңдап, "Send file" батырмасын басқанда не болады? Браузер отсылап файл сервер, мұнда php-интерпретаторлар орналастырады, оны өзінің уақытша директорию, присваивая оған кездейсоқ аты және орындайды, скрипт, көрсетілген жолында action.

Upload қалай көрінуі керек.php?

```

$uploaddir = '/var/www/uploads/';
if (move_uploaded_file($_FILES['userfile']['tmp_name'], $uploaddir.
    $_FILES['userfile']['name'])) {
    print "File is valid, and was successfully uploaded.";
} else {
    print "There some errors!";
}

```

При написании скрипта, возникает естественный вопрос: как получить жүктелген файл туралы ақпарат және файлға жету. Егер сіз PHP 4.1.0 және одан жоғары нұсқаларын

пайдалансаңыз, `$_files` жаһандық массивіне жүгініңіз. Әрбір жүктелген файл үшін келесі деректермен хэш массиві бар:

- `$_FILES['userfile']['name']` - файлды тандап, пайдаланушы көрген файлдың бастапқы атауы;

- `$_FILES ['userfile'] ['type']` - MIME/type файл, мысалы, image/gif болуы мүмкін; жүктелген файлдарды жүктеу үшін интерфейс ұсынғыңыз келсе, бұл өрісті сақтау пайдалы;

- `--_FILES['userfile']['size']` - жүктелген файл өлшемі;

- `$_FILES['userfile']['tmp_name']` - толық жолын уақытша файлды дискіде;

- * `$_FILES['userfile']['error']` - 4.2.0 нұсқасынан бастап, егер операция сәтті өтсе, 0 тең қате коды бар.

4.1.0 төмен PHP нұсқасы үшін бұл массив `$HTTP_POST_FILES` деп аталады. `$_FILES`-тен айырмашылығы, бұл массив суперглобалды емес және оған жүгінгенде, мысалы, функциядан `global $HTTP_POST_FILES`-ді анық көрсету қажет екенін ұмытпаңыз.;

Егер сіздің серверіңіздің `register_globals=on` баптауларында `$userfile_name`, `$userfile_type`, `$userfile_size` қосымша айнымалылары құрылады. Жүктелетін файлдар туралы ақпаратты алудың ең жақсы жолы - `$_FILES` массивін пайдалану.

Жүктелген файлдармен жұмыс істеу үшін, файл жүктелгенін тексеретін және оны көрсетілген қалтаға орналастыратын `is_uploaded_file()` және `move_uploaded_file ()` мүмкіндіктерін пайдалану керек. Толығырақ ақпаратты басшылықтың беттерінде таба аласыз. Велосипед ойлап тауып, уақытша файлдармен жұмыс істеуге, оларды көшіруге, жоюға болмайды. Бұл сізге және сіз үшін жасалды.

- Мен бәрін дұрыс жасадым, бірақ менде бірдеңе жұмыс істемейді. Мүмкін, сервер дұрыс конфигурацияланбаған шығар?
- Егер сіз "бәрін дұрыс жасадыңыз", бірақ сіздің кодыңыз жұмыс істемейді немесе дұрыс жұмыс істемесе, үмітсіздікке асығыңыз. Мәселе сіздің қолыңызда емес, сервердің дұрыс емес параметрлерінде болуы мүмкін. Мұнда файлдарды жүктеуге қатысы бар директивалар тізімі:
- Php файлында.ini:
- Егер сіз php орналасқан жерін білгіңіз келсе.ini, орындаңыз

```
<?php phpinfo(); ?>
```

- `file_uploads`-файлдарды жалпы жүктеуге тыйым салу немесе рұқсат ету мүмкіндігі. Әдепкі `On`.
- `upload_max_filesize`-жүктелуі мүмкін файлдың ең үлкен өлшемі. Үлкен файлдармен жұмыс істеу керек болса, осы параметрді өзгертіңіз. Әдепкі бойынша 2м. `post_max_size` өзгертуді ұмытпаңыз.
- `post_max_size`-сұрауда жіберілетін деректердің өлшеміне жоғарыдан жалпы шектеу. Үлкен файлдармен жұмыс істеу немесе бір уақытта бірнеше файлды жіберу қажет болса, осы параметрді өзгертіңіз. Әдепкі мән 8М.
- `upload_tmp_dir` - барлық жүктелетін файлдар орналастырылатын сервердегі уақытша директория. Оған қандай құқықтар қойылғанын тексеріңіз(егер осы кезеңде сізде қиындықтар туындаса, мақаланың соңында түсіндірмені қараңыз). Мұндай директория Apache орындайтын пайдаланушыда да болуы керек, сондай-ақ осы директорияға жазу құқығы болуы керек. Егер сіз `open_basedir` шектеуімен жұмыс істесеңіз, онда уақытша каталог ішінде болуы керек. Сіз оны тазалау немесе атаулардың бірегейлігіне қамқорлық қажет емес, PHP бұл мәселені сіз үшін шешеді.
- `Httpd` файлында.conf:

- Ең алдымен, Apache 1.3 веб-серверін пайдаланғаныңызға көз жеткізіңіз (мақала жазудың соңғы нұсқасы-1.3.27). Егер сіз Apache 2.0 пайдалансаңыз, құжаттаманың келесі үзіндісін оқу керек:
- Do not use Apache 2.0 and PHP in a production environment neither on Unix nor on Windows.
- Егер сіз "POST Method not Allowed" хабарын алсаңыз, бұл келесі директиваларға ұқсас нәрсені іздеу және Allow кілт сөзін пайдалану керек:

- <Limit POST>
- Order allow,deny
- Allow from all
- </Limit>

- Бинарлық файлдарды жүктеумен проблемалар-классикалық сұрақ "upload кезінде файлдарды неге ұрады". Міне, Дима Бородиннің шешімі (<http://php.spb.ru>): сценарий жатқан директорияда файлды жасаймыз .htaccess, онда жазамыз:

CharsetDisable On

Httpd файлына.conf жолдарды жазу:

```
<Location />
CharsetRecodeMultipartForms Off
</Location>
```

Бұл рецептке шағын түсініктемелер: серверге жүктелген архивтер ашылмаған кезде жоғарыда сипатталған мәселе және суреттер көрсетілмейді, өйткені Russian Apache веб-сервері пайдаланылады қандай пайда болуы мүмкін. CharsetDisable директивасы charset-processing module модулін өшіреді, яғни осы қалтадағы файлдарды жүктеу кезінде ешқандай қайта кодтау болмайды. CharsetRecodeMultipartForms директивасы Content-Type: multipart/form-data тақырыптарымен POST әдісімен берілген деректерді қайта кодтауды өшіреді. Т. е. бұл параметрмен берілген екілік деректер бастапқы түрде қалдырылады, ал қалған сайт толтыру сервердің ағымдағы параметрлеріне сәйкес қайта кодталады.

Бірақ асқынулар болуы мүмкін: кейбір жағдайларда сұраулардың мәтіндік бөліктері өзіңіз қайта кодтауға тура келетініне дайын болыңыз. Бұл туралы құжаттамада айтылған:

PL23-да пайда болған CharsetRecodeMultipartForms директивасын пайдаланыңыз, бірақ сіз сұраулардың мәтіндік бөліктерін қолмен кодтау керек. Ол үшін басқа модульдерде қол жетімді Russian Apache Apache Apache Perl API mod_perl-ден қол жетімді.

Кодтауды анықтаудың бір мысалын мына жерден таба аласыз: http://tony2001.phpclub.net/detect_charset/detect.php

Russian Apache бойынша ең жаңа құжаттама оның ресми сайтында: <http://apache.lexa.ru/>.

Кез келген конфигурацияны ауыстырғаннан кейін веб-серверді қайта іске қосу керек екенін ұмытпаңыз.

Сондай-ақ, Arach параметрлерін орнату мүмкін .htaccess:


```
php_value upload_max_filesize 50M
php_value post_max_size 50M
```

Зертханалық жұмыс №41. Сервер жағында бағдарламалау.

Кестені жасау үшін SQL өрнегін пайдалану керек " CREATE TABLE":

```
<?php
require_once 'connection.php';

$link = mysqli_connect($host, $user, $password, $database)
    or die("Ошибка " . mysqli_error($link));

$query = "CREATE Table tovars
(
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(200) NOT NULL,
    company VARCHAR(200) NOT NULL
)";
$result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($link));
if($result)
{
    echo "Создание таблицы прошло успешно";
}

mysqli_close($link);
?>
```

Сценарийді орындағаннан кейін деректер базасында tovars кестесі жасалады, ол өз анықтамасы бойынша бұрын phpMyAdmin арқылы жасалғанына ұқсас болады. Және біз phpMyAdmin интерфейсін ашып, оны көре аламыз.

Кестені жою үшін SQL өрнегін орындау керек " DROP TABLE":

```
<?php
require_once 'connection.php';

$link = mysqli_connect($host, $user, $password, $database)
    or die("Ошибка " . mysqli_error($link));

$query = "DROP TABLE tovars";
$result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($link));
if($result)
{
    echo "Удаление таблицы прошло успешно";
}

mysqli_close($link);
?>
```

Деректерді қосу үшін " INSERT":

```
$query = "INSERT INTO tovar VALUES(NULL, 'Samsung Galaxy III', 'Samsung)";
```

"INSERT" өрнегі кестеге бір жолды енгізеді. INTO кілт сөзінен кейін кестенің атауы көрсетіледі, ал VALUES кейін жақшада барлық бағандар үшін мәндер жиыны көрсетіледі. Кестеде үш баған болғандықтан, біз үш мәнді көрсетеміз.

Бұл жағдайда id бағаны үшін null мәні, name - 'Samsung Galaxy III', ал company үшін - 'Samsung' мәні беріледі.

Ол үшін белгілі бір сандық мәнді көрсету міндетті емес және NULL мәнін беруге болады, ал MySQL бағанға келесі қолжетімді мәнді береді.

Енді мысалда деректерді қосуды қарастырайық. Create файлын жасаңыз.php келесі мазмұнмен:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<?php
require_once 'connection.php'; // подключаем скрипт

if(isset($_POST['name']) && isset($_POST['company'])) {

    // подключаемся к серверу
    $link = mysqli_connect($host, $user, $password, $database)
        or die("Ошибка " . mysqli_error($link));

    // экранирования символов для mysql
    $name = htmlentities(mysqli_real_escape_string($link, $_POST['name']));
    $company = htmlentities(mysqli_real_escape_string($link,
$_POST['company']));

    // создание строки запроса
    $query = "INSERT INTO tovars VALUES(NULL, '$name', '$company)";

    // выполняем запрос
    $result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($link));
    if($result)
    {
        echo "<span style='color:blue;'>Данные добавлены</span>";
    }
    // закрываем подключение
    mysqli_close($link);
}
?>
<h2>Добавить новую модель</h2>
<form method="POST">
<p>Введите модель:<br>
<input type="text" name="name" /></p>
<p>Производитель: <br>
<input type="text" name="company" /></p>
<input type="submit" value="Добавить">
</form>
</body>
</html>
```

Мұнда деректер базасымен өзара әрекеттесу коды формалардың функционалдылығымен біріктірілген: форманың көмегімен деректерді ДБ-ға қосу үшін енгіземіз.

Зертханалық жұмыс №42. MySQL деректер қоры.

MySQL деректер базасындағы басқару жүйесінің серверіне қосылу mysqli сыныбында орындалатын құрылымды пайдалану арқылы жүзеге асырылады.

```
$ mysqli = new mysqli ('host', 'username', 'passwd', 'dbname', 'port')
```

Mysqli сыныбының конструкторын шақырғаннан кейін, MySQL серверіне қосылуды білдіретін нысанды қайтарады. Немесе:

```
$mysqli = new mysqli();
```

```
$mysqli->real_connect ('host', 'username', 'passwd', 'dbname', 'port')
```

Бұл скрипте host – қосылатын хост атауы, username – пайдаланушының аты, passwd – пайдаланушының паролі және dbname – сұраныс жасалатын деректер базасының атауы.

Серверге қосылуды аяқтау үшін \$ mysqli-> close () командасы қолданылады.

Мысалы, student пайдаланушысы (паролі ="123") үшін жергілікті сервердің деректер базасына қосылу үшін төмендегі скрипті теруі керек.

```
<?php
$mysqli = new mysqli("localhost", " student ", "123", "db");
if (mysqli_connect_errno())
{ printf ("Байланыс орнатылмады: %s\n", mysqli_connect_error());
exit();}
$mysqli->close();
?>
```

Ал егер деректер базасымен байланыс орнатылмаса, онда «Байланыс орнатылмады» хабары мен қатенің нөмірі көрсетіледі.

Практикалық жұмыс

<http://localhost/Tools/phpMyAdmin/> адресі бойынша компьютердің жергілікті деректер базасымен байланыс орнат. PhpMyAdmin интерфейсіні қолданып, еліміздегі өзендер атауы мен олардың ұзындықтары жайлы «river» атауымен деректер базасын құрастыр. Деректер базасына еліміздегі бес өзен жайлы ақпарат енгізіп, оны сақта. PHP программасы арқылы деректер базасымен байланыс орнат.

Зертханалық жұмыс №43. MySQL деректер қоры.

1. SQL Server Configuration утилиты танысу.

1.1 SQL Server Configuration Manager утилитаның оның көмегімен ушыққан қызметтің тізбесін серверде анықтайсыңдар. Осы тізбені есеп беруге жазу.

1.2 серверде мен тағайынды MS SQL Server 2008 мен көмек Services утилиты анықта MS SQL Server қызметінің ұшырымының параметрлерін және оларды есеп беруге жазып ал. (Pұқсат алу Services утилите болсам, сол арқылы SQL Server Configuration Manager).

1.3 анықтайсыңдар, мен көмек қандай желілік кітапханалардың мүмкін тағайынды мен MS SQL Server (сурет 1.25) құралым. Қандай кітапханалар ұшырымы белсенді болып табылады? Осы ақпаратты есеп беруге жазып ал.

1.4 арқылы SQL Server Configuration Manager анықтайсыңдар, бас негіз қандай желілік кітапханалардың клиент MS SQL Server қосылады. Кітапхананың тізбесін есеп беруге жазу.

2. Мен SQL құралымды сервермен тағайында.

2.1 жұмыстың бекетінде асқындыр- SQL Server Management Studio және сервердің логикалық атын сендердің компьютерінде ушыққан тізбеде деген таңдап ал-. Керек сервер тізбеде болсам, сол таңдап алу болады <Browse more for.> мүмкін орында қосу және сұрап серверді сервердің тізбесінде.

2.2 MS SQL Server аутентификациясының ақы-пұлдарының серверге қосу.

2.3 жазып қойу үшін ана чтобы жаңа сұранысты орындауға қажетке команданы New Query SQL Server Management Studio аспабының панельсінде пейілді. Арада нәтижеде келесі мүмкіндіктерді жіберіп жаңа қыстырма ашу және пайдаланушының атының, тағайындаған құралымды - тақырыптама сервердің логикалық аты, ағымдық дерекқор көрсетіледі;

- сұраныстың енгізуі үшін пайдаланатын сұраныстың облысының, жұқтыр- MS SQL Server;

- сұраныстың орындалуының нәтижелері кейіптелетін нәтиженің облысының, ал кескіннің қиюы Messages.

3. MS SQL Server конфигурациясының параметрлерімен танысу.

MSSQLServer қызметінің конфигурациялауы мүмкін орында немесе арнаулы SQL Server Management Studio утилите орындайтын сақтау рәсімі немесе осы утилиты ақы-пұлдарымен ғой жазу-сызудың қиюының, қиюның талғамында мағына емес, т.к. жазу-сызудың қиюы рұқсат алуды жүйелі деректерлерге мен көмек ғой осы сақтау рәсімін жасайды, ғана арада астам көрнекі пішінде.

Зертханалық жұмыс №44. MySQL деректер қоры.

1. Сервердің параметрінің тізбесін кейіпте.

General вкладке негізгі ақпарлар туралы жүйеде: операциялық жүйенің болжамы, жадтың көлемінің, және др. процессорының саны, ал да сервердің қызметінің ұшырымының параметрлері кейіптеледі.

Memory қыстырма жадтың қасқайт- MS SQL Server әрекетінің орындалуы үшін: либо үдемелі басқарма жадпен немесе тағайындау тіркелген өлшемді басқару қояды.

Security қыстырмасы пайдаланушының аутентификациясының үлгісі анықталады, да рұқсат алудың аудитының параметрлері серверге анықталады. Күйттеу болады сервер тағайынды MSSQLServer қызметі асқындыру болатын есептік жазбаның игерушілігіне.

Connections қыстырма қосуларды клиенттің қосуларының серверге конфигурация қояды. Пайдаланушының ең көпсаны серверге деген мезгілдес қосылу. Нәлдік мағына көрсетілген, сол оның саны 32767 келеді.

Advanced қыстырма сервердің біреудің ортақ қондырғыларын асырайды. Айталық, тіл ша умолчанию сервердің хабарламалары үшін анықталады немесе 2000 сүйеніш реттейді.

4. Дерекқор атпен Stud _<фио_студента>ДҚБЖ MS SQL Server 2008 1 ақы-пұлдарымен журналмен SQL Server Management Studio ақы-пұлдарымен және атпен Stud _<фио_студента>Query Editor 2 ақы-пұлдарымен және sp _ helpdb . рәсімінің орындалуының нәтижелерін жасау үшін сендермен ДҚ есеп беруге деген жазып ал.

5. Жаса- деген бірдың сақтық көшірмесін сендермен ДҚ жаса- және оператордың BACKUP орындалуының нәтижесін есеп беруде кейіптейсіңдер.

6. Сізбен жасау дерекқорды Stud _ қайта ата-<фио_студента> 1 ара Stud _<фио_студента> карамастан және атын өзгертудің операторының орындалуының нәтижесін есеп беруде кейіптейсіңдер.

7. Анықтау ақпараттар туралы жасау сендермен ДҚ қарызға алатын диск. Түйін дерекқорды олай, ол ғана жетімді оған ағымдық кезді 25% аяның асыраған.

8. Сендермен жасау дерекқорды атпен Stud _ алыстат-<фио_студента> 2 және аулақтаудың операторының орындалуының нәтижесін есеп беруде кейіптейсіңдер.

9. Отключить/подключить жаса- сендермен ДҚ Stud _<фио_студента> серверден. ДҚ қатал күй табағымда құралса, сол орын алмастыр-4. Жаса- дерекқор атпен Stud _<фио_студента>ДҚБЖ MS SQL Server 2008 1 ақы-пұлдарымен журналмен SQL Server Management Studio ақы-пұлдарымен және атпен Stud _<фио_студента>Query Editor 2 ақы-пұлдарымен және sp _ helpdb . рәсімінің орындалуының нәтижелерін жасау үшін ДҚ есеп беруге жаз.

5. Жасау деген бірдың сақтық көшірмесін сендермен ДҚ жасау және оператордың BACKUP орындалуының нәтижесін есеп беруде кейіптейсіңдер.

6. Дерекқорды Stud _ қайта ата-<фио_студента> 1 ара Stud _<фио_студента> карамастан және атын өзгертудің операторының орындалуының нәтижесін есеп беруде кейіптейсіңдер.

7. ДҚ анықтау ақпараттар туралы жасау.

8. Сендермен жасау дерекқорды атпен Stud _ алыстат-<фио_студента> 2 және аулақтаудың операторының орындалуының нәтижесін есеп беруде кейіптейсіңдер.

9. қосу/өшіру жасау сендермен ДҚ Stud _<фио_студента> серверден. ДҚ қатал күй құралса, сол орын алмастырады ол сақтық тасығышқа және оператордың орындалуының нәтижесін есеп беруде кейіптейсіңдер.

Зертханалық жұмыс №45. Оңтайлы жұмыс істеу үшін веб-жүйені теңшеу.

Мазмұнды басқару жүйесі негізінде динамикалық веб-сайт жобасын әзірлеу:

* Жобаның идеясын, оның мазмұнын, болашақ веб-жобаның әлеуетті келушілерін егжей-тегжейлі сипаттау

* Үлгіге негізделген тиісті дизайнды таңдау

* Жоба идеясын іске асыру үшін қажетті функционалдық компоненттер мен модульдерді сипаттау

Әдістемелік нұсқаулар:

Соңғы уақытта кеңінен таралған сайттарды басқару жүйелері күрделі веб-жүйелерді дамытуды жеңілдетеді, алайда олардың бірде-біреуі нарықты жаулап алған жоқ, себебі әрбір жүйе белгілі бір қолдану саласына арналған және бірқатар кемшіліктерге ие. Қазіргі заманғы сайтты басқару жүйелерінің жіктелуін келтіреміз, оларды құрудың әртүрлі тәсілдерінің артықшылықтары мен кемшіліктерін қарастырамыз.

Серверлік тарап Қосымшаларының платформаларының кемшіліктері, мысалы, төмен функционалдылық, визуалды басқару құралдарының болмауы, масштабтау қиындықтары іс жүзінде әрбір компания неғұрлым тиімді даму ортасын алу үшін өз құралдарын жасайды. Сондай-ақ, еркін таратылатын әзірлемелер мен жарияланған зерттеулердің жеткілікті саны бар. Барлық осындай жүйелер сайтты (немесе жүйенің архитектурасына байланысты сайттарды) басқаруды және өзгертуді жеңілдету, мазмұны мен дизайнға жедел өзгерістер енгізу үшін жобаланған, яғни базалық технологияларға жетіспейтін нәрсе, бірақ мұндай жүйелерді дамытудың екі негізгі бағытын бөліп көрсетуге болады: кәсіби емес пайдаланушылар үшін сайтты қарапайым құру және күрделі веб – жүйелерді құру үшін платформаны ұсыну.

Соңғы уақытта PHP: Php-Nuke және PostNuke негізделген жедел режимде сайтты құру және басқару құралдары кеңінен танымал болып келеді. Бұл бір жобаны дамытудың әртүрлі тармақтары болып табылатын ұқсас жүйелер. Бұл жүйелер PHP және деректер базасы (MySQL, mSQL, PostgreSQL, ODBC, Sybase немесе Interbase) үлгілерімен ұсынылған барлық мүмкіндіктерді пайдаланады, сайтты құру үшін стандартты функционалдық модульдердің массасын ұсынады: жаңалықтар, мақалалар, сауалнамалар, дауыс беру, файл менеджері, форум, іздеу модулі, сілтеме каталогы, пошта тарату және т. б. Жүйелер жақында пайда болды және әзірлеушілердің қажеттіліктеріне жауап береді. Олар сайтқа үшінші тарап модульдерін қосуға, өзімізді құруға және веб-интерфейс арқылы жүйені басқаруға мүмкіндік береді.

Әкімшілік интерфейс сайтты оңай және жылдам басқаруға мүмкіндік береді. Оны пайдалану үшін HTML немесе басқа веб-технологияларды білу қажет емес, өйткені дизайн үлгілерде, ал PHP файлдарындағы бизнес логика. Жасау қажет болған жағдайда мамандандырылған жүйесін ғана емес, орналасу қолданыстағы модульдер құру қажет өз PHP модульдер, бұл ұласады барлық құндылықтары мен кемшіліктері әзірлеу PHP, ол нашар сезімтал әзірлеу ірі жоба. Алайда, PHP кемшіліктерінен басқа, дизайнды басқару мүмкіндіктері жеткілікті дамымаған. Дайын модульдерді құрайтын және дизайндағы "тақырыптардың" бірін таңдап алатын кәсіби емес пайдаланушыларға бағдарланған, жүйе үлгілердің қашықтан өзгеруі туралы айтпағанда, дизайнды басқарудың ыңғайлы құралдарын ұсынбайды. Дизайнды өңдеу үшін серверге кіру және PHP файлдары сияқты үлгілерді өңдеу қажет. Осылайша, басқару жеделдігі ішінара болып табылады және тек сайттың мазмұнына, модульдердің жиынтығы мен параметрлеріне ғана қатысты, бірақ дизайнға емес.

Мұндай жүйелер сайтты басқару жүйелерінің келесі даму деңгейі болып табылады, үлкен перспективалар бар, бірақ қазіргі уақытта іске асыру кезінде бар браузер арқылы сайтты басқарудың барлық мүмкіндіктерін ұсынбайды (әсіресе бұл дизайнға қатысты), веб-қосымшалардың жеткіліксіз қуатты платформаларына негізделеді (көп жағдайда бұл PHP), сондай-ақ кеңейту және интеграция мүмкіндіктерінің әлсіз қолдауына ие (құрамдас архитектура, кластерлеуді қолдау). Осының барлығы осындай жүйелерді әмбебап қолдануға, әсіресе күрделі, таратылған веб-жүйелерді құруға пайдалануға кедергі келтіреді.

CMS «Joomla!» сайтты басқарудың ең қуатты жүйесі. CMS «Joomla!» жоғары интерактивті көптілді корпоративтік сайттарды, жеке беттерді, қоғамдастық сайттарын, бұқаралық ақпарат құралдары сайттарын, блогтарды, интернет-дүкендерді жылдам құруға арналған сайтты басқарудың ашық ортасы және басқару жүйесі. Атауы " Joomla!"суахилден"Бәрі бірге" деп аудару. 2007 жылдың 31 қазанында осы CMS "Open Source Content Management System Award"байқауында 2007 жылдың жеңімпазы болып жарияланды. Жүйе "Deface realtime compo" конкурсында Chaos Constructions' 2006 халықаралық фестивалінде бас тартуға және қауіпсіздікке тексерілді, бұл туралы тиісті сертификат берілді. Бұл CMS бүкіл әлемде сайттарды жасау үшін қолданылады: қарапайым жеке үй беттерінен бастап, күрделі корпоративтік веб-қосымшаларға дейін. Міне, "Joomla!"жеңімпаз атанды:

- * Корпоративтік сайттар мен порталдар
- * Жылжымайтын мүлік агенттіктерінің сайттары
- * Интернет-сауда
- Шағын бизнес сайттары
- * Коммерциялық емес ұйымдар
- Жеке және отбасылық үй беттері
- * Әлеуметтік порталдар
- * Журналдар мен газеттер

CMS «Joomla!»сіздің сайтыңыздың барлық аспектілерін басқару үшін қолданылады: сайттың мазмұны мен суреттерді қосудан бастап өнім каталогын жаңартуға дейін, нақты уақыт режимінде төлемдерді жүргізуден бастап өнімді резервтеуге дейін.

Сіздің бюджетіңіз бойынша әртүрлі нұсқаларды ұсынатын басқа CMS шешімдерінен айырмашылығы, CMS "Joomla!"барлық нұсқаларды қамтитын бір ғана шешім бар.

Көптеген CMS шешімдері кейбір жағдайларда 80-ге дейін модульдердің кең ауқымын көрсетеді. «Joomla!»көптеген кеңейтулерді ұсынады, қазіргі уақытта 4000-ға жуық. Олардың көпшілігі да тегін!

CMS «Joomla!»бұл әзірлеушілер мен достық пайдаланушылардың 100 000-нан астам (бұл клуб мүшелері ғана) бар белсенді қоғамдастық.

CMS орнатылған және іске қосылғаннан кейін, ол тіпті техникалық жағынан жарамсыз пайдаланушы үшін қарапайым: мазмұнды қосу немесе өңдеу, суреттерді немесе мультимедиа мазмұнын жаңарту, маңызды деректерді басқару. Мәтіндерді өңдеудің базалық дағдыларымен таныс кез келген адам CMS-те сайтты басқаруды оңай біле алады.

Қарапайым браузер-интерфейс арқылы сіз жаңа баспасөз хабарламаларын немесе жаңалықтарды, серіктестерге арналған жеке жабық беттерді, өз қызметтері туралы ақпаратты, тауарлардың фотосуреттерін, тауарлар каталогының жаңа позицияларын оңай қосуға, сондай-ақ сіздің сайтыңыздың беттерінде шексіз бөлімдер мен мазмұнды жасауға болады. Жүйенің барлық сұлулығы, ол бір жағынан, кең функционалдығы бар күрделі сайт құруға мүмкіндік береді, бірақ бұл ретте қысқа мерзімде және ең аз қаржылық шығындармен, ал екінші жағынан, жобаны басқаруды, оның жаңартылуын қарапайым операцияларға дейін тастауға мүмкіндік береді. Парадокс, бірақ қиындап, біз, сайып келгенде, басқарудың қарапайымдылығына, теңшеу мен жұмыстың жеңілдігіне келеміз! Қуатты, икемді, ашық көзі бар CMS пайда болуы-шын мәнінде веб-технологиялар саласындағы революция. Interwoven, Broadvision және, әрине, Microsoft сияқты фирмалар CMS нарығында 12000 еуродан басталатын ұсыныстармен басым болды. Бұл шындық болу үшін тым таңқаларлық көрінуі мүмкін, ал жоғары функционалды CMS платформалары қазіргі уақытта қол жетімді және лицензиялауға шығынсыз.

Зертханалық жұмыс №46. Коммерциялық емес жоспар жүйесінің модульдерімен жұмыс істеу әдістері.

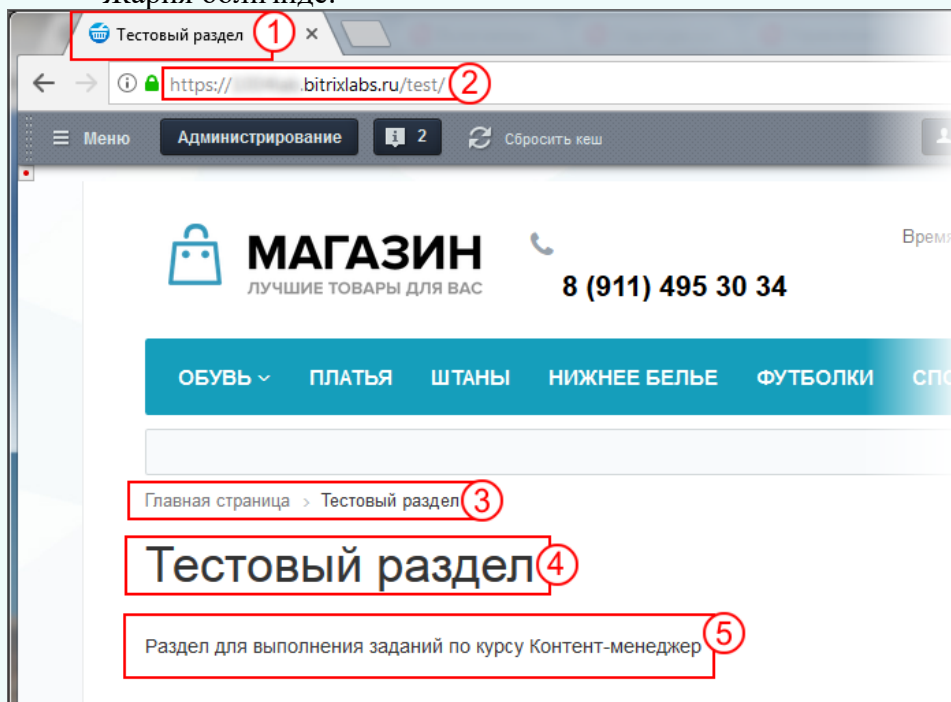
Негізгі тапсырма

Сынақ бөлімі, сынақ қалтасының атауы бар бөлім жасаңыз. Логикалық құрылымда бөлімді екінші деңгейде, физикалық құрылымда бірінші орында орналастырыңыз. Төменгі мәзірде бөлім-соңғы орында. Шолғыш терезесінің тақырыбы бөлімнің атауына сәйкес келеді. Беттің денесінде "Контент-менеджер курсы бойынша тапсырмаларды орындауға арналған бөлім" деген мәтінді көрсетіңіз.

Тапсырманы орындау нәтижесі:

Тапсырманы орындау нәтижесі:

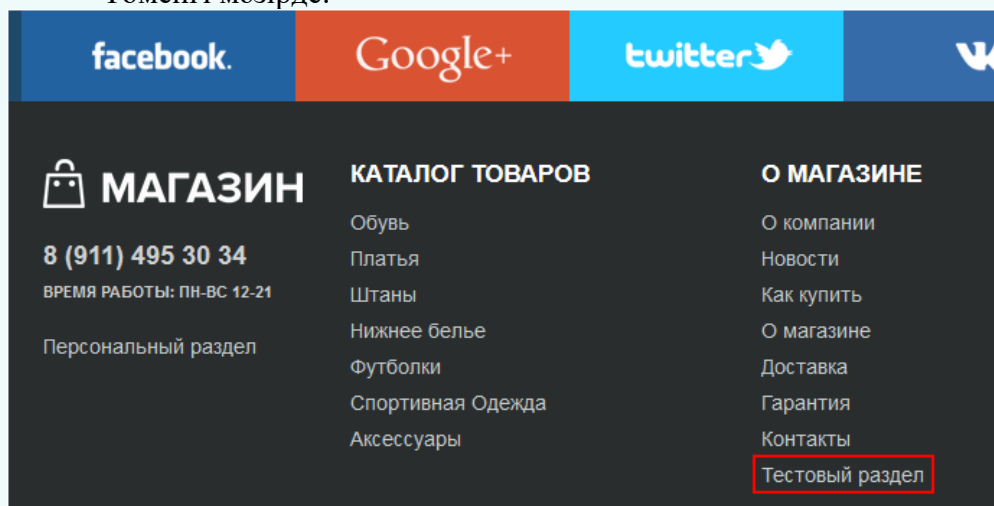
Жария бөлігінде:



Онда:

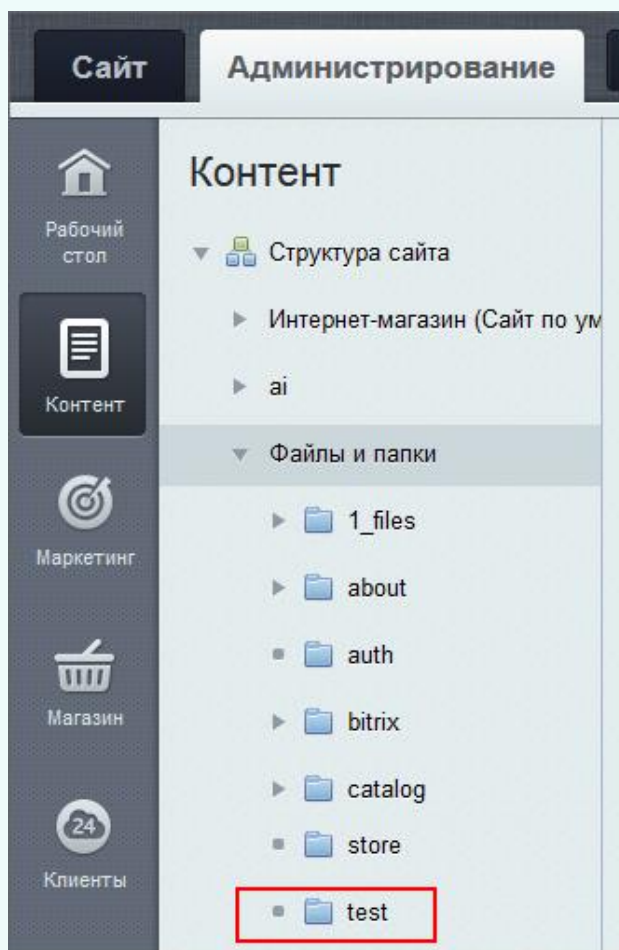
- 1-шолғыш терезесінің тақырыбындағы бөлімнің атауы;
- 2-сайттың мекен-жайы физикалық құрылымдағы test қалта сайттың түбірінде, бірінші деңгейде;
- 3-навигация тізбегінде логикалық құрылымда Екінші деңгейдегі бөлім, басты беттен кейін көрінеді;
- 4-бөлімнің индекстік бетінің атауы бөлімнің атауымен сәйкес келеді;
- 5-бөлім бетіндегі мәтін.

Төменгі мәзірде:



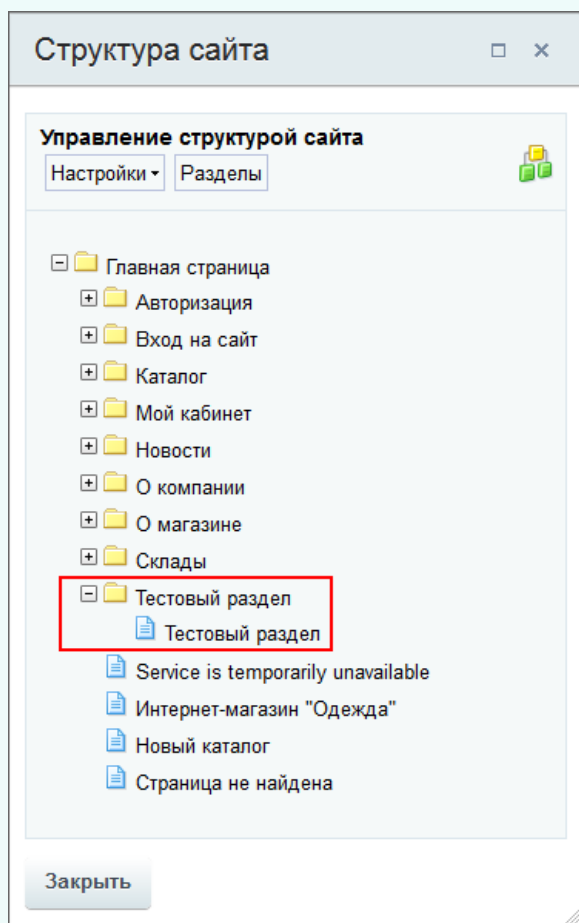
В рамках физической структуры:

Тапсырманы орындау нәтижесі:



Физикалық құрылым ішінде

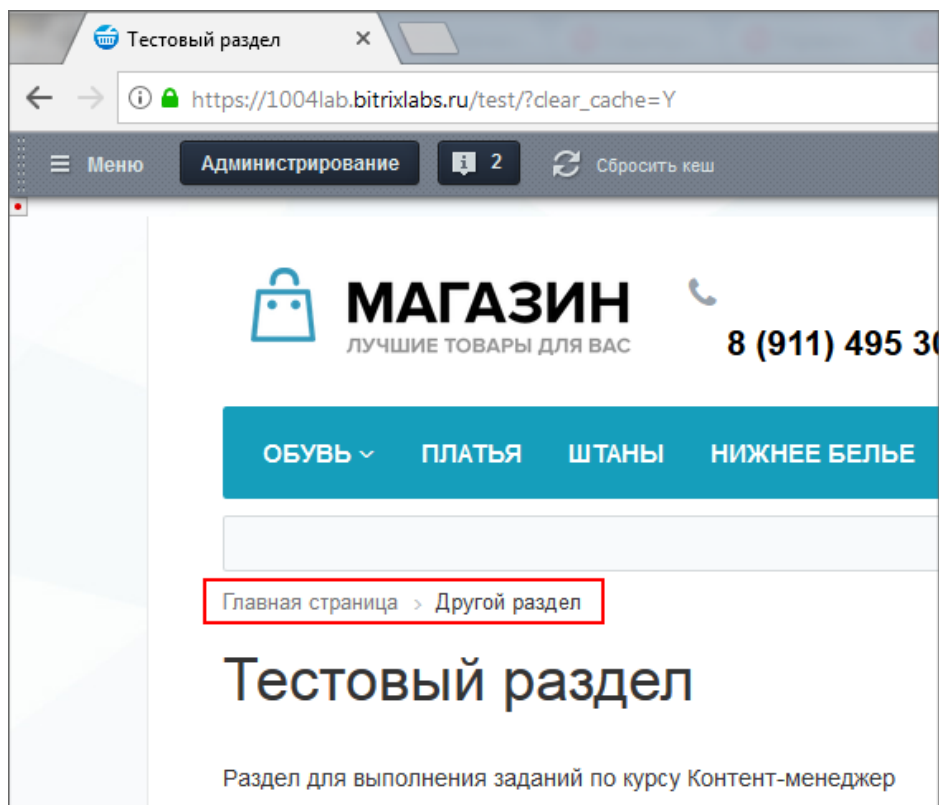
Тапсырманы орындау нәтижесі:



Қосымша

Бөлім атауын навигация тізбегіндегі сынақ бөлімін "басқа бөлімге" өзгертіңіз. Мәзір тармағы емес, бет атауы емес, браузер терезесі тақырыбының атауы емес, тек шарлау тізбегіндегі тармақ. (Содан кейін бұрынғы мәнді қалпына келтіріңіз.)

Результат замены имени раздела в Цепочке навигации:



Зертханалық жұмыс №47. LDAP сервері арқылы пайдаланушыларды импорттау әдістері.

Для добавления целого списка пользователей в систему служит специальная страница административного раздела **Импорт пользователей** (*Настройки > Пользователи > Импорт пользователей*).

Для импорта пользователей из **Active Directory / LDAP** выполните следующее:

1. На странице **Импорт пользователей** в качестве источника данных выберите **Active Directory / LDAP** и нажмите кнопку **Далее**.
2. На втором шаге импорта (закладка **Параметры импорта**) в выпадающем списке выберите сервер, который будет использован для импорта. После выбора сервера в форме появятся импортируемые поля. Снимите флажки с дополнительных полей, которые не нужно импортировать.

Источник данных | **Параметры импорта** | Импорт

Настройка параметров импорта

LDAP-сервер: Myserver [Создать](#)

Импортируемые поля:
(из настроек сервера)

- Активность
- Имя
- Фамилия
- E-Mail адрес

Дополнительно импортировать поля:

- (все)
- WWW-страница
- Телефон
- Мобильный
- Улица, дом
- Почтовый ящик
- Город
- Область / край
- Почтовый индекс
- Страна
- Наименование компании
- Департамент / Отдел
- Должность
- Рабочий телефон
- Рабочий факс
- Заметки администратора

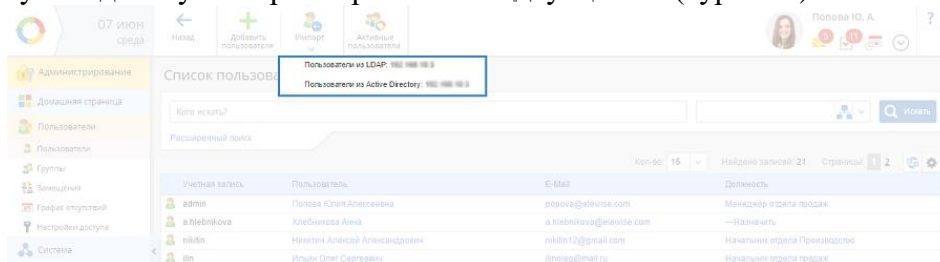
[< Назад](#) [Далее >](#)

3. Нажмите кнопку **Далее**. Мастер перейдет к третьему шагу. Будет автоматически произведен импорт, система сообщит об его завершении и количестве добавленных записей.

Примечание: Если пользователь заблокирован и существует на сайте, то его данные обновляются при импорте из AD/LDAP. Если пользователь неактивен и на сайте не существует, то он не импортируется на сайт.

Егер ELMA сервері LDAP қызметімен жұмыс істеу үшін орнатылса, осы қызмет арқылы ELMA пайдаланушылары тізімін импорттау мүмкіндігі бар. LDAP қосылымын қалай орнату керектігі туралы LDAP біріктіру бетінде толығырақ сипатталған.

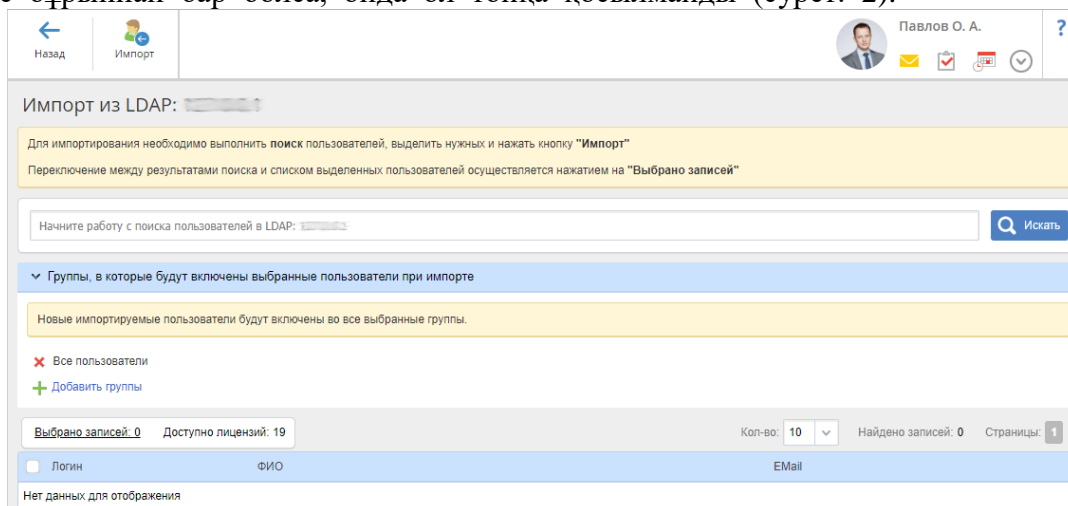
Пайдаланушыларды LDAP серверінен импорттау үшін Импорттау батырмасын басу қажет және ашылмалы мәзірден LDAP пайдаланушылары немесе Active Directory пайдаланушылары тармағын таңдау қажет (сурет. 1).



Сур. 1. LDAP пайдаланушыларды импорттау түймесі

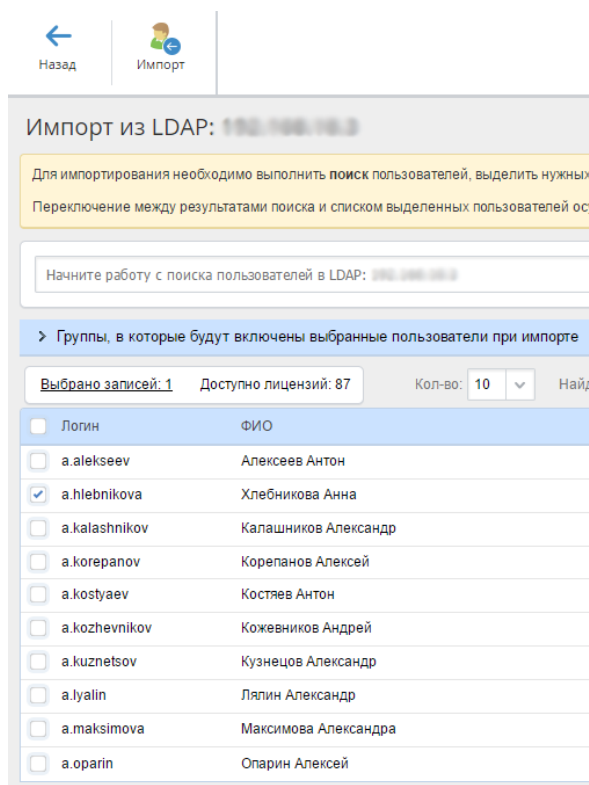
Әрбір жаңа кестеленген импортталатын пайдаланушы ELMA жүйесіндегі "барлық пайдаланушылар" тобына қосылады.

Ашылған бетте топты қосу түймесі арқылы жаңа пайдаланушылар импорттау кезінде автоматты түрде түсетін басқа топтарды қосу мүмкіндігі бар. Айта кету керек, тек жаңа пайдаланушылар таңдалған топтарға түседі. Егер пайдаланушы жүйеде бұрыннан бар болса, онда ол топқа қосылмайды (сурет. 2).



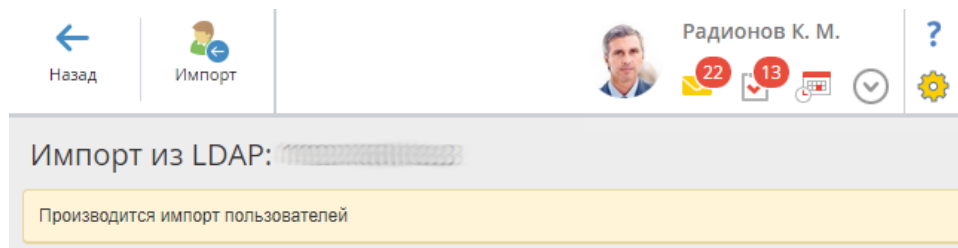
Сур. 2. LDAP пайдаланушыларды импорттаудың бірінші кезеңі

Іздеу түймесін басқаннан кейін (сурет. 3) импорт үшін қол жетімді пайдаланушылар тізімі пайда болады (сурет. 3). Тізімде қажетті пайдаланушылар құсбелгісін белгілеп, Импорттау түймесігін басу керек. Таңдалған пайдаланушыларды LDAP-дан ELMA жүйесіне импорттау процедурасы іске қосылады.

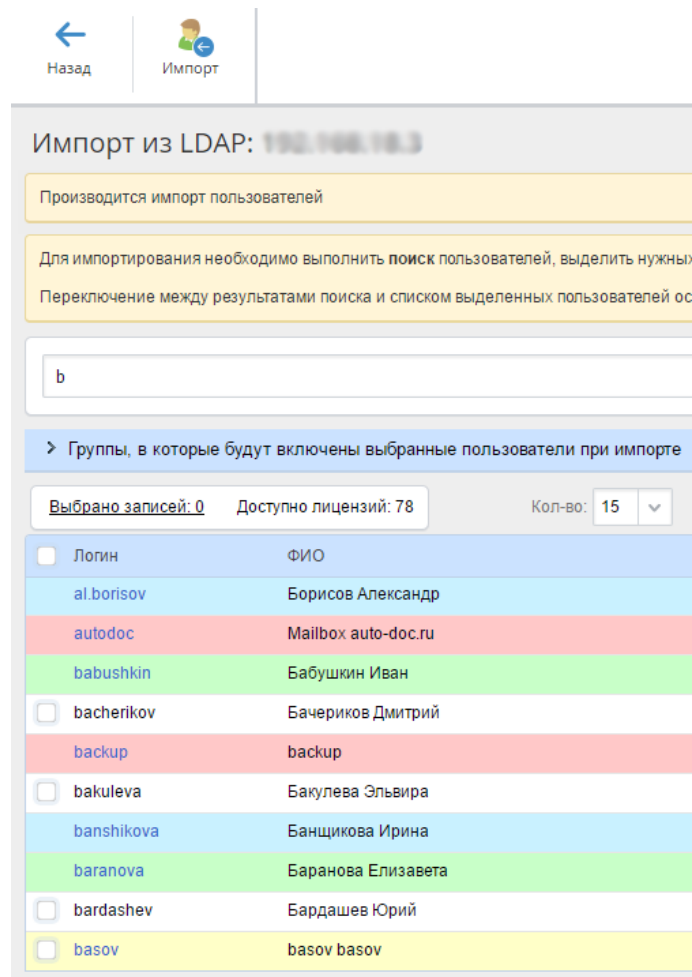


Сур. 3. LDAP-дан ELMA импорттау үшін табылған пайдаланушылар тізімі

Импорт рәсімін іске қосқаннан кейін тиісті хабарлама бейнеленеді (сурет. 4). Импорт фондық режимде жүзеге асырылады: импорттың аяқталуын күтпестен, беттегі жұмысты жалғастыруға болады.



- Сур. 4. Пайдаланушыларды импорттаудың басталғаны туралы хабарлама
-
- LDAP-дан импортталған пайдаланушылардың есептік жазбалары тікелей ELMA жүйесінде сақталмайды. Мұндай пайдаланушыларды жүйеде аутентификациялау кезінде олар енгізген ақпарат LDAP-та сақталатын деректермен салыстырылады.
- Іздеу жолында пайдаланушы логинінің бір бөлігін теріп, Іздеу батырмасын басуға болады-жүйе логиндерінде символдардың енгізілген комбинациясы бар пайдаланушылар тізімін көрсетеді –
- Егер импорт рәсімі екінші, үшінші және т.б. рет орындалса, пайдаланушылар тізімде түсі бойынша ерекшеленеді (сурет. 5):
- * жасыл-пайдаланушы ELMA-ға сәтті импортталды;
- • қызыл-пайдаланушы ELMA-ға импортталды, бірақ бұғатталған немесе белсендірілмеген, яғни жүйеде авторлана алмайды;
- * ақ-пайдаланушы әлі импортталмаған;
- * сары-пайдаланушы ELMA жүйесінде де, LDAP базасында да бар, бірақ деректер сәйкес келмейді. Импорттау кезінде жүйеде Пайдаланушының деректері (соның ішінде пароль) LDAP мәліметтеріне ауыстырылады;
- * көк-пайдаланушы ELMA жүйесінде және LDAP базасында бар, оның деректері толығымен сәйкес келеді. Пайдаланушы бұрын басқа модульден (Active Directory сияқты) импортталса, бұл түс пайда болады.
-

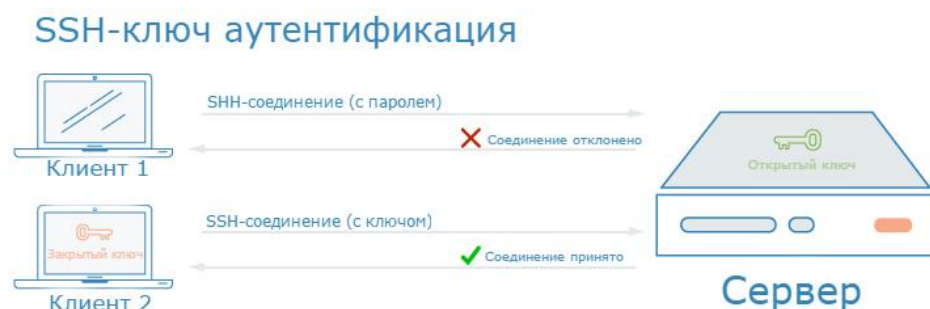


Сур. 5. Қайта импорттау кезіндегі пайдаланушылар тізімі

Зертханалық жұмыс №48. Сайтты рұқсатсыз кіруден қорғау құралдары.

SSH-ключи

Технология негізінде-пароль арқылы аутентификацияға балама ретінде түпнұсқалығын тексеру үшін қолданылатын криптографиялық кілттердің жұбы. Кіру жүйесі аутентификацияға дейін жасайтын жабық және ашық кілттерді пайдаланады. Жабық кілт сенімді пайдаланушымен құпия сақталады, ал ашық кілт қосылу керек кез келген SSH серверінен таратылуы мүмкін.



SSH кілттері арқылы аутентификацияны теңшеу үшін ашық пайдаланушы кілтін сервердегі арнайы директорияға қою керек. Пайдаланушы серверге қосылғанда, SSH байланыс сұрауын көреді. Содан кейін ол қоңырау жасау және жіберу үшін ашық кілтті пайдаланады. Қоңырау-серверге кіру үшін тиісті жауап қажет шифрланған хат. Хабарға тек жабық кілт ұстаушысы ғана дұрыс жауап бере алады. Яғни, ол тек қоңырау қабылдай алады және тиісті жауап жасайды. Ашық кілт хатты шифрлау үшін пайдаланылады, бірақ бұл хат шифрды шифрлай алмайды.

Қоңырау және жауап пайдаланушы үшін білінбей өтеді. Сізде әдетте шифрланған түрде сақталатын жабық кілт бар `~/.ssh/`, сіздің SSH клиенті серверге дұрыс жауап жібере алады.

SSH кілті қауіпсіздікті қалай арттырады?

SSH арқылы аутентификацияның кез келген түрі толығымен шифрланған. Алайда, егер пароль негізінде аутентификация рұқсат етілген болса, зиянкестер сервердің деректеріне жете алады. Қазіргі заманғы есептеу қуаттарының көмегімен серверге дұрыс пароль тапқанға дейін комбинацияны енгізе отырып, бұзу әрекеттерін автоматтандыру арқылы кіруге болады.

SSH кілттері бойынша аутентификацияны орнатқаннан кейін сіз парольдерді ұмыта аласыз. Кілттер құпия сөздерге қарағанда әлдеқайда көп деректер биттері бар, яғни взломщики таңдауға тиіс комбинациялар саны айтарлықтай көп. Көптеген SSH кілттерінің алгоритмдері қазіргі заманғы есептеу техникасымен шектелмейтін болып саналады, себебі олар сәйкестіктерді таңдау үшін тым көп уақытты талап етеді.

SSH жүзеге асыру қаншалықты қиын?

SSH кілттерін орнату өте оңай. Олар Linux және Unix серверлік орталарына қашықтан кіру әдісі ретінде жиі қолданылады. Жұп кілттер компьютерде жасалады, содан кейін бірнеше минут бойы серверлерге ашық кілтті жібере аласыз.

Міне, кілттерді орнатудың негізгі қадамдары:

1. Компьютеріңізде бірнеше кілтті жасау үшін пәрменді енгізу керек:

```
ssh-keygen -t rsa
```

2. Сіз кілттерді генерациялау командасын енгізгеннен кейін, файлды қайда сақтау және қандай негізгі фразаны таңдау сияқты бірнеше сұраққа жауап беру керек. Жалпы, нәтиже осылай көрінеді:

```
ssh-keygen -t rsa
```

```
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/demo/.ssh/id_rsa):
```

```
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
```

```
Your identification has been saved in /home/demo/.ssh/id_rsa.
```


Your public key has been saved in /home/demo/.ssh/id_rsa.pub.

The key fingerprint is:

4a:dd:0a:c6:35:4e:3f:ed:27:38:8c:74:44:4d:93:67 demo@a

The key's randomart image is:

+++[RSA 2048]-----+

| .oo. |

| . o.E |

| + . o |

| . = = . |

| = S = . |

| o + = + |

| . o + o . |

| . o |

| |

+-----+

3. Кілттер болған кезде ашық кілтті пайдаланатын виртуалды серверге орналастырыңыз.
Бұл ұқсас нәрсе алыңыз:

```

The authenticity of host '12.34.56.78 (12.34.56.78)' can't be established.

RSA key fingerprint is b1:2d:33:67:ce:35:4d:5f:f3:a8:cd:c0:c4:48:86:12.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '12.34.56.78' (RSA) to the list of known hosts.

user@12.34.56.78's password:

Now try logging into the machine, with "ssh 'user@12.34.56.78'", and check in:

~/ssh/authorized_keys

```

Егер сіз серверлерде аутентификация паролін қажет деп ойласаңыз, құпия сөзді енгізу әрекеттерінің санын шектейтін Fail2ban типті шешімдерге қараңыз.

Фаерволдар

Брандмауэр-желілік трафикті сүзетін және желіге қатынауды бақылайтын бағдарламалық немесе бағдарламалық-аппараттық қамтамасыз етудің бір бөлігі. Бұл ерекшеліктерден басқа әрбір ашық портқа кіруді бұғаттау немесе шектеу дегенді білдіреді.



Әдеттегі серверде фауервол компоненттері әдепкі бойынша іске қосылады. Оларды топтарға бөлуге болады:

- Интернетте әркім қосыла алатын ашық қызметтер жиі анонимді. Жақсы мысал-сіздің сайтқа кіруге рұқсат беретін веб-сервер.
- Тек белгілі бір жерлерден немесе өкілетті пайдаланушылардан қол жетімді жабық қызметтер. Мысал-сайтты немесе деректер қорын басқару тақтасы.
- Сервердің ішінде қол жетімді ішкі қызметтер, сыртқы көздерге қол жеткізусіз. Мысалы, тек жергілікті қосылымдарды қабылдайтын дерекқор.

Брандмауэр бағдарламалық жасақтама мен деректерге қол жеткізу жоғарыда көрсетілген санаттарға сәйкес шектелетініне кепілдік береді. Жабық қызметтер көптеген параметрлер бойынша реттелуі мүмкін, бұл қорғауды құруға икемділік береді. Пайдаланылмайтын порттар үшін көптеген конфигурацияларда блоктауды орнатуға болады.

FaerVol қауіпсіздікті қалай арттырады?

Брандмауэрлер-кез келген сервер конфигурациясының ажырамас бөлігі. Бағдарламалық жасақтама ішкі қорғаныс функционалына ие болса да, faerVol қосымша қорғаныс деңгейін қамтамасыз етеді.

Мүқият реттелген брандмауэр сіз өзіңіз ерекшелік тағайындамайтын барлық кіруді блоктайды. Шабуылға осал компоненттер, фаерволмен жабылған, серверге шабуыл бетін азайтады.

Ферволды жүзеге асыру қаншалықты қиын?

LAMP серверлерінде көптеген брандмауэр бар. Жалпы, фаерволды орнату бірнеше минутты алады және екі жағдайда қажет: серверді бастапқы теңшеу кезінде және жұмыс істеп тұрған сервердің нақты қызметтерін өзгерту кезінде.

Бұл мақалада біз нақты ферволдарды ұсынбаймыз, бұл жеке әңгіме тақырыбы.

VPN және Private networking

VPN (виртуалды жеке желі) - қашықтағы компьютерлер мен ағымдағы Байланыс арасында қорғалған байланыс жасау тәсілі. Сіз қорғалған жергілікті желіні пайдаланғандай сервермен жұмысыңызды теңшеуге мүмкіндік береді.

Виртуальная закрытая сеть (VPN)



VPN қауіпсіздікті қалай арттырады?

Егер жеке және ортақ желі арасында таңдасаңыз, бірінші нұсқа әрқашан жақсырақ. Бұл жағдайда дата-орталықтың пайдаланушылары бір желімен байланысты екенін есте сақтау керек, сіз серверлер арасындағы қауіпсіз байланыс үшін қосымша шаралар қабылдай отырып, өзіңізді тәуекелден барынша арылту керек.

VPN пайдалану, шын мәнінде, сіздің серверлеріңіз ғана көре алатын жеке желіні жасау тәсілі. Байланыс толығымен жеке және қауіпсіз болады. Сонымен қатар, VPN жеке қолданбалар мен қызметтер үшін олардың трафигі виртуалды интерфейс арқылы өтуі үшін теңшеуге болады. Осылайша, компания ішіндегі процестерді қауіпсіздендіруге болады, тек клиенттік тарап үшін қоғамдық қолжетімділік ашық, ал сервер жұмысының ішкі бөлігін VPN жасыру. Насколько сложно реализовать VPN?

Дата-орталықтардың жеке желілері, қызмет ретінде-бұл оңай. Күрделілік тек серверіңіздің параметрлерімен, оның интерфейсімен, фаерволымен және сіз пайдаланатын бағдарламалармен шектеледі. Дата-орталықтарда көптеген серверлерді біріктіретін үлкен жеке желілер пайдаланылатынын есте сақтаңыз.

VPN болсақ, бастапқы орнату сәл қиын, бірақ көп жағдайда жоғары қауіпсіздік жұмсалған қаражат. VPN-дегі әрбір сервер қорғалған байланыс үшін қажетті Жалпы деректер мен қауіпсіздік конфигурацияларын пайдаланып Орнату және орнату керек. VPN іске қосылғаннан кейін, VPN туннелін пайдалану үшін бағдарламалық жасақтаманы теңшеу керек.

PKI және SSL/TLS шифрлау

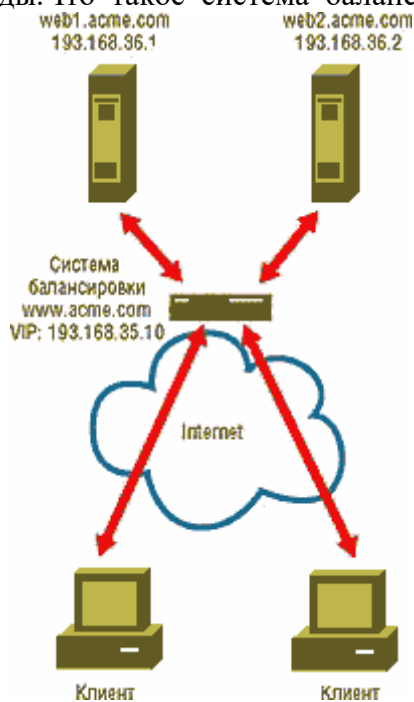
Ашық кілттердің инфрақұрылымы (PKI) - адамдарды сәйкестендіру және берілетін деректерді шифрлау үшін сертификаттар жасауға, басқаруға және тексеруге арналған жүйелер жиынтығы. Аутентификациядан кейін, олар шифрланған байланыс үшін де пайдаланылуы мүмкін.



Зертханалық жұмыс №49. Жоғары жүктеме кезінде сайттың өміршеңдігін қамтамасыз ету құралдары.

Сайттың жылдамдығын арттырудың бірнеше жолы бар: өткізу жолағын ұлғайтуға, жоғары өнімді желілік жабдықты орнатуға, Web үшін тиімді қосымшаларды әзірлеуге, Web-сервердің бағдарламалық және аппараттық компоненттерін оңтайландыруға және жаңартуға, сондай-ақ Web ортасында кәштеу технологиясын қолдануға болады. Web торабының өнімділігін арттырудың тағы бір жолы Web-серверлер мен сайттар санын көбейту және оларға материалдардың "айналы" көшірмелерін орналастыру болып табылады. Осылайша, жалпы жүктемені жүйенің барлық компоненттері бойынша бөлуге және сервердің клиенттік сұраныстарды өңдеудің ішкі рәсімдерін орындау кезінде ақпаратты қайтару уақытын қысқартуға болады. Сонымен қатар, қолданыстағы серверлер де сақталады, өйткені оларды пайдаланудан шығару және жаңамен ауыстыру қажет емес. Сонымен қатар, бір-бірімен байланысқан, бір-бірімен байланысқан, бір-бірімен байланысқан, бір-бірімен байланысқан, бір-бірімен байланысқан. Web-серверлер арасында жүктемені бөлу үшін әдетте кез келген клиентке сайтты құрайтын кез келген Web-сервердің IP-мекенжайын айналмалы түрде беру мүмкіндігін қарастыратын циклдік таңдау (round-robin feature) деп аталатын DNS функциясы қолданылады; нәтижесінде барлық Web-серверлер трафикпен теңдей жүктеледі. Алайда, бұл механизм жекелеген Web-серверлердің аппараттық және бағдарламалық компоненттерінің мүмкіндіктері тең емес орталарда жеткілікті тиімді емес. Pentium III екі 450 мегагерц процессорымен және 1 Гбайт сыйымдылығы бар жадымен жабдықталған жүктемелі ортада Windows 2000 Server жүйесі бір 300 мегагерц Pentium II және 256 Мбайт жады процессорымен Windows NT Server жүйесіне қарағанда ауыр жүк көтере алады. Алайда, осы жүйелер арасында DNS қызметінің циклдік үлгісі процедурасы бойынша ешқандай айырмашылық жоқ; сонымен қатар, бұл функция қандай да бір Web-сервердің қол жетімділігі туралы "көріністері жоқ", себебі компьютердің жұмыс істеп тұрғанын немесе істен шыққанын анықтай алмайды.

Көп ұзамай Жабдықтаушылар жүктемені теңестіру жүйелерін - бағдарламалық өнімдерді шығаруды ретке келтірді, олар жүктемені бірнеше серверге бөледі. Сонымен қатар, олар Web-серверлердің бас тартуға төзімділігін арттырады: бір машина істен шыққан жағдайда деректер пакеттерін басқа серверге немесе сайтқа жібереді. Осылайша, күту уақыты азаяды, ал өңделмеген сұраулар саны азаяды. Жүктемені теңгеру жүйесін тек бір Web-сайт болған кезде де, бірнеше тораптармен жұмыс істеген кезде де пайдалануға болады. Жүктемені теңестіру жүйесі қандай және олар қалай жұмыс істейтіні туралы түсінік ала отырып, жүктемені теңестіру құралын таңдау кезінде ескеру қажет олардың ең маңызды сипаттамаларын анықтауға болады. Что такое система балансировки нагрузки



1-ЭКРАН. Бір сайт үшін теңгеру жүйесі.

Web-серверлердің жүктемесін теңгеру жүйесі - бұл ақпараттық ресурстың айналы көшірмелері сақталатын машиналар тобынан ең аз жүктелген немесе неғұрлым қолайлы Web-серверге клиенттік сұраныстарды бағыттауға арналған құрал. Клиент серверлердің тұтас тобына хабарласады деп күдіктенбейді: олардың барлығы оған бір бірыңғай виртуалды сервер түрінде ұсынылады. Мысалы, біз бір Web-сайтқа қызмет көрсетеміз және екі Web-сервер бар делік: web1.acme.com IP мекенжайы 193.168.36.1 және web2.acme.com IP мекенжайы 193.168.36.2. Бұл конфигурацияны 1-суретті көрсетеді. Internet пайдаланушыларына біздің сайтты ұсына отырып, жүктемені теңестіру жүйесі виртуалды компьютердің атауын пайдаланады (бұл атау болсын www.acme.com), сондай-ақ виртуалды IP-мекен - жайы (VIP-мекен-жайы-айталық, 193.168.35.10). Виртуалды жүйенің атын және тиісті VIP-мекенжайды біздің екі веб-сервермен байланыстыру үшін жүйенің атын және оның VIP-мекенжайын DNS серверінде жариялауымыз керек. Жүктемелерді теңгерімдеу жүйесі үнемі жүктемені және әрбір Web-сервердің дайындық деңгейін бақылайды. Торапқа қашан www.acme.com келуші қарайды, оның сұрауы Web-серверлердің біріне емес, жүктемені теңгеру жүйесіне түседі. Бұл жүйе сұранысты қай серверге жіберу туралы шешім қабылдайды. Бұл ретте, ол әрбір қамқорлықтағы серверді жүктеу сияқты критерийлерді басшылыққа алады, сондай-ақ әкімші қалыптастырған шарттар мен ережелерді сақтайды. Содан кейін жүктемені теңгеру жүйесі клиенттің сұрауын тиісті серверге жібереді (әдетте, ол сервердің жауабын клиентке жібереді, бірақ бұл нақты іске асыруға байланысты).

Жүктемені бөлу жүйелері бірнеше Web-сайттардың жүктемелерін теңестіруді қамтамасыз ете алады. Бірнеше сайттарды қолдану серверлерді-"дублерлерді" (айналы Web-серверлерді) сайтқа келушілерге жақын орналастыруға және сайт пен клиенттер арасында ақпарат алмасу кезінде кідірістерді қысқартуға мүмкіндік береді. Сонымен қатар, бірнеше Web-сайттар Бар болған жағдайда олардың арасында жүктемені біркелкі бөлу, сондай-ақ сайт жұмысында бұзушылықтар болған жағдайда (энергиямен қамтамасыз ету жүйесіндегі іркілістер немесе есептеу орталығындағы Internet-пен байланысты жоғалтудан болған болса да) олардың дайындығының жоғары деңгейін және бас тартуға төзімділігін қамтамасыз ету мүмкіндігі пайда болады. 2-суретте көрсетілгендей бірнеше сайттар болған жағдайда, барлық сайттардағы жүктемені теңгеру жүйелерінің бір жалпы виртуалды жүйенің атауы бар, бірақ әр түрлі VIP-адрестері бар. Мәселен, Нью-Йорктегі 1 сайтында 1 теңестіру жүйесі виртуалды басты машинаның аты берілді www.acme.com VIP-мекен-жайы 193.168.35.10. Ал виртуалды негізгі машинаның атын пайдаланатын Лос-Анджелесте 2 сайтында 2 теңгерушісі басқа VIP-мекен-жайы бар-193.200.1.10. Барлық баланстау жүйелерінің жергілікті Web-серверлермен байланысы бір сайтпен болған жағдайдағыдай орнатылады. Жергілікті серверлердің жүктемелеріне мониторинг жүргізумен қатар, теңгеру жүйелері өздерінің "әріптестерімен" басқа сайттарда конфигурация және жүктеу туралы ақпаратпен алмасады; бұл ретте сайттың дайындық деңгейі де тексеріледі. Нәтижесінде барлық теңгеру жүйелері өз иелігінде жүктемелерді бөлудің және әртүрлі тораптардың жұмысқа дайындығының жалпы көрінісі болады. Бірнеше сайттар Бар болған жағдайда жүктемелерді теңестіру жүйесі жиі параллельді тағы бір тапсырманы орындайды: виртуалды жүйенің атауына қызмет көрсететін DNS серверлерінің рөлін өзіне алады. Осы атауды көрсеткен клиенттен DNS сигналын алып, теңгерімдеу жүйесі клиентке жүктеменің ағымдағы деңгейін, клиенттен алшақтық дәрежесін және басқа параметрлерді ескере отырып, ең қолайлы сайттың VIP-мекенжайын қайтарады. Содан кейін клиент автоматты түрде осы торапқа қол жеткізе алады.

Зертханалық жұмыс №50. Іздеуді оңталайндыру және сайтқа кіруді бақылау құралдары.

Серверді таңдау

Серверлердің сыртқы және ішкі мониторингі барысында алынған ақпаратты ескере отырып, жүктемені теңестіру жүйесі клиенттік сұранысты өңдеуді басқалардан жақсы шеше алатын серверді бөлектей алады. Егер пулдың барлық серверлерінің аппараттық және бағдарламалық құрамдастарының жұмыс сипаттамалары бірдей болса, ол сервердің күйін ескере отырып, айналмалы тізім қағидаты бойынша кезекті сұрауды өңдеу үшін серверді бөліп алатындай жүктемені теңгеру жүйесін баптауға болады. Бірақ егер баланс жүйесі Pentium III процессоры бар серверді және Pentium Pro процессоры орнатылған тағы бір машинаны басқарса, бұл жүйені басқа режимде жұмыс істеу үшін реттеуге болады: қуатты сервер сұраулардың көп санын жеткізеді. Басқаша айтқанда, бұл тек салмақ коэффициенттерін ескере отырып, айналмалы тізімнің бірдей принципі туралы.

Жүктемені теңгерудің заманауи жүйелері әкімшіге серверді таңдау ережелерін өз қалауы бойынша анықтауға мүмкіндік береді. Мысалы, бұл ережелерге ЦП және жадты жүктеу коэффициенті, TSP ашық қосылыстарының саны және сол немесе басқа сервердің желілік интерфейстік төлеміне түсетін пакеттер саны сияқты критерийлерді қосуға болады. Баланстау жүйесі серверлердің жүктелу деңгейін анықтайтын Формула шамамен былай көрінуі мүмкін: $(10 * \text{ЦП пайдалану деңгейі}) + (3 * \text{жадты пайдалану деңгейі}) + (6 * \text{ашық TSP-қосылыстар саны}) + (3 * \text{берілген пакеттер саны}) = \text{серверді жүктеу}$. Клиенттен

сұрау алған кезде жүктемені теңгеру жүйесі осы формула бойынша әрбір сервердің жүктемесін есептейді және ең аз жүктемемен серверге сұрау жібереді.

Кейбір жағдайларда теңгерімдеу құралы клиенттің сұратуын орындау үшін серверді тағайындағаннан кейін және Клиент бастапқы қосылымды орнатқаннан кейін, қолданбалы бағдарламаны орындауды қамтамасыз ету үшін теңгерімдеу жүйесі клиент пен сервер арасындағы трафикті үнемі ұстап тұру қажет. Мұндай байланыс тұрақты деп аталады. Электрондық кітап дүкеніне келуші сатып алуды таңдайды және виртуалды себетке үш кітапты қояды делік. Егер клиент сұранымын өңдейтін сервер себеттің мазмұны туралы ақпаратты жергілікті кэштесе, жүктемені теңгеру жүйесі клиентпен кейінгі алмасу үшін басқа серверді бөле алмайды, тіпті егер торап серверлері жүктемелерінің өзгерген арақатынасын талап етсе де. Өйткені, мұндай бағыттау кезінде клиенттің виртуалды қоржынының мазмұны туралы деректер жоғалады, өйткені жаңа серверде бұл мәліметтер жоқ. Демек, теңгерімдеу жүйесі әрбір клиент қандай сервермен байланысатыны туралы ақпаратты тіркеп, оны белгілі бір уақыт ішінде жадта сақтауы тиіс (сақтау уақытын әкімші клиенттердің әдеттегі мінез-құлқы мен қолданбалы бағдарламаның ерекшеліктері сияқты факторларды ескере отырып көрсетеді). Тұрақты (тұрақты) қосылыстар функциясын белсендірген кезде, ол жүктемені теңгерудің басқа ережелерін үнемі алып тастайды.

Тұрақты қосылымды ұйымдастыру кезінде жүйенің негізгі міндеті Клиентті анықтау және тиісті идентификаторды сұрау серверімен байланыстыру болып табылады. Әдетте, жүктемені теңгеру жүйелері клиенттің идентификаторы ретінде ол қолданатын жіберушінің IP-мекенжайын пайдаланады. Бірақ барлық мәселе-жіберушінің мекен-жайы міндетті түрде клиенттің нақты IP-мекен-жайына сәйкес келмейді. Көптеген компаниялар мен провайдерлер Web-трафикті бақылауға және өз пайдаланушыларының IP-адрестерін бөгде көздерден жасыруға тырысып, делдал серверлерді орнатады. Егер біздің гипотетикалық Web торабына America Online (AOL) қызметінің 500 клиентінен, сондай-ақ басқа компанияны ұсынатын 10 клиенттен сұрау түскен жағдайда, серверлер жүктемесінің балансы бұзылады. Теңгерім жүйесі жөнелтушінің бір мекен — жайы бар AOL-ден 500 Клиентті бір сервермен, ал қалған 10 клиентті екіншісімен байланыстырады. Бұл мәселені шешу үшін, егер TCP порттарының IP мекенжайларын және нөмірлерін идентификациялау құралдарымен жабдықталған жүктемені теңгеру жүйесін пайдалансаңыз, шешуге болады. Мұндай жүйелер соңғы бір проху-сервер арқылы Internet-ке шыққан жағдайда да клиенттерді тануға қабілетті. Мұндай идентификация мүмкін, себебі әрбір TCP байланыс бірегей жіберушінің IP-мекен-жайы және TCP порт нөмірі бар. HTTP хаттамасы бойынша қорғалған байланыс сеансын өткізетін Клиентті сәйкестендірудің тағы бір тәсілі Secure Sockets layer (SSL) хаттамасы бойынша пайдаланушының байланыс сеансының сәйкестендіру нөмірін тіркеу болып табылады. SSL протоколы әрбір орнатылған байланыс сеансын арнайы идентификаторды тағайындайды және виртуалды дүкендер үшін қолданбалы бағдарламалар жиі осы Хаттаманы пайдаланады. Тұрақты қосылыстарды қолдаудың ең заманауи құралы — бұл желі арқылы таратылатын Web cookie-файлдар. Бұл файлдарда клиент туралы мәліметтер де, басқа да деректер де бар (мысалы, клиент соңғы рет қандай сервермен байланысқаны туралы). Cookie файлдарының мазмұнын талдау жүктемені теңестіру жүйесіне клиенттерді анықтауға және олар үшін ең қолайлы серверді таңдауға көмектеседі. Cookie-файлдармен жұмыс істеу құралдарымен жабдықталған жүктемені теңгеру жүйесін жеткізушілер қатарына Alteon WebSystems, ArrowPoint Communications, F5 Networks және Resonate сияқты компаниялар кіреді.

Ақырында, серверді таңдаудың тағы бір жолы бар-тікелей байланыс (immediate binding) деп аталады. Осы әдіске сәйкес жүктемені теңгеру жүйесі клиент үшін серверді таңдайды және жүйе клиенттен TCP SYN пакетін алатын сәтте оған сұрау жібереді. Сонымен қатар, баланстау жүйесі сервердің жүктемені бөлудің берілген ережелерін, сондай-ақ клиенттен алынған TCP SYN пакетінде қамтылған IP-мекенжайын басшылыққа

ала отырып, серверді таңдайды. Бұл әдіс жоғары жылдамдықты қамтамасыз етеді, бірақ оны қолданғанда теңгеру жүйесі басқа ақпаратты талдай алмайды: атап айтқанда, SSL протоколы бойынша байланыс сеансының идентификаторы, cookie-файлдың мазмұны, URL мекенжайы және қолданбалы бағдарлама деректері. Клиент туралы егжей-тегжейлі ақпарат алу үшін және, тиісінше, серверді дәлірек таңдау үшін, жүктемені теңестіру жүйесі қолданба деңгейінің ақпаратын талдау үшін уақыт қажет. Кейінге қалдырылған байланыстыру әдісі бойынша серверді таңдау кезінде жүктемені теңгерімдеу жүйесі растаушы хабарламалармен үш кезеңді алмасу аяқталғаннан кейін және оның және Клиент арасындағы байланыс орнатылғаннан кейін ғана серверді тағайындау туралы шешім қабылдайды. Егер клиент үшін серверді таңдағанға дейін қолданбалы бағдарлама деңгейінің ақпаратын зерттесе, жүйе жарияланатын материалдардың мазмұнын ескеруі мүмкін.

Трафикті бағыттау

Сонымен қатар, Gigabit Ethernet технологиясы, Gigabit Ethernet, Gigabit Ethernet, Gigabit Ethernet, Gigabit Ethernet, Gigabit Ethernet, Gigabit Ethernet, Gigabit Ethernet, Gigabit Ethernet, Gigabit Ethernet, Gigabit Ethernet, Gigabit Ethernet, Gigabit Ethernet және т.б. Осы әдістердің әрқайсысы жүктемені теңестіру арқылы трафикті қайта бағыттауды қарастырайық.

МАС. Бұл әдіс әрбір Web-сервер өзінің Жеке IP-мекен-жайымен бірге кері байланыстың интерфейстік мекен-жайы (loopback interface address) теңгеру жүйесінің VIP-мекен-жайын пайдаланған жағдайда жүктемені теңгеру жүйесімен іске асырылуы мүмкін. Клиенттен пакетті алып және оған тиісті серверді тағайындап, теңгеру жүйесі осы пакетте MAC-адресін алушының Mac-адресін тиісті сервердің адресімен ауыстырады, содан кейін пакетті бөлінген серверге жібереді. Пакетте клиенттің IP-мекен-жайы бар, сондықтан клиентке тікелей жауап беру үшін сервер алушының IP-мекен-жайы ретінде клиенттің бастапқы IP-мекен-жайын пайдаланады. Алайда, жіберушінің IP-мекенжайы ретінде сервер жүктемені теңгеру жүйесінің VIP-мекенжайын көрсетеді, егер трафик клиентке дәл одан келіп түссе. Осылайша, клиенттен келесі пакет оған жауап бермеген серверге, жүктемені теңгеру жүйесіне жіберіледі.

НАТ. Бұл әдісті қолданған кезде теңгерімдеу жүйесі клиенттен алынған пакетті пакеттің үстінен бірнеше операцияларды орындағаннан кейін ғана тағайындалған серверге жібереді: біріншіден, ол пакетте алушының мекенжайын (яғни, өзінің VIP-адресін) тағайындалған сервердің IP-адресімен алмастырады, ал екіншіден-жіберушінің IP-адресін өзінің VIP — адресіне ауыстырады. Бұл әдіс клиенттерден Web-серверлердің IP-адрестерін жасыруға мүмкіндік береді, сондықтан соңғылары кез келген IP-адрестерді, соның ішінде жеке IP-адрестерді пайдалана алады. Бұл ретте Web-серверлер теңгеру жүйесімен тікелей қосылуы міндетті емес (басқаша айтқанда, сол ЖТС сегментіне кіру); олар статикалық немесе желілік маршруттау хаттамалары бойынша қосылымдарды орнатуы жеткілікті.

ТСР шлюзі. Жүктемені теңгеру жүйесін тікелей (дереу) байланыстыруды орнату кезінде трафикті 2 немесе 3 деңгейлерде MAS немесе Nat әдістері бойынша бағыттай алады. Бірақ егер бұл кейінге қалдырылған байланыс болса, баланс жүйесі ТСР деңгейінде және жоғары деңгейде трафикті басқаруы керек. Кейінге қалдырылған байлау деп болжайды жүйесі жүктемені теңдестіру және клиент белгілейді байланысы ТСР болатындай жүйе еді осы қосымшаның тағы тағайындалғанға дейін сервер. Содан кейін теңгеру құралы тағайындалған сервермен ТСР байланысын орнатады және оған клиенттік сұрау жібереді. Одан кейін теңгеру жүйесі клиентке сервердің жауабын береді, ол үшін тағы да "теңгеру жүйесі-клиент"ТСР-қосылымы қолданылады. Сипатталған функция ТСР шлюзы деп аталады. Resonate компаниясының мамандары оны жүктемені теңгеру жүйесінде арнайы агенттің көмегімен іске асырады (бұл агент клиент пен теңгеру құралы рөлін атқаратын сервер арасында тікелей ТСР-қосылуды қамтамасыз ететін серверде орнатылады).

Жеткізуші компания ұсынған терминология бойынша бұл іске асыру TCP-қосылыстар (TCP connection hop) транзиті бар жүйе деп аталады.

Зертханалық жұмыс №51. Іздеуді оңталайндыру және сайтқа кіруді бақылау құралдары.

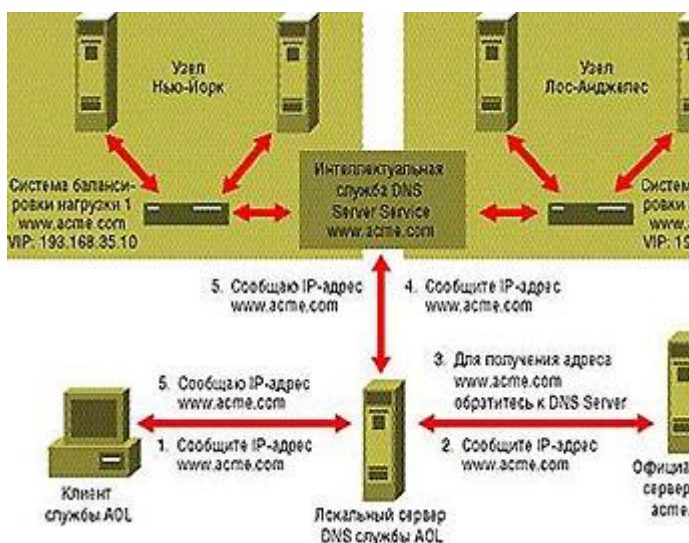
Сайтты таңдау және трафикті жаһандық деңгейде басқару

Ақпараттық ресурстар бірнеше айналы тораптарда орналастырылған жағдайларда жүктемені теңгеру жүйелері (сонымен қатар жүктемені теңдестірудің жаһандық жүйелері деп аталады) серверді таңдаудың сипатталған тетіктерінің көмегімен клиент үшін қолайлы торапты анықтайды. Сонымен қатар, сайтты таңдау критерийі ретінде баланстаудың ғаламдық жүйесі сайт пен клиент арасындағы қашықтық (транзиттік учаскелер саны мен желілік кідіріс ұзақтығы көрсетілген) сияқты көрсеткішті пайдалана алады. Ең қолайлы сайтты анықтау кезінде, теңгерім жүйесі жиі клиенттің трафигін DNS зияткерлік функциясы арқылы тиісті торапқа жібереді.

Бұл үдерістің механизмін жақсы түсіну үшін, біздің мысалымызға оралайық. Аты бар виртуалды жүйе www.ame.com екі сайтта ұсынылған. Бұл машина үшін DNS серверлерінің функцияларын жүктемені теңгерудің екі жүйесі орындайды: біреуі — Нью-Йоркте, екіншісі — Лос-Анджелесте. FTP және электрондық пошта сияқты қызметтерге, сондай-ақ Internet-дегі басқа серверлер мен компьютерлерге арналған атауларды DNS доменінің ресми сервері ажыратады ame.com. Міне, әкімші поддоменнің атын бере алады www.ame.com Internet доменінің бөлігі болып табылады ame.com, екі теңгеру құралы, содан кейін екі жүктеме теңестіру жүйесі тегеурінді серверлер болады www.ame.com. осындай конфигурацияны алу үшін әрбір теңгерім жүйесі үшін DNS атауын анықтау қажет www.ame.com және оны өз Жергілікті VIP мекенжайына сәйкес қойыңыз. Екі баланс жүйесі конфигурация және жүктеме туралы ақпаратпен алмасады, екеуі де виртуалды жүйе екенін "түсінеді" www.ame.com екі VIP-мекен-жайы бар (яғни. екі түйінмен ұсынылған). Демек, екі сайттардың әрқайсысының жүктемесі мен қолжетімділігі туралы ақпарат бар.

AOL клиенті мекен-жай бойынша хабарласқанда www.ame.com 3-суретте көрсетілгендей, қоңырау рәсімі клиент жергілікті DNS серверінен компьютердің IP мекенжайын сұрайды www.ame.com. егер AOL қызметінің DNS жергілікті серверінің кэшінде сұралған IP мекенжайы туралы деректер болмаса, бұл сервер DNS доменінің ресми серверіне сұрау жібереді ame.com. Қалай есте DNS Домен сервері ame.com аты тағайындады www.ame.com екі жүктеме теңгеру жүйесі, сондықтан домен ame.com AOL IP мекен-жайы қызметінің DNS жергілікті серверіне екі жүктемені теңестіру жүйесінің атау сервері ретінде қайтарады www.ame.com. (3 — суретте мен DNS серверінің зияткерлік қызметін белгілеп, оны жеке тіктөртбұрыш деп белгіледім-кейбір жеткізушілер бұл технологияны жеке серверде іске асырады. AOL қызметінің DNS жергілікті сервері жүктемені теңдестірудің екі жүйесінің бірінің атын алуға сұрау жібереді. Екі жүйе атаулардың серверлерін білдіреді, сондықтан бірінші серверден жауап алмай, AOL қызметінің DNS жергілікті сервері екінші машинаға қайта сұрау жібереді. Жүктемені теңестіру жүйесі сайт үшін берілген критерийлердің негізінде ең қолайлы серверді таңдап, сайт серверінің AOL VIP-мекенжайын AOL қызметінің DNS жергілікті серверіне қайтарады. AOL VIP қызметінің DNS жергілікті серверінен бас машинаның мекен-жайын алып www.ame.com. клиент өзінің HTTP трафигін таңдалған сайттың жүктемесін теңгеру жүйесіне (мысалы, нью-Йорк) жібереді және бұл жүйе клиент үшін жергілікті серверлердің бірін таңдайды. Жергілікті DNS сервері қол жетімді DNS жазбасын Time To Live (TTL) параметріне сәйкес келетін мерзімге кәштейтіндіктен, жеткізушілер әдетте

клиент жаңа VIP-мекенжайды тез алуға және басқа қол жетімді сайтқа ауысуға мүмкіндік беретін TTL үлкен мәндерін белгілеуді ұсынады.



Бірнеше сайттар үшін DNS сұрау қайта бағыттау.

Жүйесі жүктемені теңдестіру мүмкін таңдау, ыңғайлы сайт, сондай-ақ бағыттауға трафик және басқа әдіспен: көмегімен бағыттау құралдарымен хаттама HTTP. Бұл әдіс жүктемені теңгеру жүйесінің DNS-функцияларын пайдалануды қарастырмайды. Оның орнына (біздің мысалымызға қайтадан жүгінемін) әкімші DNS серверінде жазба анықтайды www.asme.com және оған сәйкес VIP-мекенжайлар. Клиент www VIP-мекенжайын алған кезде.asme.com HTTP хаттамасы бойынша жүктемені теңгеру жүйесіне сұрау жібереді, соңғысы оған ең қолайлы торапты таңдайды. Егер таңдаулы сайт қашықтағы болмаса (тым алыс емес), теңгерім жүйесі клиенттің браузеріне HTTP-командасын бағыттайды және браузер көрсетілген сайтпен байланыс орнатады. Бұл әдіс жүктемені теңгеру жүйесіне торапты таңдағанға дейін клиент туралы толық ақпаратты (мысалы, оның IP-мекен-жайы) алуға мүмкіндік береді. Дегенмен, клиент DNS серверімен қайтарылған VIP-мекенжайды пайдалана алады және сұрауларға жауап бермейтін тораппен байланыс орнатуға тырысады.

Сонымен қатар, бұл жүйенің негізгі мақсаты-белгілі бір уақыт аралығында белгілі бір уақыт ішінде Мысалы, бізде Еуропада айна Web торабы бар. Бізге Еуропадан келген клиенттер бүкіл уақытта еуропалық сайтқа түсуі керек-ол істен шыққан және жүктемені теңгерімдеу жүйесі еуропалық трафикті АҚШ-та орналасқан торапқа бағыттайтын жағдайларды қоспағанда. Администратордың жүктемені теңгеру жүйесін еуропалық IP-мекен-жайы бар клиенттен кез келген сұрау алдымен Еуропадағы сайтқа бағытталатындай етіп реттеуге мүмкіндігі бар. (Тиісті баптауларды орындау үшін, бүкіл еуропалық IP блогын теңгерімдеу жүйесіне енгізу қажет.) Мұндай нұсқауды алғаннан кейін жүктемені теңгеру жүйесі бірінші кезекте еуропалық торапқа еуропалық кері мекен-жаймен кез келген сұрау жібереді және осыдан кейін ғана сайтты таңдаудың басқа да белгіленген критерийлерін есепке алады.

Зертханалық жұмыс №52. Әр түрлі CMS мысалында контентті басқару жүйесі үшін үлгілер жасау.

1. "Құрылымды басқару" тарауында тапсырмаларды орындау кезінде жасалған Тест бетін өңдеңіз":

1. Еркін мәтіннің ежесін салып, еркін сөздерді пішімдеңіз:

- майлы бөліңіз;
- асты сызыңыз;
- мәтіннің түсі мен фонын өзгертіңіз.

2. 3 тақырып мәнеріне мәтін тақырыбын қосыңыз;

3. Мәтінде сайтқа сілтемені орнатыңыз <http://1c-bitrix.ru/> кез келген сөзден;

4. Мәтін жолын қосып, курсивпен бөлектеңіз және оны оң жағынан пішімдеңіз;

5. Жергілікті компьютерден жүктелген еркін суретті бетке орналастырыңыз және оған 30 масштабты орнатыңыз%;

6. 5 бағаннан және 6 жолдан кесте жасаңыз:

- Кесте үшін ені мен биіктігін орнатыңыз: 200x300;
- Жоғарғы жолды тақырып жасаңыз;
- Тегістеу: ортасында;
- Кесте ұяшықтарына кез келген деректерді енгізіңіз.

2. С니ппет жасаңыз және оны бетке орналастырыңыз. ҚНЖЕ коды:

```
1. <table cellpadding=0 cellspacing=0>
2.   <tbody>
3.     <tr bgcolor="white">
4.       <td bgcolor="#8C8C8C">
5.         <table cellpadding="5" cellspacing="1">
6.           <tbody>
7.             <tr bgcolor="silver"><td colspan="2"><b>Заголовок
таблицы</b></td></tr>
8.
9.             <tr bgcolor="white"><td><b>1</b></td><td>2</td></tr>
10.
11.             <tr bgcolor="#EEEEEE"><td><b>3</b></td>
<td>4</td></tr>
12.           </tbody>
13.         </table>
14.       </td></tr>
15.     </tbody>
```

```
</table>
```

Зертханалық жұмыс №53. Әр түрлі CMS мысалында контентті басқару жүйесі үшін үлгілер жасау.

Тапсырмаларды дәйекті орындау қажет.

1. Солтүстік-батыс аймағында инфоблок бос және инфоблок Бос түрін жасаңыз (негізгі параметрлері стандартты).

Инфоблок параметрлерінде элемент анонсының суреті егжей-тегжейлі суреттен жасалуы керек және 70x70 пиксель өлшемі болуы керек.

Келесі кіру құқықтарын теңшеңіз:

- * әкімшілер үшін-толық қол жеткізу;
- * мазмұнды редакторлар үшін-Өзгерту;
- барлық басқа пайдаланушылар үшін-оқу.

2. Солтүстік-батыс аймағындағы инфоблокта Бос орындар екі бөлімді жасаңыз, ал әрбір бөлімде – бір элементтен (Бос орындар):

- бөлім техникалық қызметкерлер, веб-әзірлеуші бос орын;
- * бөлім Әкімшілік персонал, бос есепші.

Әрбір элемент (бос орын) анонстың сипаттамасы, толық сипаттамасы және сурет болуы керек.

3. Қоғамдық бөлімде жаңа бетті және бос орын мәзірінің төменгі тармағын жасаңыз. Бұрын жасалған бос жұмыс орындарын кешенді жаңалықтар компоненті арқылы жариялаңыз .

4. * Екі жаңа бос жұмыс орны бар CSV файлын жасаңыз және келесі талаптарды ескере отырып, оны сайтқа жүктеңіз (UTF-8 сайты кодтау):

- бос орын техникалық қолдау бөлімінің инженері техникалық қызметкерлер бөлімінде болуы тиіс;
- бос орын маркетинг менеджері әкімшілік қызметкерлер бөлімінде болуы керек;
- әрбір элемент (бос орын) анонстың сипаттамасы, толық сипаттамасы және сурет болуы керек;
- * элементтер жүктелгеннен кейін бірден белсенді болуы керек;
- бұрын жасалған элементтер жойылмайды;
- * элементтің Анонсы үшін сурет толық суреттен қалыптасуы керек.

Зертханалық жұмыс №54. Сайттың жұмыс қабілетін тексертеді

Тапсырмаларды дәйекті орындау қажет.

1. Қазір бүкіл сайттың бірегей үлгісі интернет-дүкеннің бейімделген үлгісі бар.

Жаңа оқу бөлімін жасаңыз және оған оқу курсынан өту үлгісін белгілеңіз.

2. Құрылған оқыту бөлімі үшін сайттың белгішесін ауыстырыңыз.

3. Сайттағы телефон нөмірін өзгертіңіз (қосылатын аймақта және контактілер бөлімінде).

4. Сайт логотипін ауыстырыңыз.

5. Дүкен туралы бөлім үшін сайттың фонын өзгертіңіз.

ДИАГНОСТИКО-КОНТРОЛИРУЮЩИЙ БЛОК

Описание проверочного испытания

Проверочное испытание № 1 , вариант № _____	
Название:	Web-бағдарламалау және интернет технологиялар негіздері
Описание задачи:	Web-беттерді әзірлеуге арналған бағдарламалаық тілдерді қолданып білу
Нормированное время.	45 мин
Результаты обучения	1. Таратылған ақпараттарды өңдеуді жүзеге асырудың ұймдастырылған қағидттары мен механизмдерін қолданады. 2. Ақпаратты жүйедегі сервердің функциялары, клиент, деректер туралы түсінігі бар. 3. Жұмысқа қазіргі заманғы web-технологияларды қолданады.
Критерии оценки выполнения	1. Желілік қызметтің функцияларын икемдейді. 2. Интернеттің қолданбалы хаттамасының жұмысын икемдейді.
Порядок выполнения задания	1. Web-беттерді құру құралдарын сипаттау; 2. Гипермәтіндік құжаттарды құру технологияларын сипаттау; 3. Сервер тұжырамдамасын сипаттау.
Перечень необходимого оборудования и инструментов.	Қағаз, қалам
Место проведения:	Каб 223
Дата проведения:	« » _____ 20 г.

ОЦЕНОЧНЫЙ ЛИСТ

Дисциплина (модуль) « _____ »

Тип проверочного испытания								
Результаты обучения/ Критерии оценки	Тест	ЛПЗ	Задание	Курсовой проект	Производственное обучение			
Результат обучения 1	Web-беттерді әзірлеуге арналған бағдарламалаық тілдерді қолданып біледі							
А-1			+					
А-2			+					

--	--	--	--	--	--	--	--	--

Проверочное испытание № 2 , вариант № _____	
Название:	HTML және CSS белгілеу тілі
Описание задачи:	HTML және CSS қолдана отырып, орташа күрделіліктегі Веб-парақшаны түптеуді орындау.
Нормированное время.	2ч30мин
Результаты обучения	1) HTML гипермәтіннің өлшеу тілін және CSS стильдерінің каскадты кестелерін қолданады.
Критерии оценки выполнения	1. Мәтіндік деректермен, кестелермен, графикалармен жұмыс істей алады. 2. Кез келген редактордағы гипермәтіндік құжаттарды құрайды.
Результаты обучения	1) CSS стилінің каскадты кестесін қолдана отырып, Веб- парақшаны құру және сайттарды жобалау технологиясын қолдана отырып, Веб-сайтты әзірлеу
Критерии оценки выполнения	1. Стильдік каскадты кестесінің ережесін қолданады. 2. Стильдік файлдары құрады. 3. Сайтқа стильдік кестелерді енгізу тәсілдерін, CSS қасиеттерімен жұмыс істеуді CSS-тің негізгі блоктары мен қосалқы сервистерді қолданады.
Порядок выполнения задания	1. Notepad++ программасын ашып, HTML гипермәтіндік тілді қолданып веб беттің бірінші бетінің қаңқасын құрастыру. 2. Таңдалған тақырыпқа байланысты веб-бетке суреттер мен ақпаратты енгізу. 3. Веб бетті блокты, кестелі немесе фреймдік құрылымды істеу . 4. Веб беттің тақырыбына байланысты, қосымша 3-4 бетті жасап, оларды басты бетпен гиперсілтемелер арқылы байланыстыру. 5. CSS стильдің каскадты кестесіне веб-беттің түсін, мәтін өлшемін, графикалық редакциялау үшін, селекторларды жазу. 6. HTML мен CSS байланыстыру.
Перечень необходимого оборудования и инструментов.	ДК, Notepad++
Место проведения:	Каб 223
Дата проведения:	« » _____ 20 г.

ОЦЕНОЧНЫЙ ЛИСТ

Дисциплина (модуль) « _____ »

Тип проверочного испытания								
Результаты обучения/ Критерии оценки	Тест	ЛПЗ	Задание	Курсовой проект	Производственное обучение			
Результат обучения 1								
Критерии оценивания 1			+					
Критерии оценивания 2			+					
Критерии оценивания 3			+					
Критерии оценивания 4			+					
Критерии оценивания 5			+					
Результат обучения 2								
Критерии оценивания 1			+					
Критерии оценивания 2			+					
Результат обучения 3								
Критерии оценивания 1			+					
Критерии оценивания 2			+					
Критерии оценивания 3			+					
Критерии оценивания 4			+					
Критерии оценивания 5			+					

**Критерии оценивания студентов в процессе обучения
с учетом модульно-компетентного подхода.**

Характеристика решения	баллы
Тапсырма аяқталды, студент шешімді түсіндіреді, қосымша тапсырмалар орындалады	10
Тапсырма аяқталды, студент шешімді түсіндіреді.	9
Тапсырма аяқталды, студент шешімді түсіндіреді, бірақ кейбір сұрақтарға жауап беру қиын.	8
Тапсырма толығымен аяқталды, бірақ бағдарлама барысында жұмыс уақытында қате пайда болады.	6-7
Тапсырма 70% -ға аяқталды, жұмыс уақытында қате пайда болды	4-5
Тапсырма 50% -ға аяқталды, бағдарлама барысында жұмыс уақытында қателігі пайда болады	3
Тапсырма 25% -ға аяқталды, бағдарламаның жұмыс барысында жұмыс уақыты қателігі пайда болады	2
Тапсырма 10% -ға орындалды, бағдарлама барысында жұмыс уақыты қателігі пайда болады	1
Тапсырма орындалмады	0

Критерии оценки модуля

№	Виды контроля	недели																	мак балл
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
1	Посещаемость																		
2	Домашняя работа																		
3	Практическая работа																		
4	Рубежный контроль																		
5	Тестовые задания																		
6	Контрольная работа																		
7	Реферат																		
8	Презентация																		
9	Промежуточный контроль (экзамен)																		
10	Всего:																		

КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНЫЕ МАТЕРИАЛЫ

Тест тапсырмалары

1. HTML құжаттары қандай кеңейтілуінде сақталады?
 - A) .php или .asp
 - B) .txt или .doc
 - C) .doc
 - D) .html или .htm
2. Қандай тег көмегімен HTML құжаттары сурет қоюға болады?
 - A) <pic>
 - B)
 - C) <picture>
 - D) <image>
3. Дұрыс синтаксиспен жазылған сілтемені таңдаңыз?
 - A)
 - B) wisdomweb.ru
 - C) wisdomweb.ru
 - D) wisdomweb.ru
4. Қандай тегтің көмегімен мәтінді қою қаріппен жазуға болады?
 - A)
 - B) <i>
 - C) <big>
 - D) <h2>
5. Қандай тегтің көмегімен мәтінге тақырыпты жасауға болады?
 - A)
 - B) <small>
 - C) <h2>
 - D)
6. Кестені жасайтын тегті таңдаңыз
 - A) <tab>
 - B) <table>
 - C) <tr>
 - D) <tabs>
7. HTML құжаттының қай бөлігінде <title>, <link> и <meta> жазылады?
 - A) берілген тегтер құжатының әрбір бөлігіне де жазылынады
 - B) <body> бөлігіне
 - C) <head> бөлігіне
8. ең үлкен тақырып қандай атрибутпен жазылады
 - A)
 - B) <h6>
 - C) <h1>
 - D) <p>
9. HTML – бұл
 - A) қосымша(приложение)
 - B) программалау тілі
 - C) мәтіндік редактор
 - D) гипертекстік белгілеу тілі
10. Қай тег көмегімен құжатқа суретті қосуға болады?
 - A) <imgt>
 - B)

 - C) <title>

D) P

11. background атрибуты береді

A) беттің фонын

B) ұяшықтың фонын

C) кесте фонын

D) барлық жауап дұрыс

12. Жолды аудару тегін таңдаңыз:

A) Img

B) Src

C) Tr

D) Br

13. text атрибуты body тегінде береді

A) мәтіннің әріптүрін

B) мәтіннің өлшемін

C) мәтіннің түсін

D) мәтіннің стилін

14. Қайсысы font тегіне жатпайды

A) Size

B) color

C) face

D) border

15. align атрибутына жатпайды

A) Left

B) Right

C) Top

D) center

16. border атрибутының мәні бола алады

A) Green

B) Left

C) Top

D) 2

17. Көрсетілген тегтердің қайсысы негізгі тег болмайды?

A) Html

B) caption

C) body

D) head

18. Қандай тег кестенің жолын көрсетеді?

A) td

B) tr

C) th

D) tu

19. Қандай тег кестенің ұяшығын көрсетеді?

A) Tr

B) Th

C) Tu

D) Td

20. Қандай тег азат жолын (абзац) көрседі?

A) P

B) Allign

C) Td

D) br

**Рефераттар мен баяндамалардың тақырыптары.
Тематика рефератов и докладов.**

1. HTML тілі. Даму тарихы, мүмкіндіктері
2. Веб-дизайн және веб-дизайн бойынша заманауи әдебиеттерді шолу
3. Веб-дизайн және Веб-дизайн бойынша веб-ресурстарды шолу
4. Веб-дизайндағы үлгілер: қолдану аясы, артықшылықтары, кемшіліктері, үздік ресурстар
5. Контентті басқару жүйелері (CMS) - жалпы шолу жасауға немесе нақты CMS сипаттауға болады.
6. Веб-сайттарды жылжыту
7. Веб-сайттағы мультимедиа құралдары
8. Эргономика (юзабилити) веб-сайт
9. Онлайн веб-редакторлар (Яндекс халық, ucoz және т. б.) - жалпы сипаттама. Нақты редакторларға шолу
10. Жетекші редакторлар (нақты бағдарламалардың мүмкіндіктеріне шолу жасауға болады)
11. Виджеттер
12. Сайттардың түс схемасы: таңдау мәселесі
13. Динамикалық технологиялар: JavaScript, AJAX, PHP, Java, Flash, Silverlight және т. б.)
14. Блог-платформалар
15. RSS технологиясы
16. Тегін HTML редакторлары
18. Блог-платформалар
19. Веб-редактор мен әзірлеу құралдарын таңдау
20. Әлеуметтік қызметтер және оларды кітапханалардың веб-сайттарында қолдану
21. Тегін мазмұнды басқару жүйелері
22. Эргономикалық дизайн ережелері
23. Сайттарға арналған дайын іздеу жүйелері
24. Домендік атауды таңдау және сатып алу
25. Веб-сайттардағы жылдам хабар алмасу жүйелері (чаттар)
26. Мәтіндік контентті дайындау тәсілдері
27. Веб-сайтты тестілеу әдістемесі

Қорытынды бақылауға арналған сұрақтар
Вопросы для итогового контроля

1. HTML тілі не үшін қолданылады?
2. HTML құжаты қандай тэг арасында?
3. Құжаттың тақырыбы тэг болып табылады...
4. Құжаттың денесі тэге...
5. Қаріпті пішімдеу Тэг...
6. Мәтінді жаңа жолға көшіру үшін тег қолданылады...
7. Web-беттер қандай кеңейтім бар?
8. HTML-дағы түс он алтылық таңбалардың қайсысы кодталады?
9. Қандай селектор сізге веб-беттің әрбір элементіне жүгінуге мүмкіндік береді?
10. Элементті жасыру үшін қандай сипат береді, бірақ беттегі орналасқан кеңістікті сақтауға болады?
11. Font style сипаттары төрт жарамды мәнге ие. Олардың үшеуі-inherit, normal және italic. Ал қандай төртінші?
12. Түсті білдіретін қандай кілт сөз әрқашан таңдалған элемент немесе элементтер үшін есептелген түс мәніне тең болады?
13. CSS-де @ ережесі стильдер кестесіндегі таңбалардың кодтауын анықтауға мүмкіндік береді?
14. JavaScript тілі қандай тіл?
15. Скриптілік тілдер не үшін қолданылады?
16. Java Script тілінің мүмкіндіктері қандай?
17. JavaScript тілін кім ойлап тапқан?
18. Скрипт немесе сценарий деген не?
19. JavaScript тілінің қандай объектілері бар?
20. Объектілік модель неден құралады?
21. Айнымалылардың қандай типтері бар?
22. JavaScript тегтері қайтегтердің арасында орналасады?
23. <NOSCRIPT> тегінің атқаратын қызметі?
24. Ақпаратты шығарудың қандай тәсілдері бар?
25. JavaScript-тің қандай объектілері бар?
26. Date ішкі объектісі және оның тәсілдері?
27. Array объектісінің қызметі?
28. Math математикалық класының қызметі?
29. Document объектісі қызметі?
30. Жаңа терезе ашу үшін қандай объект қолданылады?
31. Терезені сипаттау параметрлері?
32. JavaScript-тің alert функциясының қызметі?
33. Click оқиғасының қызметі?
34. Тышқанды шерту, курсорды қозғалту, пернелерді басудың қандай оқиғалары бар?
35. ДҚБЖ MS SQL Server 2008 болжамының мінездемесі.
36. MS SQL Server 2008 аспаптың ақы-пұлының мінездемелері.
37. ДҚ SQL Server 2008 өзімен нени ұсынады?

38. Нысандар дерекқор SQL Server 2008.
39. ДҚБЖ MS SQL Server 2008 деректердің әдіс-айлалары?