ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

М.ӘУЭЗОВ атындағы ОҢТҮСТІК ҚАЗАҚСТАН МЕМЛЕКЕТТІК УНИВЕРСИТЕТІ

«Информатиканы оқытудың әдістемесі мен теориясы» кафедрасы

Нысанов Е.А., Құрақбаев Ж.С., Ибрагимов О.М.

WEB-БАҒДАРЛАМАЛАУ НЕГІЗДЕРІ

оқу құралы

Шымкент, 2015

УДК 004.42 (075.8)

Нысанов Е.А., Құрақбаев Ж.С., Ибрагимов О.М. Web-бағдарламалау негіздері. Оқу құралы. –Шымкент: ОҚМУ, 2015. –206 бет.

Оқу құралында динамикалық гипермәтіндік құжаттарды құру технологиясы қарастырылған. Онда HTML, JavaScript және PHP бағдарламалау тілдерінің синтаксисі, тэгтері, операторлары мен функциялары қаралған. Web-құжатта CSS технологиясы және JavaScript, PHP скрипттерден пайдалану әдістері көрсетілген. Оқу құралында осы аталған мәселелер мысалдармен көрнекі түрде сипатталған. Сонымен қатар оқу құралында берілген жеке тапсырмалар мен тест сұрақтары, оның мазмұнын түсінуге көмектеседі.

Оқу құралы университеттің 5В060200 – «Информатика» мамандығы бойынша оқитын студенттерге арналған.

Пікір жазғандар: Оразов И. – ф.-м.ғ.к., доц., М.Әуезов атындағы Оңтүстік Қазақстан мемлекеттік университетінің «Информатика» кафедрасының меңгерушісі;
Бараев А.Б. – ф.-м.ғ.д., Оңтүстік Қазақстан мемлекеттік педагогикалық институтының «Информатика» кафедрасы профессоры.
Ибрагимов Р.И. – п.ғ.д., проф., Қазақстан инженерлі-педагогикалық Халықтар достығы университетінің «Информатика және математика» кафедрасының меңгерушісі.

М.Әуезов атындағы Оңтүстік Қазақстан Мемлекеттік университетінің оқу-әдістемелік кеңесі баспаға ұсынған, хаттама №__«__» _____ 2015 ж.

© М.Әуезов атындағы Оңтүстік Қазақстан Мемлекеттік университеті

Мазмұны

Кіріспе.....5

1 Интернет және WWW жүйесі 7
1.1 Интернет технологиялар 7
1.2 IP-адрес және URL-адрес 10
1.3 Интернет қызметтері 13
1.4 OSI эталондық үлгісі мен TCP/IP хаттамасы 19

2 HTML тілінің негіздері 27
2.1 Web-құжат құру технологиясы 27
2.2 Web-құжат құрылымы 30
2.3 HTML тілінің мәтінді пішімдеу тэгтері 34
2.4 HTML тілінде тізімдерден пайдалану 39
2.5 Web-құжатқа сілтемелер, сурет және объекттер енгізу 41

3 Web-құжатты безендіру 47
3.1 Web-құжатта кестелерден пайдалану 47
3.2 Web-құжатта қалыптардан пайдалану 51
3.3 Web-құжатта фреймдерден пайдалану 56
3.4 Web-құжаттың интерактивтік интерфейсі 59

4 CSS технологиясы 64
4.1 CSS технологиясының синтаксисі 64
4.2 CSS технологиясының қасиеттері 75
4.3 CSS стильдерін Web-құжатқа енгізу әдістері 83
4.4 CSS технологиясының алгоритмдері 89

5 JavaScript тіліне кіріспе 96
5.1 JavaScript тілінің қарапайым элементтері 96
5.2 JavaScript тіліндегі амалдар 100
5.3 Шартты операторлар 104
5.4 Цикл операторлары 107

6 **JavaScript-дің алгоритмдік мәселелері** 111
6.1 Web-құжатқа скриптті орналыстыру әдістері 111
6.3 Интерактивтік интерфейстің қалып скрипттері 112
6.3 Web-құжатқа ақпаратты енгізу-шығару скрипттері 121
6.4 JavaScript-ке мысалдар 135

7 PHP тіліне кіріспе 139
7.1 PHP тілінің қарапайым элементтері 139
7.2 Деректердің типтері және амалдар 141

7.3 Операторлар 146
7.4 Функциялар 151

8 PHP-скрипттерді құру технологиясы 156
8.1 Арасне локальді серверін орнату 156
8.2 PHP препроцессорын орнату 162
8.3 PHP-скрипттерге мысалдар 165
8.4 «Web-дизайн» электрондық оқулығын құру технологиясы 175

Жеке жұмысқа тапсырмалар 191

Тест сұрақтары 195

Пайдаланылған әдебиеттер 201

Қосымшалар 202

Кіріспе

Қазіргі таңда Қазақстан Республикасы алдында ақпараттық қоғам құру мәселесі туындап отырғаны белгілі. Ақпараттық қоғам – басым көпшілігі ақпараттық өндіріспен, яғни ақпараттың жоғарғы формасы білімді сақтау, өңдеу және тасымалдау істерімен айналысатын қоғам. Ақпараттық қоғамды құру критерийлерінің бірі қажетті ақпаратты жедел және нақты түрде жеткізу, тасымалдау. Бұл критерийдің Интернет-технологияларға байланысты тұстары бар.

XX ғасырдың аяғында бүкіләлемдік ауқымды желі Интернеттің қарқынды дамуы ғылымда жаңа бағыттардың пайда болуына себеп болды. Солардың бірі Web-бағдарламалау технологиясы. Қазіргі таңдағы Интернет қызметтері, мысалы, бүкіләлемдік өрмек (WWW), электрондық пошта (E-mail), нақты уақыт аралығында ақпарат алмасу (IRC), сандық телевидение (iDTV) және т.б. осы бағдарламалау технологиясының жемісі.

Web-бағдарламалау технологиясын көбінесе Web-дизайн деп атайды. Бұл термин Web (WWW қызметінің қысқартылған атауы) және дизайн (ағыл. design) сөздерінің тіркесі болып, «Web-құжатты көркем жобалау, құру» деген мағынаны білдіреді.

Дизайн сөзі күнделікті өмірде - объектті жобалау, яғни жаңа құрылғыны, құрал-саймандарды көркем құрастыру деген көп мағыналы термин ретінде қолданылады. Оның негізгі мақсаты – объектті қолдануына қарай көркемдеу, пішімдеу. Классикалық дизайн көркемөнерінен Web-дизайнның айырмашылығы, мұнда қалам, бояу, мольберт сияқты құрал-саймандардың орнына, компьютерлік құрал-саймандар - бағдарламалау жүйелері, Web-редакторлар, графикалық процессорлар, аниматорлар, цифрлік фотоаппараттар, сканерлер т.с.с. пайдаланылады.

Сонымен, Web-дизайн – IT-технологиялар (компьютер және жаңа ақпараттық технологиялар), арнайы бағдарламалау тілдері (HTML, JavaScript, PHP, Java, Perl, C++) және желілік технологиялар (CSS, CGI, SSI) негізінде құрылған көркемөнердің жаңа түрі болып есептеледі. Кейбір әдебиеттерде компьютерлік дизайн, Web-шебер терминдері де кездеседі.

Web-дизайншы деп, әдетте, Web-сайт дизайнымен айналысушы бағдарламашыны айтады. Бірақта олар дизайнмен қатар, жүйелік администратор және бағдарламашы міндеттерін де атқарады. Кейде сайт модераторлары да Web-дизайншы деп аталады.

Web-дизайншы Web-құжат құруда гипермәтіндерден, анимациялық суреттерден, фото, видео және аудио файлдардан пайдаланады. Олар коммерциялық және жарнамалық Web-жобалар: мемлекеттік мекеме немесе коммерциялық ұйым Web-сайты, Интернет-магазин, электрондық басылым, ақпаратты іздеу жүйелерін құрады. Web-дизайншы өзінің кәсіби міндетін атқаруда декоративті-қолданбалы өнер, көркем фотография, архитектура, скульптура, граффити сияқты көркемөнер салаларында жинақталған тәжірибеден пайдаланады. Сонымен қатар оның өзіне тән

жаңа қасиетері, заңдылықтары бар. Олар Интернеттің техникалық, бағдарламалық және ақпараттық компоненттерінің дамуымен өзгеріп отырады.

Қазіргі таңда Web-құжаттарда гипермәтін мен статикалық суреттермен қатар, қозғалушы объекттер, яғни баннерлер, анимациялар, видеоклиптер қолданылады. Интернетте тұтынушымен интерактивті байланыста болатын динамикалық Web-сайттар, ақпаратты іздеу жүйелері жиі кездеседі. Сонымен қатар, тұтынушының менталитеті мен психологиясын, креативті идеялар мен тақырыптарды есепке ала отыру қажеттігі туындап отыр. Мұның барлығы Web-дизайншыдан әдебиет, журналистика, кино, телевидения, фото т.с.с. өнер салаларының туындыларынан орынды пайдалануды талап етеді.

Интернеттің WWW жүйесінде гипермәтін, графика, фото, аудио, видеоклип, көркем фильм, баннер, анимация, т.с.с. өте ауқымды ақпарат сақталады. WWW жүйесінің негізгі қызметі қажетті ақпаратты тұтынушы сұранысы бойынша іздеу, сұрыптау және мониторға жинақы түрде шығаруды ұйымдастыру. WWW жүйесінің ақпараты Web-құжат түрінде дайындалып сақталған гипермәтіндік құжаттар. Мұндай гипермәтіндік құжаттың қарапайым мәтіндік құжаттан айырмашылығы, оның жазылу пішімінде, яғни HTML тілінде жазылған. Сонымен қатар HTML тілінің мүмкіндіктерін кеңейтуге арналған қосымша Интернет-технологиялар дамуда. Мұндай технологияларға төмендегілер кіреді:

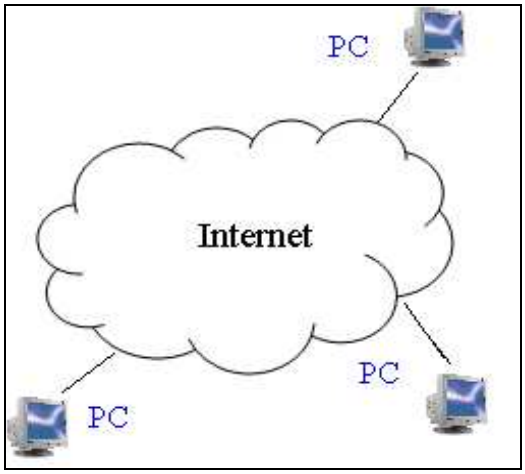
- CSS – каскадты стильдер кестесі;
- JavaScript, VBScript, ActionScript тілдерінде жазылған скрипттер;
- Java тілінде жазылған апплеттер;
- PHP тілінде жазылған PHP-скрипттер;
- Perl, C++ тілдерінде жазылған CGI-скрипттер;
- GIF, JPEG, PNG форматындағы суреттер;
- SWF-файлдар және Flash-анимациялар;
- AVI, MPEG форматындағы видеоклиптер;
- WAV, MP3, MIDI форматындағы аудиоклиптер.

Ұсынылып отырған оқу құралында динамикалық гипермәтіндік құжаттарды құру технологиясы қарастырылған. Онда HTML, JavaScript және PHP бағдарламалау тілдері, CSS технологиясынан пайдалану әдістері берілген. Оқу құралында осы аталған мәселелер мысалдармен көрнекі түрде сипатталған. Оқу құралында берілген жеке тапсырмалар материалды терең түсінуге көмектеседі.

1 Интернет және WWW жүйесі

1.1 Интернет технологиялар

Интернет (Internet - Interconnecting Networks) - бүкіл әлемдегі миллиондаған шағын компьютерлерді бір-бірімен байланыстырып тұрған кең ауқымды компьютерлік желі, желілердің желісі. Ақпаратты тарату мақсатында мемлекеттік және жекеменшік мекемелердің, жеке тұлғалардың, университеттердің, ұйымдардың т.с.с. желілерін бір жүйеге біріктіріп TCP/IP хаттамаларын пайдалана отырып жұмыс істейді. Желідегі компьютерлерде мәтін, графика, фото, аудио, видео, анимация т.б сияқты өте үлкен ақпарат көлемі сақталады. Желіні пайдаланудың негізгі қызметтеріне – WWW (World Wide Web) - бүкіләлемдік өрмек, E-mail - электрондық пошта, ISQ (I seek you) - нақты уақыт кезінде хат-хабарлар алмасу және FTP (File Transfer Protocol) - файлдарды тасымалдау т.б. жатады [1]. (сурет 1).



Сурет 1 – Интернет желісі

Интернетке қойылатын негізгі талап – байланыс арналары елеулі өзгеріске ұшырағанымен, кез келген жағдайда ақпаратты бір компьютерден екінші компьютерге шығынсыз жеткізу қажеттілігі. Желідегі байланыс арналарының бірсыпыра бөлігі істен шықса да, қалғандары өз жұмысын атқара беруі тиіс.

Қазіргі таңда ақпаратты тасымалдау сенімділігі мен жеткізу жылдамдығын арттыру мақсатында дестелерді коммутациялау технологиясы қолданылады. Оның артықшылығы – ақпараттың ара қашықтыққа тәуелсіздігі, тасымалдау жылдамдығының жоғарылығы, ақпарат шығынының жоқтығы және ақаусыз икемді жұмыс істеуі. Дестелерді коммутациялау технологиясының мағынасы тасымалданатын ақпарат бірнеше шағын дестелерге (package) бөлінеді. Әдетте дестелердің ұзындығы тұрақты және алдын ала анықталады. Ақпарат жеткізілетін серверге дейінгі аралықтарға көптеген түйінді компьютерлер орналасады.

Олар арасында дестелер бір–біріне тәуелсіз, тек өзіне тән байланыс арналары арқылы жылдам тасымалданады, яғни әрбір дестe тасымалданатын соңғы серверге дейін тек өз маршрутымен (route) жеткізіледі. Қабылдау серверінде дестелер қайтадан біртұтас ақпаратқа біріктіріліп, өз иесіне, яғни ақпарат жіберілген абонементке беріледі. Интернет желісінің сенімді жұмыс атқаруын амалға асыру үшін, оны бір орталықтан басқармай, барлық түйінді компьютерлерді бірыңғай тең құқықты етіп жасаған. Интернет желісі торабындағы әрбір компьютер ақпаратты қабылдау-жөнелту жұмыстарын орындаумен қатар, басқа тораптардан келіп түскен ақпаратты да әрі қарай тасымалдау кезінде, маршрутты өзгерте алады.

Әр түрлі құрылғылар мен программалар арасында мәлімет алмасу қалай және қандай түрде атқарылатыны жайлы ережелер жиыны хаттама деп аталады. Хаттамалар мынадай топтарға бөлінеді:

- базалық
- қолданбалы

Интернет базалық TCP/IP хаттамасын пайдаланады. Қолданбалы хаттамалар Интернеттің арнайы қызмет баптарының хаттамаларынан, мысалы, http, ftp, telnet, т.с.с тұрады.

Қазіргі таңда Қазақстан Республикасы алдында ақпараттық қоғам құру мәселесі туындап отырғаны белгілі. Ақпараттық қоғам – басым көпшілігі ақпараттық өндіріспен, яғни ақпараттың жоғарғы формасы білімді сақтау, өңдеу және тасымалдау істерімен айналысатын қоғам. Ақпараттық қоғамды құру критерийлерінің бірі қажетті ақпаратты жедел және нақты түрде жеткізу, тасымалдау. Бұл критерийдің Интернет-технологияларға байланысты тұстары бар.

HTML (HyperText Markup Language) – гипермәтінді белгілеу тілі мәтіндерді Web-құжат (HTML-құжат) түрінде бейнелеуге арналған тэгтер жиынын анықтайды. HTML тэгтері құжатты туралауға, пішімдеуге және безендіруге мүмкіндік береді. Интернетке арналған мұндай гипермәтінді Web-құжаттар баспаға шығаруға емес, тек монитор экранында көруге арналған. Сондай-ақ, HTML тілі Web-құжатқа гипермәтін мен қозғалмайтын объекттерді бейнелеу мүмкіндігін береді.

DHTML (Dynamic HTML) тілі HTML-дің мүмкіндіктерін кеңейтуге арналған гипермәтінді белгілеу тілі. Бұл тіл, атынан көрініп тұрғандай, Web-құжатқа қозғалыс енгізеді, яғни мәтіннің түсін өзгертеді, кейбір элементті жасырып, кейбірін экранға шығарады т.с.с. Тілді MicroSoft және NetsCape корпорациялары бір-бірінен тәуелсіз жасап шығарған. Сондықтан DHTML-де жазылған код Internet Explorer мен NetsCape Navigator браузерлерінде әр түрлі жұмыс істейді. Соңғы кезде W^3C консорциум тілдің бірыңғай стандартын шығарған.

HTML (HyperText Markup Language - гипермәтіндік белгілеу тілі) - қарапайым мәтіндерді гипермәтіндік түрде бейнелеуге арналған тэгтер жиыны мен ережелерін анықтайды [2].

HTML тілінің тэгтері мәтінді мониторда бейнелеуге, пішімдеуге және безендіруге мүмкіндік береді. Сондай-ақ Web-құжатқа сілтеме, сурет, кесте, баннер, фрейм сияқты нысандарды енгізе алады. Сонымен қатар басқа бағдарламалау тілдерінде құрылған скрипт, апплет, анимация және т.б. нысандарды да пайдалануға болады. Жалпы, тек қана HTML тілін пайдаланып Web-құжат құру жеткіліксіз. Қазіргі таңда HTML тілінің мүмкіндіктерін кеңейтуге арналған қосымша Интернет-технологиялар дамуда. Мұндай технологияларға төмендегілер кіреді:

- CSS – каскадты стильдер кестесі;
- JavaScript, VBScript, ActionScript тілдерінде жазылған скрипттер;
- Java тілінде жазылған апплеттер;
- PHP тілінде жазылған PHP-скрипттер;
- Perl, C++ тілдерінде жазылған CGI-скрипттер;
- GIF, JPEG, PNG форматындағы суреттер;
- SWF-файлдар және Flash-анимациялар;
- AVI, MPEG форматындағы видеоклиптер;
- WAV, MP3, MIDI форматындағы аудиоклиптер.

CSS (Cascade Style Sheets – каскадты стильдер кестесі) технологиясы желідегі бірнеше Web-құжат мәтіндерін бірыңғай пішімдеуге арналған. Бұл технология Web-сайт парақтарының монитор экранында бірдей бейнеленуін және жылдам жүктелуін қамтамасыз етеді. CSS технологиясын пайдаланып Web-сайттағы барлық құжаттағы мәтіндердің шрифтін, түсін, өлшемін, туралау пішімдерін т.с.с. өзгертуге болады.

Java-апплет деп Java тілінде жазылған бағдарламаны айтады. Бұл бағдарлама компиляцияланып, пайда болған кодты байт-код деп атайды. Байт-код клиенттік компьютерлерде Java виртуал машинасы көмегімен орындалады. Java-апплеттер көлемі бойынша ықшам және компьютер платформасына тәуелсіз. Сондықтан Java-апплет Web-құжатқа енгізілсе, браузер оны автоматты түрде Java виртуал машинасы көмегімен орындайды.

Скрипт (Script) – HTML тілінің мүмкіндіктерін кеңейтуге арналған, арнайы бағдарламалау тілінде жазылған ықшам бағдарлама. Бағдарламалау тілі ретінде JavaScript, VBScript, Java, Perl, C++, т.с.с. тілдері қолданылады. Скрипттер әртүрлі оқиғаларды өңдейді, мысалы, тышқанды шерту, курсордың экран бетіндегі қозғалысы. Оқиғаның жауабы ретінде Web-құжаттың экрандағы түрінің өзгеруі, клиент-компьютерден серверге ақпарат жіберу сияқты іс-әрекеттер орындалады.

Скрипттер Web-жобада әртүрлі оқиғаларды өңдейді, мысалы, тышқанның пернесін шерткендегі іс-әрекет, курсорды мониторда қозғалту, терезе өлшемдерін өзгерту. Скрипттерді орындалу ортасына байланысты екі топқа бөледі: клиенттік (JavaScript, VBScript, ActionScript); серверлік (CGI-скрипт, PHP-скрипт).

Клиенттік скрипттер Web-жобаның HTML тілінде жазылған кодының ішіне тікелей жазылады. Клиенттік скрипттердің негізгі қасиеті, олар тұтынушы компьютерінде, яғни Web-браузерде HTML-кодпен бірге орындалады. Скрипттер компиляцияны қажет етпейді, яғни компилятордың орнына Web-браузер жеткілікті. Бұл скрипт құруды жеңілдетеді, уақытты үнемдейді. Клиенттік бағдарламалау тілдеріне JavaScript (Oracle Corporation), VBScript (Microsoft), ActionScript (Adobe Systems) тілдері жатады. Олардың ішінде JavaScript тілі қазіргі заманғы барлық Web-браузерлерде қолданылады, ал VBScript тілін тек қана Internet Explorer браузері пайдаланады. Жалпы, JavaScript тілі Java-ның, ал VBScript тілі Visual Basic-тің скрипттік версиялары болып саналады.

Скрипттердің негізгі кемшілігі, олардың орындалуы Web-браузерге тәуелді. Біріншіден, егер тұтынушы қаламаса скриптті орындауға жібермей қоюға болады. Ол үшін Web-браузердің арнаулы опциясын баптау жеткілікті. Екіншіден, скрипт әртүрлі Web-браузерлерде кейде әртүрлі орындалады. Сондықтан скриптті бірнеше Web-браузерде қатар тексеру қажеттігі туындайды.

PHP (Hypertext Preprocessor – гипермәтінді препроцессор) тілі серверде орындалатын сценарий жазуға арналған тіл. Браузер URL-адресте (Uniform Resource Locator – ықшамдалған ресурстар көрсеткіші) көрсетілген PHP тілінде жазылған Web-құжатты орындау үшін Web-серверге сұраныс жасайды. Web-сервер PHP-процессорын жүктеп, PHP-кодты орындайды ды, нәтижені Web-серверге қайтарады. Ал Web-сервер өз кезегінде нәтижені браузерге терезеде бейнелеу үшін жібереді.

1.2 IP-адрес және URL-адрес

Бүкіләлемдік немесе корпоративтік желіге қосылған әрбір компьютердің жеке IP-адресі (Internet Protocol) болатыны белгілі. Бұл сіздің ұялы телефоныңыздың номері сияқты. Басқа абоненттер сізді осы номер бойынша тауып алады. IP-адрес нүктелермен бөлінген төрт ондық бүтін саннан тұрады. Сандардың өзгеру аралығы 0-ден 255-ке дейін. Себебі бір байтқа осынша екілік сан сияды. Мысалы: М.Ауезов атындағы Оңтүстік Қазақстан мемлекеттік университеті серверінің адресі 192.168.32.1.

IP-адресі екіге бөлінеді:

- тұрақты;
- динамикалық.

Желіге қосылған Web-серверлер тұрақты IP-адреске ие болады. Олар күні-түні жұмыс істейді, яғни тұтынушыға тұрақты түрде қызмет көрсетеді. Компьютерді осылай адрестеу 256^4=4,3 млрд физикалық IP-

адрес бөлу мүмкіндігін береді екен. Мұндай адрес 32-разрядты екілік сан түрінде (4 байт) жедел жадыда сақталады.

Интернеттегі адрестеудің екінші түрін URL-адрес (Uniform Resource Locator – Ықшамдалған ресурстар көрсеткіші) деп атайды. Кейде домендік адрес (domain, облыс, аймақ, регион) деп те аталады. Мұндай URL-адрес компьютерлік желідегі әрбір компьютерге және әрбір құжатқа беріледі. Мұның негізгі себебі желідегі әрбір компьютердің IP-адрестерін жаттау тұтынушыға қиындық тудырады. Сондықтан IP-адрестермен қатар, мағынасы бар URL-адрестер енгізілген. Бұл ұялық телефон нөмірінің жанына оның иесінің атын жазған сияқты.

Домендік URL-адрестің жазылу ережесі мынадай [3]:

хаттама://компьютер аты. домен аты [/каталог/ішкі_каталог/файл аты]

Бірінші болып қолданылатын хаттама көрсетіледі, мысалы, http. Келесі компьютер аты жазылады да, одан кейін домен аты тұрады, мысалы, ukgu.kz. Адрес соңында кейде сервердегі каталог аттары мен файл аттары жазылады, мысалы inform/dissert.html. Жалпы Интернетте домен, каталог, бума және файл атын жазуда 8.3 форматын ұстаған дұрыс. Сонымен қатар, тік жақша ішіндегі адрес бөлігі жазылмаса да болады. Бірнеше нақты адрестермен мысал келтірейік:

http:// www.ukgu.kz/ inform/dissert.html
http:// www.google.kz

Әдетте URL-адрестер жазуда браузерлерге хаттаманы көрсетпесе де болады. Браузерлер хаттаманы Интернеттің қызметіне қарап, мысалы www, анықтайды. Сондықтан көрсетілген мысалдарды былай қысқаша жазуға да болады:

www.ukgu.kz/ inform/dissert.html
www.google.kz

Сонымен, домендік URL-адрестің жазылу ережесі төменде беріліп отыр:

- сервер жұмыс істеп тұрған хаттаманы, егер Интернет қызметі көрсетілсе көрсетпесе де болады (WWW қызметі үшін http хаттамасы);
- Интернетте бір адресті көрсету жеткілікті: компьютердің IP-адресі немесе URL-адресі;
- сервер өңдейтін TCP портын көрсетпесе де болады, мысалы, WWW қызметі үшін порт номері 8080. Компьютердің өзі автоматты түрде тауып алады;
- Web-сервердегі HTML-құжаттың адресін көрсетпесе де болады, Web-сервердің бір адресін көрсету жеткілікті.

Мысалы: М.Ауезов атындағы Оңтүстік Қазақстан мемлекеттік университеті адресі былай жазылады:

- www.ukgu.kz
- 92.168.32.1

Енді DNS (Domain Name System – домендік атаулар жүйесі) жүйесін атап кетейік. DNS жүйесі IP-адрес пен URL-адрес арасында байланыс орнататын, оларды бірінен-біріне ауыстыратын жүйе. DNS жүйесі сақталатын компьютерді DNS-сервер деп атайды [5].

DNS жүйесі бойынша домендік адрес солдан оңға қарай тізбектеп оқылады. Сол жақтағы домен ең жоғарғы домен болып есептеледі. Әдетте домен атауы үш-бес атаудан тұрады (стандарт бойынша жетеуге дейін деп берілген). Домен атауы екі түрде беріледі:

- қызмет бабына қарай;
- аймақтық.

Қызмет бабына қарай қойылған домендік атаулар алғаш рет АҚШ-та пайда болған болатын. Интернеттің де АҚШ-та пайда болғаны сияқты. Бұл домендік атаулар мекеме типін көрсетіп, үш әріптен тұрады. Олар төменде берілген:

- com – коммерциялық ұйымдар;
- edu – білім мекемелері, университеттер;
- int – халықаралық ұйымдар;
- gov – мемлекеттік ұйымдар;
- net – желілік ұйымдар;
- mil – әскери ұйымдар;
- org – басқа мекемелер.

Содан бері, WWW жүйесінің ауқымды дамуына байланысты аймақтық домендер мемлекет атына сәйкес қойыла бастады. Олар екі әріптен тұрады. Аймақтық домен атаулары мысалдар келтірейік.

- us - АҚШ
- ru - Ресей
- kz – Қазақстан
- jp – Жапония
- ua - Украина
- uz - Өзбекстан

Осы айтылған атаулардан басқа кейде альтернативті домендік атаулар да кездеседі. Оларды ішінде АҚШ мемлекеті жанында құралған ICANN (Internet Corporation for Assigned Names and Numbers) мемлекеттік емес ұйымын атауға болады. Осы ұйымның домен атауларына мысалдар келтірейік.

- museum — музейлер

- name — жеке есімдер
- aero — авиакомпании
- coop — бірлескен, коммерциялық
- biz — бизнес
- info - ақпараттық
- pro — кәсіпқой
- tv - телевидения

1.3 Интернет қызметтері

Қазіргі таңда Интернеттің тұтынушыға көрсететін көптеген сервистік қызмет түрлері (web service) бар. Интернеттің әр қызметі ақпарат таратудың, тасымалдаудың және алмасудың әртүрлі функцияларын атқарады. Сондай-ақ, мұндай қызметтердің кейбіреуі ескіріп жойылса, кейбіреуі жаңадан пайда болып жатыр. Интернеттің сервистік қызметтері ішінен негізгілерін қысқаша қарастырып өтейік.

Қолданыстағы Интернеттің қызметтері көп, жыл сайын жаңа қызметтер қосылуда. Соның ішінде ең көп тарағаны және қолданылатыны WWW жүйесі.

WWW (W3, World Wide Web – бүкіләлемдік өрмек) - Интернеттегі гипермәтіндік ақпаратты іздеу жүйесі, кейде оны қысқартып Web деп те атайды. WWW негізінде HTTP (Hyper Text Transfer Protocol) - гипермәтінді тасымалдау хаттамасы жұмыс істейді.

WWW жүйесін 1989 жылы Тим Бернерс-Ли (Timothy John Berners-Lee) мен Роберт Кайо (Robert Cailliau) құрастырған. Сондай-ақ Тим Бернерс-Ли HTTP хаттамасы, URI/URL идентификаторы мен HTML технологиясының авторы болып саналады. Сонымен қатар, алғашқы Web–сервер мен браузерді де құрастырған [6].

1994 жылдан бастап жүйені Интернет консорциум (World Wide Web Consortium, W3C) дамытып келеді. Бұл консорциум негізінен Интернет және WWW жүйесі үшін бірыңғай технология мен стандарттарды дайындау және енгізумен айналысады. Ал осы стандарттарды есептеу техникасы мен бағдарламалық жабдықтама құрушы корпорациялар пайдаланады. Нәтижеде әр түрлі корпорациялардың техникалық және бағдарламалық өнімдері бір-бірімен үйлеседі. Сондықтан Интернет технологиялары ыңғайлы және әмбебап. Айта кетейік, Интернет консорциум өз технологиялары мен стандарттарын тегін таратады. Олардан кез-келген заңды немесе физикалық тұлға, ешқандай қаржылық міндетсіз пайдалана алады.

WWW жүйесінің мүмкіндіктері оның төменде көрсетілген құрамдас бөліктерімен сипатталады [7]:

- URL (Universal Resource Locator) – ықшамдалған ресурстар көрсеткіші;
- HTML (Hyper Text Markup Language) – гипермәтіндік белгілеу тілі;
- HTTP (Hyper Text Transfer Protocol) - гипермәтіндік ақпаратты тасымалдау хаттамасы;
- CGI (Common Gateway Interface) – жалпы шлюздік интерфейс.

WWW жүйесінде ақпарат Web-құжаттарда (гипермәтіндік) сақталады деп айтып өттік. Гипермәтіндік құжаттарда бірінен екіншісіне ауысу үшін гиперсілтемеден пайдаланады. Гиперсілтеме деп ішіне арнаулы код, яғни екпінді элемент (anchor) орналасқан мәтінді айтады. Сондай-ақ, гиперсілтеме келесі Web-құжатқа немесе осы Web-құжаттың басқа бір бөлігіне өтуге арналған белгіленген сөз. Гиперсілтеме мәтіннің ішінде асты сызылған сөз түрінде орналасқан. Егер курсорды сол сөздің үстіне әкеліп, тышқан пернесін шертсек, онда ол сілтеп тұрған басқа Web-құжатқа ауысамыз. Бұл құжаттар ағымдағы компьютерде немесе Интернеттегі кез келген Web–серверде орналасқан болуы мүмкін.

Web-құжаттарды браузер (browser) деп аталатын арнайы программалар өңдейді. Олар Web-құжаттарды мониторда бейнелеп, парақтап көру, оқу мүмкіндігін береді. Браузерлер ішінде Қазақстанда ең көп тарағандары Internet Explorer (Microsoft, 1995), Opera (Opera Software, 1994), Mozilla FireFox (Mozilla Corporation, 2004) және Google Chrome (Google Inc., 2008) болып саналады.

E-mail (Электрондық пошта). Электрондық пошта - желі қызметтерінің ішіндегі электрондық ақпарат алмасудың ең негізгісі. Электрондық пошта арқылы барлық құрылықтарда тұратын адамдар бір-біріне электрондық хаттар мен файлдар жібере алады. Электрондық поштаның жұмысы кәдімгі күнделікті өмірдегі поштаға ұқсас болғандықтан, оның жұмыс істеу үрдістерін меңгеру тұтынушылар үшін аса көп қиындық туғыза қоймайды. Оның негізгі айырмашылығы – физикалық нысандар емес, электрондық файлдар жіберіледі. Сонымен қатар, электрондық пошта географиялық адрестеуді емес, электрондық адресті қолданады. Электрондық поштаның адресі төмендегі түрде жазылады.:

тұтынушы_аты@домен_аты.

Мысалы: info@mail.ru, us-ibr@rambler.ru, us-ibr@nur.kz

Тегін пошталық серверлерге мысалдар:

- www.gmail.com
- www.mail.ru
- www.rambler.ru
- www.yandex.ru

- www.ok.kz
- www.nur.kz

FTP (File Transfer Protocol – файлдарды тасымалдау хаттамасы). Қашықта орналасқан компьютердің каталогтары және файлдарымен жұмыс істеуге арналған хаттама мен программа. FTP программалары FTP-сервердің каталогтары мен файлдарын көріп, олардың бірінен біріне ауысып, қажеттісін көшіріп алуға, тасымалдауға және жаңартуға мүмкіндік береді. Олардың жұмыс істеу үрдісі мен интерфейсі файл менеджерлерге ұқсас. Тіпті кейде олар файл менеджерлердің ішіне опция ретінде орнатылады.

Telnet (TErminaL NETwork – желілік терминал). Бұл атаумен клиенттің қашықта орналасқан Web-сервермен байланысын қамтамасыз ететін хаттама мен программаны айтады. Олардың арасында байланыс орнатылғаннан кейін, тұтынушы қашықтағы компьютердің амалдық жүйесіне кіріп, сол компьютерде орнатылған программалармен өз программалары сияқты жұмыс істей алады. Telnet қызметі негізінен telnet хаттамасынан пайдаланады, кейде басқа TCP деңгейіндегі хаттамаларға (транспорттық деңгей) тікелей байланыс жасау үшін TCP, FTP, IRC хаттамаларын да қолданады.

Usenet (User Network - тармақталған дискуссия клубы, телеконференция, жаңалықтар тобы). Бұл қызметтің электрондық поштадан айырмашылығы – жіберілген ақпаратты бір абономент ғана емес, бірнеше абоненттер тобы (телеконференция) алады. Әрбір телеконференция бір тақырыпқа (әлеуметтік, ғылым, өнер, жаңалық, хобби, спорт, т.б.) арналады және оның өзіндік атауы болады. Мұндағы қарастырылатын тақырыптар ауқымы өте кең және әр бағытта бола береді. Сондықтан, өте ауқымды тақырыптан тек қана бір ғана сұраққа дейін қамтылады. Тақырыптар иерархиялық құрылымға ие. Әртүрлі тақырыптарды талқылау кезінде телеконференцияға қатысушылар тең құқықты болып келеді.

Бүгінгі таңда Usenet он мыңнан астам дискуссиялық топтарды (NewsGroups) немесе телеконференцияларды құрайды. Сондықтан ол пікір алмасу құралы болып тұр. Телеконференцияларды бірнеше топтарға бөледі:

- news – телеконференцияларға байланысты мәселелер;
- comp – компьютерлер мен программалық жабдықтама;
- rec – хобби, ойын-сауықтар мен өнер;
- sci – ғылыми-зерттеу жұмыстары;
- soc - әлеуметтік мәселелер;
- talk – спортқа байланысты мәселелер;
- misc – басқа мәселелер.

IRC (Internet Relay Chat) – нақты уақыт кезеңінде телеконференция өткізудің бір түрі, тұтынушылардың бір-бірімен әңгіме-дүкен құруға болатын қызметі. IRC-серверлері мен IRC-клиенттері көмегімен тұтынушылар бір-бірімен «виртуальды» әңгіме өткізуді пернетақтада сөздерді теру арқылы жүргізеді.

ISQ (I seek you – мен сені іздеймін) – желі тұтынушыларына нақты уақыт кезеңінде хат-хабарлармен алмасып, бірден чат (chat- әңгіме) құрып, файл жіберіп, әңгімелесуге мүмкіндік беретін қызмет түрі.

IP-телефония (ағылшынша VoIp, Voice over ip). Интернетте дыбыстық сигналдар арқылы ақпарат алмасуды ұйымдастыру қызметі. Ол тек IP хаттамасы негізінде ақпарат алмасудың жеке бір түрі. Дыбыс цифрлық түрге түрлендіріледі де, желідегі қарапайым ақпарат дестелері тәрізді тасымалданады.

Ағындық мультимедиа (stream media). Мультимедиа – бұл провайдерден тұтынушыға жеткізілетін үздіксіз ағындық ақпарат. Ағындық ақпарат деп телекоммуникация арқылы таралатын радио, телевидения және видео т.б. ақпаратты айтады. Жалпы, мультимедиа үлкен көлемді жадыны қажет етеді. Сондықтан оларды сақтау және тасымалдау шығыны үлкен.

Қазіргі таңда Интернеттің жоғарыда аталған сервистік қызметтері аясында жаңа қызмет түрлері кездеседі. Осы қызметтерді қысқаша қарастырайық.

Уики (wiki) – Web-сайттағы программа арқылы оның құрылымы мен контентін тұтынушылардың өздері өзгерте алатын қызмет. Мәтінді пішімдеу және оған әртүрлі нысандарды қосу мүмкіндіктері бар. Осы технология негізінде Уикипедия (Wikipedia) энциклопедиясы жобасы құрылған.

Web-форум - Web-сайттың тұтынушылары бір-бірімен пікір алмасатын қосымша, скрипт. Форумда ақпарат иерархиялық түрде - бөлімдер, тақырыптар, хабарлар болып бөлінеді. Тұтынушы администраторға немесе басқа тұтынушыларға хабарлар, ұсыныстар, жарнама жазуы мүмкін. Форум тақырыбын тұтынушылардың өздері таңдайды. Форумның қысқартылған нұсқасын қонақ кітабы деп атайды.

Блог (blog – Интернет-күнделік, онлайн-күнделік). Күнде жеке тұлға тарапынан ақпаратпен толықтырылып отыратын Web-сайт. Ақпарат ретінде қысқа жазбалар, яғни мәтін, сурет, аудио және видео қолданылады. Блог авторларын блоггер деп атайды. Блогтың кәдімгі күнделіктен

айырмашылығы оқырман өз пікірін, түсініктемесін жазуы немесе блоггермен дебатқа кірісуі мүмкін. Ең көп таралған блог-хостингдерге «LiveJournal» (АҚШ), Dirty.ru (Ресей), Blogger (Google Inc.), Twitter (АҚШ) жатады. Блогтар төмендегі түрлерге бөлінеді:

- авторы бойынша (дербес, ұжымдық, корпорациялық);
- тақырып бағыты бойынша (саясат, тұрмыс, саяхат, білім, мода, музыка, спорт, кино, денсаулық және т.б.);
- мультимедиа түрі бойынша (мәтіндік, фотоблог, артблог, блогкастинг, видеоблог, микроблог);
- контент ерекшілігі бойынша (контент-блог, мониторинг-блог, цитата-блог, спам-блог, dofollow-блог, фэйк-блог);
- техникалық негізі бойынша (автономды, блог-хостинг).

Интернет-дүкен (e-shop). Интернет желісі арқылы тауар сататын Web-сайт. Тұтынушыдан он-лайн режимінде тауарға сұраныс қабылдайды, рәсімдейді, төлем түрін және жеткізу жолын ұсынады.

Әлеуметтік желілер (social networking service) - Тұтынушыларды біріктіретін әлеуметтік қоғамдастық Web-сайттар. Әлеуметтік желілердің негізгі сипаттамалары:

- тұтынушыға жеке профиль береді; профильде нақты деректер (аты-жөні, суреті, туылған жылы, мекен жайы, жұмыс орны, т.с.с.) көрсетіледі;
- ақпарат алмасуға мүмкіндіктер береді (суреттер орналастыру, хабар жазу, түсініктеме беру, және т.б.):
- басқа тұнушылармен қарықатынас жасау (достасу, әлеуметтік топ құру, жеке хабарлармен алмасу және т.б.).

Әлеуметтік желілерге мысалдар:

- Facebook
- Twitter
- MySpace
- YouTube
- Вконтакте
- Одноклассники.ru
- Мой Мир
- В кругу друзей
- Мой Круг

Интернет-радио (Web-радио) - үздіксіз ағындық радиосигналдарды тарату қызметі. Сонымен қатар Web-радио деп ағындық радиосигналдарды таратушы радиостанцияларды айтады.

Интернет-телевидение (on-line TV, IP-TV - Internet Protocol Television, IP-телевидение) – IP хаттамасы негізінде таратылатын сандық телевидение технологиясы. Клиенттік құрылғы ретінде компьютер, медиа-плеер, SMART TV қолданылады.

Қазақстан Республикасы телекоммуникациясының ұлттық операторы болып саналатын «Қазақтелеком» ААҚ-ы iDTV (integrated digital television – енгізілген сандық телевидения) байланыс түрін қолданады.

Іздеу серверлері мен каталогтары – Web-интерфейсті программалық-техникалық кешен. Оның мақсаты тұтынушы сұранысы бойыншы Интернеттен ақпарат іздеу. Әдетте іздеу сервері дегенде, осы іздеу серверлері мен каталогтары орналасқан Web-порталды түсінеміз. Ал іздеу жүйесінің программалық бөлігін іздеу машинасы деп атайды. Іздеу серверлері мен каталогтарына мысалдар келтірейік.

Шет елдік іздеу серверлері мен каталогтары:

- www.google.com
- www.yahoo.com
- www.ask.com
- www.altavista.com
- www.msn.com

Ресейдің іздеу серверлері мен каталогтары:

- www.google.ru
- www.mail.ru
- www.yandex.ru
- www.rambler.ru
- www.aport.ru

Қазақстанның іздеу серверлері мен каталогтары:

- www.google.kz
- www.yandex.kz
- www.qazaq.kz/izdew.php
- www.catalog.site.kz
- www.resurs.kz
- www.catalog.tabu.kz
- www.zero.kz
- www.poisk.kz
- www.catalog.desk.kz
- www.kazakh.ru
- www.catalog.awstudio.kz

1.4 OSI эталондық үлгісі мен TCP/IP хаттамасы

1984 жылы ISO (International Standards Organization) ұйымы ашық жүйелердің өзара қарым-қатынасынынң эталондық үлгісін OSI (Open System Interconnection basic reference model) ұсынған болатын. Содан бері ол Интернет ащық жүйесінің халықаралық стандартына айналып отыр. Желілік өнімдерді шығаруда корпорациялар осы стандарттан кең пайдаланады. Сондықтан бұл бөлімде осы OSI эталондық үлгісін қарастырайық.

OSI эталондық үлгісі жүйелердің қарым-қатынасының әртүрлі деңгейлерін анықтап, әрбір деңгейдің функционалдық мүмкіндіктерін көрсетеді. Сондай-ақ әрбір деңгейге спецификациялар тағайындайды. Ашық жүйелерді құру барысында спецификацияларды пайдалану, осы жүйелер үшін әртүрлі техникалық және бағдарламалық құрал-жабдықтарды кешенді түрде жасайтын корпорацияларға жаңа типтегі өнімдерді құрастыруға және дамытуға жол ашады. Жалпы, нақты жүйелер үшін толық ашықтық қол жетпейтін идеал болып саналады. Әдетте, ашық деп аталатын жүйелердің өздерінде де, ашықтық ұғымы, осы жүйелердің сыртқы интерфейстерді қолдайтын кейбір бөлімдеріне ғана қатысты. Мысалы, UNIX амалдық жүйесінің туыстарының ашықтығы, барлығынан бұрын, ядро мен қосымшалар арасындағы стандартты бағдарламалық интерфейске негізделген. Мұндай ашықтық қосымшаларды Unix-тің бір версиясынан екіншісіне жеңіл көшіруге жағдай жасайды.

OSI эталондық үлгісі бойынша өзара қарым-қатынас құралдары жеті деңгейге бөлінеді (жоғарыдан төменге):

- қолданбалы (application);
- ұсыну (presentation);
- сеанстық (session);
- транспорттық (transport);
- желілік (network);
- арналық (data link);
- физикалық (physical).

Әдебиеттерде OSI эталондық үлгісі деңгейлерін жетінші деңгейден бастап сипаттау қалыптасқан [8, 9]. Ең жоғарғы 7-ші деңгей қолданбалы деп аталады, ал ең төменгісі, яғни 1-ші деңгей физикалық деп аталады. OSI эталондық үлгісі деңгейлерінің функциялары төмендегі 1-кестеде берілген.

Кесте 1 - OSI деңгейлерінің функциялары

деңгей (layer)	**функциялары**
қолданбалы	желілік қызметтерге қолжетімділік
ұсыну	деректерді ұсыну және шифрлау
сеанстық	байланыс сеансын басқару
транспорттық	соңғы торап нүктесімен тікелей байланыс

	және сенімділік
желілік	маршрутты анықтау және логикалық адрестеу
арналық	физикалық адрестеу
физикалық	тасымалдау ортасымен, сигналдармен және екілік деректермен жұмыс жасау

OSI үлгісінің әр деңгейінің өзінің хаттамалары бар. Әрбір хаттама өз деңгейіндегі немесе бір деңгей жоғары немесе төменгі деңгейдегі хаттамалармен өзара әрекеттеседі. Өз деңгейіндегі хаттамалармен қарым қатынасын горизонталь, ал бір деңгей жоғары немесе төменгі деңгейдегі хаттамалармен қарым қатынасын вертикаль өзара әрекеттесу деп атайды.

OSI үлгісінің кез келген хаттамасы тек қана өз деңгейінің функциясын атқарады. Олар басқа деңгейдегі хаттамалардың немесе басқа альтернативті үлгідегі хаттамалардың функциясын орындамайды. Сол сияқты, әрбір деңгейдің өз деректер типі анықталған. Хаттамаларда тек қана осы деректер типімен, яғни деректердің логикалық бөлінбейтін элементімен ғана жұмыс орындалады. Физакалық деңгейде ең кіші элемент ретінде – бит, арналық деңгейде – кадр, желілік деңгейде – пакет (датаграмма), транспорттық деңгейде – сегмент анықталған. Сол сияқты, деректердің кез келген фрагменті, яғни кадр, пакет, сегмент - хабар түрінде анықталады. Осы хабар - сеанстық, ұсыну және қолданбалы деңгейлердің деректер типі болып саналады (2-кесте). Базалық желілік технологиялар ретінде физикалық және арналық деңгейлер алынады.

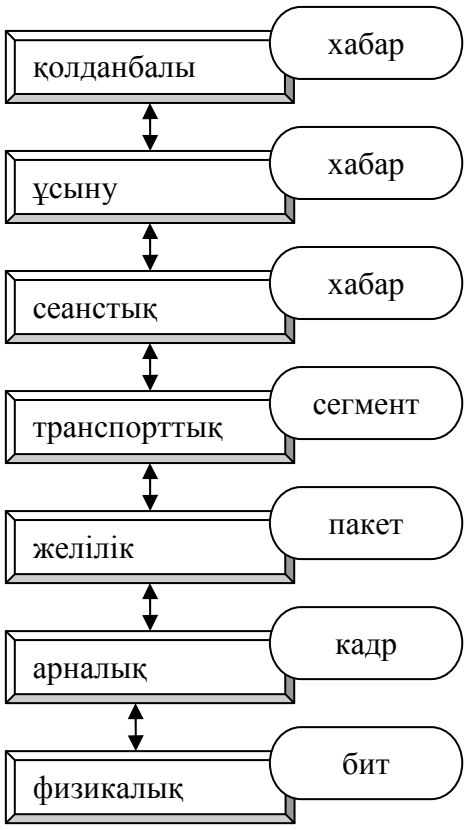
Кесте 2 - OSI деңгейлерінің деректер типі

деңгей	**деректер типі**
қолданбалы	хабар
ұсыну	хабар
сеанстық	хабар
транспорттық	сегмент
желілік	пакет (датаграмма)
арналық	кадр
физикалық	бит

Әрі қарай, осы деңгейлерді жеке-жеке қарап шығайық (сурет 2).

Қолданбалы деңгей (application layer). OSI үлгісінің ең жоғарғы деңгейі, ол тұтынушы қосымшаларының желімен өзара қарам-қатынасын анықтайды. Бұл деңгей тұтынушы қосымшаларын тікелей қолдап, олардың желілік қызметтерді пайдалануын, алыстағы деректер қоры мен файлдарға тікелей қолжетімділікті, электрондық поштаны тасымалдауды қамтамасыз етеді. Сонымен қатар, қызметтік түрдегі ақпараттың тасымалдануына жауап береді, қосымшаларға қателер туралы ақпаратты жеткізеді, ұсыну

деңгейіндегі хаттамаларға сұраныстар ұйымдастырады. Төменде орналасқан деңгейлер қолданбалы деңгейдің сұраныстарын орындайды. Қолданбалы деңгей қосымшалардың желі ресурстарына жалпы енуін, деректер ағындарын, қателерді өңдеуді басқарады. Қолданбалы деңгейдегі деректер әдетте хабарлар деп аталады. Қолданбалы деңгейдегі хаттамаларға HTTP, SMTP, POP3, FTP, TELNET т.б. жатады.



Сурет 2 – OSI үлгісінің деңгейлері

*Ұсыну деңгейі (presentation layer)***.** OSI үлгісінің бұл деңгейі деректерді түрлендіру форматтары мен оларды шифрлау-дешифрлау хаттамаларын анықтайды. Қолданбалы деңгейдегі қосымшалардан алынған сұраныстарды желідегі тасымалдау форматтарына түрлендіреді, ал желіден алынған деректерді қоымшалардың форматтарына түрлендіреді. Осы деңгейде деректерді сығу және ашу немесе шифрлау және дешифрлау орындалады. Сол сияқты, сұраныстар өңделмейтін болса, оларды басқа желілік ресурстарға қайта жібереді. Сондықтан бұл деңгейді аудармашы десек те болады.

Ұсыну деңгейі көрші деңгейлер үшін ақпаратты түрлендірудің аралық деңгейі болып есептеледі. Қолданбалы деңгейден түскен деректер осы деңгейде жалпыға бірдей аралық формат түріне көшіріледі. Тасымалдану орындалғасын деректер аралық форматтан қолданбалы деңгейдегі форматқа қайта түрлендіріледі. Ұсыну деңгейі хаттамаларды түрлендіруге, деректерді тсымалдауға, оларды шифрлауға, шифрден шығаруға,

қолданылатын символдар кодын ауыстыруға қатысады. Сонымен қатар, тасымалданатын биттер санын азайту үшін деректерді сығуға да қатысады.

Осы айтылғандарды түсіну үшін, айталық, екі желі берілген болсын. Біреуі EBCDIC – ақпаратты алмасу үшін кеңейтірілген екілік кодын пайдалансын. Әдетте IBM мэйнфреймі осы кодтан пайдаланады. Екіншісі ASCII – ақпаратты алмасудың американдық стандарт кодын пайдалансын. Бұл кодты көптеген корпорациялардың дербес компьютерлері қолданады. Егер осы екі желі арасында ақпарат алмасу қажеттілігі туындаса, онда деректердің форматтарын түрлендіреді, яғни бірінен-біріне өткізеді. Сол сияқты, бір желіде графикалық стандарт формат ретінде TIFF (Tagged Image File Format) алынған, ал екіншісінде JPEG (Joint Photographic Expert Group) болуы мүмкін. Мұндай жағдайда графикалық форматтар түрлендірілуі тиіс.

Ұсыну деңгейдегі хаттамаларға AFP (Apple Filing Protocol), ICA (Independent Computing Architecture), LPP (Lightweight Presentation Protocol), NCP (NetWare Core Protocol), NDR (Network Data Representation), XDR (eXternal Data Representation), X.25 PAD (Packet Assembler/Disassembler Protocol) жатады.

Сеанстық деңгей (session layer). Бұл деңгей әртүрлі компьютерлердегі екі қосымша үшін сеанс деп аталатын қосылысты құрады, байланыстырады және осы байланысты ұзақ уақыт сақтайды. Деңгей сеансты құру және аяқтауды, ақпарат алмасуды басқарады. Осы деңгейде тұтынушылар сұраныстарын деректер ағындарының арасына бақылаушы нүктелер қою арқылы синхронизациялау жүзеге асырылады. Желілік қателік туған жағдайда, соңғы бақылау нүктесінен кейін орналасқан деректерді ғана қайта жібереді. Сонымен қатар, осы деңгейде өзара әсерлесетін үрдістер арасында интерактивтивтік басқару, яғни жіберу қай жақтан орындалып жатқандығы, қашан және қанша уақытқа созылатындығын анықтау амалға асырылады.

Сеанстық деңгейдегі хаттамаларға ADSP (AppleTalk Data Stream Protocol), ASP (AppleTalk Session Protocol), H.245 (Call Control Protocol for Multimedia Communication), ISO-SP (OSI Session Layer Protocol (X.225, ISO 8327)) т.б. хаттамалары жатады.

Транспорттық деңгей (transport layer). OSI үлгісінің бұл деңгейі деректерді жіберушіден қабылдаушыға дейін жеткізуді қамтамасыз етуге арналған. Жіберушіден қабылдаушыға барар жолда деректер өзгеруі, типті жоғалып кетуі мүмкін. Сондықтан деңгей деректердің жоғалмай, қайталанбай, қатесіз жетуін сенімді түрде қамтамасыз етеді. Бұл жерде сенімділік кең мағыналы болуы мүмкін. Транспорттық деңгейдің хаттамалары көптеген сыныптарға бөлінеді. Кейбіреулері тек қана негізгі функциясын атқарады, яғни деректерді тек жеткізіп береді, оның қабалданғанын тексермейді. Ал кейбір хаттамалар сыныбы деректерді жеткізуге толық кепілдік береді. Бұл деңгейде ұзын хабарлар бірнеше

пакеттерде бөлінеді, ал қысқалары бір пакетке біріктіріледі. Бұл пакеттердің желі бойынша тиімді тасымалдануына қажет болады. Қабылдаушы компьютер пакеттерді ашып, алғашқы қалпына келтіреді де, олардың дұрыс жеткендігі жайлы сигналдар жіберіледі. Сондай-ақ транспорттық деңгей деректер ағындарын басқарады, қателерді тексереді, пакеттерді жіберу мен қабылдауда туатын мәселелерді тиімді шешеді.

Транспорттық деңгейдегі хаттамаларға ATP (AppleTalk Transaction Protocol), CUDP (Cyclic UDP), DCCP (Datagram Congestion Control Protocol), FCP (Fiber Channel Protocol), NBF (NetBIOS Frames protocol), NCP (NetWare Core Protocol), SCTP (Stream Control Transmission Protocol), SPX (Sequenced Packet Exchange), SST (Structured Stream Transport), UDP (User Datagram Protocol) хаттамалары кіреді.

*Желілік деңгей (*network layer*).* Бұл деңгей деректерді тасымалдау жолдарынын анықтауға арналған. Деңгей логикалық адрестер мен атауларды физикалық адрестер мен атауларға алмастырады. Нақты желілік шарттарға, мысалы, қызметтер приоритетіне және басқа факторларға байланысты жіберуші мен қабылдаушы компьютерлер арасындағы қысқа маршруттарды анықтайды. Сондай-ақ, пакеттерді коммутациялау, бағыттау сияқты желілік трафикке байланысты туатын мәселелер де осы деңгейде өз шешімін табады. Егер бағыттаушы желілік адаптер (маршрутизатор) деректердің үлкен бөліктерін жібере алмаса, онда осы деңгейде ол бөліктер бөлшектенеді. Ал қабылдаушы компьютер оларды жинақтап, алғашқы қалпына келтіреді. Осы деңгейде жұмыс істеуші бағыттаушы желілік адаптерлер шартты түрде үшінші деңгейдегі құрылғылар (деңгейдің OSI үлгісіндегі нөміріне байланысты) деп аталады.

Желілік деңгейдегі хаттамаларға IP/IPv4/IPv6 (Internet Protocol), IPX (Internetwork Packet Exchange), IPsec (Internet Protocol Security) хаттамалары жатады.

*Арналық деңгей (*data link layer*).* OSI үлгісінің бұл деңгейі желілердің өзара қарым-қатынасын физикалық деңгейде амалға асырады және пайда болатын қателіктерді қадағалайды. Физикалық деңгейден алынған биттерді кадрға біріктіріп, бүтіндігін тексереді. Егер қателік табылса, сол кадрды қайтадан құрады. Солай құрылған кадрларды желілік деңгейге жібереді. Жіберу барысында бұзылған немесе оларды қабылдаған жайлы хабар түспеген кадрлар қайта жіберіледі. Жалпы, кадр - деректерді орналастыруға арналған логикалық ұйымдасқан құрылым. Бұл желілік деңгейге деректерді жіберуді желілік бірігу арқылы қатесіз оқуға мүмкіндік жасайды. Көбінесе бірнеше өзара қарым-қатынас жасайтын компьютерлер бөліп пайдаланылатын желілерде, физикалық тасымалдау ортасы бос болмауы мүмкін екендігі есепке алынбайды. Сондықтан да арналық деңгейдің мақсаттарының бірі тасымалдау ортасының бостығын анықтау болып табылады. Осы деңгейде жұмыс істеуші коммутаторлар, көпірлер т.б. шартты түрде екінші деңгейдегі құрылғылар (деңгейдің OSI

үлгісіндегі нөміріне байланысты) деп аталады. Дегенмен, әр түрлі топологиялы желілерде хабарларды сапалы тасымалдау үшін арналық деңгейдің функциясы жеткіліксіз. Сондықтан OSI эталондық үлгісінде бұл мәселе келесі желілік және транспорттық деңгейлерге жүктеледі.

Арналық деңгейдегі хаттамаларға CAN (Controller Area Network), Econet, Ethernet, EAPS (Ethernet Automatic Protection Switching), FDDI (Fiber Distributed Data Interface), PPP (Point-to-Point Protocol), PPPoE (Point-to-Point Protocol over Ethernet) т.б. хаттамалары кіреді.

*Физикалық деңгей (*physical layer*).* Ең төменгі физикалық деңгей сигналдарды тасымалдауға арналған. Физикалық деңгей өзінен жоғары жатқан барлық деңгейлерден түсетін деректерді тасымалдайды. Бұл деңгейде құрылымдық емес деректер, яғни биттер ағындары физикалық орта арқылы, мысалы, желілік кабель, радиоэфир арқылы тасымалданады. Сондай-ақ, желілік тасымалдаушы мен желілік құрылғы интерфейстері арасында электрлік, толқындық, оптикалық, механикалық және функционалдық байланыс орнатылады.

Сонымен, физикалық деңгей бір компьютерден екіншісіне биттерді жіберуге арналған. Биттердің мазмұнының бұл деңгейге қатысты жоқ. Физикалық деңгей желілік кабель арқылы жіберілетін әрбір биттің ұзақтығын және оның сәйкес электрлік немесе оптикалық импульсқа көшуін айқындайды. Бұл деңгейде разъемдердің типтері, концентраторлар, сигналдарды қайтарушы және медиаконверторлар анықталады. Жалпы, физикалық деңгей желіге жалғанған барлық құрылғыларда қолданылады. Компьютерде физикалық деңгей функциясын желілік адаптер немесе тізбектелген порттар атқарады.

Физикалық деңгейдегі хаттамаларға IEEE 802.15 (Bluetooth), IRDA, EIA RS-232, EIA-422, EIA-423, RS-449, RS-485, DSL, ISDN, SONET/SDH, 802.11 Wi-Fi, Etherloop, GSM Um radio interface, ITU-T, TransferJet, ARINC 818, G.hn/G.9960 жатады.

Әрі қарай TCP/IP хаттамасын (Transmission Control Protocol / Internet Protocol) қарастырайық. TCP/IP хаттамасы Интернетте қолданылады кең қолданылады. Бұл хаттама ашық жүйеге негізделген және жоғарыда қарастырған OSI үлгісінің жүйелік архитектурасына сәйкес келеді.

TCP/IP хаттамасы әртүрлі орталы байланысты қамтамасыз ететін хаттамалар тобынан тұрады, яғни әртүрлі типті компьютерлер арасындағы үйлесімділікті орнатады. Бұл дегені, TCP/IP хаттамалар тобы кез келген есептеу палатформасында, кез келген амалдық жүйеде, кез-келген компьютер моделінде жұмыс істейді. Бір сөзбен айтқанда, TCP/IP хаттамалары арқылы кез-келген компьютерді желіге қосуға болады. TCP/IP хаттамалары бір-бірімен стекте (stack) жұмыс жасайды, яғни жоғарғы деңгейде жұмыс жасаушы хаттама төменгі деңгейдегі хаттамаларды өз ішіне алады (инкапсуляция). Сонымен қатар TCP/IP

хаттамалары Интернет ресурстарын пайдалануға мүмкіндік береді және желі үшін өнеркәсіп аумағындағы бағыттаушы хаттама болып есептеледі. Көбінесе TCP/IP желіаралық хаттама ретінде кең пайдаланылады. Сондықтан TCP/IP желілік қарым-қатынас жасаудың стандартына айналып отыр.

TCP/IP хаттамалар стегі, OSI эталондық үлгісі сияқты деңгейлерге бөлінеді. Тек қана бұл жерде деңгеңлер саны төртеу (жоғарыдан төменге):

- қолданбалы (application);
- транспорттық (transport);
- желілік (network);
- арналық (data link).

Осы дейгейлердің хаттамалары OSI эталондық үлгісінің функционалдық мүмкіндіктерін толық атқарады. TCP/IP хаттамалар стегінде IP-желісіндегі барлық тұтынушылардың қарым-қатынасы құрылады. Стек деректердің физикалық таралу ортасына байланысты емес.

Әрбір деңгей өзінің функционалдық жүктемесін орындайды. Әрі қарай осы деңгейдерді қарап шығайық.

Арналық деңгей (link layer). Бұл деңгейді кейде желілік интерфейс деңгейі деп те атайды. Оған компьютердегі желілік тақша мен амалдық жүйедегі сәйкес құрылғы драйвері жатады. Олар қосыла отырып, компьютердің желімен физикалық байланысының техникалық жабдықтамасын құрайды.

Арналық деңгейдегі хаттамаларға Ethernet, IEEE 802.11 Wireless Ethernet, SLIP, Token Ring, ATM және MPLS жатады.

Желілік деңгей (network layer). Бұл деңгейді кейде желіаралық қарым қатынас деңгейі деп те атайды. Ол пакеттерді желі бойынша таратумен айналысады. Пакеттерді маршруттарға бөлу осы деңгейде амалға асырылады.

TCP/IP хаттамалар стегіндегі пакеттерді коммутациялау қағидасы тасымалданатын деректерді бірнеше шағын пакеттерге бөлуден тұрады. Пакеттер біріне бірі тәуелсіз, тек қана өзіне тән қасиеті бар байланыс арналары арқылы, жылдам тасымалданатын етіп ұйымдастырылады. Әрбір пакет баратын орнына тек өз маршруты бойынша жеткізіледі. Қабылдау орнында пакеттер қайтадан біртұтас дерекке біріктіріліп, өз иесіне, яғни арналған адресатына берілуі тиіс.

Сондай-ақ, желінің жұмыс істеуін сенімді басқару мақсатында, оларды бір орталықтан басқармай, барлық түйінді компьютерлер бірдей, тең құқықты етіп жасалады. Яғни әрбір желі торабындағы компьютер, деректерді қабылдау/жіберу істерін атқарумен қатар, басқа тораптардан келіп түскен хабарларды әрі қарай жөнелту кезінде, оларды басқаша адрестеу ісін де жүргізе алады.

Желілік деңгейдегі хаттамаларға IP (Internet Protocol), ICMP (Internet Control Message Protocol) және IGMP (Internet Group Management Protocol) хаттамалары кіреді.

Транспорттық деңгей (transport layer). TCP/IP хаттамалар стегіндегі бұл деңгей екі компьютер арасындағы деректер ағынының тасымалдынуы мен қолданбалы деңгейдің жұмыс істеуін қамтамасыз етеді. TCP/IP хаттамалар тобында екі транспорттық хаттама бар: TCP (Transmission Control Protocol) және UDP (User Datagram Protocol).

TCP хаттамасы екі компьютер арасында деректер ағынын сенімді түрде тасымалдайды. Деректер желілік деңгей мүмкіндігіне қарай пакеттерге бөлінеді, пакеттердің қабылданғаны туралы хабар келетін тайм-аут уақыты орнатылады. Осы деңгейде деректер ағыны сенімді жеткізілгендіктен, қолданбалы деңгейде сенімділік мәселесі қаралмайды.

UDP хаттамасы қарапайым жұмыс атқарады. Бұл хаттамада деректер датаграмма (datagram) деп аталатын пакеттерде жібереді, бірақта олардың қабылданғаны туралы хабарды күтпейді. Дегенмен бұл хаттаманы да пайдаланатын қосымшалар бар. Сондықтан сенімділік мәселесі қолданбалы деңгейге жүктеледі.

Қолданбалы деңгей (application layer). Бұл деңгейде негізінен Интернеттің әр түрлі қызметтері, сервистері жұмыс істейді. Бұл қызметтердің деректерді тасымалдайтын меншікті хаттамалары бар, мысалы, WWW қызметі үшін HTTP хаттамасы, файлдарды тасымалдау үшін FTP хаттамасы, электрондық пошта үшін SMTP хаттамасы, қашықтықтағы терминал үшін Telnet хаттамасы, қашықтағы компьютермен қауіпсіз байланыс үшін SSH хаттамасы және т.б.

Бұл деңгейдегі хаттамалардың көпшілігі компьютердің нақты портына біріктіріліп қойылған, мысалы:

- HTTP хаттамасына, порт 80 немесе 8080;
- FTP хаттамасына, порт 20 (деректерді жіберу үшін) және 21 (басқару нұсқаулары үшін);
- SSH хаттамасына, порт 22.

Сонымен бөлімде OSI эталондық үлгісі мен TCP/IP хаттамасын қарастырдық. Кейінгі кезде OSI эталондық үлгісі өте көп сыналып жүр. Қазіргі таңда OSI үлгісі теориялық стандарт ретінде қаралғанымен, іс жүзінде TCP/IP хаттамасы қолданылып жүр. Сондықтан Интернеттің негізгі хаттамасы ретінде TCP/IP хаттамасы аталады.

2 HTML тілі негіздері

2.1 Web-құжат құру технологиясы

Интернеттің WWW (World Wide Web) жүйесінің негізгі қызметі қажетті ақпаратты іздеу, жинау және экранға шығаруды ұйымдастыру. Оның экранда көрсететін мәліметтері Web-беттер түрінде дайындалып сақталған, электрондық құжаттар. Электрондық құжаттың кәдімгі құжаттан айырмашылығы, оның жазылу форматында. WWW қызметіндегі барлық гипермәтінді құжаттардың жалпы бір қасиеті, олардың барлығы HTML тілінде жазылған. HTML (HyperText Markup Language) –гипермәтінді белгілеу тілі мәтіндерді Web-құжат (HTML-құжат) түрінде бейнелеуге арналған нұсқаулар жиынын анықтайды. Сонымен қатар Web-құжатқа суреттерді, кестелерді түрлі түспен бейнелеу мүмкіндігін береді. Web-құжаттар Интернеттің WWW қызметімен бірге дами отырып, оның соңғы технологияларымен үнемі толықтырылып отырылады. Төменде тілдің Интернет консорциум ұсынған HTML 4.0 версиясын қарастырамыз [1-4].

HTML тілінде жазылған гипермәтіндік құжаттар браузер терезесінде ықшам түрде көрсетілгенімен, HTML тілі мәтіндерді пішімдеп көрсету тіліне жатпайды. Ол гипермәтіндердің атқаратын қызметін анықтап, оларды тұтынушыға бейнелеп жеткізетін тіл болып есептеледі. Web-құжаттар тұтынушы мониторының мүмкіндіктеріне қарап, браузермен әртүрлі бейнеленуі мүмкін. Сонымен, гипермәтіндерді осылай байланыстыра отырып белгілейтін мүмкіндікті беретін HTML тілі. Оның дұрыс нәтиже алуды қамтамасыз ететін өз заңдылықтары мен ережелері бар.

Web-құжат файлдары .html немесе .htm кеңеймесі бар мәтіндік файлдар. Оларды құруда қарапайым мәтіндік редакторлардан пайдалануға болады. Мысалы: MS Windows ортасында Notepad (Блокнот) редакторы.

Бірақта Web-құжаттарды арнайы HTML-редакторларда құру ыңғайлы. HTML-редакторлар екі топқа бөлінеді:

- ASCII-редакторлар (ASCII-editors, Source editors);
- WYSIWYG-редакторлар (WYSIWYG-editors).

ASCII-редакторлар қарапайым мәтіндік редакторлар. Тек олар HTML-құжаттарды құруға арналған қосымша мүмкіндіктерге ие. Оларға Webber, Macromedia HomeSite, DarekWare HTML Author, WebEdit редакторлары жатады.

WYSIWYG-редакторлар HTML-құжаттарды дайындалу барысында терезеде көру, құжатқа алдын-ала дайындалған шаблондарды қосу мүмкіндігін береді. Редакторлар қосымша терезе – браузерлермен жабдықталған. Оларға Microsoft FrontPage, Netscape Navigator Gold редакторлары жатады.

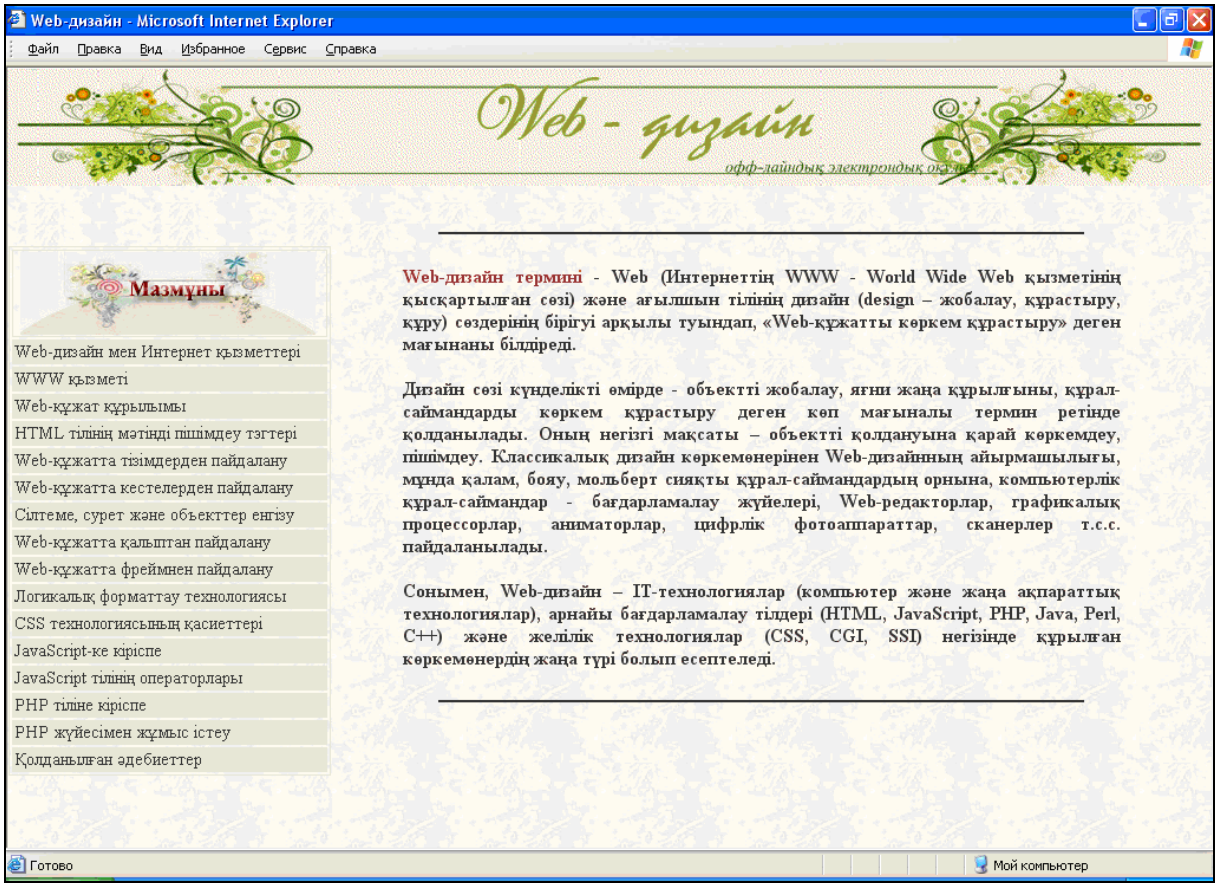
Web–браузерлер HTML тілінде Internet желісіне арнап жазылған Web-сайттарды парақтап көру, оқу мүмкіндігін береді. Web–браузерлер ішінде

ең көп таралғандары Internet Explorer (Microsoft), Opera (Opera Software ASA) және Mozilla FireFox (Mozilla Corporation) (3-5 суреттер).

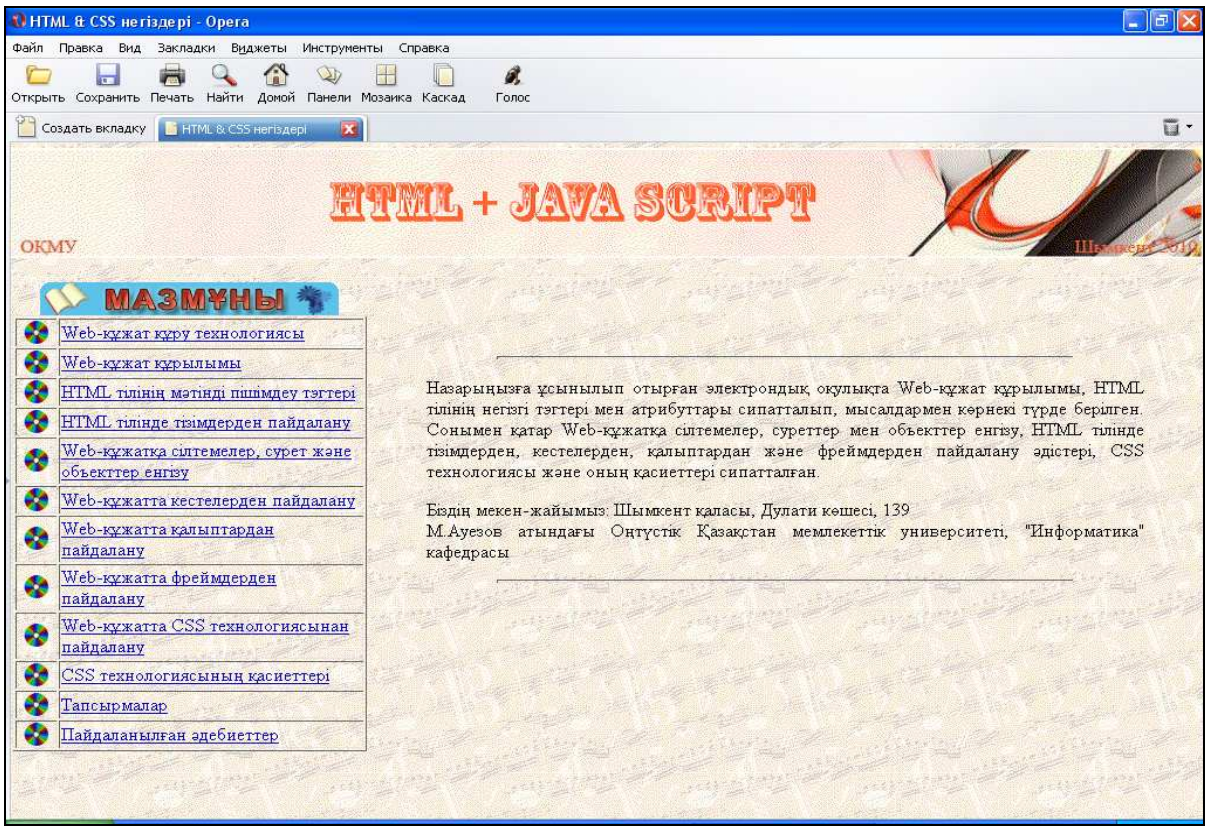
Браузерлердің негізгі атқаратын қызметтері:

Браузердің негізгі функционалдық қызметтері төменде көрсетілген [4]:

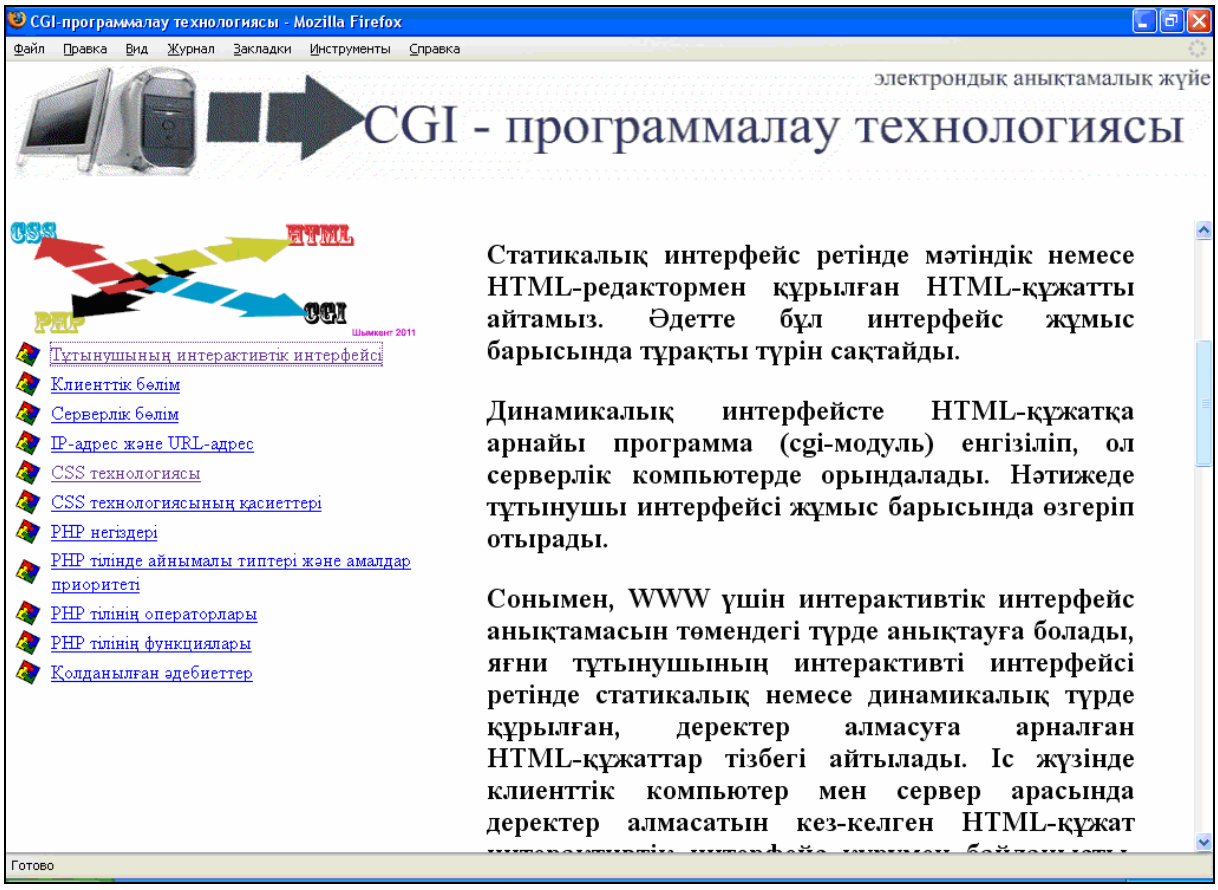
- инфрақұрылым арналары арқылы Web-сайт орналасқан WWW-сервермен HTTP хаттамасы арқылы клиент-сервер типіндегі байланыс орнату;
- Көрсетілген URL-адрестері бойынша Web-сайтты барлық Web-парақтарымен жедел жадыға жүктеу;
- Web-сайттың HTML-кодындағы тэгтерді атрибуттарымен қоса нұсқау түрінде орындау;
- Web-сайтты браузер баптауына сәйкес монитор экранында бейнелеу;
- Web-сайтты URL-адрестері бойынша орындалуын бақылау және жүктелген Web-парақтарын қайта жүктеуді жеңілдету;
- жүйенің өзге сервистік қызмет хаттамаларымен үйлесімді жұмыс атқаруды ұйымдастыру.



Сурет 3 - Microsoft Internet Explorer браузері



Сурет 4 - Opera браузері



Сурет 5 - Mozilla FireFox браузері

HTML тілінің мәтіндерді белгілеу нұсқаулары тэг (tag) деп аталады. Тэг символдар тізбегінен тұрады. Барлық тэг < (кіші) символынан басталып, > (үлкен) символымен аяқталады. Осы символдар ішіне *қызметші сөз* – тэг орналасады. Әрбір тэг өзіне тән бір арнаулы қызмет атқарады. Оларды жазуда бас әріпті де, кіші әріптерді де қатар қолдануға болады, әріптер бірдей қабылданады. Браузер мәтінді монитор терезеында көрсеткенде тэгтер бейнеленбейді, тек олардың құжат мәтініне тигізетін әсері ғана білініп тұрады.

HTML тілінің әрбір тэгі әдетте құжаттың белгілі бір бөлігіне ғана әсер етеді. Осыған орай екі тэг қатар қолданылады: бірі – тэгті *ашады*, екіншісі – *жабады*. Ашатын тэг белгілі бір нұсқауды бастайды, ал жабатын тэг – оны аяқтайды. Жабу тэгінің алдына / (қиғаш сызық) символы жазылуы тиіс. Кейбір тэгтер мағынасына орай жабылмауы да мүмкін.

Тэгтердің мәтінге тигізетін әсерлерін түрлендіруде олардың *атрибуттарынан* пайдаланады. Әдетте атрибуттар ашылу тэгтерінде кездесіп, қосымша қызметші сөздерден тұрады. Көбінесе атрибуттар " (қос тырнақша) символдары ішіне жазылған мәнімен беріледі. Атрибут мәні оның қызметші сөзінен = (теңдік) символы арқылы бөлініп жазылады.

Web-құжатта *түсініктеме* жазу үшін <!-- comment --> тэгінен пайдаланамыз. Түсініктемеде жазылған мәтін немесе тэг өңделмейді, яғни браузер терезесіне шықпайды. Түсініктемені құжаттың кез келген жеріне жазуға болады.

2.2 Web-құжат құрылымы

Web-құжатты құрудан алдын, оның қай тілде жазылғанын браузер үшін анықтау қажет. Ол үшін құжаттың басына

<!doctype html public"-//W3C//Dtd HTML 4.0//EN">

тэгін орналастырамыз. Бұл қатар құжаттың HTML 4.0 тілінде жазылғанын білдіреді.

Web -құжат құрылымы төмендегі негізгі тэгтерден тұрады:

```
<html>
<head>
<title> тақырып </title>
</head>
<body>
        негізгі бөлім
</body>
</html>
```

Web-құжат <html> тэгімен басталып, сәйкес </html> тэгімен аяқталады. Осы екеуінің арасына құжаттың *тақырыбы* және *негізгі бөлімі* орналасады.

Құжаттың тақырыптық бөлімінде төмендегі тэгтер кездеседі:

- <head>
- <title>
- <base>
- <meta>
- <style>
- <link>

Құжаттың тақырыптық бөлімі <head> және </head> тэгтерінің ортасына орналасады ды, жалпы құжат жайлы мәлімет береді. Бұл бөлім браузер терезесінде бейнеленбейді. Әдетте бұл бөлімде <title> және </title> тэгтерімен шектелетін құжаттың ресми тақырыбы жазылады. Браузерлер бұл атауды браузердің терезе тақырыбында жазылатын файл аты есебінде пайдаланады.

<base> – тэгі Web-құжаттың ішкі бумалары мен файлдары үшін негізгі URL-адресін жазуға арналған. <base> тэгінің жабылу тэгі жоқ. Тэг төмендегі атрибуттарға ие:

- href – атрибуты құжаттың негізгі URL-адресін анықтайды. URL-адрес толық көрсетілуі тиіс;
- target – атрибуты сілтемеде көрсетілген құжаттың бейнелену фрейм терезесін анықтайды. Егер атрибут жазылмаса негізгі терезе түсініледі.

Мысалы: <base href ="http://domen.ru" target=="frame">

<meta> – тэгі Web-құжат жайындағы ақпарат беруге арналған. Бұл ақпаратты іздеу серверлері құжатты индекстеуде пайдаланады. <meta> тэгінің жабылу тэгі жоқ. Тэг төмендегі атрибуттарға ие:

- http-equiv="refresh" – атрибуты браузерге content атрибутында берілген уақыт аралығында көрсетілген құжатқа өту керектігін көрсетеді; content=”number” – берілген уақыт; url="url-адрес" – http-equiv атрибуты үшін адресі;
- name="description/keywords" – атрибуты браузерге content атрибутында құжатты сипаттау немесе түйінді сөздер тізімі берілгенін анықтайды; content="text/мәндер тізімі" – атрибут мәні description болғанда құжаттың сипатталуы, ал атрибут мәні keywords болғанда түйінді сөздер тізімі беріледі.

Мысалы: <meta http-equiv="Content-Type" content="text/html">.

<style> тэгі құжатқа стильдер кестесін (CSS – Cascade Style Sheet) енгізуге арналған. Стильдер кестесінде абзац, мәтін, тақырып сияқты түсініктердің жалпы пішімдеу стильдері анықталады. Тэг </style> тэгімен жабылады. Тэг төмендегі атрибуттарға ие:

- type – міндетті атрибут. Ол стильдер кестесінің MIME-типін анықтайды. Әдетте атрибуттың мәні ретінде "text/css" жазылады;
- title – атрибуты пайдаланылатын стильдер кестесінің атын анықтайды.

Мысалы: <style type="text/css" title="style1"> стильдер </style>

<link> тегі құжатты желідегі басқа Web-құжаттармен байланыстырады. Тэгтің жабылу тэгі жоқ. Тэг төмендегі атрибуттарға ие:

- href – атрибуты байланысу файлының url-адресін анықтайды;
- type – атрибуты байланысу файлының MIME-типін анықтайды;
- rel – атрибуты url-адрестегі файлмен байланысу типін көрсетеді. rel атрибуты төмендегі мәндердің бірін қабылдайды:
  - stylesheet – стильдер кестесімен байланысуды көрсетеді. Браузер href атрибутында көрсетілген стильдер кестесін жүктеп, құжатта пайдаланады;
  - home – сайттың негізгі Web-құжатын анықтайды;
  - toc, contents – құжаттың негізгі мазмұны сақталатын файлды анықтайды;
  - index – индекстік іздеу серверлеріне кілттік сөздер тізімін сақтаушы файлды көрсетеді;
  - glossary – құжаттқа тиісті терминдер тізімі сақталатын файлды анықтайды;
  - copyright – сайттың құжат авторлары жайында деректерді сақтаушы файлын көрсетеді;
  - up, parent – иерархиялық ағашта өзінен жоғары тұрған құжатты анықтайды;
  - child – иерархиялық ағашта өзінен төмен тұрған құжатты анықтайды;
  - next – құжаттар тізбегіндегі келесі құжатты көрсетеді;
  - previous – құжаттар тізбегіндегі өзінен алдыңғы құжатты көрсетеді;
  - last, end – құжаттар тізбегіндегі соңғы құжатты көрсетеді;
  - first – құжаттар тізбегіндегі бастапқы құжатты көрсетеді;
  - help – түсініктеме сақталатын файлды анықтайды.

Мысалы: <link rel="home" href="index.html">

<link rel=="stylesheen" type=="text/css" href=="files/us-ibr.css">

Құжаттың негізгі бөлімі <body> және </body> тэгтерінің ортасында жазылады. Негізгі бөлімде құжат мәтіні, кестелер, суреттер, сілтемелер сәйкес тэгтермен өрнектеліп жазылады. Сонымен қарапайым құжат көрінісі мынадай түрде болады:

```
<html>
<head>
<title> Web-бағдарламалау </title>
</head>
<body>
        HTML негіздері
</body>
</html>
```

Жоғарыда көрсетілген <html>, <head>, <body> тэгтері HTML тілінде бекітілгенімен, оларды жазбаса да болады, бірақта алдын-ала тұтынушының қандай браузер пайдаланатыны белгісіз. Сондықтан Web-құжат құрылымы олардың толық болуын талап етеді. Әйтпесе кейбір браузерлер Web-құжатты терезеде дұрыс бейнелемеуі мүмкін.

Негізгі бөлімде қолданылатын <body> тэгінің атрибуттарын қарастырайық:

- text="түс" – мәтін түсін анықтау;
- bgcolor="түс" – фон түсін анықтау;
- link="түс" – қаралмаған сілтеме түсі;
- vlink="түс" – қаралған сілтеме түсі;
- alink="түс" – тұтынушы таңдаған уақыттағы сілтеме түсі;
- background="URL-адрес" – фондық сурет адресі.

Мысалы: <body background="files/graf01.gif" text="#0000FF">. Мұнда фондық обой ретінде graf01.gif файлы алынған.

Түсті анықтауда түстердің RGB (red, green, blue) моделінен пайдаланып, түстің ағылшынша атын немесе он алтылық санақ жүйесіндегі мәнін береміз. Түс red-қызыл, green-жасыл, blue-көк үш бояудың араласуынан тұрады. Араласу мөлшері он алтылық сан 00-ден басталып FF-санымен аяқталады. Он алтылық сан алдына # символы жазылады (сурет 6).

Мысалы: <body bgcolor="#008000">. Терезенің фоны ретінде жасыл түсі алынады.

түсі	аты	сандық мәні	түсі
қара	black	#000000	
күміс	silver	#C0C0C0	
күлгін	gray	#808080	
ақ	white	#FFFFFF	
қоңыр	maroon	#800000	
қызыл	red	#FF0000	
сирень	fuchsia	#FF00FF	
сия көк	purple	#800080	
жасыл	green	#008000	
ашық жасыл	lime	#00FF00	
зайтүн	olive	#808000	
сары	yellow	#FFFF00	
көк	blue	#0000FF	
тоқ көк	navy	#000080	
жасыл-күлгін	teal	#008080	
ауа көк	aqua	#00FFFF	

Сурет 6 - Негізгі түстер кестесі

2.3 HTML тілінің мәтінді пішімдеу тэгтері

Егер бірнеше қатарлы мәтінді құжаттың ішінде теріп, оны браузермен қарасақ, мәтін бір түрлі пішімді шрифтте жазылып, қатарды тасымалдау амалдары орындалмайды. Мәтін браузер терезесіне сиғанша жазылып, кез-келген жерінен келесі қатарға тасымалданады, яғни мәтін пішімдеуді талап етеді.

Мәтіндік бөлікті пішімдеу тэгтерін қарастырайық. Оларға төмендегі тегтер жатады:

- <hn> (headers)
- <p> (paragraph)
- (division)
- <address>
- <blokequote>
-
 (break)
- <hr> (horisontal rule)
- <pre> (preformatted text)
-

<hn> тэгі мәтін тақырыптарын деңгейге бөледі. Барлығы болып шрифт өлшемімен бөлектенетін алты деңгей бар, яғни $n = 1..6$. Бірінші деңгей ең үлкен өлшемді шрифттермен беріледі. Олардың көмегінде тақырыптар мағынасы бойынша реттеледі. Тақырып </hn> тэгімен жабылады. Тэгтің align атрибуты тақырыпты left - сол жақ шетке, right - оң жақ шетке және center -ортаға туралайды.

Мысалы:

```
<h1 align="center"> тақырып 1</h1>
<h2 align="center"> тақырып 2</h2>
<h3 align="center"> тақырып 3</h3>
<h4 align="center"> тақырып 4</h4>
<h5 align="center"> тақырып 5</h5>
<h6 align="center"> тақырып 6</h6>
```

деп жазсақ, браузер терезесінде төмендегі сөздер пайда болады:

тақырып 1

тақырып 2

тақырып 3

тақырып 4

тақырып 5

тақырып 6

 тэгі жаңа абзацты ұйымдастырады. Мәтінді келесі қатарға көшіріп, абзацты алдыңғы мәтін қатарларынан бөліп тұрады. Абзацты тэгі жабады. Тэгтің align атрибуты абзацты left-сол жақ шетке, right-оң жақ шетке, center-ортаға және justify-екі жақ шетке туралайды.

Мысалы: мәтін .

 тэгі мәтінді логикалық түрде бөлектеу мүмкіндігін береді. Браузер бұл бөлікті логикалық объект деп түсінеді. Ол объектпен динамикалық іс-әрекеттер орындай алады. Мәтінді тэгі жабады. Тэгтің align атрибуты мәтінді left-сол жақ шетке, right-оң жақ шетке, center-ортаға және justify-екі жақ шетке туралайды.

Мысалы: мәтін .

<address> тэгі мәтінді пошталық адрес ретінде сипаттайды. Көп браузерлер мәтінді курсив шрифтпен бейнелейді. Мәтінді </address> тэгі жабады.

Мысалы: <address> Шымкент, 21-мкрн, 14-12 </address>

<blokequote> тэгі мәтінді цитата ретінде сипаттайды. Браузер терезеде цитатаны абзац сияқты бейнелейді. Цитатаны </blokequote> тэгі жабады.

Мысалы: Абайға пародия
 <blokequote>
Қараша, желтоқсан мен сол бір-екі ай
Қыстың басы бірі ерте, біреуі жай
Осы екі айда бір қызбен таныстырып
Ғашық қылып, ажыратқан, құдайым-ай!
</blokequote>

 тэгі қатарды тасымалдау амалын орындайды. Бұл тэг жабылу тэгісіз жазылады. Мысалы: мәтін 1
 мәтін 2
 мәтін 3 .

<hr> тэгі мәтінге горизонталь сызық енгізеді. Тэг жабылу тэгісіз жазылады. Тэг төмендегі атрибуттарға ие:

- align – атрибуты сызықты left-сол жақ шетке, right-оң жақ шетке және center-ортаға туралайды;
- size – атрибуты сызықтың қалыңдығын пиксельде анықтайды;
- width – атрибуты сызық ұзындығын пиксельде немесе терезе енінің пайызында анықтайды. Егер көрсетілмесе 100%, яғни терезенің толық ені алынады;
- noshade – атрибуты сызықты көлеңкесіз шығарады;
- color – сызық түсін түстердің RGB моделінде анықтайды.

Мысалы: <hr align="left" width="80%" size="2" noshade>.

<pre> тэгі абзацты сол жазылған пішімделген күйінде, өзгеріссіз браузер терезесінде бейнелейді. Мәтін шрифті моношрифт типінде, барлық әріптердің ені бірдей өлшемді болады. Абзацтағы бос орындар саны мен сөздерді тасымалдау өзгеріссіз жазылады. Абзацты </pre> тэгі жабады.

Мысалы: <pre> мәтін </pre>.

 тэгі мәтін фрагментін бөлектеп, браузер терезесінде ортаға жазуға арналған. Мәтінді тэгі жабады.

Мысалы: мәтін .

Мәтінді пішімдеу тэгтерін қарастырайық. Оларға төмендегі тегтер жатады:

- <basefont>
-
- <i> (italic)
- (emphasis)
- <b> (bold)
- (stronger emphasis)
- <u> (underline)
- <s> (strike)

- <big>
- <small>
- <sup> (superscript)
- <sub> (superscript)
- <code>, <samp> (sample)
- <tt> (tele type)
- <kbd> (keyboard)
- <var> (variable)
- <cite> (citation)

<basefont> тэгі құжат мәтінінің негізгі шрифтін анықтайды. Бірақта мәтінннің кез-келген жерінде шрифтті тэгімен өзгертуге болады. Тэгтің жабылу тэгі жоқ. Тэг төмендегі атрибуттарға ие:

- size – міндетті атрибут. Негізгі шрифттің өлшемін анықтайды. Қабылдайтын мәндері:
  - абсолют мән: 1-ден 7-ге дейінгі бүтін сандар. Шрифт өлшемі браузерге байланысты түрде атрибуттың мәніне сәйкес үлкейеді;
  - салыстырмалы мән: -7-ден +7-ге дейінгі бүтін сандар. Шрифт өлшемі браузердің негізгі шрифт өлшеміне салыстырмалы түрде өзгереді. Егер + болса үлкейеді, ал – болса кішірейеді;
- face – атрибуты негізгі шрифттің типін анықтайды.

Мысалы: <basefont size=2 face="Times New Roman"> .

 тэгі мәтін шрифтінің типін, өлшемін және түсін өзгертеді. Тэг тэгімен жабылады. Тэг жабылғасын шрифт <basefont> тэгіндегі мәндерді қайтадан алады. Тэг төмендегі атрибуттарға ие:

- size – атрибуты шрифттің өлшемін анықтайды. Мәндері:
  - абсолют мән: 1-ден 7-ге дейінгі бүтін сандар;
  - салыстырмалы мән: -7-ден +7-ге дейінгі бүтін сандар;
- face – атрибуты шрифттің типін анықтайды;
- color – атрибуты шрифттің түсін анықтайды.

Мысалы: мәтін .

Мәтін шрифті Arial типінде қызыл түспен жазылады.

Мысалы: мәтін .

Мәтін шрифті Verdana типінде көк түспен жазылады.

 тэгі мәтінді курсив шрифтпен бейнелейді. Тэг тэгімен жабылады. Мысалы: мәтін .

 логикалық екпін тэгі. Мәтінді логикалық түрде бөлектеу мүмкіндігін береді. Әдетте мәтін браузер терезесінде курсив шрифтпен жазылады. Мағынасы жағынан тэгіне жақын. Тэг тэгімен жабылады.

Мысалы: мәтін .

 тэгі мәтінді жуан шрифтпен бейнелейді. Тэг тэгімен жабылады. Мысалы: мәтін .

 күшейтірілген бөлектеу тэгі. Мәтінді логикалық түрде бөлектеу мүмкіндігін береді. Әдетте мәтін браузер терезесінде жуан шрифтпен бейнеленеді. Мағынасы жағынан тэгіне жақын. Тэг тэгімен жабылады.

Мысалы: мәтін .

<u> тэгі мәтіннің астын сызады. Тэг </u> тэгімен жабылады. Мысалы: <u> мәтін </u>.

<s> тэгі мәтінді жазып, үстінен сызады. Тэг </s> тэгімен жабылады. Мысалы: <s> мәтін </s>.

<big> тэгі мәтіннің өлшемін шамалы үлкейтіп бейнелейді. Тэг </big> тэгімен жабылады.

Мысалы: <big> мәтін </big>.

<small> тэгі мәтіннің өлшемін шамалы кішірейтіп бейнелейді. Тэг </small> тэгімен жабылады.

Мысалы: <small> мәтін </small>.

^{тэгі мәтіннің өлшемін шамалы кішірейтіп, дәреже сияқты жоғарыға бейнелейді. Тэг} тэгімен жабылады.

Мысалы: ^{мәтін}.

_{тэгі мәтіннің өлшемін шамалы кішірейтіп, индекс сияқты төменге бейнелейді. Тэг} тэгімен жабылады.

Мысалы: _{мәтін}.

<code> тэгі формула немесе программа кодын жазуда қолданылады. Тэг </code> тэгімен жабылады.

Мысалы: <code> y=a*x^2+b*x+c </code>.

<tt> тэгі мәтінді моношрифт типінде, барлық әріптердің енін бірдей етіп бейнелейді. Тэг </tt> тэгімен жабылады.

Мысалы: <tt> мәтін </tt>.

<kbd> тэгі тұтынушы пернетақтадан енгізуі қажет болған мәтінді жазуда қолданылады. Мәтін браузер терезесінде моношрифт типінде бейнеленеді. Тэг </kbd> тэгімен жабылады.

Мысалы: <kbd> мәтін </kbd>.

<var> тэгінен мәтінде айнымалыны белгілеуде пайдаланады. Мәтін браузер терезесінде курсив түрінде бейнеленеді. Тэг </var> тэгімен жабылады.

Мысалы: <var> мәтін </var>.

<cite> тэгі қысқа мәтінді цитата ретінде сипаттайды. Цитата браузер терезесінде курсив түрінде бейнеленеді. Тэг </cite> тэгімен жабылады.

Мысалы: <cite> мәтін </cite>.

2.4 HTML тілінде тізімдерден пайдалану

Web-құжатта ақпаратты ықшам түрде бейнелеуде тізімдерден пайдаланады. Тізімдер екі түрге бөлінеді:

- нөмірленген;
- маркерленген.

Тізімдерді құру тэгтерін қарастырайық. Оларға төмендегі тегтер жатады:

- unsorted list
- ordered list
- list item
- <menu> <dir>
- <dl> definition list
- <dt> definition term
- <dd> definition description

 тэгі маркерленген тізім құрады. Тэг ішінде бірнеше <li> тэгтері қолданылуы тиісті. Тэг тэгімен жабылады.

Мысалы:

<p> «Гамлет» трагедиясын кім жазған:
  Шекспир
 Монтень
 Гомер

```
<li> Рабле </li>
<li> Софокл </li> </ul> </p>
```

Браузер терезесінде төмендегі маркерленген тізім көрінеді:

«Гамлет» трагедиясын кім жазған:

- Шекспир
- Монтень
- Гомер
- Рабле
- Софокл

 тэгі нөмірленген тізім құрады. Тэг ішінде бірнеше <li> тэгтері қолданылуы тиісті. Тэг тэгімен жабылады. Тэг төмендегі атрибуттарға ие:

- start – атрибуты нөмір қай саннан басталатынын анықтайды. Мән ретінде тек қана бүтін санды алады;
- type – атрибуты нөмірлеу силін анықтайды. Қабылдайтын мәндері:
  - "A" – латын бас әріптері: A, B, C, …
  - "a" – кіші латын әріптері: a, b, c, …
  - "I" – үлкен рим цифрлары: I, II, III, …
  - "i" – кіші рим цифрлары: i, ii, iii, …
  - "1" – араб цифрлары: 1, 2, 3, …

Егер type атрибуты берілмесе, онда нөмір араб цифрларымен жазылады.

Мысалы:

```
<p> «Гамлет» трагедиясын кім жазған:
<ol type="a"> <li> Шекспир </li>
<li> Монтень </li>
<li> Гомер </li>
<li> Рабле </li>
<li> Софокл </li> </ol> </p>
```

Браузер терезесінде төмендегі нөмірленген тізім көрінеді:

a. Шекспир
b. Монтень
c. Гомер
d. Рабле
e. Софокл

 тэгі тізім қатарын құрады. Тэг <ul> және <ol> тэгтері ішінде қолданылады. Тэгтің value атрибуты тек қана <ol> тэгінде қолданылады. Атрибут қатар нөмірінің басталу ретін көрсетеді. Тэг тэгімен жабылады.

Мысалы:

 «Гамлет» трагедиясын кім жазған:
<ol type="a"> <li value="3"> Шекспир
 Монтень
 Гомер
 Рабле
 Софокл

Браузер терезесінде төмендегі нөмірленген тізім көрінеді:

c. Шекспир
d. Монтень
e. Гомер
f. Рабле
g. Софокл

<menu> және <dir> тэгтері HTML 4.0 тілі тэгтеріне жатпайды. Олар нөмірленген тізім құруда <ol> тэгінің орнына пайдаланылуы мүмкін.

<dl> тэгі анықтамалар тізімін құрады. Анықтамалар термин (<dt> тэгі) және оның сипаттамасынан (<dd> тэгі) тұрады. Тэгтің compact атрибуты тізімді ықшам түрде бейнелейді. Атрибуттың мәні жоқ. Тэг </dl> тэгімен жабылады.

Мысалы:

<dl compact>
<dt> HTML <dd> гипермәтінді белгілеу тілі
<dt> WWW <dd> бүкіләлемдік өрмек
</dl>

<dt> тэгі анықтамалар тізімінің терминін анықтайды. Ол <dl> тэгінің ішінде қолданылады. Тэг жабылу тэгісіз жазылады.

<dd> тэгі анықтамалар тізімінің сипаттамасын анықтайды. Ол <dl> тэгінің ішінде қолданылады. Тэг жабылу тегісіз жазылады.

2.5 Web-құжатқа сілтемелер, сурет және объекттер енгізу

HTML тілі Web-құжатқа суреттерді, кестелерді енгізу мүмкіндігін береді. Сонымен қатар желідегі басқа Web-құжаттармен байланысуда гиперсілтемелерден пайдаланады. Web-құжатқа сілтеме, сурет, объект, апплет және скрипттер енгізуді қарастырайық. Оларға төмендегі тэгтер жатады:

- <a> (anchor)
- <img> (image)
- <embed>
- <noembed>

- <applet>
- <param>
- <object>
- <marquee>

<a> тэгі гиперсілтемелерді құруда және пайдалануда қолданылады. Тэг тэгімен жабылады. Тэг төмендегі атрибуттарға ие:

- name – атрибуты бастапқы және жабылу тэгтері арасындағы мәтінді сілтеме жасау объекті деп анықтайды. Мән ретінде қайталанбайтын кез-келген сөзді латын әріптерімен жазуға болады.
- href – атрибуты бастапқы және жабылу тэгтері арасындағы мәтінді немесе суретті гиперсілтеме деп анықтайды. Қабылдайтын мәндері:
  - "http://..." – Web-құжатқа сілтеме жасайды;
  - "ftp://..." – ftp-сайтқа немесе құжатқа сілтеме жасайды;
  - "mailto:..." – пошталық клиент-программаны жүктейді. Ашылған өріске электрондық адрес пен жіберілетін мәтін жазылады;
  - "news:..." – жаңалықтар серверінің конференциясына сілтеме жасайды;
  - "telnet://…" – қашықтағы компьютерге telnet-программасымен байланысқа сілтеме жасайды;
  - "wais://…" – wais-серверге сілтеме жасайды;
  - "gopher://..." – gopher-серверге сілтеме жасайды.
- title – атрибуты ағымдағы түсініктемені анықтайды.
- target – атрибуты сілтемеде көрсетілген құжаттың бейнелену фрейм терезесін анықтайды. Қабылдайтын мәндері:
  - "_self" – href атрибутында анықталған құжат ағымдағы фреймде бейнеленеді;
  - "_parent" – href атрибутында анықталған құжаттың ағымдағы фреймнің аналық фреймінде бейнелеуі қажет екенін көрсетеді. Басқаша айтқанда ағымдағы фреймді қосушы frameset тегі сипатталған терезені көретеді;
  - "_top" – href атрибутында анықталған құжаттың бастапқы фреймде бейнеленуі қажет екенін көрсетеді;
  - "_blank" – href атрибутында анықталған құжаттың жаңа терезеде бейнеленуі қажет екенін көрсетеді.

Егер байланыс түрі мен URL-адрес көрсетілмесе бастапқы адрес ретінде ағымдағы бума алынады. Бұл салыстырмалы сілтеме жасауға мүмкіндік береді:

Мысалы: Информатика .

Бұл жерде ағымдағы буманың files ішкі бумасындағы text10.html файлына сілтеме жасалған.

Мысалы: соңына
тэгін құжат басына, ал
 соңы
тэгін құжат соңына жазайық. Браузер терезесінде құжат басында асты сызылған соңы сөзі басқа түспен шығып тұрады және ол құжаттың соңына сілтеме жасайды.

Мысалы: құжаттың кез-келген жеріне
 информатика
тэгі жазылсын, онда
 информатика
тэгі сол «информатика» сөзі жазылған бөлімге сілтеме жасайды. Браузер терезесіндегі басқа түспен бөлектенген информатика сөзін тышқан көрсеткішімен басу арқалы, сол бөлімге келеміз.

Мысалы: Информатика

Бұл жерде ағымдағы буманың files ішкі бумасындағы text10.html файлына сілтеме жасалған. Аталған файл жаңа терезеде бейнеленеді.

<img> тэгі Web-құжатқа суретті енгізуде қолданылады. Сурет jpg, gif немесе png форматтарының бірінде болуы тиіс. Тэг жабылу тэгісіз жазылады. Тэг төмендегі атрибуттарға ие:

- src – міндетті атрибут. Енгізілетін суреттің URL-адресін анықтайды;
- height – атрибуты суреттің биіктігін пиксельде анықтайды;
- width – атрибуты суреттің енін пиксельде анықтайды;
- hspace – атрибуты суреттің құжаттағы басқа объекттен горизонталь бойынша қашықтығын пиксельде анықтайды;
- vspace – атрибуты суреттің құжаттағы басқа объекттен вертикаль бойынша қашықтығын пиксельде анықтайды;
- align – міндетті атрибут. Құжатта суретті браузер терезесіне туралау түрін көрсетеді. Қабылдайтын мәндері:
- left – суретті сол жақ шетке туралайды. Ағымдағы мәтін суретті оң жағынан қоршайды;
- right – суретті оң жақ шетке туралайды. Ағымдағы мәтін суретті сол жағынан қоршайды;

- top – суреттің жоғарғы шетін ағымдағы мәтіннің негізгі сызығына туралайды;
- bottom – суреттің төменгі шетін ағымдағы мәтіннің негізгі сызығына туралайды;
- middle – суреттің ортасын ағымдағы мәтіннің негізгі сызығына туралайды;
- absmiddle – суреттің ортасын ағымдағы мәтіннің ортасына туралайды;
- absbottom – суреттің төменгі шетін ағымдағы мәтіннің төменгі сызығына туралайды;
- name – атрибуты суретті сілтеме жасау объекті деп анықтайды. Мән ретінде қайталанбайтын кез-келген сөзді латын әріптерімен жазуға болады;
- alt – атрибуты қосалқы мәтінді анықтайды. Егер браузер сурет файлын таба алмаса, онда терезеде суреттің орнында қосалқы мәтін пайда болады;
- border – атрибуты сурет рамкасының қалындығын пиксельде анықтайды;
- usemap – атрибуты суретті <map> тэгінде анықталған навигациялық картамен байланыстырады. Атрибуттың мәні <map> тэгінің name атрибуты мәнімен сәйкес болуы тиіс.
- ismap – атрибуты суретті сервер өңдейтін навигациялық картамен байланыстырады. Егер тұтынушы суретті тышқанмен шертсе, суреттің сол жерінің x, y координаталарын пиксель түрінде құжат орналасқан серверге жіберіледі. Сервер жіберілген координаталарды өңдеп, қажетті құжатты немесе суретті көрсетеді. Атрибуттың мәні жазылмайды;
- title – атрибуты ағымдағы түсініктемені анықтайды.

Мысалы: <img border="1" src="files/graf10.gif" width="700" height="79" align="left"> – браузер терезесінің сол жақ шетіне files бумасындағы graf10.gif суреті шығады.

Мысалы: <img border="1" src="files/graf10.gif" width="700" height="79" align=left hspace="10"> – браузер терезесінде мәтін files бумасындағы graf10.gif суретінің оң жағынан 10 пиксель ара қашықтық тастап жазылады.

Мысалы: <img src="files/graf10.gif" border="1" width="700" height="79" align=center> – браузер терезесінде graf10.gif суреті files бумасындағы text00.html құжатына сілтеме жасайды.

<embed> тэгі құжатқа басқа объекттерді (html-емес құжаттар) енгізуде қолданылады. Бірақта көптеген браузерлер бұл тэгті қолдамайды. Тэг

</embed> тэгімен жабылады. Тэг <img> тэгі сияқты орындалады. Сондықтан src, height, width, hspace, vspace, align, name, alt, border атрибуттары бірдей. Басқа қосымша атрибуттары:

- pluginspace – атрибуты енгізілген объектті өңдейтін плагин (plug-in) орналасқан URL-адресті анықтайды. Мысалы: mp3 файлдары үшін қажетті плэйерді (player) көрсету қажет;
- palette – атрибуты енгізілген объект үшін түс палитрасы түрін анықтайды. Бұл атрибут тек қана Windows жүйесінде 256 түсті палитрада ғана қолданылады. Қабылдайтын мәндері:
  - background – объектті background-палитрасымен бейнелейді (үндеместен қолданылады);
  - foreground – объектті foreground-палитрасымен бейнелейді;
- hidden – атрибуты енгізілген объектті көрсету қажет пе, жоқ па, соны анықтайды. Қабылдайтын мәндері:
  - true – объектті көрсетпейді;
  - false – объектті көрсетеді (үндеместен қолданылады);
- type – атрибуты браузерге объектті өңдеуші плагинді анықтау мақсатында, енгізілетін объект файлының типін ашық көрсетеді. Атрибуттың мәні ретінде тіркелген MIME-тип файлы жазылады;
- title – атрибуты ағымдағы түсініктемені анықтайды.

<noembed> тэгі, егер <embed> тэгі жұмыс істемесе пайдаланылатын мәтінді анықтайды. Тэг </noembed> тэгімен жабылады.

Мысалы: <embed src="music.mid" hidden="true" width="700" height="100"> <noembed> Сіздің браузеріңізде бұл тэг орындалмайды; </noembed> </embed>.

<applet> тэгі құжатқа Java-апплет (Java тілінде құрылған программалар) енгізуде қолданылады. Тэг </applet> тэгімен жабылады. Браузерде Java-апплет құжатта енгізілген сурет сияқты бейнеленіп, орындалады. Сондықтан <img> тэгі сияқты src, height, width, hspace, vspace, align, name, alt, border атрибуттарының қолданылуы бірдей. Басқа қосымша атрибуттары:

- code – міндетті атрибут. Сілтеме жасалған Java-апплеттің атын анықтайды;
- codebase – атрибуты Java-апплет файлының негізгі URL-адресін анықтайды. Егер атрибут жазылмаса ағымдағы құжаттың URL-адресін қабылдайды.
- align – міндетті атрибут. Java-апплетті суретті браузер терезесіне туралау түрін көрсетеді;
- title – атрибуты ағымдағы түсініктемені анықтайды.

Мысалы:

<applet code="JumpingGirl.class" width="30" height="40" align="left">
мәтін </applet> – браузер терезесінде JumpingGirl.class файлы орындалады.

<param> тэгі <applet> тэгінің ішінде Java-программаға мән енгізуде қолданылады. Тэг жабылу тэгісіз жазылады. Мысалы:

```
applet code="JumpingGirl.class" width="30" height="40" align="left">
<param name="width" value="10">
Java-апплет
 </applet>
```

<object> тэгі құжатқа басқа объекттерді (html-емес құжаттар) енгізуде қолданылады. HTML 4.0 тілінде бұл тег <embed> және <applet> тэгтерінің орнына енгізілген.

<marquee> тэгін құжатқа жүгіруші мәтін қатарын енгізуде пайдаланады. Тэг </marquee> тэгімен жабылады. Тэг төмендегі атрибуттарға ие:

- behavior – атрибуты мәтіннің браузер терезесінде көріну режимін анықтайды. Қабылдайтын мәндері:
- alternate – мәтін терезе шетіне келіп артқа қайтады;
- scrool – мәтін терезеде қайта-қайта айналып жүгіріп отырады (үндеместен қолданылады);
- slide – мәтін терезенің сол жағына келіп тоқтайды.
- bgcolor – фон түсін анықтайды;
- loop – мәтіннің қайталану санын көрсетеді. Қабылдайтын мәндері:
  - 0 - 9 – қайталану саны;
  - infinite – шексіз (үндеместен қолданылады);
- direction – жүгіруші мәтіннің бағытын көрсетеді. Мәндері:
  - down – жоғарыдан төменге;
  - left – оңнан солға (үндеместен қолданылады);
  - right – солдан оңға;
  - up – төменнен жоғарыға;
- lang – мәтіннің негізгі тілін анықтайды. Бұл ақпарат іздеу серверлерінде, дыбыстық синтезаторларда пайдаланылады. Мысалы: en – ағылшынша; ru – орысша;
- scrolldelay – жүгіруші мәтіннің жылдамдығын миллисекундда анықтайды;
- scrollamount – жүгіруші мәтіннің жылдамдығын пиксельде анықтайды.

Мысалы: <marquee direction="right"> жүгіруші мәтін қатары </marquee>

3 Web-құжатты безендіру

3.1 Web-құжатта кестелерден пайдалану

Web-құжаттармен жұмыс істеу барысында реттелген ақпаратты жиі пайдаланамыз. Осындай реттелген, ықшам ақпаратты сақтаудың ең қолайлы тәсілі кестелер болып табылады. Кестедегі қатар мен баған қиылысқан орынды ұяшық деп атаймыз. Ұяшықтың ішінде мәтінмен қатар, сурет, шағын кесте де болуы мүмкін. HTML тілінің кестелермен жұмыс істейтін тэгтерін қарастырайық. Оларға төмендегі тэгтер жатады:

- <table>
- <caption>
- <tr> (table row)
- <td> (table data)
- <th> (table head)

<table> тэгі Web-құжатқа кесте құру үшін қажет. Тэг </table> тэгімен жабылады. Кестенің ұяшықтары <caption>, <tr>, <td> және <th> тэгтерімен құрылады. Тэг төмендегі атрибуттарға ие:

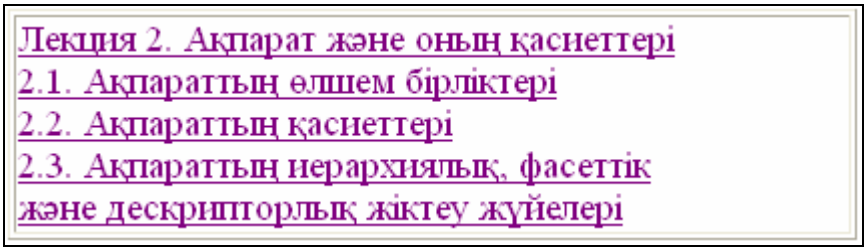
- align – атрибуты кестені браузер терезесінде left-сол жақ шетке, right-оң жақ шетке және center-ортаға туралайды. Үндеместен left мәні қолданылады;
- valign – атрибуты кестені браузер терезесінде тігінен туралайды. Көптеген браузерлер бұл атрибутты бейнелемейді. Мәндері:
  - top – төменнен;
  - bottom – жоғарыдан;
  - middle – ортадан (үндеместен қолданылады);
- border – атрибуты кесте рамкасының қалындығын пиксельде анықтайды. Егер атрибут жазылмаса, онда рамка көрінбейді;
- cellpadding – атрибуты кестенің әр бір ұяшығындағы мәтіннің ұяшық рамкасына дейінгі қашықтықты пиксельде анықтайды;
- cellspacing – атрибуты кестенің жалпы рамкасы мен әр бір ұяшығының рамкасы арасындағы қашықтықты пиксельде анықтайды;
- width – атрибуты кестенің енін анықтайды. Кестенің ені пиксельде немесе браузер терезесіне сәйкес пайызда беріледі. Егер атрибут жазылмаса, кестенің ені енгізілген мәтін ұзындығы бойынша анықталады;
- height – атрибуты кестенің биіктігін анықтайды. Кестенің биіктігі пиксельде немесе браузер терезесіне сәйкес пайызда беріледі. Егер атрибут жазылмаса, кестенің биіктігі енгізілген мәтін бойынша анықталады;

- bgcolor – атрибуты кестенің фон түсін анықтайды. Түс RGB (red, green, blue) моделімен беріледі;
- background – атрибуты кесте фоны ретінде суретті алады. Атрибуттың мәні ретінде суреттің URL-адресі беріледі;
- title – атрибуты ағымдағы түсініктемені анықтайды.

Мысалы:

```
<table width="50%" border="1">
<tr> <td><p>
<a href=files/text10.html>Лекция 2. Ақпарат және оның
қасиеттері</a></br>
<a href=files/text11.html>2.1. Ақпараттың өлшем бірліктері</a></br>
<a href=files/text12.html>2.2. Ақпараттың қасиеттері</a></br>
<a href=files/text13.html>2.3. Ақпараттың иерархиялық, фасеттік
</br>
және дескрипторлық жіктеу жүйелері</a></p></td></tr>
</table>
```

Браузер терезесінде «Ақпарат және оның қасиеттері» кестесі пайда болады (сурет 7):



Сурет 7 – «Ақпарат және оның қасиеттері» кестесі

<caption> тэгі кестенің тақырыбын анықтайды. Тақырып тек қана мәтіннен тұрады. Тэг </caption> тэгімен жабылады. Тэгтің align атрибутының top мәні (үндеместен қолданылады) тақырыптың кесте басында, ал bottom мәні соңында болуын анықтайды.

<tr> тэгі кестенің ұяшықтар қатарын анықтайды. Тэг </tr> тэгімен жабылады. Ұяшықтар <td> және <th> тэгтерімен құрылады. Тэг төмендегі атрибуттарға ие:

- align – атрибуты ұяшық мәтінін left –сол жақ шетке, right –оң жақ шетке және center –ортаға туралайды. Үндеместен left мәні қолданылады;
- valign – атрибуты ұяшық мәтінін браузер терезесінде тігінен туралайды. Қабылдайтын мәндері:
  - top – төменнен;
  - bottom – жоғарыдан;
  - middle – ортадан (үндеместен қолданылады);

- bgcolor – атрибуты кесте ұяшығының фон түсін анықтайды.

<td> тэгі кестенің ағымдағы қатарында ұяшық құрады. Тэг </td> тэгімен жабылады. Тэг төмендегі атрибуттарға ие:

- align – атрибуты ұяшық мәтінін браузер терезесіне туралайды. Егер атрибут жазылмаса, онда <tr> тэгінің align атрибутының мәні үндеместен алынады. Қабылдайтын мәндері:
  - left – сол жақ шетке;
  - right – оң жақ шетке;
  - center – ортаға;
- valign – атрибуты ұяшық мәтінін браузер терезесінде тігінен туралайды. Егер атрибут жазылмаса, онда <tr> тэгінің valign атрибутының мәні үндеместен алынады. Қабылдайтын мәндері:
  - top – төменнен;
  - bottom – жоғарыдан;
  - middle – ортадан (үндеместен қолданылады);
- width – атрибуты ұяшықтың енін анықтайды. Ұяшықтың ені пиксельде немесе кесте еніне сәйкес пайызда беріледі;
- height – атрибуты ұяшық биіктігін анықтайды. Ұяшықтың биіктігі пиксельде немесе кесте биіктігіне сәйкес пайызда беріледі;
- colspan – атрибуты кесте ұяшығы қанша бағанды біріктіретінін анықтайды. Егер жазылмаса атрибуттың мәні үндеместен бірге тең;
- rowspan – атрибуты кесте ұяшығы қанша қатарды біріктіретінін анықтайды. Егер жазылмаса атрибуттың мәні үндеместен бірге тең;
- nowrap – атрибуты ағымдағы ұяшықта мәтінді тасымалдауға тиім салады;
- bgcolor – атрибуты кесте ұяшығының фон түсін анықтайды;
- background – атрибуты ұяшықтың фоны ретінде суретті алады. Атрибуттың мәні ретінде суреттің URL-адресі беріледі.

<th> тэгі <td> тэгі сияқты кестенің ағымдағы қатарында ұяшық құрады, тек ұяшық мәтінін тақырып деп анықтайды. Мұндай анықтама браузерге тақырып пен мәтінді бөлектеу мүмкіндігін береді. Тэгтің атрибуттары <td> тэгінің атрибуттарына ұқсас. Тэг </th> тэгімен жабылады.

Кесте құруға мысалдар қарастырайық. Қарапайым кесте құрамыз:

```
<table width="50%" border="1">
<tr>
   <td> 1 </td>
   <td> 2 </td>
   <td> 3 </td>
```

```
</tr>
<tr>
    <td> 4 </td>
    <td> 5 </td>
    <td> 6 </td>
</tr>
</table>
```

Браузер терезесінде келесі қарапайым кесте пайда болады (сурет 8):

1	2	3
4	5	6

Сурет 8 – қарапайым кесте

Кестеге жалпы фон түсін, ал кейбір ұяшықтарға басқа фон түсін беруге болады. Кейбір ұяшықтардың мәтінін пішімдейік.

```
<table width="50%" border="1" >
<tr>
    <td align="center"> 1 </td>
    <td align="center"> 2 </td>
    <td align="center"> 3 </td>
</tr>
<tr>
    <td bgcolor="#ff0000"> 4 </td>
    <td bgcolor="#ffffff"> 5 </td>
    <td bgcolor="#0000ff"> 6 </td>
</tr>
</table>
```

Браузер терезесінде келесі пішімделген кесте пайда болады (сурет 9):

1	2	3
4	5	6

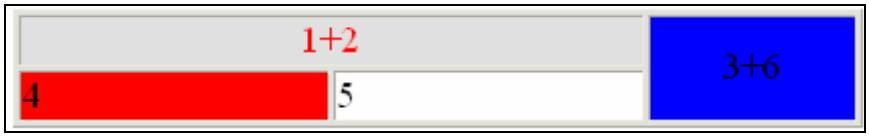
Сурет 9 – пішімделген кесте

Кесте ұяшықтарын бір-біріне қосуға болады. Кейбір ұяшық мәтіндерінің түсін өзгертіп көрейік.

```
<table width="50%" border="1" bgcolor="#e0e0e0">
<tr>
    <td align="center" colspan="2"> <font color="#ff0000"> 1+2
    </font></td>
    <td align="center" rowspan="2" width="25%"
```

```
    bgcolor="#0000ff">3+6
    </td>
</tr>
<tr>
    <td bgcolor="ff0000"> 4 </td>
    <td bgcolor="#ffffff" > 5 </td>
</tr>
</table>
```

Браузер терезесінде келесі ұяшықтары қосылған кесте пайда болады (сурет 10):



Сурет 10 – ұяшықтары қосылған кесте

3.2 Web-құжатта қалыптардан пайдалану

Web-құжатта тұтынушымен ақпарат алмасу қажет болса, онда қалыптан пайдаланады. Әдетте қалыптан тұтынушыны тіркеуде, әр-түрлі мәліметтер жинауда, сұранысты қабылдауда пайдаланады. Қалып тұтынушыға арнайы тек толтыруға рұқсат етілген өрістерге деректер енгізуді ұсынады. Енгізілген деректерді Web-сервердегі арнайы программа (скрипт) өңдейді. HTML 4.0 тілінің қалыптармен жұмыс істейтін тэгтерін қарастырайық. Оларға төмендегі тэгтер жатады:

- <form>
- <textarea> (text area)
- <select>
- <option>
- <input>

<form> тэгі қалып құрушы негізгі тэг. Тэг </form> тэгімен жабылады. Тэгтің ішінде HTML тілінің көптеген тэгтерінен пайдалануға болады. Тэг төмендегі атрибуттарға ие:

- name – атрибуты қалыптың өзіндік атын анықтайды. Атрибуттан құжатта бірнеше қалып болғанда пайдаланады;
- action – міндетті атрибут. Қалыпқа енгізілген деректерді өңдеуші скрипттің URL-адресін анықтайды;
- method – атрибуты қалыптың мазмұнын сервердегі скриптке жіберу әдісін таңдайды. Қабылдайтын мәндері:

        - get – параметрлерді ашық жіберу (үндеместен қолданылады);
        - post – параметрлерді жасырын жіберу;
    - enctype – атрибуты қалыптың мазмұнын сервердегі скриптке жіберудің post әдісіндегі деректердің MIME-типін анықтайды;
    - target – атрибуты қалып мазмұнын өңдеген сервердегі скрипттің нәтижелері бейнеленетін фреймнің атын анықтайды. Ал егер <base> тэгі қоданылса, онда үндеместен қолданылатын фреймнің атын анықтайды. Қабылдайтын мәндері:
        - _self – нәтиже ағымдағы фреймде бейнеленеді;
        - _parent – нәтиже ағымдағы фреймнің аналық фреймінде бейнеленуі қажет екенін көрсетеді;
        - _top – нәтиже бастапқы фреймде бейнеленуі қажет екенін көрсетеді; (үндеместен қолданылады);
        - _blank – нәтиже жаңа фреймде бейнеленуі қажет екенін көрсетеді;
        - нақты көрсетілген фреймнің аты;
    - accept-charset – атрибуты құжатқа енетін нәтиженің код стандартын көрсетеді. Қабылдайтын мәндері:
        - ISO-8859-1 – жалпы стандарт;
        - Windows-1250 – ағылшын тіліндегі мәтіндер үшін;
        - Windows-1251 – орыс тіліндегі мәтіндер үшін;
    - dir – атрибуты мәтіннің бейнелену бағытын көрсетеді. Қабылдайтын мәндері:
        - ltr – солдан оңға;
        - rtl – оңнан солға;
    - lang – мәтіннің негізгі тілін анықтайды. Мысалы: en – ағылшынша; ru – орысша;
    - title – атрибуты ағымдағы түсініктемені анықтайды.

<textarea> тэгі қалыпта мәтіндік аймақты анықтайды. Тэг <form> тэгінің ішінде жазылып, </textarea> тэгімен жабылады. Тэг төмендегі атрибуттарға ие:

- name – міндетті атрибут, өрістің өзіндік атын анықтайды;
- rows – атрибуты өріс терезесінде мәтіннің көрінуге тиісті қатар санын анықтайды;
- cols – атрибуты өріс терезесінде мәтіннің көрінуге тиісті баған санын анықтайды;
- wrap – атрибуты мәтіннің сөздерін тасымалдау әдісін анықтайды. Қабылдайтын мәндері:
    - off – мәтін сөздері тасымалданбайды;
    - virtual – мәтін сөздері тасымалданады, бірақта сервердегі скриптке тасымалданбай жіберіледі;
    - physical – мәтін сөздері тасымалданады;

- title – атрибуты ағымдағы түсініктемені анықтайды;
- lang – мәтіннің негізгі тілін анықтайды;
- dir – атрибуты мәтіннің бейнелену бағытын көрсетеді. Қабылдайтын мәндері:
  - ltr – солдан оңға;
  - rtl – оңнан солға;
- align – атрибуты мәтінді браузер терезесіне туралайды. Қабылдайтын мәндері:
  - left – сол жақ шетке;
  - right – оң жақ шетке;
  - baseline – негізгі сызыққа;
  - top – жоғары шетке;
  - bottom – төменгі шетке;
  - center – ортаға;
  - justify – еніне;

Мысалы:

```
<form action="mailto:us-ibr@rambler.ru" method=post>
<textarea name="comments" wrap="virtual" cols="40" rows="3">
</textarea><br>
<input type="submit" value="ok">
</form>
```

Браузер терезесінде келесі қарапайым қалып пайда болады (сурет 11):



Сурет 11 – қарапайым қалып

Қалыпқа енгізілген мәтінді, браузер элекрондық поштаға жібереді. Көрсетілген мысалда mailto: – скрипттің аты, us-ibr@rambler.ru – электрондық адрес.

<select> тэгі қалыпта мәзірді анықтайды. Тұтынушы мәзір ішінен қажеттісін таңдай алады. Тэг <form> тэгінің ішінде жазылып, </select> тэгімен жабылады. Тэг ішінде бірнеше <option> тэгі жазылуы тиіс. Тэг төмендегі атрибуттарға ие:

- name – атрибуты мәзірдің өзіндік атын анықтайды;
- title – атрибуты ағымдағы түсініктемені анықтайды;
- lang – мәзірдің негізгі тілін анықтайды;

- dir – атрибуты мәтіннің бейнелену бағытын көрсетеді. Қабылдайтын мәндері:
  - ltr – солдан оңға;
  - rtl – оңнан солға;
- align – атрибуты мәзір мәтінін браузер терезесіне туралайды. Қабылдайтын мәндері:
  - left – сол жақ шетке;
  - right – оң жақ шетке;
  - baseline – негізгі сызыққа;
  - top – жоғары шетке;
  - bottom – төменгі шетке;
  - center – ортаға;
  - justify – еніне;
- multiple – атрибуты мәзірдің бірнеше тақырыбын таңдау мүмкіндігін береді. Бірнеше тақырыпты таңдауда Ctrl пернесін басып тұру қажет. Атрибуттың мәні жоқ. Егер атрибут жазылмаса, мәзірдің тек қана бір тақырыбын таңдау мүмкін.

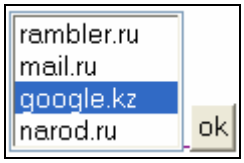
<option> тэгі қалыптағы мәзірдің тақырыптарын анықтайды. Тэг <select> тэгінің ішінде жазылып, жабылу тэгі болмайды. Тэгтің атрибуттары:

- value – атрибуты ағымдағы тақырыпқа берілетін мәнді анықтайды. Егер атрибут берілмесе мән ретінде тақырыптың мазмұны алынады;
- selected – логикалық атрибуты ағымдағы тақырыпты бөлектеп, мазмұнын бастапқы мән ретінде қабылдайды.

Мысалы:

```
<form action="mailto:us-ibr@name" method=post>
  <select name="firstname" multiple>
    <option value="rumbler">rambler.ru
     <option value="mail">mail.ru
     <option value="google" selected>google.kz
     <option value="narod">narod.ru
   </select>
  <input type="submit" value="ok">
</form>
```

Браузер терезесінде келесі тақырыптары анықталған қалып пайда болады (сурет 12):



Сурет 12 – тақырыптары анықталған қалып

<input> тэгі қалыптың өрісін анықтайды. Өрістің мазмұнын тұтынушы анықтайды немесе орындауға жібереді. Тэг <form> тэгінің ішінде жазылып, жабылу тэгі болмайды. Тэгтің атрибуттары:

- name – атрибуты өрістің өзіндік атын анықтайды;
- type – атрибуты өрістің типін анықтайды. Қабылдайтын мәндері:
  - text – мәтіндік өріс;
  - textarea – мәтіндік аймақ;
  - file – қалыпқа файлды біріктіреді;
  - password – мәтіндік өріс. Енгізілген мәтіннің орнына * символы шығады;
  - checkbox – чекбокс, логикалық типті (ия/жоқ) немесе бірнеше пернелі тізімді өріс. Бірнеші өрісті қатар белгілеуге болады. Таңдалған әрбір өрістің өзіндік name/value мәнін серверге жібереді. Осы типті өрістің міндетті түрде name, value және міндетті емес checked атрибуттары болуы тиіс;
  - radio – берілген тізімнің біреуін таңдау өрісі. Тізімдегі барлық пернелер (radio buttons) бірдей аталуы тиіс, олардың тек қана біреуі белгіленеді. Таңдалған өрістің name/value мәні серверге жіберіледі. Осы типті өрістің міндетті түрде name, value және міндетті емес checked атрибуттары болуы тиіс;
  - submit – перне өрісі. Құрылған пернені бассақ, қалып деректері серверге жіберіледі. Value атрибутының мәні перне үстіндегі жазуды көрсетеді;
  - image – графикалық перне өрісі. Графикалық суреттің URL-адресі src атрибутының мәні ретінде көрсетіледі. Серверге тышқан көрсеткішімен шертілген суреттегі орынның x, y координаталары жіберіледі. Координаталар өсі терезенің сол жақ жоғары бұрышынан басталады. Серверге жіберілетін координаталар name.x/value және name.y/value атауына ие.
  - reset – перне өрісі; Құрылған пернені бассақ, қалып өрістеріндегі мәтіндер өшеді, деректер серверге жіберілмейді. Value атрибутының мәні перне үстіндегі жазуды білдіреді;
  - hidden – жасырын өріс, терезеде бейнеленбейді. Өрістің мазмұны name/value басқа мәндер сияқты, серверге жіберіледі. Әдетте тұтынушыға білдірмей, скрипттен скриптке мәлімет жіберуде қолданылады;
  - button – перне өрісі, скрипттің функционалдық перне қызметін атқарады. Value атрибутының мәні перне үстіндегі жазуды білдіреді;
- value – атрибуты өрістің атын, перне үстіндегі жазуды анықтайды;
- checked – логикалық атрибуты тізімдегі тақырыпты бөлектеп, мазмұнын бастапқы мән ретінде қабылдайды;
- size – атрибуты мәтіндік өріс өлшемін символда анықтайды;

- maxlength – атрибуты өріске енгізілетін мәтіннің ұзындығын анықтайды;
- src – атрибуты графикалық перненің URL-адресін анықтайды;
- align – атрибуты графикалық пернені браузер терезесіне туралау түрін көрсетеді. type="image" атрибутымен бірге қолданады. Қабылдайтын мәндері:
  - left – графикалық пернені сол жақ шетке туралайды. Ағымдағы мәтін пернені оң жағынан қоршайды;
  - right – графикалық пернені оң жақ шетке туралайды. Ағымдағы мәтін пернені сол жағынан қоршайды;
  - top – графикалық перненің жоғарғы шетін ағымдағы мәтіннің негізгі сызығына туралайды;
  - bottom – графикалық перненің төменгі шетін ағымдағы мәтіннің негізгі сызығына туралайды;
  - middle – графикалық перненің ортасын ағымдағы мәтіннің негізгі сызығына туралайды;
  - absmiddle – графикалық перненің ортасын ағымдағы мәтіннің ортасына туралайды;
  - absbottom – графикалық перненің төменгі шетін ағымдағы мәтіннің төменгі сызығына туралайды;
- accept – атрибуты файлдың өрістің MIME-типін анықтауда, type="file" атрибутымен бірге қолданылады. Мысалы: accept="text/html", "image/png", "image/gif", "text/tcl", "text/javascript", "text/vbscript"
- title – атрибуты ағымдағы түсініктемені анықтайды.
- alt – атрибуты қосалқы мәтінді анықтайды. Егер браузер файлды таба алмаса, онда терезеде өрістің орнында қосалқы мәтін пайда болады;
- dir – атрибуты өрістегі мәтіннің бейнелену бағытын көрсетеді. Қабылдайтын мәндері:
  - ltr – солдан оңға;
  - rtl – оңнан солға.

3.3 Web-құжатта фреймдерден пайдалану

Web-құжатта фреймдерден пайдалану, құжатты рәсімдеу мүмкіндіктерін арттырады. Фреймдер браузер экранын бірнеше терезеге бөледі. Әр бір терезеде құжаттың жеке бір парағы бейнеленеді. Осы технология негізінде браузерге бірнеше Web-құжатты бір уақытта жүктеу мүмкін. HTML 4.0 тілінің фреймдермен жұмыс істейтін тэгтерін қарастырайық. Оларға төмендегі тэгтер жатады:

- <frameset> (frame set)

- <frame>
- <noframes>

<frameset> тэгі құжаттың фреймдік құрылымын сипаттайды. Тэг фреймдердің өлшемін, олардың экрандағы орнын анықтайды. Тэг </frameset> тэгімен жабылады. Бұл тэг құжатта <body> тэгінің орнына, яғни құжаттың негізгі бөлімі ретінде қолданылады. Тэгтің ішінде <frame> және <noframe> тэгтері жазылады. Тэг төмендегі атрибуттарға ие:

- rows – атрибуты браузер терезесіндегі горизонталь фреймдер саны мен өлшемдерін анықтайды. Фреймдер өлшемі тізбектеліп, төмендегі тәсілдердің бірімен беріледі:
  - браузер терезесі биіктігіне сәйкес пайыз – % түрінде. Мысалы: "30%, 30%, 40%";
  - браузер терезесінің барлық көлемі – * түрінде. Мысалы: "25%, 25%, *";
  - пиксельде – бүтін сан түрінде. Мысалы: "200, *";
- cols – атрибуты браузер терезесіндегі вертикаль фреймдер саны мен өлшемдерін анықтайды. Фреймдер өлшемі тізбектеліп, төмендегі тәсілдердің бірімен беріледі:
  - браузер терезесі еніне сәйкес пайыз – % түрінде. Мысалы: "30%, 30%, 40%";
  - браузер терезесінің барлық көлемі – * түрінде. Мысалы: "25%, 25%, *";
  - пиксельде – бүтін сан түрінде. Мысалы: "200, *";
- border – атрибуты фреймдердің рамкасын пиксельде анықтайды;
- frameborder – атрибуты фреймдердің рамкасы бар немесе жоқтығын анықтайды. Мәндері: yes, no немесе 1, 0;
- framespacing – атрибуты фреймдер арасының қашықтығын (бос аймақ) пиксельде анықтайды;

<frame> тэгі фреймді және оның қасиеттерін анықтайды. Тэг жабылу тэгісіз, <frameset> тэгінің ішінде жазылады. Тэг төмендегі атрибуттарға ие:

- src – міндетті атрибут, фреймде бейнеленетін html-файлдың URL-адресін анықтайды;
- name – атрибуты фреймнің өзіндік атын анықтайды. Бұл атау басқа құжаттан сілтеме жасалғанда target атрибутының мәні ретінде беріледі;
- marginwidth – атрибуты фрейм мазмұнының терезеден сол жақ және оң жақ шеттен ара қашықтығын пиксельде анықтайды;
- marginheight – атрибуты фрейм мазмұнының терезеден жоғарғы және төменгі шеттен ара қашықтығын пиксельде анықтайды;
- scrolling – атрибуты фрейм терезесінің айналдыру сызғышын анықтайды. Қабылдайтын мәндері:
  - yes – айналдыру сызғышы бейнеленеді;

- no – бейнеленбейді;
- auto – қажет болса бейнеленеді;

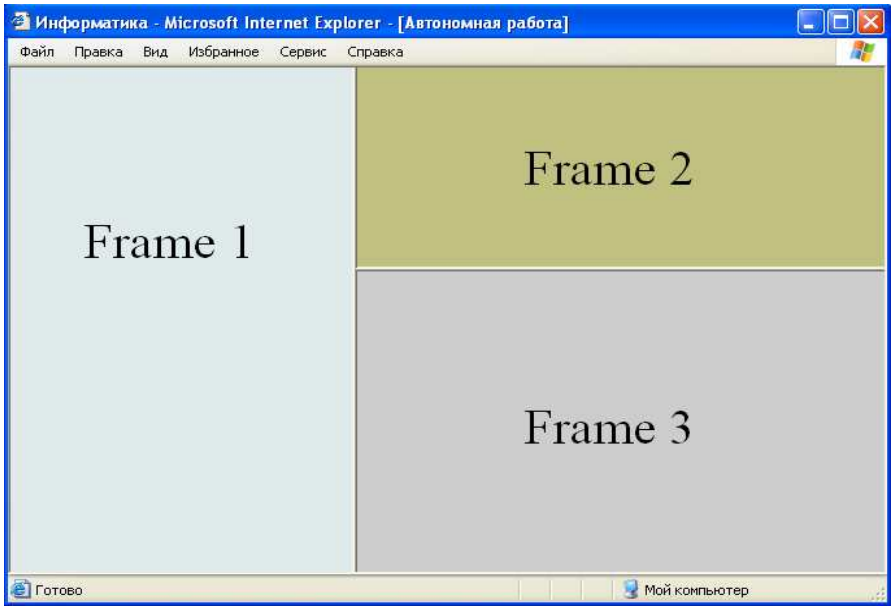
- noresize – атрибуты фрейм өлшемдерін өзгертуге тиім салады;
- frameborder – атрибуты фреймнің рамкасын анықтайды. Мәндері: yes, no немесе 1, 0;

<noframe> тэгі, егер браузер фреймді қолдамаса, терезеге шығатын мәтінді анықтайды. Тэг </noframe> тэгімен жабылады. Тэг <frameset> тэгінің ішінде атрибутсыз жазылады.

Мысалы:

```
<frameset frameborder="1" framespacing="0" border="1" cols="250,*">
  <frame src="info1.html" name="page1">
    <frameset rows="160,*">
      <frame src="info2.html" name="page2" marginwidth="0">
      <frame src="info3.html" name="page3" marginwidth ="0">
    </frameset>
     <noframes> фреймдер </noframes>
</frameset>
```

Браузер терезесінде келесі 3 фреймді терезе пайда болады (сурет 13):



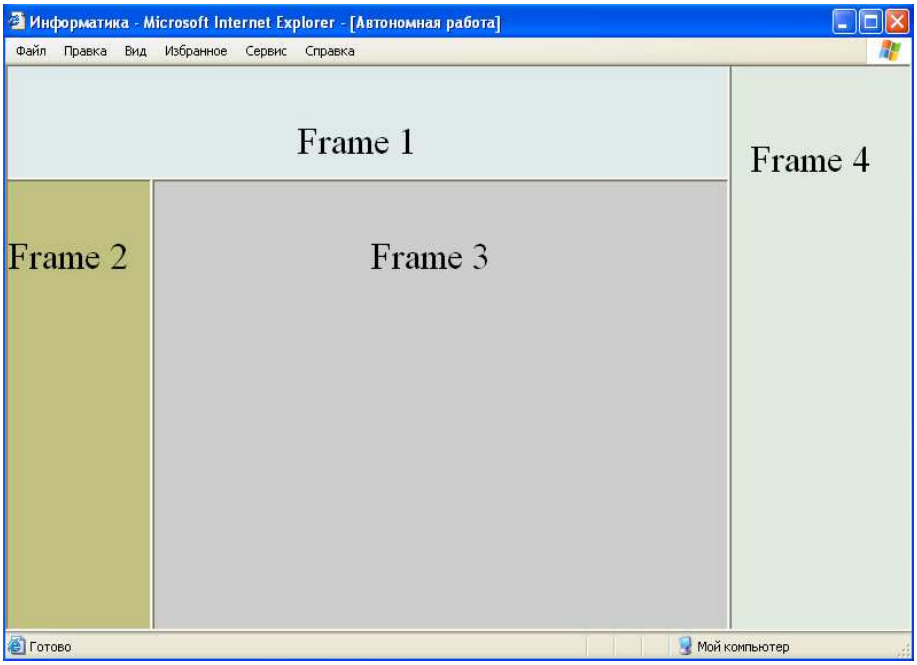
Сурет 13 – үш фреймді терезе

Мысалы:

```
<frameset frameborder="1" framespacing="0" border="1" cols="80%,*">
  <frameset rows="20%,*">
    <frame src="info1.html" name="page1" marginwidth="0">
      <frameset cols="20%,*">
       <frame src="info2.html" name="page2" marginwidth ="1">
       <frame src="info3.html" name="page3" marginwidth ="1">
      </frameset>
  </frameset>
  <frame src="info4.html" name="page4">
```

<noframes> сіддің браузер фреймді көрсетпейді </noframes>
</frameset>

Браузер терезесінде келесі фреймдер пайда болады (сурет 14):



Сурет 14 – төрт фреймді терезе

3.4 Web-құжаттың интерактивтік интерфейсі

CGI (Common Gateway Interface) технологиясы Web-сервер мен клиенттік компьютер арасында деректер алмасуды ұйымдастыруға арналған. Web-сервер клиенттік компьютерден алынған деректерді арнайы бағдарлама (cgi-модуль) көмегінде өңдеп, нәтижені клиенттік компьютерге қайтарады. Сонымен қатар, CGI технологиясы ешқандай тілге тәуелді емес, яғни әмбебап. Сондықтан CGI технологиясы арқылы Web-сервермен кез келген бағдарламалау тілінде жазылған бағдарлама байланыса алады.

В 1993 жылы WWW жүйесінде Web-сервер құрастырушылар арасында тұтынушының интерактивтік интерфейсін құру мәселесі көп талқыланады. Осы талқылауға қатысқандар: NCSA HTTPD серверінің авторы Роб Мак-Кул (Rob McCool), GN серверінің авторы Джон Франкс (John Franks), HTTPD CERN серверінің авторы Ари Luotonen (Ari Luotonen), Plexus серверінің авторы Тони Сандерс (Tony Sanders) және Британдық Колумбия университеті серверінің негізін қалаушы Джордж Филлипс (George Phillips). Осы талқылау нәтижесінде кейбір келісімдер жасалып, CGI технологиясы пайда болды.

Әдетте, Web-серверден алынған деректерді CGI-модуль Web-құжаттың HTML-кодына генерациялайды. Сосын осы HTML-код тұтынушының

сұранысына жауап ретінде тұтынушы компьютеріне қайтарылады. Мысалы, тұтынушы «Интернет туралы кітаптарды табу» сұранысын берді дейік. Осы сұранысқа жауап ретінде, CGI-модуль деректер қорынан барлық кітаптар тізімін құрастырады да, тізімді Web-серверге қайтарады. Ал Web-сервер өз кезегінде тізімді HTML-кодына өткізіп, тұтынушыға жібереді. Сонымен, тұтынушы «Интернет туралы кітаптарды табу» сұранысына жауапты Web-серверден алады. Тұтынушы браузерінде кітаптар тізімі пайда болады.

CGI-модуль Web-серверде арнайы каталогта сақталады, мысалы, "http: //my- Web-site-name.kz/ cgi-bin /script.cgi". Web-құжаттан CGI-модульге параметрлер екі әдіс бойынша жіберіледі (Web-сервер арқылы): GET және POST.

GET әдісінде параметр URL-адресте беріледі. Әдетте, іздеу серверлері, мысалы, Google, Yandex осы әдістен пайдаланады. Мысалы, тұтынушыға Web-құжат пайдаланатын тілді анықтау үшін мынадай өрнек қолданылады:

"http://my-Web-site-name.kz/cgi-bin/script.cgi?lang=kz"

Бұл жерде тілдің атауы URL қатарында "?" белгісінен кейін, "параметр=мән" түрінде беріліп отыр; "lang=kz". Егер параметрлер көп болса, онда олар "&" белгісімен бөлектенеді. Праметрлерге CGI-модульде талдау жасалып, генерациялау нәтижесі браузерге жіберіледі.

Кейбір жағдайларда GET әдісін пайдалану қауіпсіз емес. Мысалы, тұтынушы реквизиттерін (логин мен пароль) жіберуде осы әдісті қолдансақ, онда

"http://my-Web-site-name.kz/cgi-bin/script.cgi? username=shymkent&
password=usa01"

Бұл қатар URL-адрес болып есептеледі, сондықтан ол Web-серверде log-файлда сақталып қалады. Ал онда тұтынушы реквизиттерін әр түрлі бағдарламалар пайдаланып кетуі мүмкін.

POST әдісінде параметрлер HTML тілінде қалыпты құру тэгінен пайдаланады. Мысалы:

```
<form method="post" action="/cgi-bin /script.cgi">
<input type="text" name="shymkent" value="usa01">
<input type="submit" value="ok">
</form>
```

Көрсетілген мысалда тұтынушы реквизиттері сұраныстың ішінде беріледі. Қауіпсіздік мәселесі бойынша GET әдісінен тиімді, дегенмен деректер ашық арнамен жіберілетінін ескерген жөн.

Жалпы, CGI технологиясы тұтынушының интерактивтік интерфейсін құрумен тығыз байланысты.

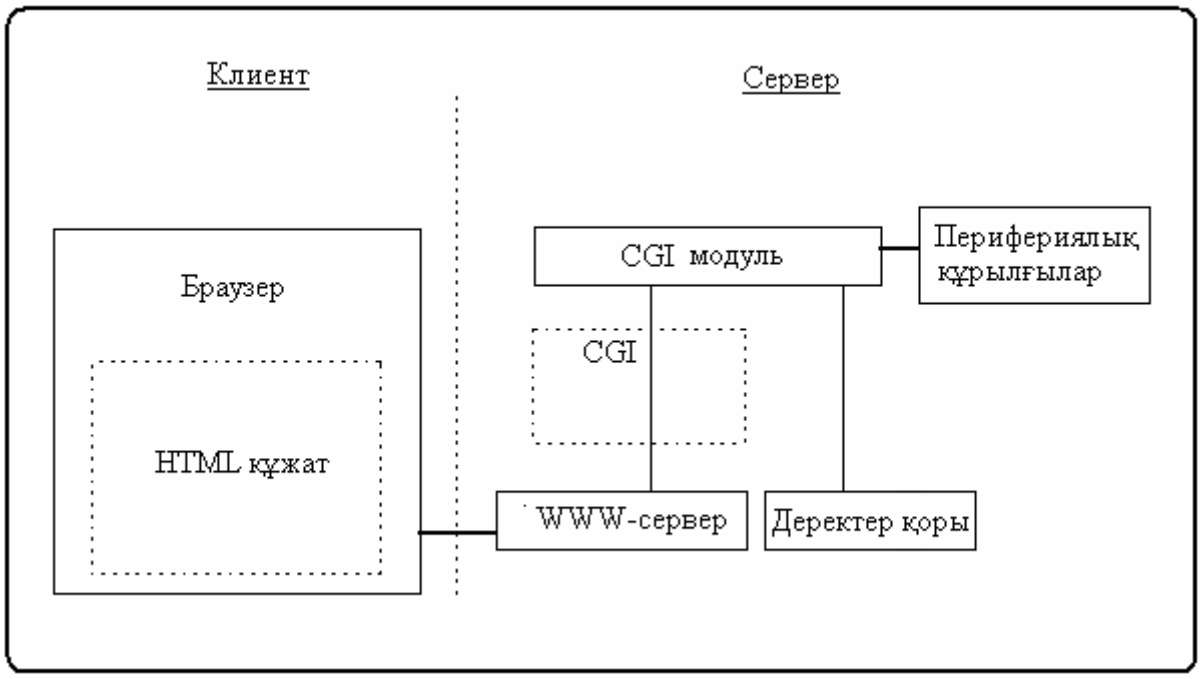
Тұтынушының интерактивтік интерфейсі - тұтынушы мен Web-құжаттың өзара қарым-қатынасын ұйымдастыратын интерфейс. Интерфейс WWW жүйесінде HTML-құжаттардың тізбегі түрінде анықталады [10].

Интерактивтік интерфейсті HTML-құжатты құрастыру үрдісі бойынша шартты түрде екі түрге бөлуге болады: статикалық және динамикалық. Статикалық интерфейс ретінде қарапайым HTML-құжатты айтамыз. Әдетте бұл интерфейс жұмыс барысында тұрақты түрін сақтайды. Динамикалық интерфейсте HTML-құжатқа арнайы бағдарлама коды (CGI-модуль) енгізіліп, ол Web-серверде орындалады. Нәтижеде тұтынушы интерфейсі жұмыс барысында түрленіп отырады.

Сонымен, WWW жүйесі үшін интерактивтік интерфейс анықтамасын төмендегі түрде анықтайды. Тұтынушының интерактивті интерфейсі - деп статикалық немесе динамикалық түрде құрылған, деректер алмасуға арналған Web-құжатты айтады. Іс жүзінде тұтынушы компьютері мен Web-сервер арасында деректер алмасатын кез-келген Web-құжат интерактивтік интерфейс құрумен байланысты. Интерактивтік интерфейс мәселеріне SQL-сервермен байланысу, деректер қорымен байланыс жасау, перифериялық құрылғылардан ақпарат алу немесежіберу, клиенттік компьютермен байланыс орнату, электрондық поштадан пайдалану т.с.с. жатады. Бұл айтылғандардың барлығы CGI технологиясын пайдалану арқылы орындалады.

Жалпы, Web-құжатта тұтынышының интерактивтік интерфейс құру үрдісі негізгі екі бөлімнен тұрады (сурет 15):

- клиенттік бөлім;
- серверлік бөлім.



Сурет 15 - Тұтынушының интерактивтік интерфейсі

Клиенттік бөлім. Интерактивтік интерфейстің клиенттік бөлімін құру үшін тұтынушы интерфейсі сипатталған HTML-құжат құру қажет. Ол HTML тілінде қалыпты құру тэгінен пайдаланамыз. Әдетте қалыптан тұтынушыны тіркеуде, әр-түрлі мәліметтер жинауда, сұранысты қабылдауда пайдаланады. Қалып тұтынушыға арнайы тек толтыруға рұқсат етілген өрістерге деректер енгізуді ұсынады. Енгізілген деректерді WWW-сервердегі арнайы программа (cgi-модуль) өңдейді. HTML 4.0 тілінің қалыптармен жұмыс істейтін тэгтерін қарастырайық. Оларға төмендегі тэгтер жатады: <form>, <textarea> (text area), <select>, <option> және <input>.

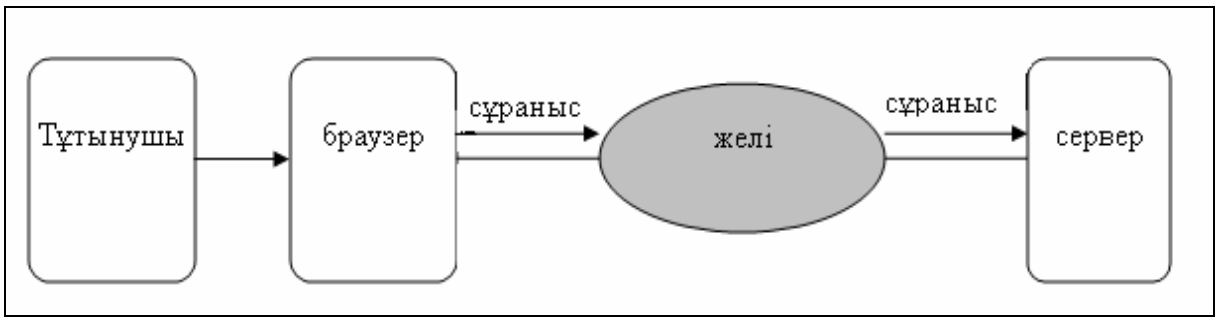
Серверлік бөлім. Серверлік бөлім клиенттік бөлімнен түсетін деректерді өңдеу сұранысын орындаушы модульден тұрады. Ол HTML тілінде жауапты өңдейді. Мұндай модуль cgi-модуль деп аталады. HTML-құжат WWW-сервермен «клиент-сервер» байланыста болуы үшін, олардың өзара қандай HTTP әдісті пайдалануын анықтау қажет. Жалпы алғанда, сұраныс – бұл клиенттің серверге жіберген хабарлары.

WWW-сервер – бұл желі тұтынушыларына серверде орналасқан гипермәтіндік құжаттарды пайдалану мүмкіндігін беретін бүкіләлемдік немесе корпоративтік желінің бір бөлігі. Тұтынушы қашықтағы WWW-сервермен байланысу үшін браузер программасы қажет. Браузер программасы монитор экранында гипермәтіндік құжаттарды URL-адрес бойынша тікелей бейнелеп, алдыңғы немесе келесі парақты оқу, HTML-кодты көру, баспаға шығару т.б. іс-әрекеттерді орындайды.

Web-сервердің жұмыс үрдісін қарап шығайық [4]:

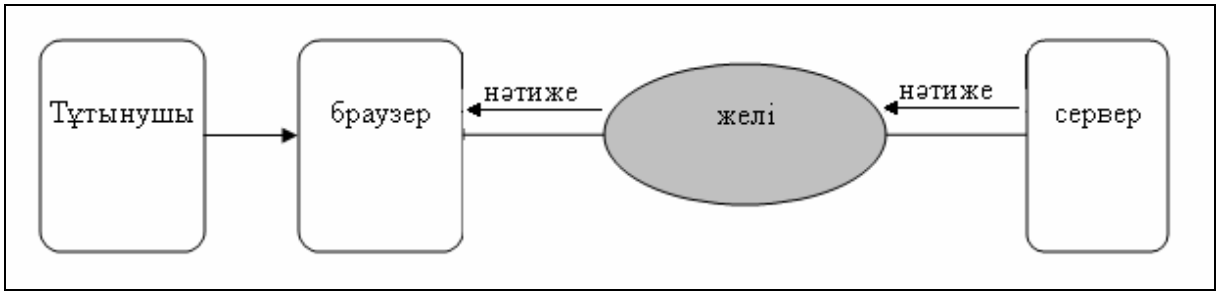
1.Тұтынушы клиенттік компьютердегі браузерді жұмысқа қосады. Клиенттік компьютер міндетті түрде желіге жалғанған болуы тиіс. Содан соң браузер төмендегі функцияларды іске асырады (сурет 16):

- WWW-сервермен HTTP хаттамасы арқылы байланыс жасайды;
- көрсетілген URL-адрестері бойынша Web-сайтты іздейді (іздеу машиналары арқылы);
- табылған Web-сайтты монитор экранында бейнелейді (интерпретациялау);
- тұтынушының монитор экранындағы Web-сайтқа тиісті клиенттік іс-әрекеттерін міндетті түрде орындайды (манипуляциялау);
- тұтынушыға браузер программасының ішкі сервистік қызметтерін ұсынады;
- WWW-серверге тұтынушы сұранысын дестелеп коммутациялау технологиясы әдісімен жібереді.



Сурет 16 - Клиенттен серверге сұраныс жіберу

2.WWW-сервер тұтынушы сұранысын CGI технологиясы арқылы орындап, нәтижені HTML-коды түрінде клиенттік компьютердегі браузерге HTTP хаттамасы арқылы қайта жібереді (сурет 17).



Сурет 17 - Серверден клиентке нәтижені қайтару

3.Клиенттік компьютердегі браузер HTTP хаттамасы арқылы жіберілген HTML-кодты қабылдап, монитор экранында форматтап бейнелейді және тұтынушының келесі нұсқауларын күтеді. Тұтынушының нұсқаулары төмендегі түрде болуы күтіледі:

- Браузердің адрестік қатарына жаңа Web-сайттың URL-адрестерін енгізеді;
- ағымдағы Web-сайтта контентті іздеу, масштабтау, баспаға шығару, скриптті орындауға рұқсат беру және т.б. іс-әрекеттер жасайды (манипуляциялау);
- ағымдағы Web-сайтта гиперсілтеме арқылы басқа Web-сайтпен байланыс орнатады.

Ескерту: Бұл жерде бірінші және үшінші жағдайларда әр-түрлі құжат қолданылады.

4 CSS технологиясы

4.1 CSS технологиясының синтаксисі

CSS технологиясының пайда болуы. CSS каскадты стильдер технологиясы W3C Интернет консорциумы ұсынған «Web стандарттары» деп аталатын технологиялардың бірі болып есептеледі.

1990 жылдары барлық браузерлердің Web-сайттарды бейнелеуде өз стильдері болды. XML және HTML тілдері осы жылдары қарқынды дамыды және сол кездегі ақпаратты бейнелеу мен рәсімдеу үшін жеткілікті болды.

1990 жылдардың ортасында бағдарламашы мен Web-дизайнерлер үшін Web-сайттарды жобалауда бірыңғай ережелерден пайдалану қажеттігі туындады. Осылай HTML тілімен қатар CSS технологиясы пайда болды [11].

1994 жылы Хокон Виум Ли (Håkon Wium Lie, Норвегия) «каскадты стильдер кестесі» терминін ұсынды және Берт Боспен (Bert Bos) бірге осы технологияны дамытты. Бұл технологияның басқа стильдерден негізгі ерекшелігі – стильдерді сипаттауда мұрагерлік қасиетінен пайдаланылған, яғни бағдарламашы жаңа стильдерді құруда өзінің алдыңғы стильдерінен пайдалана алады.

Бұл технологияны одан әрі дамытуға W3C Интернет консорциумы қызығушылық танытып, 1996 жылы CSS1 спецификациясын ұсынды. Ал 1998 жылы өңделген және толықтырылған CSS2 спецификациясы шықты [12].

2011 жылы 7 июньда W3C консорциумы CSS2 спецификациясының орнына CSS 2.1 спецификациясын ұсынды. Қазіргі таңда Интернеттегі барлық Web-сайттар осы спецификацияға өту және қолдану қажеттігі туындап отыр.

Синтаксис. CSS 2.1 спецификациясының синтаксисі CSS стильдер кестесінің алдыңғы версияларын да өз ішіне алады. Сонымен қатар, CSS технологиясының келешекте құрылатын жаңа версияларында да осы синтаксистің ядросы сақталатыны анық, тек қана жаңа синтаксистік құрылымдар қосылуы мүмкін. Мұның барлығы Web-дизайнерлерге кез келген гипермәтіндік құжатты синтаксистік талдау мүмкіндігін береді. Дизайнерлер стильдер кестесін құруда өзіне таныс емес стиль қасиеттерін де пайдалана алады.

Белгілер (Labels). CSS стильдер кестесін лексикалық талдау деңгейіне қарасақ, ол белгілер тізбегінен тұрады. Белгілердің анықтамасында регуляр өрнектер Lex-стилінде жазылады. Сегіздік санақ жүйесіндегі кодтар ISO 10646 стандартымен анықталады. Төмендегі кестеде CSS спецификациясында кездесетін белгілер тізімін келтіреміз (кесте 1) [12].

Кесте 1 – CSS белгілері мен анықтамасы

белгі	анықтамасы
IDENT	{ident}
ATKEYWORD	@{ident}
STRING	{string}
INVALID	{invalid}
HASH	#{name}
NUMBER	{num}
PERCENTAGE	{num}%
DIMENSION	{num}{ident}
URI	url\({w}{string}{w}\) | url\({w}([!#$%&*-~] | {nonascii} | {escape})*{w}\)
UNICODE-RANGE	U\+[0-9a-f?]{1,6}(-[0-9a-f]{1,6})?
CDO	<!--
CDC	-->
;	;
{	\{
}	\}
(	\(
)	\)
[	\[
]	\]
S	[\t\r\n\f]+
COMMENT	\/*[^*]**+([^/*][^*]**+)*\/
FUNCTION	{ident}\(
INCLUDES	~=
DASHMATCH	|=
DELIM	көрсетілген анықтамаларға жатпайтын символ

Төмендегі кестеде 1-кестеде көрсетілген фигуралық жақша ішіне жазылған макростар анықтамасын келтіреміз (кесте 2).

Кесте 2 – CSS макростары мен анықтамасы

макрос	анықтамасы
Ident	[-]?{nmstart}{nmchar}*
Name	{nmchar}+
Nmstart	[_a-z] | {nonascii} | {escape}
Nonascii	[^\0-\177]
Unicode	\\[0-9a-f]{1,6}(\r\n | [\n\r\t\f])?
Escape	{unicode} | \\[^\n\r\f0-9a-f]

Nmchar	[_a-z0-9-] | {nonascii} | {escape}
Num	[0-9]+ | [0-9]*\.[0-9]+
String	{string1} | {string2}
string1	\"([^\n\r\f\\"] | \\{nl} | {escape})*\"
string2	\'([^\n\r\f\\'] | \\{nl} | {escape})*\'
Invalid	{invalid1} | {invalid2}
Invalid1	\"([^\n\r\f\\"] | \\{nl} | {escape})*
Invalid2	\'([^\n\r\f\\']|\\{nl} | {escape})*
Nl	\n|\r\n|\r|\f
W	[\t\r\n\f]*

Лексика. Төменде синтаксистік ережелердің ядросын (грамматикалық сипаттау) келтіреміз:

```
stylesheet    : [ CDO | CDC | S | statement ]*;
сипаттау      : ережелер | at-ережелер;
at-ережелер   : ATKEYWORD S* кез келген* [ бөлім | ';' S* ];
бөлім         : '{' S* [ кез келген | бөлім | ATKEYWORD S* | ';' S* ]* '}' S*;
ережелер      : селектор? '{' S* жариялау? [ ';' S* жариялау? ]* '}' S*;
селектор      : кез келген+;
жариялау      : DELIM? қасиет S* ':' S* мәні;
қасиет        : IDENT;
мәні          : [ кез келген | бөлім | ATKEYWORD S* ]+;
кез келген    : [ IDENT | NUMBER | PERCENTAGE | DIMENSION | STRING
                | DELIM | URI | HASH | UNICODE-RANGE | INCLUDES
                | DASHMATCH | FUNCTION S* кез келген* ')'
                | '(' S* кез келген* ')' | '[' S* кез келген* ']' ] S*;
```

Грамматикалық сипаттауда оқуға жеңіл болу үшін COMMENT белгісі қолданылмайды, өйткені түсініктемені кез келген жерде қолдануға болады. Сол сияқты, S белгісі бос орынды білдіреді. Бос орын ретінде "пробел" (U+0020), "азат жол басы" (U+0009), "қатарды ауыстыру" (U+000A), "каретканы ауыстыру" (U+000D) және "қағаз беру" (U+000C) символдары кездесуі мүмкін.

Мысалы: STRING белгісін қарастырайық. Ол {string} макросынан тұрады. Ол өз кезегінде {string1} немесе {string2} түрінде жазылады. Ал {string1} анықтамасы \"([^\n\r\f\\"]|\\{nl}|{escape})*\" түрінде берілген. Сол сияқты {string2} анықтамасы \'([^\n\r\f\\']|\\{nl}|{escape})*\' түрінде берілген т.с.с.

Осы ережелерден келіп шығатыны қатарды жазуда кез келген символды, яғни әріп, сан, арнайы символдарды (Unicode коды ISO 10646 кодына эквивалент) қолдануға болады. Сондай-ақ, қатар ' (тырнақша)

немесе " (қос тырнақша) символдары ішіне жазылады. Қатарды жазуда тырнақша мен қос тырнақша символдарының аралас пайдаланып немесе олардың алдына \ (бэкслэш) символын жазуға да болады.

' "Информатика" кафедрасы'
" 'Информатика' кафедрасы"
"\"Информатика\" кафедрасы"

Көрсетілген мысалдың бірінші жолында тырнақша қолданылып, «информатика» сөзі қос тырнақша ішіне алынған. Екінші жолда осының барлығы керісінші орындалған. Ал үшінші жолда тек қана қос тырнақша қолданылған, тек қана ішкі тырнақшалар бэкслэш символымен ерекшеленген.

Идентификатор. CSS стильдер кестесінде идентикатор әріптерден (a-z), цифрлардан (0-9), '-' (тире) және '_' (төменгі сызықша) символдары тізбегінен тұрады. CSS стильдер кестесі регистрге қарамайды, яғни бас әріп пен кіші әріпті бірдей қабылдайды. Сонымен қатар идентификаторда ISO 10646 стандартының кейбір арнайы символдары (U+00A1–ден жоғары) және \ (бэкслэш) символы қолданылады.

Мысалы: ocean – дұрыс жазылған
zaru_01 – дұрыс жазылған
R&B – дұрыс жазылған

Әдетте \ (бэкслэш) символы арнайы символдардың іс-әрекетін тоқтату үшін пайдаланылады. Егер ол әріптен алдын келсе, онда оны жоқ деп есептеуге де болады. Мысалы, «apple» идендификаторын «a\pple» немесе «a\ppl\e» деп жазуға болады.

Кілт сөздер. CSS стильдер кестесінде кілт сөздер идентификатор түрінде кездеседі. Кілт сөздер тырнақша ішіне алынбайды ("..." немесе '...').

Мысалы: color – дұрыс жазылған
"color" – қате жазылған (қатар ретінде түсініледі)

Кейбір кілт сөздер '-' (тире) немесе '_' (төменгі сызықша) символдарымен басталуы да мүмкін. Әдетте мұндай кілт сөздер ұйым, мекеме атауларына қалдырылған. Спецификация жазылған уақытта төмендегі префикстер белгілі болған:

-ms- Microsoft
-mso- Microsoft Office
-moz- Mozilla
-o- Opera Software
-atsc- Advanced Television Standards Committee
-wap- WAP

Өрнектер. CSS стильдер кестесі өрнектер тізімінен тұрады. Өрнектер 2 топқа бөлінеді:

- at-ережелер
- ережелер тобы

At-ережелер. CSS стильдер кестесінде at-ережелер «at» кілт сөзінен басталады, яғни @ символынан кейін идентификатор тұрады. Мысалы, @media, @import.

Мұндай атау @ («эт» деп оқылады) символының атынан келіп шыққан. Орыс тілді Интернетте бұл символды «собачка» деп те атайды. Оны қазақшаға «күшік» деп аудару ыңғайсыз шығар.

At-ережелер ; (нүктелі үтір) символы кездескенше барлық құрылымдық бөлімді өз ішіне алады.

Мысалы: @import "ukgu.css";
h1 { color: red }

At-ережелерден басқа барлық құрылымдар ережелер тобына жатады.

Бөлімдер. CSS стильдер кестесінде бөлім {} (фигуралық жақша) символдары ішіне жазылады. Бөлімде кез келген белгілер (labels) жазылады. Бөлімнің ішінде ( ) қарапайым, [ ] тік және { } фигуралық жақшалар тек қана жұбымен және біріне-бірі кіретіндей жазылуы қажет. Сол сияқты, ' (тырнақша) немесе " (қос тырнақша) символдары да жұбымен және біріне-бірі кіретіндей жазылады.

Мысалы:

```
negizgi_fon {
        background-image: url ("files/fon_cyreti.gif");
        font-family: "Times New Roman", shadow;
        font-size: 12 pt;
        color: "#000080";
        }
tizim   {
        font-family: "Times New Roman", serif;
        font-size: 14 pt;
        color: "#000080";
        font-style: normal;
        }
```

Көрсетілген мысалда «negizgi_fon» және «tizim» бөлімдері берілген. Бөлімде кездесетін қарапайым жақшалар ішпе-іш орналасқан. Ал "files/fon_cyreti.gif" және "Times New Roman" өрнектері қатар (string) ретінде қаралады.

Ережелер тобы. CSS стильдер кестесінде ережелер селектор және сипаттау бөлімінен тұрады.

Селектор деп бірінші фигуралық жақшаға дейінгі кездесетін идентикатор немесе идентификаторлар тобы айтылады. Селектор әрқашан өзінің сипаттау бөлімімен жазылады. Егер селектордың синтаксисі қате жазылса, онда оның сипаттау бөліміне талдау жасалмайды.

Сипаттау бөлімі фигуралық жақшалар ішіне жазылады. Бөлімде жариялаулар нүктелі үтірмен бөлектенеді. Бөлім бос болуы да мүмкін.

Мысалы:

```
h1, h2 {color: yellow}
h3 & h4, h5 {color: red}
```

Көрсетілген мысалда h1, h2, h3, h4, h5 селекторлар, ал фигуралық жақшалар ішіндегі оның сипаттау бөлімі. Мысалдағы бірінші жол синтаксисі дұрыс берілген. Ал екінші жол қате, яғни & символы қате белгі болып есептеледі. Сондықтан тұтынушы агенті селектор үшін қызыл реңді қабылдамайды.

Жариялау және қасиеттер. CSS стильдер кестесінде жариялау қасиет идендификаторы, : (қос нүкте) символы және қасиеттің мәнінен тұрады. Егер селекторда бірнеше жариялау қатары кездессе, онда олардың арасына ; (нүктелі үтір) символы жазылады.

Қасиет идентификаторының мәні ретінде кез келген белгі алынады. Қасиет мәнінің ішінде қарапайым, тік және фигуралық жақшалар, сонымен қатар тырнақша және қос тырнақша символдары тек қана жұбымен және біріне-бірі кіретіндей жазылуы тиіс. Егер нүктелі үтір қатарға жатпаса, онда ол бэкслэш символымен бірге жазылады.

Мәндер синтаксисі әрбір қасиет үшін бөлек анықталады. Жалпы, мәндер идентификаторлардан, қатарлардан, сандардан, түстерден т.б. құралады.

Мысалы:

```
tizim {
        font-family : "Times New Roman", serif;
        font-size : 14 pt;
        color : "#000080";
        font-style : normal;
        }
```

Көрсетілген мысалда «font-family», «font-size», «color», «font-style» - қасиет идентификаторлары, ал «"Times New Roman", serif», «14 pt», «"#000080», «normal» - мәндері.

Селекторлардағы жариялауларды біріктірмей бөлек жазуға да болады. Онда селектордың идентификаторы әр қатарда жазылуы тиіс.

Мысалы: Төменде берілген ереже

```
tizim {
        font-family : "Times New Roman", serif;
        font-size : 14 pt;
```

}

мына ережелерге эквивалент

```
tizim  {font-family : "Times New Roman", serif}
tizim  {font-size : 14 pt }
```

Түсініктеме. CSS стильдер кестесінде түсініктеме /* және */ символдары арасына жазылады. Түсініктемені кез келген жерде жазуға болады.

Мысалы: /* CSS технологиясы түсініктемесі*/

Түсініктемені SGML стилінде, яғни <!—және --> символдары арасында да беруге болады. Бірақта олардың қолдану аясы шектелеген (әдетте HTML 3.2-ден төменгі версияларда).

Синтаксистік талдаудың қателерін өңдеу ережесі. Егер CSS стильдер кестесінде қателер табылса, онда тұтынушы агенті оларды өңдеуі қажет. Бұл жерде өңдеу дегенде қатесі бар стильді синтаксистік талдаудан өткізіп, оның басы мен аяғын тауып, стильді осы қатесі бар өрнегін тұтынушы агенті орындамауын айтамыз.

CSS стильдер кестесінің келесі версияларында жаңа қасиеттер мен олардың мәндері енгізілуі мүмкін деп тұтынушы агенті келесі ережелерді өңдеуі тиіс:

а) *Белгісіз қасиет*. Тұтынушы агенті белгісіз қасиетпен берілген жариялауды өңдемеуі қажет.

Мысалы: Стильдер кестесі берілсін (rotation : 70 minutes – қасиет қате берілген):

```
ukgu {color : blue; rotation : 70 minutes}
```

Тұтынушы агенті стильдер кестесі төменде берілгендей етіп өңдейді:

```
ukgu {color : blue}
```

ә) *Қасиеттің мүмкін емес мәні*. Тұтынушы агенті мүмкін емес мән кездесетін жариялауды өңдемеуі қажет.

Мысалы: Стильдер кестесі берілсін:

```
img {float : lefet}       /* lefet мәні қате берілген */
img {background : "blue"} /* blue кілт сөзі қос тырнақша ішіне
жазылған */
```

б) *Malformed жариялау*. Тұтынушы агенті синтаксистік талдау нәтижесінде кездесетін күтілмеген белгілерді, (), [], {}, "", және " символдарының жұп келмеуі т.с.с. қателерді мүмкін болса қалпына келтіруі қажет.

Мысалы: Стильдер кестесі берілсін:

```
ukgu {color : blue; color } /* қос нүкте символы және мәні берілмеген
*/
ukgu {color : }blue}  /* фигуралық жақша - күтілмеген белгі */
```

в) *Күтілмеген at-ережелер*. Тұтынушы агенті күтілмеген at-ережелерді нүктелі үтір символы кездескенше немесе келесі құрылымдық бөлім біткенше өңдемеуі қажет.

Мысалы: Стильдер кестесі берілсін:

```
@tree-md
{
@import "tizim.css";
ukgu {color: red}
}
ukgu {color: blue}
```

Көрсетілген мысалда @tree-md деген at-ереже CSS 2.1-де анықталмаған, сондықтан келесі құрылымдық бөлім өңделмейді, тұтынушы агенті берілген стильдер кестесін төмендегідей етіп қысқартады:

```
ukgu {color: blue}
```

г) *Күтілмеген стильдер кестесінің соңы*. Тұтынушы агенті барлық аяқталмаған құрылымдық бөлімдерді (бөлімдер, жақшалар, қатарлар) жабуы қажет.

Мысалы: Берілген стильдер кестесі

```
@media screen
{
h1 : before {content: 'Шымкент      /* стильдер кестесі аяқталмаған */
```

төмендегідей орындалады:

```
@media screen
{
h1 : before {content: 'Шымкент';}
}
```

д) *Күтілмеген қатардың соңы*. Тұтынушы агенті барлық ашық қатарларды жабуы, сосын осы құрылымды (жариялау, ереже) өңдемеуі қажет.

Мысалы: Берілген стильдер кестесі

```
p1 {
color : maroon;
font-family : 'Times New Roman   /* нүктелі үтір қойылмаған */
color : red;
color : olive;
```

```
}
```

төмендегідей орындалады:

```
p1 { color: maroon; color: olive; }
```

Көрсетілген мысалда «font-family : 'Times New Roman» қатары нүктелі үтір қойылмағандықтан орындалмайды. Келесі «color:red;» қатарында қате жоқ, бірақта қатардан алдыңғы нүктелі үтір жоқ болғандықтан орындалмайды (синтаксистік қате).

Бүтін және нақты сандар. CSS стильдер кестесінде қасиеттердің мәндерінің типі бүтін (<бүтін> деп белгіленеді) немесе нақты (<сан> деп белгіленеді) сандармен өрнектеледі. Бүтін және нақты сандар тек қана ондық санақ жүйесінде анықталған. Бүтін сан бір немесе бірнеше «0» ден «9»-ға дейінгі цифрлерден тұрады. Нақты сан бүтін және бөлшек бөліктерден құралады. Олардың арасына . (нүкте) қойылып жазылады. Көптеген қасиеттер тек қана оң таңбалы сандарды ғана қабылдайды. Кейбір жағдайда бүтін және нақты сандар алдына «-» немесе «+» таңбалары жазылуы мүмкін.

Ұзындықтың өлшем бірліктері. CSS стильдер кестесінде ұзындықтың өлшем бірліктері көлденең және тік өлшемдерді анықтауда қолданылады. Өлшем форматы (<ұзындық>) төмендегідей: <сан>, өлшем бірлігі идентификаторы (px, em, ex). Егер мән нольге тең болса өлшем бірлігі жазылмайды.

Кейбір қасиеттер сол таңбалы мәндерді қолдануға да рұқсат береді, бірақта оларды іс жүзінде пайдалану техникалық құрылғылармен қарама-қайшылықтарға әкелуі мүмкін. Ондай жағдайда тұтынушы агенті стильді өңдемейді.

Ұзындықтың өлшем бірліктері екі түрге бөлінеді:

- салыстырмалы
- абсолют

Салыстырмалы өлшем бірліктері ұзындық мәнін басқа ұзындық қасиетіне салыстыра отырып анықтайды. Әдетте мұндай бірліктер бір типті құрылғыдан екіншісіне өтуде баптау жеңіл болу үшін қолданылады. Мысалы, монитор экранынан принтерге өткізу.

Салыстырмалы өлшем бірліктері:

- em /* сәйкес шрифттің 'font-size' қасиеті */
- ex /* сәйкес шрифттің 'x-height' қасиеті */
- px /* сәйкес құрылғының пикселі */

а) «em» өлшем бірлігінің мәні ағымдағы шрифттің 'font-size' қасиетінің мәніне сәйкес анықталады. Ол көлденең және тік өлшемдерде қолданылады.

Мысалы: Стильдер кестесі берілсін:

```
h1 { line-height: 1.2em }
```

Көрсетілген мысалда h1 селекторы шрифтінің биіктігі ағымдағы шрифттің биіктігінен 20% ұзын болуы тиіс.

ә) «ex» өлшем бірлігінің мәні ағымдағы шрифттің бас «x» әрпінің мәнімен анықталады. Бірақта ағымдағы шрифтте «x» әрпі болмаса да анықталуы тиіс.

Шрифттің «x-height» мәнін анықтаудың бірнеше әдісі бар. Әдетте шрифттердің сенімді «x-height» метрикасы болады. Егер ол болмаса, онда тұтынушы агенті шрифттің бас әріптерінің биіктігіне сәйкес анықтайды.

Мысалы:

```
h1 { margin: 1ex }
```

б) «px» өлшем бірлігінің мәні ағымдағы құрылғының (монитор, принтер т.б.) пикселімен анықталады. Әдетте ол монитордың рұқсат берілген мүмкіндігін анықтау үшін қолданылады.

Абсолют өлшем бірліктері құрылғының физикалық параметрлері белгілі болғанда қолданылады.

Абсолют өлшем бірліктері:

- in /* дюйм – 1 дюйм 2,54 сантиметрге тең */
- cm /* сантиметр */
- mm /* миллиметр */
- pt /* нүкте – 1 нүкте 1/72 дюймге тең */
- pc /* пика – 1 пика 12 нүктеге тең */

Мысалы:

```
h1 { margin: 0.5in }          /* дюйм  */
h2 { line-height: 3cm }       /* сантиметр */
h3 { word-spacing: 4mm }      /* миллиметр */
h4 { font-size: 12pt }        /* нүкте */
h5 { font-size: 1pc }         /* пика */
```

Пайыздық мәндер. CSS стильдер кестесінде кейбір қасиеттердің мәні пайыз түрінде де анықталады. Пайыздық мәннің форматы:

<пайыз> : <сан>%

Пайыздық өлшем бірліктері мәндерін басқа қасиеттің ағымдағы мәніне сәйкес пайызбен анықтайды.

Мысалы:

```
p { line-height: 120% }
```

URI-мән. CSS 2.1 спецификациясында URI-мән (Uniform Resource Identificator) <uri> деп белгіленеді. Қасиеттерге URI-мәнді меншіктеуде «url()» құрылымында қажетті URL-адрес (Uniform Resource Locator – Ықшамдалған ресурстар көрсеткіші) көрсетіледі.

Мысалы:

```
body { background: url ("http://www.ukgu.kz/logo.gif") }
```

Түстер. CSS стильдер кестесінде түсті анықтауда түстердің RGB (red, green, blue) моделінен пайдаланып, түстің ағылшынша атын немесе он алтылық санақ жүйесіндегі мәнін береміз. Түс red-қызыл, green-жасыл, blue-көк үш бояудың араласуынан тұрады. Араласу мөлшері он алтылық сан 00-ден басталып FF-санымен аяқталады. Он алтылық сан алдына # символы жазылады.

Мысалы:

```
fon {
background-color: "#e0e0e0";
font-family: "Times New Roman";
font-size: 12 pt;
color: "#000080";
}
```

Көрсетілген мысалда монитор экранының фоны "#e0e0e0" түсімен, ал шрифт "#000080" (тоқ көк) түсімен берілген.

Жалпы, RGB форматында мәндерді үш түрде беруге болады:

- он алтылық санақ жүйесі;
- ондық санақ жүйесі;
- пайыздық.

Ондық санақ жүйесінде 0-ден 255-ке дейінге сандар қолданылады (он алтылық санақ жүйесінде 00-ден FF-дейін). Пайыздық жүйеде 255 санына 100% сәйкес келеді, яғни RGB (255, 255, 255) немесе RGB (100%, 100%, 100%) ақ түсті береді.

Мысалы:

```
em { color : rgb (255,0,0) }
em { color : rgb (100%, 0%, 0%)}
```

CSS стильдер кестесін ұсыну. CSS стильдер кестесі символдардың әмбебап тобынан тұрады (ISO 10646). Бұл символдарды тасымалдау және сақтау үшін, оларды ASCII символдарын қолдайтын кодтау жүйелеріне ауыстыру қажет (мысалы, UTF-8, Windows-1251).

Егер стильдер кестесі басқа құжатқа енгізілетін болса, мысалы, HTML-құжаттың «style» атрибуты, онда стильдер кестесі осы құжатта қолданылған кодтау жүйесінен пайдаланады.

Егер стильдер кестесі жеке файлда сақталса, онда тұтынушы агенті символдарды кодтау жүйесін анықтаудың келесі приоритетін қолдануы тиіс (жоғарыдан төменге):

- «charset» HTTP-параметрінің «Content-Type» өрісі (немесе басқа хаттамадағы сәйкес параметр)
- @charset
- <link charset=""> немесе сілтеменің басқа метадеректері
- құжаттан немесе стильдер кестесінен алынған символдар тобы

@charset ережесін қолданушылар, ережені стильдер кестесінің басына орналастыруы тиіс. «@charset» ережесінен кейін символдарды кодтау жүйесінің атауы тырнақша ішіне жазылады.

Мысалы:

```
@charset " Windows-1251";
```

Тұтынушы агенті стильдер кестесінің басында келмеген кез келген @charset ережесін қабылдамауы тиіс.

4.2 CSS технологиясының қасиеттері

Бұл бөлімде CSS каскадты стильдер кестесінің қасиеттерімен танысамыз. CSS стильдер кестесінде қасиет «property-name» деп аталады. Қасиеттің мәндері кілт сөздермен беріледі. Кілт сөздер < (кіші) және > (үлкен) символдары арасына жазылады.

Кілт сөздер арасында кездесетін | символы, қасиеттің осы мәндердің тек біреуін қабылдайтынын білдіреді. Сол сияқты, кілт сөздер арасында кездесетін || символы, қасиеттің осы мәндердің бірнешеуін, кез келген тәртіпте қабылдайтынын білдіреді. Ал кілт сөздерді топтау үшін [ ] символдары қолданылады.

Төменде осы технологияның негізгі қасиеттерін қарастырамыз [2].

background-color : <түс> | transparent – құжаттың фон түсін анықтайды. Фонның түсін анықтаумен қатар, transparent арқалы фон асты түсін де (paddings) көрсетуге болады. Сыртқы өріс (margins) ашық түсте қалады. Үндеместен transparent алынады.

Мысалы:

```
fon {
background-color: gray;
}
```

Көрсетілген мысалда фон үшін күлгін түс таңдалған.

background-image: url (<url>)│none│inherit – фон ретінде суретті файлдан алады. none – фондық сурет берілмейді. Inherit – фонның мәнін аналық қасиеттен алады. Үндеместен none мәні алынады.

Мысалы:

```
fon  {
background-color : "#e0e0e0";
background-image : url ("files/logo_fon.gif");
font-family : "Times New Roman", shadow;
font-size : 12 pt;
color : "#000080";
}
```

Көрсетілген мысалда фондық сурет ретінде «files» бумасындағы «logo_fon.gif» файлы алынады. Сонымен қатар «background-color» қасиетімен «#e0e0e0» түсі берілген. Егер қандай-да бір себеппен «logo_fon.gif» файлы жүктелмесе, онда фон «#e0e0e0» түсін алады. Фонда мәтін айқын көріну үшін «Times New Roman» шрифті, өлшемі 12 нүктемен берілген. Әдетте фондық суретті осылай фондық түспен беру бағдарламалаудың тамаша үлгісі болып есептеледі. Бұл жерде фондық түс фондық суреттің астында қалады.

background-position : top │ center │ bottom │ left │ center │ right │<пайыз>│ <ұзындық> │ inherit – фондық суреттің басталу координаталарын анықтайды. Фондық сурет координаталары кілттік сөздермен көлденеңінен және тігінен беріледі.

Фондық суреттің көлденеңінен басталуы төмендегіше беріледі:

- left – солдан;
- center – ортадан;
- right – оңнан.

Фондық суреттің тігінен басталу координаталары төмендегіше беріледі:

- top – жоғарыдан;
- center – ортадан;
- bottom – төменнен.

Координаталар көлденеңінен бастап, сонан соң тігінен көрсетіледі.

Мысалы:

```
fon  {
background-color : "#e0e0e0";
background-position: right top;
background-image : url ("files/logo_fon.gif");
}
```

Көрсетілген мысалда фондық сурет жоғарғы оң жақ бұрышқа орналасады.

Координаталар пиксельде – (px), сантиметрде – (cm) немесе пайызда – (%) берілуі де мүмкін.

Мысалы:

```
fon  {
background-color : "#e0e0e0";
background-position: 50px 50px;
background-image : url ("files/logo_fon.gif");
}
```

Көрсетілген мысалда фондық сурет жоғарғы сол жақ бұрыштан тігінен 50 пиксель және енінен 50 пиксель тастап орналасады.

Егер координаталар пайызда берілсе, онда 0% 0% - жоғарғы сол бұрышты, ал 100% 100% - төменгі оң бұрышты анықтайды.

Мысалы:

```
fon  {
background-color : "#e0e0e0";
background-position: 50% 50%;
background-image : url ("files/logo_fon.gif");
}
```

Көрсетілген мысалда фондық сурет монитор экранының ортасынан бастап орналасады.

Жалпы, фондық суретті орналастыруда төмендегі мәндер алынады:

- жоғарғы сол бұрыш - left top, 0% 0% немесе top left;
- жоғарғы ортасы - center top, 50% 0%, top center;
- жоғарғы оң бұрыш - right top, 100% 0%, top right;
- сол жақтың ортасы - left center, 0% 50%, center left;
- центр - center, 50% 50%, center center;
- оң жақтың ортасы - right center, 100% 50%, , center right;
- төменгі сол бұрыш - left bottom, 0% 100%, bottom left;
- төменгі ортасы - center bottom, 50% 100%, bottom center;
- төменгі оң бұрыш - right bottom, 100% 100%, bottom right;
- inherit – фонның координаталарын аналық қасиеттен алады;
- үндеместен 0% 0% алынады.

background-repeat : repeat │ repeat-x │ repeat-y │ no-repeat │ inherit – экранда фондық суретті қайталайды. Үндеместен repeat мәні алынады. Мұндағы:

- repeat-x – фондық сурет көлделеңінен қайталанады;
- repeat-y – фондық сурет тігінен қайталанады;
- no-repeat-y – фондық сурет бір-ақ рет көрсетіледі, қайталанбайды;
- repeat – фондық сурет көнделеңінен де, тігінен де қайталанады;
- inherit – мәнді аналық қасиеттен алады.

Мысалы:

```
fon  {
background-color : "#e0e0e0";
background-position: 50% 50%;
background-image : url ("files/logo_fon.gif");
background-repeat : no-repeat
}
```

background-attachment: scroll | fixed – фонды мәтінге байланыстырады. Үндеместен scroll алынады. Мұндағы:

- scroll – егер мәтін терезеге симаса, онда фон мәтінмен қоса жүреді;
- fixed – фон орнында тұрады, ал мәтін оның үстінде қозғалады.

background: <background-color> ‖ <background-image> ‖ <background-repeat> ‖ <background-attachment> ‖ <background-position> – жоғарыда сипатталған қасиеттердің жалпыламасы, яғни фондық сурет қасиеттерін стильдер кестесінің бір жерінде анықтау мүмкіндігін береді. Қасиеттердің қабылдайтын мәндері жоғарыдағыдай. Егер кейбір қасиеттер көрсетілмесе, онда олар үндеместен алу принципі бойынша таңдалады.

Мысалы:

а)
```
fon  {
background-color : "#e0e0e0";
background-image : url ("files/logo_fon.gif");
background-repeat : no-repeat;
background-attachment: fixed;
background-position: right top;
}
```

б) fon {background: "#e0e0e0" url ("files/logo_fon.gif") no-repeat fixed right top}

Көрсетілген мысалдағы а) және б) варианттар эквивалент.

border-width: thin ‖ medium ‖ thick ‖ inherit – рамка бордюрінің 4 жағының қалындығын бірден анықтайды. Үндеместен medium алынады. Бордюр қалындығы ретінде төмендегі мәндер алынуы мүмкін:

- thin – жұқа рамка (әдетте 2 пиксель);
- medium – орташа рамка (әдетте 4 пиксель);
- thick – қалын рамка (әдетте 6 пиксель);
- inherit – мәнді аналық қасиеттен алады.

Мәндердің тізімінде бір, екі, үш немесе төрт мән кездесуі мүмкін. Егер бір мән кездессе, онда рамканың барлық төрт жағына бірдей әсер етеді.

Егер екі мән кездессе, онда рамканың тек қана жоғарғы және төменгі жақтарына әсер етеді. Егер үш мән кездессе, онда біріншісі рамканың жоғарғы, екіншісі сол және оң жақтарына, үшіншісі төменгі жақғына әсер етеді. Ал егер төрт мән кездессе, онда рамканың сәйкес жоғарғы, сол, оң және төменгі жақтарына әсер етеді.

Мысалы:

а) p { border-width: 4px}
б) p { border-width: thin thick}
в) p {border-width: 3px 2px 4px 1px}

Көрсетілген мысалдағы а) вариантта: рамка бордюрінің төрт жағының қалындығыда 4 пиксельден тұрады; б) вариантта: рамка бордюрінің жоғарғы жағы 2 пиксель, ал төменгі жағының қалындығы 6 пиксельден тұрады; в) вариантта: рамка бордюрінің жоғарғы жағы 3 пиксель, сол жағы 2 пиксель, оң жағы 4 пиксель, ал төменгі жағының қалындығы 1 пиксельден тұрады.

border-style: solid ‖ none ‖ hidden ‖ dotted ‖ dashed ‖ double ‖ groove ‖ ridge ‖ inset ‖ outset ‖ inherit - рамка бордюрінің 4 жағының стилін бірден анықтайды. Бордюр стилі ретінде төмендегі мәндер алынуы мүмкін:

- solid – қарапайым сызық;
- none – бордюр жоқ, оның қалындығы нольге тең;
- hidden – жасырын, бордюр жоқ;
- dotted – жіңішке пунктир сызық;
- dashed – пунктир сызық;
- double – екі еселенген сызық;
- groove – көлемді, қысылған сызық ;
- ridge – көлемді, қалындатылған сызық;
- inset – көлемді, қысылған сызықтың басқа варианты;
- outset – көлемді, қалындатылған сызықтың басқа варианты;
- inherit – мәнді аналық қасиеттен алады.

Мәндердің тізімінде бір, екі, үш немесе төрт мән кездесуі мүмкін. Олардың әсері жоғарыдағы «border-width» қасиетіндегі ретпен анықталады.

Мысалы:

а) p { border-style: none}
б) p { border-style: solid double solid }
в) p {border-style: ridge}

Көрсетілген мысалдағы а) вариантта: рамка бордюрі жоқ, ол көрінбейді; б) вариантта: рамка бордюрінің жоғарғы және төменгі жағы қарапайым сызықтан, ал сол мен оң жағы екі еселенген сызықтан тұрады; в) вариантта: рамка бордюрінің төрт жағы да көлемді, қалындатылған сызықтан тұрады.

border-color: <түс>│transparent│inherit - рамка бордюрінің түсін анықтайды. Түс CSS стильдер кестесінің синтаксисінде анықталған стандарт әдістермен беріледі. Бордюрдің түсін анықтаумен қатар, transparent арқалы бордюр асты түсін де (paddings) көрсетуге болады. Ал inherit мәні түсті аналық қасиеттен алады.

Түс мәндері тізіммен берілсе, тізімде бір, екі, үш немесе төрт мән кездесуі мүмкін. Олардың әсері жоғарыдағы «border-width» қасиетіндегі ретпен анықталады.

Мысалы:

```
p { border-color: red}          /* бордюр түсі қызыл */
```

border: <border-width> ‖ <border-style> ‖ <border-color> - қысқартылған қасиет, элементтің рамкасының бордюрінің бірнеше атрибуттарын, яғни енін, стилін және түсін бір қатарда анықтайды. Қасиет рамканың төрт жағына бірдей әсер етеді.

Мысалы:

а)
```
p {border: 2px double navy}
```

б)
```
p {
  border-top: 2px double navy;
  border-bottom: 2px double navy;
  border-left: 2px double navy;
  border-right: 2px double navy;
 }
```

Көрсетілген мысалдағы а) және б) варианттар эквивалент. Рамканың бордюрі төрт жағында бірдей: қалындығы 2 пиксель, стилі екі еселенген және түсі тоқ көк болып анықталады.

font-family: <шрифттер тізімі> – Шрифттер немесе олардың типтері үтірмен бөлектеніп көрсетіледі. Браузер шрифттерді көрсетілген ретпен компьютерде кездескенше іздеп, табылған шрифтті пайдаланады. Қасиеттің шрифт типтері түрінде төмендегі мәндер алынуы мүмкін:

- serif – түртпесі бар шрифттер, мысалы: Times;
- sans-serif – түртпесі жоқ шрифттер, мысалы: Helvetica;
- cursive – курсивті шрифттер, мысалы: Zarf-Chencery;
- fantasy – декоратив шрифттер, мысалы: Western;
- monospace – моношрифттер, мысалы Courier.

Егер шрифт типі бірнеше сөзден тұрса, онда шрифт типін тырнақша ішінде жазу қажет, мысалы: “Times New Roman Cyr”.

Егер бірнеше шрифт │ белгісімен көрсетілсе, браузер олардың біреуін үндеместен алу принципі бойынша таңдайды.

font-style: normal │italic│ oblique – шрифт стилдерін көрсетеді. Үндеместен normal алынады. Мұндағы:

- normal – қалыпты;
- italic – курсив;
- oblique – қиғаш.

font-size: <абсолют>│<салыстырмалы>│<нақты>│<пайыз> – шрифт өлшемін анықтайды. Қасиеттің мәні ретінде абсолют, салыстырмалы, нақты өлшемдер немесе пайыз көрсетіледі. Үндеместен normal алынады.

font-weight: normal│bold│bolder│lighter – шрифттің енін анықтайды. Үндеместен normal алынады. Мұндағы:

- normal – қалыпты;
- bold – жартылай жуан;
- bolder – жуан;
- lighter – жіңішке.

font-variant: normal │small-caps – шрифттің бас әріптерінің өлшемін анықтайды. Үндеместен normal алынады. Мұндағы:

- normal – қалыпты;
- small-caps – кіші бас қаріп.

font: <font-style>‖<font-variant>‖<font-weight>‖<font-size>‖<font-family> – жоғарыда сипатталған қасиеттердің жалпыламасы, яғни шрифт қасиеттерін стильдер кестесінің бір жерінде анықтау мүмкіндігін береді. Қасиеттердің қабылдайтын мәндері жоғарыдағыдай. Егер кейбір қасиеттер көрсетілмесе, онда олар үндеместен алу принципі бойынша таңдалады.

Мысалы:

```
h1 {
font-family: Arial;
font-size: 30 pt;
font-weight: bold;
font-style: italic
}
p {
font: bold italic large Times New Roman, serif
}
```

word-spacing: normal│<ұзындық> – сөздер арасындағы аралық беріледі. Үндеместен normal алынады.

letter-spacing: normal│<ұзындық> – символдар арасындағы аралық анықталады. Үндеместен normal алынады.

text-decoration: underline │overline │line-through │blink │none – Символ гарнитурасы көрсетіледі. Мұндағы:

- underline – астын сызу;
- overline – төбесін сызу;
- line-through – үстін сызу;
- blink – жанып-өшу;
- none – алдынғы орнатылған мәндерді жою.

text-transform: capitalize │uppercase │lowercase │none – мәтінді бейнелеуде регистрді түрлендіруді анықтайды. Үндеместен none алынады. Мұндағы:

- capitalize – әрбір сөздің бірінші қаріпі бас қаріпке түрленеді;
- uppercase – барлық әріптер бас қаріпке түрленеді;
- lowercase – барлық әріптер кіші қаріпке түрленеді;
- none – алдынғы орнатылған мәндерді жояды.

text-align: left │right │center │justify – мәтінді терезе шеттеріне теңестіреді. Мұндағы:

- left – солға;
- right – оңға;
- center – ортаға;
- justify – екі шеттен.

text-indent: <абсолют бірліктер> │<пайыз> – мәтіннің шегінуін көрсетеді. Әдетте шегіну абзацтың бірінші қатарының сол жағына ғана тиісті. Егер бірлік мәні сол таңбалы болса, онда салыстырмалы түрде солға қарай жылжыйды. Абсолют бірліктер төмендегідей анықталады:

- px – логикалық пиксельдер;
- pt – логикалық нүктелер;
- em – шрифт өлшемінің бірліктері;
- cm – сантиметрлер;
- mm – миллиметрлер.

Мысалы: text-indent: -3em.

vertical-align: baseline │sub │super │top │text-top │middle │bottom │text-botton │<пайыз> – элементтің тік орналысуын анықтайды. Үндеместен baseline алынады. Мұндағы:

- baseline – элементті негізгі элементтен төменге орналастырады;
- middle – элементтің орта сызығын «негізгі + жартылай биіктікке» теңестіреді;
- sub – элементті қатар астына теңестіреді (әдетте font-size: smaller қасиетімен бірге беріледі);
- super – элементті қатар төбесіне теңестіреді (әдетте font-size: smaller қасиетімен бірге беріледі);

- text-top – элементің жоғарғы шетін негізгі элементтің төбесімен теңестіреді;
- text-bottom – элементтің төменгі шетін негізгі элементтің астымен теңестіреді;
- top – элементтің төбесін негізгі элементтің ең жоғарғы шетімен теңестіреді;
- bottom – элементтің астын негізгі элементтің ең төменгі шетімен теңестіреді.

line-height: normal | <сан> | <ұзындық> | <пайыз> – қатар биіктігін көрсетеді. Үндеместен normal алынады.

color:<түс> – құжат мәтінінің түсін көрсетеді. Үндеместен құжатта анықталған түсті алады. Түстің мәні бірнеше тәсілмен анықталады:

- түс аттары бойынша, мысалы: red (қызыл), blue (көк);
- он алтылық сан түрінде, мысалы: #000000 (қара);
- нөлдері алып тасталған он алтылық сан түрінде, мысалы: #fff (ақ);
- RGB моделі бойынша, мәндері 0-ден 255-дейін, мысалы: rgb(255,255,255) (ақ);
- RGB моделі бойынша пайызбен берілген, мысалы: rgb(100%, 0%, 0%) (қызыл).

Мысалы:

```
h1 {color: green}          /* түс аттары бойынша берілген (жасыл) */
h1 {color: #008000}        /* түс он алтылық сан түрінде берілген (жасыл) */
```

4.3 CSS стильдерін Web-құжатқа енгізу әдістері

CSS стильдер кестесін Web-құжатқа енгізудің бірнеше әдістері белгілі. Олар бір-бірінен мағынасы және мүмкіндіктерімен бөлектенеді. Бұл бөлімде осы әдістерді қарастырамыз.

Браузерлер стильдер тізбегін олардың орындалу приоритетіне қарай төмендегіше қолданады [4]:

- ішкі стильдер (құжатқа <style> атрибуттарымен енген стильдер);
- енгізілген стильдер (құжатқа <body> тегімен енген стильдер);
- глобаль стильдер (құжатқа тақырып бөлімінде <style> тегімен енген стильдер);
- байланысқан стильдер - құжатқа <link> тегімен енген стильдер;

- импорт стильдер - құжатқа <style> тегінің @import қасиетімен енген стильдер.

Браузерлер Internet Explorer (Microsoft), Opera (Opera Software ASA), Mozilla FireFox (Mozilla Corporation) Web-құжаттарды талдау жасағанда стильдерді осы приоритет бойынша таңдап алады. Сондықтан серверде Web-жоба үшін тек қана бір CSS стильдер кестесі құрылады.

Ішкі стильдер. CSS стильдер кестесі технологиясы HTML тілінің кез келген тегіне өзіндік стильді визуалды (inline style) енгізе алады. Ішкі стиль мағынасы бойынша HTML тілі тэгінің кеңейтілген түрі болып есептеледі. Стильді анықтау үшін HTML тілі тэгінің «style» атрибутынан пайдаланады. Сонымен қатар ішкі стиль тек қана осы тэг ішінде ғана орындалады.

Мысалы:

```
<!doctype html public"-//W3C//Dtd HTML 4.01//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
<title>Ішкі стиль</title>
</head>
<body>
<p style="font-family: Helvetica, shadow; font-size: 14 pt; color: purple;
font-weight: normal;"> мәтін </p>
</body>
</html>
```

Көрсетілген мысалда стиль <p> тэгінің «style» атрибуты көмегімен енгізілген. Стильдің қасиеттері нүктелі үтірмен бөлектенген. Мәтіннің өлшемі 14, шрифты Helvetica, түсі сия көк болып анықталған.

Жалпы, ішкі стильдерден пайдаланбаған немесе шектелген түрде қолданған дұрыс. Себебі олар Web-құжаттардың жалпы көлемін ұлғайтады. Нәтижеде браузердің Web-құжатты жүктеу уақыты ұзаяды және Web-құжатқа өзгеріс енгізу қиындайды.

Енгізілген стильдер. CSS стильдер кестесі Web-құжаттың негізгі бөлімінде енгізілген стильден пайдаланады. Бұл әдісте стиль HTML-дің <body> тегіне орналасады, сондықтан бұл стильден барлық тэгтер CSS стильдер кестесінің мұрагерлік қасиеті бойынша пайдалана алады.

Мысалы:

```
<!doctype html public"-//W3C//Dtd HTML 4.01//EN">
<html>
<head>
```

```
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
<title>Енгізілген стиль</title>
</head>
<body style="font-family: "Times New Roman",serif;
font-size: 14 pt; color: blue;">
<p> мәтін </p>
</body>
</html>
```

Көрсетілген мысалда Web-құжаттың барлық мәтінінің өлшемі 14, шрифты Times New Roman, түсі көк болып анықталған. Сонымен қатар Times New Roman шрифты Serif эффектімен берілген. Егер тұтынушы компьютерінде бұл шрифт болмаса, онда оны браузер Serif эффекті бар сәйкес шрифтпен алмастырады.

Глобаль стильдер. Бұл әдісте стильдер кестесі Web-құжаттың тақырыбына жазылады. Мағынасы жағынан енгізілген стильге ұқсас. Стильді анықтауда HTML тілі тэгінің «style» атрибутынан емес, <style> тэгінен пайдаланады. Егер Web-құжатта ішкі стиль қолданылса, онда сол тэг CSS стильдер кестесінің полиморфизм қасиетімен сипатталған ішкі стильді алады.

Мысалы:

```
<!doctype html public"-//W3C//Dtd HTML 4.01//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
<title>Глобаль стиль</title>
<style type="text/css">
h1 {
font-family: Helvetica, shadow;
font-size: 14 pt;
color: yellow;
font-weight: normal;
}
</style>
</head>
<body>
<h1> мәтін </h1>
</body>
</html>
```

Көрсетілген мысалда Web-құжат мәтінінің өлшемі 14, шрифты Helvetica, түсі сары болып анықталған. Сонымен қатар Helvetica шрифты shadow эффектімен берілген.

CSS стильдер кестесі технологиясында аталған әдістерді тәуелсіз қолдануға да, сонымен қатар бірге қолдануға да болады. Web-құжатта бірнеше стильді қолдану мәселесін көрейік. Бұл үшін стильдер кестесін құрамыз.

```
<!doctype html public"-//W3C//Dtd HTML 4.01//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
<title>Стильдерді қатар қолдану</title>
<style type="text/css">
p {
background-image: url("files/fon1.gif");
background-color:"#e0e0e0";
font-family: "Times New Roman", shadow;
font-size: 12 pt;
color: "#000080";
}
</style>
</head>
<body style="font-family: " Arial",serif;
font-size: 16 pt; colour: blue;">
<p style="font-family: Helvetica, shadow; font-size: 12 pt; color: purple;
font-weight: normal;"> мәтін </p>
</body>
</html>
```

Көрсетілген мысалда Web-құжаттың фоны глобаль стильдер әдісімен, Web-құжат енгізілген стильдер әдісімен, ал абзац ішкі стильдер әдісімен берілген. Нақты атасақ, негізгі мәтін 16 өлшемді Times New Roman шрифтімен, бас тақырып 16 өлшемді Arial шрифтімен, ал абзац 12 өлшемді Helvetica шрифтімен берілген. Сондай-ақ, негізгі шрифттің түсі тоқ көк, тақырыптың түсі көк, ал абзац сия көкпен берілген.

Байланысқан стильдер. Бұл стильдің жоғарыдағы әдістерден негізгі айырмашылығы стильдер кестесі Web-құжатқа емес, бөлек стильдік файлға жазылады. Стильдік файл «.css» файл кеңеймесімен сақталады. Әдетте бұл файлды түзетуде қарапайым ASCII-редактордан (мысалы, Блокнот) пайдаланған дұрыс.

Мысалы:

```
/* ukgu.css стильдік файлы */
```

```
negizgi
{
font-family: "Times New Roman", serif;
font-size: 12 pt;
color: green;
}
fon
{
background-image: url ("files/fon1.gif");
border-top: 2px double navy;
border-bottom: 2px double navy;
border-left: 2px double navy;
border-right: 2px double navy;
background-color:"#e0e0e0";
font-family: "Times New Roman", shadow;
font-size: 12 pt;
color: "#000080";
}
abzatch1
{
font-family: Arial, sans-serif;
font-size: 18 pt;
color: red;
font-weight: bold;
font-style: italic;
}
abzatch2
{
font-family: Helvetica, shadow;
font-size: 14 pt;
color: yellow;
font-weight: normal;
}
tizim
{
font-family: "Times New Roman", shadow;
font-size: 12 pt;
color: "#000080";
}
```

Көрсетілген мысалда «negizgi», «fon», «abzatch1», «abzatch2», «tizim» стильдері анықталған. Байқасаңыз <style> тегі стильдер кестесінде көрсетілмейді. Браузер стильдік файлды «.css» файл кеңеймесіне қарап таниді. Web-құжатты стильдер файлымен төмендегі түрде байланыстыру қажет:

```
<!doctype html public"-//W3C//Dtd HTML 4.01//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
<title>Байланысқан стильдер</title>
<link rel="style sheet" type = "text/css" href="ukgu.css">
</head>
<body>
<p> мәтін </p>
</body>
</html>
```

Көрсетілген мысалда стильдер кестесін (rel="stylesheet") файл форматымен (type="text/CSS") ағымдағы бумадан (href="ukgu.CSS") байланыстыру көрсетілген. Жалпы, стильдік файлды кез-келген URL-адресте сақтауға болады.

Байланысқан стильдер әдісінде стильдер кестесі CSS синтаксисінен басқа ешқандай дерек сақтамайды. Сондай-ақ Web-құжатта да ешқандай стильдер көрсетілмейді, тек қана стильдік файлға сілтеме жасалады. Нәтижеде Web-құжат коды мен Web-құжатты рәсімдеу бөлектенеді. Сонымен қатар жалғыз стильдер файлын бірнеше Web-құжатпен бір уақытта байланыстыруға болады. Сондықтан байланысқан стильдер әдісі CSS технологиясын қолданудың әмбебап әдісінің бірі болып есептелуде.

Импорт стильдер. Web-құжаттағы ағымдық стильдерге стильдік файлды енгізу үшін <style> тегінің @import қасиетінен пайдаланады. Бұл әдісті байланысқан немесе глобаль стильдермен бірге қолдануға болады, ал ішкі стильдермен қатар қолданудың қажеті жоқ.

Web-құжатты стильдер файлымен төмендегі түрде байланыстыру қажет:

```
<!doctype html public"-//W3C//Dtd HTML 4.01//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
<title>Импорт стильдер</title>
<style>
@import url("files/matin.css");
@import url("files/tizim.css");
</style>
</head>
<body>
```

 мәтін
</body>
</html>

Көрсетілген мысалда «matin.css» және «tizim.css» стильдік файлдары ағымдағы құжатқа «files/» бумасынан енгізіледі.

4.4 CSS технологиясының алгоритмдері

Web-жоба құру үшін HTML тілімен танысу бірінші кезең болып табылады. HTML тілінде Web-жоба құрылғанымен, HTML тілі тек қана гипермәтіндерді белгілеу тілі, яғни тілде Web-құжатты форматтау және рәсімдеу құралдары жеткіліксіз. Келесі кезең CSS стильдер кестесін құру және енгізу болып есептеледі. Сондықтан бұл бөлімде CSS стильдер кестесін құру алгоритмдерін қарастырамыз. Тұтынушы HTML тілі тэгтерін біледі деп есептейміз.

Қарапайым Web-құжат құрайық (сурет 18).

<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title> CSS технологиясының қасиеттері</title>
</head>
<body>
<h1 align=center> CSS технологиясының қасиеттері</h1>
 Web-қ ұ жаттың қ ұрылымы мен сипаттауын бөлектеуді CSS (Cascade Style Sheets – каскадты стильдер кестесі) технологиясын қолдана отырып орындаймыз. CSS технологиясы Web-қ ұ жатты сипаттауда қ осымша файлдардан, шаблондар мен стильдерден пайдалануға көмектеседі.
 Стильді анықтау құжаттың түрін стиль қасиеттерін енгізу арқалы өзгертетеді.
Стиль қасиеттерінің мәні ретінде кілттік сөздер алынады. Кілттік сөздер арасында кездесетін ¦ символы, қасиеттің осы немесе келесі мәнді қабылдайтынын білдіреді. Төменде осы технологияның негізгі қасиеттерін қарастырайық.
</body>

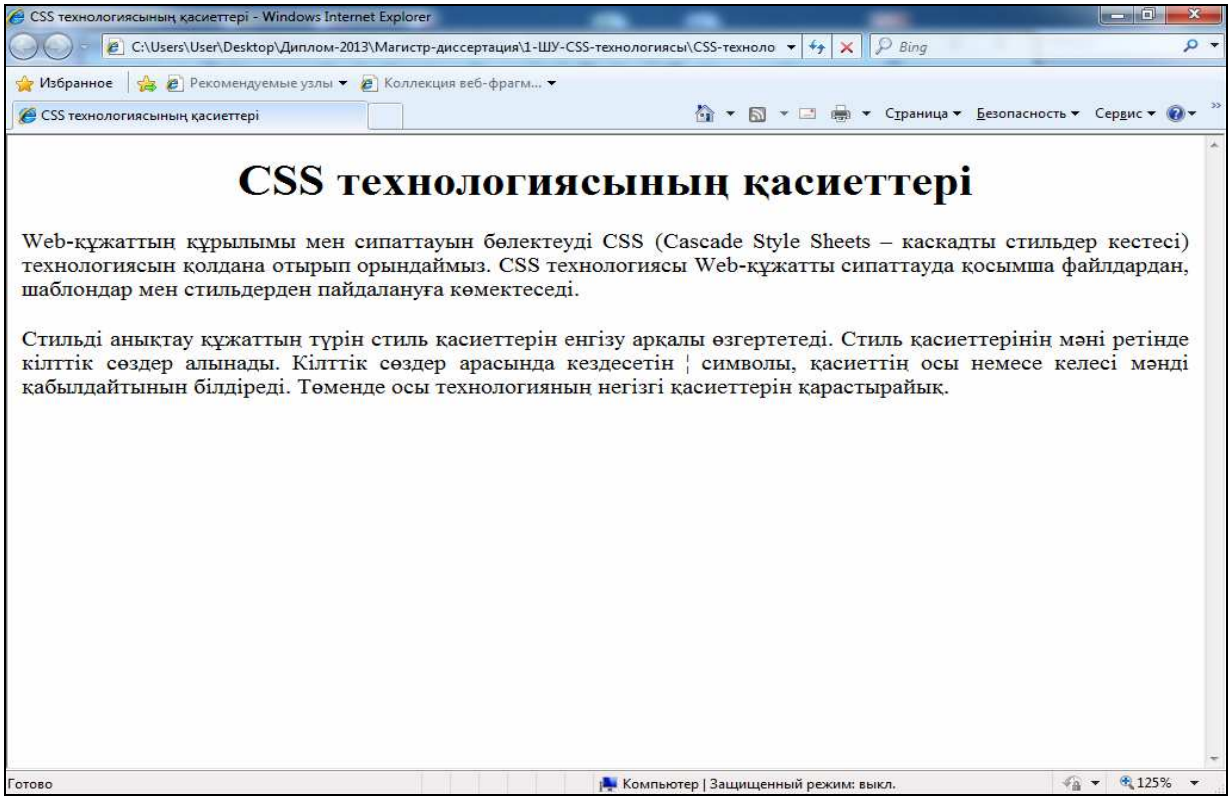
</html>

Web-құжатқа стильдер кестесін енгізу үшін HTML тілінің «style» тэгінен пайдаланамыз.

Мысалы:

```
<style type="text/css">
h1 {
font-family: Arial;
font-size: 30 pt;
color: blue;
font-style: italic
}
</style>
```

CSS синтаксисі бойынша стиль екі негізгі бөлімнен тұрады: h1 – селектор және сипаттау (font-family: Arial; font-size: 30 pt; color: blue; font-style: italic).



Сурет 18 - Қарапайым Web-құжат

Сонымен, Web-құжат коды төмендегі түрде жазылады.

```
<!doctype html public"-//W3C//Dtd HTML 4.01//EN">
<html>
<head>
```

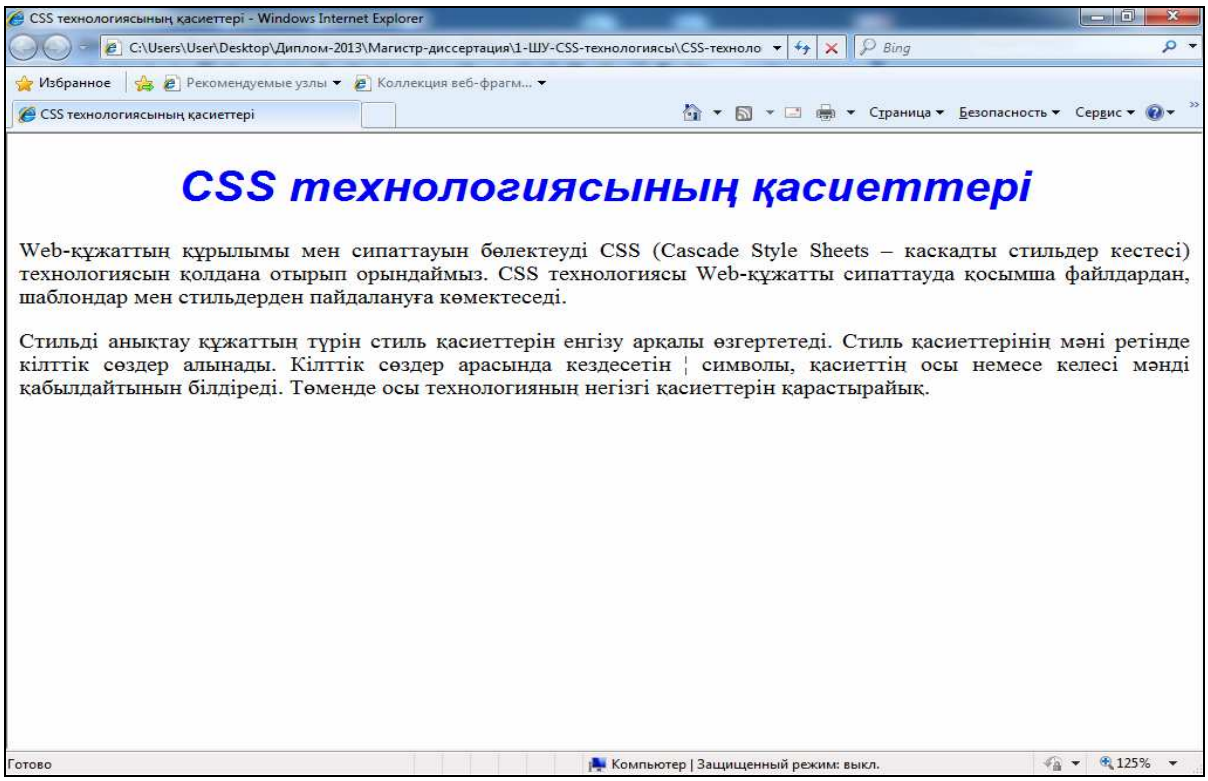
```
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title> CSS технологиясыны&#1187; &#1179;асиеттері</title>
<style type="text/css">
h1 { font-family: Arial; font-size: 30 pt; color: blue; font-style: italic }
</style>
</head>
<body>
<h1 align=center> CSS технологиясыны&#1187; &#1179;асиеттері</h1>
<p align="justify" > Web-&#1179;&#1201;жатты&#1187; &#1179; &#1201; рылымы мен сипаттауын б&#1257;лектеуді CSS (Cascade Style Sheets – каскадты стильдер кестесі) технологиясын &#1179;олдана отырып орындаймыз. CSS технологиясы Web-&#1179;&#1201;жатты сипаттауда &#1179;осымша файлдардан, шаблондар мен стильдерден пайдалануға к&#1257;мектеседі.</p>
<p align="justify"> Стильді аны&#1179;тау &#1179;&#1201;жатты&#1187; т&#1199;рін стиль &#1179;асиеттерін енгізу ар&#1179;алы &#1257;згертетеді. Стиль &#1179;асиеттеріні&#1187; м&#1241;ні ретінде кілттік с&#1257;здер алынады. Кілттік с&#1257;здер арасында кездесетін &#166; символы, &#1179;асиетті&#1187; осы немесе келесі м&#1241;нді &#1179;абылдайтынын білдіреді. Т&#1257;менде осы технологияны&#1187; негізгі &#1179;асиеттерін &#1179;арастырайы&#1179;.
</p>
</body>
</html>
```

Көрсетілген мысалда h1 стиль «style» тэгі көмегімен енгізілген. Стильдің қасиеттері нүктелі үтірмен бөлектенген. Мәтіннің өлшемі 30pt, шрифты Arial, түсі көк, стили қиғаш болып анықталған (сурет 19). Стильдер кестесінде стильдер блокка жинақталған. Блок фигуралық жақшалардың {...} ішіне жазылған. Сондай-ақ әрбір сипаттау нүктелі үтірмен аяқталады. Мысалда фигуралық жақша жабылудан алдын нүктелі үтір қойылмаған, жалпы, нүктелі үтір жазылуы тиіс.

Жоғарыда Web-құжатқа енгізілген ішкі стиль тек қана бір тэгке ғана әсер етеді. Сонымен қатар, ол басқа стильдермен қосылып қаралады. Стильді анықтау үшін HTML тілі тэгінің «style» атрибутынан да пайдалануға болады.

Мысалы, жоғарыдағы Web-құжатты бірінші абзацына <p> тэгінің «style» атрибутымен жаңа стиль енгізейік.

```
<p align="justify"; style="font-family: Helvetica, shadow; font-size: 14 pt;
    color: purple; font-weight: normal;">
```



Сурет 19 – Стильдер «style» тэгімен енгізілген Web-құжат

Көрсетілген мысалда стиль <p> тэгінің «style» атрибуты көмегімен енгізілген. Стильдің қасиеттері нүктелі үтірмен бөлектенген. Мәтіннің өлшемі 14, шрифты Helvetica, түсі сия көк болып анықталған.

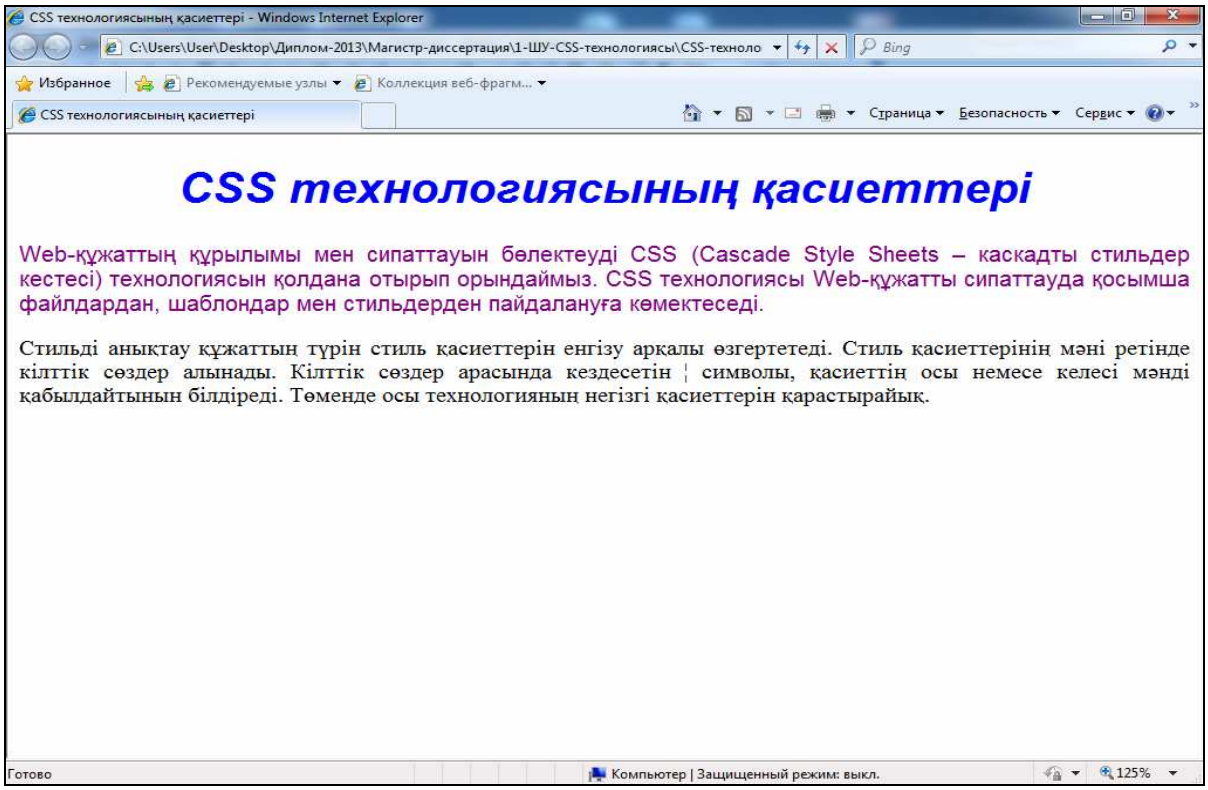
Осы мысалдың толық коды төмендегі түрде жазылады (сурет 20).

```
<!doctype html public"-//W3C//Dtd HTML 4.01//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
<title> CSS технологиясыны&#1187; &#1179;асиеттері
</title>
<style type="text/css">
h1 {
font-family: Arial;
font-size: 30 pt;
color: blue;
font-style: italic
}
</style>
</head>
<body>
```

<h1 align=center> CSS технологиясының қасиеттері</h1>
<p align="justify"; style="font-family: Helvetica, shadow; font-size: 14 pt; color: purple; font-weight: normal;"> Web-құжаттың қ ұ рылымы мен сипаттауын бөлектеуді CSS (Cascade Style Sheets – каскадты стильдер кестесі) технологиясын қолдана отырып орындаймыз. CSS технологиясы Web-құжатты сипаттауда қосымша файлдардан, шаблондар мен стильдерден пайдалануға көмектеседі.

 Стильді анықтау құжаттың түрін стиль қасиеттерін енгізу арқалыөзгертетеді. Стиль қасиеттерінің мәні ретінде кілттік сөздер алынады. Кілттік сөздер арасында кездесетін ¦ символы, қасиеттің осы немесе келесі мәнді қабылдайтынын білдіреді. Төменде осы технологияның негізгі қасиеттерін қарастырайық.

</body>
</html>



Сурет 20 - Стильдер тэгінің «style» атрибутымен енгізілген Web-құжат

CSS технологиясында стильдер мұрагерлік және полиморфизм қасиеттерін сақтайды. Жалпы, CSS технологиясында байланысқан стильді қолданған дұрыс. Бұл стильдің жоғарыдағы әдістерден негізгі

айырмашылығы стильдер кестесі Web-құжатқа емес, бөлек стильдік файлға жазылады. Стильдік файл «.css» файл кеңеймесімен сақталады.

Мысалы: style.css файлы

```
h1
    {
    font-family: Arial;
    font-size: 30 pt;
    color: blue;
    font-style: italic
    }
p
    {
    font-family: Helvetica, shadow;
    font-size: 14 pt;
    color: purple;
    font-weight: normal
    }
```

Байқасаңыз <style> тегі стильдер кестесінде көрсетілмейді. Браузер стильдік файлды «.css» файл кеңеймесіне қарап таниді. Web-құжатты стильдер файлымен байланыстыру үшін HTML тілінің <link> тэгінен пайдаланамыз.

Мысалы:

```
<link type="text/css" rel="stylesheet" href="style.css">
```

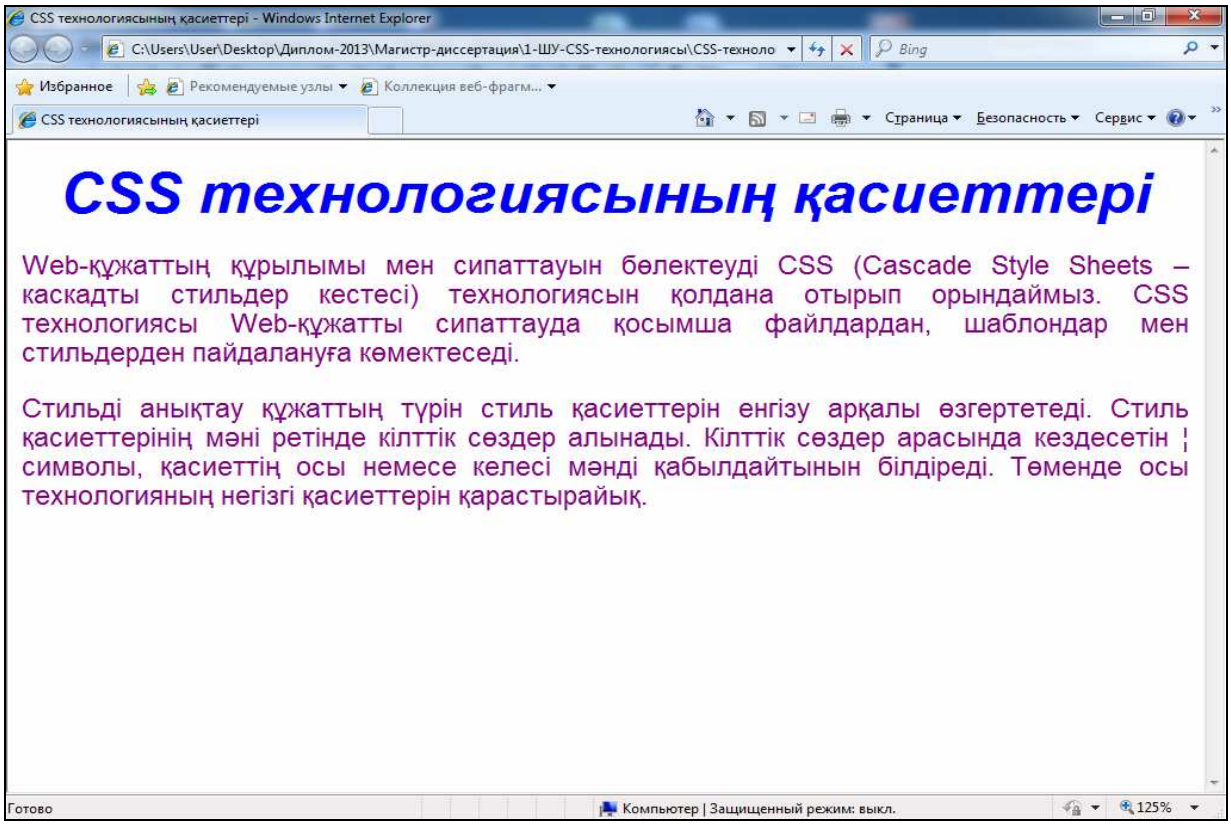
Көрсетілген мысалда стильдер кестесін (rel="stylesheet") файл форматымен (type="text/CSS") ағымдағы бумадағы стильдік файлмен (href="style.css") байланыстыру көрсетілген.

Web-құжаттың толық коды төмендегі түрде жазылады (сурет 21).

```
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title> CSS технологиясыны&#1187; &#1179;асиеттері</title>
<link type="text/css" rel="stylesheet" href="style.css">
</head>
<body>
<h1 align=center> CSS технологиясыны&#1187; &#1179;асиеттері</h1>
<p align="justify">Web-&#1179;&#1201;жатты&#1187; &#1179;&#1201;рылымы мен сипаттауын б&#1257;лектеуді CSS (Cascade Style Sheets – каскадты стильдер кестесі) технологиясын &#1179;олдана отырып орындаймыз. CSS технологиясы Web-&#1179;&#1201;жатты
```

сипаттауда қосымша файлдардан, шаблондар мен стильдерден пайдалануға көмектеседі.
 Стильді анықтау құжаттың түрін стиль қасиеттерін енгізу арқалы өзгертетеді.
Стиль қасиеттерінің мәні ретінде кілттік сөздер алынады. Кілттік сөздер арасында кездесетін ¦ символы, қасиеттің осы немесе келесі мәнді қабылдайтынын білдіреді. Төменде осы технологияның негізгі қасиеттерін қарастырайық.

</body>
</html>



Сурет 21 - Стильдер style.css файлымен енгізілген Web-құжат

5 JavaScript тіліне кіріспе

5.1 JavaScript тілінің қарапайым элементтері

JavaScript бағдарламалау тілін 1995 жылы Брендан Айх (Sun Microsystems) жасап шығарды. Тілдің синтаксисисін жазуда C++ және Java тілдері пайдаланылған. Бұл тіл алғашқыда Netscape Navigator браузеріне арналған болатын. Қазіргі таңда JavaScript тілін барлық браузерлер қолданады.

JavaScript тілі – динамикалық Web-құжаттарды дайындауға арналған объектті-оқиғалы тіл. Бұл тіл негізінен клиенттік компьютерде орындалатын скрипт құруға арналған.

Скрипт (Script) – HTML тілінің мүмкіндіктерін кеңейтуге арналған, арнайы бағдарламалау тілінде жазылған ықшам бағдарлама. Кейбір әдебиеттерде сценарий термині де кездеседі, яғни бұл екі сөз синоним.

Скрипттер HTML-құжаттың HTML тілінде жазылған кодының ішіне тікелей орналасады. Олар компиляцияны қажет етпейді, яғни компилятордың орнына интерпретатор жеткілікті. Бұл бағдарлама құруды жеңілдетіп, уақытты үнемдейді. Клиенттік скрипттердің негізгі қасиеті, олар клиент компьютерінде, яғни браузерде HTML-кодпен бірге орындалады.

Біз төменде JavaScript тілін қысқаша қарастырамыз [13-16].

Алфавит. JavaScript тілінің алфавиті төмендегі белгілерден тұрады:

- 26 латын алфавиті әріптері: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z.
- 10 араб цифрлары: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- арнайы белгілер:

+	қосу	~	тильда
-	алу	!	леп белгісі
*	жұлдызша	@	коммерциялық эт
/	қисық сызық	#	нөмір белгізі
=	тең	$	ақша бірлігі
,	үтір	%	пайыз
.	нүкте	^	көрсеткіш
:	қос нүкте	&	амперсанд
;	нүктелі үтір	|	тік таяқша
>	үлкен	\	кері қисық сызық
<	кіші	()	жақша
?	сұрақ белгісі	[]	тік жақша
_	сызықша	{}	фигуралық жақша
␣	бос орын		
‘	апостроф		
“	тырнақша		

JavaScript интерпретаторы регистрды есепке алады, яғни бас және кіші әріптерді бөлек деп қабылдайды. Сонымен қатар орыс немесе қазақ алфавиті әріптерін тек қана мәтін қатарында қолдануға болады.

Алфавит белгілерінен қызметші сөз, идентификатор, литерал, оператор т.с.с. тіл элементтері құрылады.

Қызметші сөздер деп JavaScript тілінің жұмысына қажетті нұсқауларды айтады. Әрбір сөздің өз мағынасы, атқаратын қызметі бар. Қызметші сөздерді жұмыс барысында біртіндеп мағынасы қарай түсіндіріп отырамыз.

Идентификатор деп әріп немесе _(сызықша) белгісінен басталатын әріп, цифр және _ белгілерінің тізбегін айтады. Идентификатор - литерал, оператор, функция сияқты қарапайым элементтерді атауға қажет. Идентификаторды жазуда бас және кіші әріптер әртүрлі деп есептеледі. Сонымен қатар идентификатор ретінде қызметші сөздерді қолдануға болмайды. Мысалы:

Дұрыс жазылған идентификаторлар:

- Java_Script – сызықша орынды қолданылған;
- _a1 – сызықшадан басталады;
- X – латын әрібі қолданылған.

Қате жазылған идентификаторлар:

- Java-Script –басқа белгі қолданылған;
- For – қызметші сөз;
- 1a – цифрден басталған;
- Скрипт –орыс әріптері қолданылған.

Литерал деп әдетте JavaScript тілінде қолданылатын деректерді айтады. Литерал *тұрақты* және *айнымалы* деп екіге бөлінеді. Тұрақты литерал деп скриптте мәні өзгермейтін деректі айтамыз. Ал мәні өзгеретін деректі айнымалы литерал дейміз. Литералдың атауы идентификатормен беріледі.

Деректердің типтері. JavaScript тілінде қолданылатын деректер типтерін қарастырайық:

- бүтін;
- нақты;
- логикалық;
- қатарлық;
- null.

Бүтін типті деректерге бүтін сандар жатады. Сандар ондық, сегіздік және он алтылық санақ жүйесінде берілуі мүмкін. Сегіздік санақ жүйесіндегі сандардың алдына 0 (нөл) таңбасы, ал он алтылық сандардың алдына 0x таңбасы жазылады. Мысалы: -100, 0175, 0xA1.

Нақты типті сандар екі түрде жазылады:

- тұрақты нүктелі;
- жылжымалы нүктелі.

Тұрақты нүктелі сандар, нүктемен бөлектенген бүтін және бөлшек түрінде жазылады. Санда бөлшек бөлім болмаса да, нүкте қойылады. Мысалы: 3.1415, -100.

Жылжымалы нүктелі сандар, E (үлкен әріп) символымен бөлектенген мантисса және дәреже түрінде жазылады. Мысалы: 1E-5, 3.14E3.

Логикалық типті деректер тек қана true (ақиқат, 1) false (жалған, 0) мәндерін ғана қабылдайды.

Қатарлық типке мәтін жатады. Мәтін апостроф (‘) немесе қос тырнақша (“) ішіне жазылады. Мысалы: “TurboPascal тіліне ұқсас”.

JavaScript тілінде қатарлық типке арнайы қызметші және баспаға шықпайтын символдарды енгізуге болады.

\b	бос орын (backspace)
\f	жаңа бетке өту (PageDown)
\n	жаңа қатарға өту
\r	енгізу (enter)
\t	табуляция (tab)
\’	апостроф
\”	тырнақша
\\	кері қисық сызық (backslash)

Мысалы:

my_file	-идентификатор;
12345	-бүтін сан;
012345	-сегіздік бүтін сан;
0x12345	-он алтылық бүтін сан;
2.71828	-тұрақты нүктелі нақты сан;
21.	-тұрақты нүктелі нақты сан (бөлшек бөлігі жоқ);
3.14E5	-жылжымалы нүктелі нақты сан ($3{,}14 \cdot 10^5$);
1.2E-4	-жылжымалы нүктелі нақты сан ($1{,}2 \cdot 10^{-4}$);
“Сауле”	-қатар.

Регуляр өрнектер. Регуляр өрнек деп қатардан берілген символдар тізбегін іздеуге арналған шаблондарды айтады. Олардың қарапайым іздеу функцияларынан айырмашылығы, регуляр өрнектің іздеу операциясының күрделілігінде. Регуляр өрнектер / (слэш) символдары ішіне жазылады.

Мысалы: file*.txt - шаблоны атауы file сөзінен басталатын барлық txt кеңеймесі бар файлдарды іздейді. Осы шаблонды регуляр өрнек түрінде жазайық.

file*.txt → / [fF] ile . * \. txt /

Жазылған регуляр өрнек [fF] - f немесе F символдарынан басталатын, ile – тізбегін және .* - кез келген символдар тізбегін, \. – нүктені, txt – тізбегін іздеу мағынасын береді.

Регуляр өрнектерде кездесетін литералдарды қарастырайық:

\	арнайы символдарға жатады, келесі символ литерал емес деген мағынаны білдіреді, мысалы: \n –қатардың соңы, жаңа қатарға өту. Сонымен қатар регуляр өрнектерде литералды символ түрінде жазуда қолданылады, мысалы: \(-регуляр өрнекте жақша кездеседі.
^	қатардың басы
$	қатардың соңы
*	алдыңғы символ көп кездеседі немесе кездеспейді
+	алдыңғы символ міндетті түрде бір рет немесе одан көп кездеседі
?	алдыңғы символ тек қана бір рет кездеседі немесе кездеспейді
{n}	алдыңғы символ міндетті түрде n -рет кездеседі
\w	әріп, цифра немесе _ (сызықша белгісі)

Мысалдар:

/ (\w+) @ ([\ w \ . _] +) / - кез-келген электрондық поштаның адресін білдіреді. Адресті тұтынушы аты және сервердің атауы ретінде екіге бөледі.

/ (\w + : \ / \ /) ? ([^ /] +) (. *) ? / -кез-келген HTML-құжаттың URL-адресін білдіреді. Адресті хаттама, домен және файл аты деп үш бөлікке бөліп, жедел жадыда сақтайды. Сонымен қатар, хаттама мен файл атын көрсетпесе де болады.

RegExp сыныбы қатарды регуляр өрнектер көмегімен өңдеуде қолданылады. Жалпы форматы:

RegExp ({регуляр өрнек} [, {флагтар}])

бұл жерде: регуляр өрнек міндетті компонент;

флагтар: g -глобальды іздеу;

i -символдар регистрін есептемеу;

m -көп қатарлы іздеу.

5.2 JavaScript тіліндегі амалдар

JavaScript тілінде амалдар (операторлар) төмендегі түрлерге бөлінеді:

- арифметикалық;
- логикалық;
- салыстыру;
- биттік (екілік);
- меншіктеу;
- қатарлық;

Арифметикалық амалдар. Бүтін және нақты типті деректер үстінде төмендегі арифметикалық амалдар орындалады:

+	қосу
-	алу
/	бөлу
*	көбейту
-	инверсия – таңбаны өзгерту
%	арифметикалық модуль – бүтін бөлу амалының қалдығы
++	инкремент – операнд мәнін бір бірлікке арттыру
--	декремент - операнд мәнін бір бірлікке азайту

Мысалы:c = 2 * a + b; - бірнеше арифметикалық амал қолданылған;
c = c++; - инкремент амалы операндтан кейін жазылған;
c = ++c, - инкремент амалы операндтан алдын жазылған;
x = -a; - инверсия амалы;

Арифметикалық амалдағы нәтиженің типі, амалда қатысатын деректердің типімен анықталады.

Логикалық амалдар. Логикалық амалдар тек қана логикалық *true* және *false* мәндері үстінде орындалады. Логикалық теріске шығару амалы бір аргументті, ал қалған амалдар екі аргументті.

!	теріске шығару
&&	конъюнкция (логикалық көбейту)
||	дизъюнкция (логикалық қосу)

a	b	!a	!b	a && b	a || b
false	false	true	true	false	false
false	true	true	false	false	true
true	false	false	true	false	true
true	true	false	false	true	true

Мысалы: (x <= 1) && (x >= 7) амалдарын нәтижесін табайық.

- x = 0 болса, нәтиже *false;*
- x = 5 болса, нәтиже *true;*
- x = 9 болса, нәтиже *false;*

Салыстыру амалдары. Деректер үстінде төмендегі салыстыру амалдары орындалады:

>	үлкен
>=	үлкен немесе тең
<	кіші
<=	кіші немесе тең
=	тең
!=	тең емес
= = =	қатаң түрде тең
!= =	қатаң түрде тең емес

Салыстыру амалының нәтижесі *true* немесе *false* мәндеріне ие. Бұл амалдарда деректер кез келген типке, ал нәтиже тек қана логикалық типке тиісті.

Мысалы: *a = 5; b = 3; болсын, онда*
a >= b - true;
a = b - false болады.

Егер салыстырылатын деректер әр түрлі типке ие болса, онда қатаң түрде тең (тең емес) амалдарын қолданған жөн.

Биттік амалдар. JavaScript тілінде биттік амалдар төменгі деңгейлі бағдарламалауда қолданылады.

~	инверсия – теріске шығару
&	конъюнкция – логикалық көбейту
|	дизъюнкция – логикалық қосу
^	логикалық алу
<<	солға жылжыту
>>	оңға жылжыту
>>>	оңға таңбасыз жылжыту

Мысалы:
```
c = ~a;          - инверсия амалы;
c = a & b;       - конъюнкция амалы;
c = a << b;      - a-ны солға b-разрядқа жылжыту;
c = a >> b;      - a-ны оңға b-разрядқа жылжыту;
c = a >>> b;     - a-ны оңға b-разрядқа жылжыту (үлкен
                   разрядқа нөлдер жазылады).
```

Биттік амалдардың мәндері:

a	b	~a	~b	a & b	a | b	a ^ b
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	1	0

Меншіктеу амалы (=) литералға мән беруде қолданылады. Мән ретінде тұрақты немесе өрнек жазылады. Мысалы: a = 1; a = a++;

JavaScript тілінде меншіктеу амалының қысқартылған түрлері кездеседі. Мысалы: a = a +10; және a += 10; өрнектерінің мәні бірдей.

жазылуы	*мағынасы*
a += b;	a = a + b;
a -= b;	a = a - b;
a *= b;	a = a * b;
a /= b;	a = a / b;
a &= b;	a = a & b;
a >>= b;	a = a >> b;

Қатарлық амалдар. Қатарлық амалға конкатенация (+) амалы жатады. Бұл амал қатарларды қосуда қолданылады.

Мысалы: a = ”Java”; b = ”Script”; c = a + b;

Өрнектер - деректер үстінде амалдарды және олардың орындалу тәртібін анықтайды. JavaScript тілінде өрнек литералдар және амалдар тізбегінен тұрады. Өрнектер арифметикалық және логикалық болып екіге бөлінеді.

Арифметикалық өрнектер - арифметикалық амалдармен бөлектелген тұрақтылар, айнымалылар және стандарт функциялар тізбегінен тұрады. Өрнектің нәтижесінің типі деректердің типімен анықталады.

Мысалы: 2*a + b*x + c

Логикалық өрнектер – логикалық деректер, логикалық және салыстыру амалдары тізбегінен тұрады. Салыстыру амалдарында арифметикалық, логикалық және символдық типті деректер кездеседі. Логикалық амалдардың нәтижесі true және false мәндеріне ие.

Мысалы: (x > 0) && (x < 13)

Typeof амалы өрнектің типін анықтауда қолданылады. Егер тип бүтін немесе нақты болса «number», қатарлық болса «string» және логикалық болса «boolean» мәндерін қабылдайды. Егер өрнектің типі анықталмаған болса «undefined» мәні шығады.

Мысалы: t = typeof (b);

Өрнектердегі амалдардың орындалу тәртібі амалдар приоритетімен анықталады. Амалдар приоритетін жақшалармен өзгертуге болады. Өйткені жақша ішіндегі амалдар бірінші орындалады. Өрнектегі приоритеті бірдей амалдар солдан оңға қарай орындалады.

приоритет	*аталуы*	*амалдар*
I	унарлық	-, ++, --, ~
II	мультипликатив	*, /, %, &&, &
III	аддитив	+, -, ||, |
IV	салыстыру	=, >, <, >=, =, !=, = = =, != =
V	меншіктеу	=, +=, *=, …

Өрнектерді құруда төмендегі ережелерді орындау керек:

- өрнекті бір қатарға жазу;
- тек қана қарапайым жақшалардан пайдалану;
- қатарынан екі амалды жазбау.

JavaScript тілінде бірнеше өрнекті жинақтап жазуға болады. Жинақталған өрнектер фигуралық жақшалар {} ішіне жазылады.

Мысалы:

```
{
i = i++;
a += i;
}
```

Мысалы:

```
{
i = ++i;
a += i;
}
```

JavaScript тіліндегі бағдарлама кодына түсініктеме төмендегідей жазылады:

```
// түсініктеме
/* түсініктеме */
```

5.3 Шартты операторлар

Шартты оператор деректерді өңдемейді, өңдеуді басқарумен айналысады. JavaScript тілінде шартты оператор төмендегі түрлерге бөлінеді:

- if-else тармақталу операторы;
- ? шартты операторы;
- switch таңдау операторы.

if-else тармақталу операторы логикалық шарттың мәніне қарай бағдарлама кодының бір бөлігін орындайды. Операторлың форматы:

a) if шарт өрнек1 else өрнек2;

b) if шарт өрнек1;

бұл жерде: if, else –қызметші сөздер; шарт –логикалық өрнек;

Егер шарттың мәні true болса, онда өрнек1 орындалады, ал кері жағдайда, яғни шарттың мәні false болса, онда өрнек2 орындалады.

Мысал 1:
```
if (x>0 && x<13)
f = 1
else
f = 0;
```

Мысал 2:
```
if str1==”Нурдаулет” str2=”Сәлем Нурдаулет”;
```

Мысал 3:
```
if (x==0)
{
f  = 11;
k = 22;
}
else
{
f = 33;
k = 44;
}
```

Мысал 4:
```
if (x>=1)
{
f  = 5;
k = 6;
}
else
{
```

```
f = 7;
k = 8;
}
```

? шартты операторы логикалық шарттың мәніне қарай орындалады. Операторды форматы:

```
шарт ? өрнек1 : өрнек2;
```

бұл жерде: ? –қызметші сөз; шарт –логикалық өрнек;

Егер шарттың мәні true болса, онда өрнек1 орындалады, ал кері жағдайда, яғни шарттың мәні false болса, онда өрнек2 орындалады.

Мысалы: `f = (x>0 && x<13) ? 1 : 0;`

switch таңдау операторы бірнеше тармақталу операторларын біріктіру мүмкіндігін береді. Операторды форматы:

```
switch (өрнек)
{
case мән 1  :  оператор 1;
break;
case мән 2  :  оператор 2;
break;
case мән 3  :  оператор 3;
break;
case мән 4  :  оператор 4;
break;
- - - - - - - - - - - - - - - - -
case мән n  :  оператор n;
break;
default  :  оператор;
}
```

Операторды орындалуы switch қызметші сөзінен кейінгі өрнекті өңдеуден басталады. Сонан соң өрнектің мәні case қызметші сөздерінен кейінгі мәндермен салыстырылады. Қай мән сәйкес келсе, сол қатардағы оператор орындалады. Егер ешқандай мән сәйкес келмесе, онда default қатарындағы оператор орындалады.

Мысалы: берілген day айнымалының мәні, егер 1-ге тең болса, онда out айнымалысы ”дүйсенбі” мәнін қабылдайды, ал егер 2-ге тең болса, онда out айнымалысы ”сейсенбі” мәнін қабылдайды, т.с.с. Ал егер day айнымалының мәні басқа сан болса, онда out айнымалысы ”апта күні емес” мәнін қабылдайды.

```
switch (day)
```

```
{
case 1  :  out = ”дүйсенбі”;
break;
case 2  :  out = ”сейсенбі”;
break;
case 3  :  out = ”сәрсенбі”;
break;
case 4  :  out = ”бейсенбі”;
break;
case 5  :  out = ”жұма”;
break;
case 6  :  out = ”сенбі”;
break;
case 7  :  out = ”жексенбі”;
break;
default  :  out = ”апта күні емес”;
}
```

Таңдау операторындағы default қызметші сөзін жазбауға болады. Мұндай жағдайда, switch қызметші сөзінен кейінгі өрнектің мәні case қызметші сөздерінен кейінгі мәндердің ешқайсысына сәйкес келмесе, ешқандай іс-әрекет орындалмайды.

Сол сияқты, break қызметші сөзін де жазбауға болады. Мұндайда, switch қызметші сөзінен кейінгі өрнектің мәні сәйкес келген case қызметші сөзінен кейінгі операторлардан бастап, барлығы орындалады.

Мысалы:

```
switch (day)
{
case 1  :  out = ”дүйсенбі”;
case 2  :  out = ”сейсенбі”;
case 3  :  out = ”сәрсенбі”;
case 4  :  out = ”бейсенбі”;
case 5  :  out = ”жұма”;
case 6  :  out = ”сенбі”;
case 7  :  out = ”жексенбі”;
default  :  out = ”апта күні емес”;
}
```

Бұл операторлардың нәтижесінде out айнымалысы тек қана ”апта күні емес” мәнін қабылдайды. Себебі day айнымалының мәні апта күніне сәйкес келседе, соңында default қызметші сөзінен кейінгі опреатор орындалады. Сондықтан break қызметші сөзін жазған дұрыс.

5.4 Цикл операторлары

Цикл операторы бағдарлама кодының кейбір бөлігін бірнеше рет қайта-қайта орындауды ұйымдастырады. JavaScript тілінде цикл операторы төмендегі түрлерге бөлінеді:

- for - цикл операторы;
- do-while цикл операторы;
- while цикл операторы;
- break және continue операторлары.

for цикл операторы қайталаулар саны белгілі болғанда қолданылады. Цикл бағдарлама кодындағы логикалық шарт true мәнін қабылдаған кезде орындалады. Оператордың форматы:

for (a, v, b) {<цикл денесі>}

мұндағы: for –қызметші сөз;
a -есептеуішті инициалдау;
v –логикалық шарт;
b -өрнекгі инкременттеу немесе декременттеу.

Бұл оператор бағдарлама кодының бірнеше рет қайта-қайта орындауын ұйымдастыру үшін қажет. Сондықтан кодтың неше рет орындалғанын анықтайтын бүтін типті айнымалы *есептеуіш* деп аталады. Ал, бағдарлама кодының бірнеше рет қайта-қайта орындалушы бөлігін *цикл денесі* атайды. Цикл денесі логикалық шарт false мәнін қабылдағанша орындалады.

Мысалы: S = 1 + 2 + … + 10.

а) for (i = 1; i <= 10; i ++) S += i;

б) for (i = 10; i > 0; i --) S += i;

Көрсетілген мысалда есептеуіш мәнін өзгерту үшін а)-да инкремент және б)-да декремент амалы қолданылған.

Мысалы:

```
for ( i = 2;  i < 21;  i += 2)
{
a += 3;
b = 2 * i + 1;
}
```

Көрсетілген мысалда есептеуіш мәні 2, 4, 6, ..., 20 мәндерін қабылдайды, ал цикл 10 рет орындалады.

do-while цикл операторы қайталаулар саны логикалық шартқа байланысты болған жағдайда қолданылады. Бұл оператор for операторына ұқсас, тек мұнда есептеуіш жоқ. Операторды форматы:

```
do
    цикл денесі
while (шарт);
```

мұндағы: do, while –қызметші сөздер; шарт –логикалық өрнек.

Цикл бағдарлама кодындағы логикалық шарт true мәнін қабылдаған кезде орындалады. Логикалық шарт цикл соңында орналасқан, сондықтан цикл шарт тексерілгенше бір рет орындалады.

Мысалы:

```
do
{
a = a * i + 2;
i = i ++;
}
while (i < 100);
```

do-while цикл операторын қайталаулар саны белгілі болғанда да қолдануға болады.

Мысалы: S = 1 + 2 + … + 10.

```
var  i = 1; S = 0;
do
{
S += i;
i = i ++;
}
while (i > 10);
```

while цикл операторы do-while операторына ұқсас, тек қана логикалық шарт цикл басында тексеріледі. Операторды форматы:

```
while (шарт);
    цикл денесі
```

мұндағы: while –қызметші сөз; шарт –логикалық өрнек.

Цикл логикалық шарт true мәнін қабылдаған кезде орындалады. Егер логикалық шарт басынан-ақ false мәнін қабылдаса, онда цикл міндетті түрде орындалмайды.

Мысалы:

```
while (i < 100)
```

```
{
a = a * i + 2;
i = i ++;
}
```

do-while цикл операторын қайталаулар саны белгілі болғанда да қолданса болады.

Мысалы: S = 1 + 2 + … + 10.

```
var  i = 1; S = 0;
do
{
S += i;
i = i ++;
}
while (i > 10);
```

break және continue операторлары. Кейде циклді орындалу барысында тоқтату қажеттілігі туындайды. Мұндай жағдайда JavaScript тілінде break және continue операторларынан пайдаланамыз.

break операторы циклдың орындалуын тоқтатып, басқаруды келесі операторға немесе белгісі бар операторға жібереді. Операторлың форматы:

```
break [белгі];
```

мұндағы: break –қызметші сөз; белгі –идентификатор.

Белгі – бұл бағдарлама кодында кездесетін басқа бір операторға қойылатын идентификатор. Басқару белгі орналасқан операторға жіберіледі. break операторында белгіні қолданбаса да болады.

Мысалы:

```
while ( a < 100 )
{
a = a * i + 2;
if ( a > 50) break;
i = ++ i;
}
```

Көрсетілген мысалда, егер а-ның мәні 50-ден көп болса, цикл тоқтайды (бағдарлама орындала береді).

continue операторы циклдың орындалуын тоқтатып, басқаруды цикл басына немесе белгі орналасқан операторға жібереді. Операторлың форматы:

```
continue [белгі];
```

мұндағы: continue –қызметші сөз; белгі –идентификатор.

Мысалы:

```
while ( a < 100 )
{
i = ++ i;
if ( i > 9 && i < 21) continue;
a = a * i + 2;
}
```

Көрсетілген мысалда, a-ның мәні цикл есептеуіші i-дің 10 мен 20 мәндері аралығында есептелмейді.

JavaScript тілінде *қабаттасқан циклден* де пайдалануға болады. Қабаттасқан цикл сыртқы және ішкі цикл болып бөлінеді. Ішкі цикл сыртқы циклдің ішіне орналасады. Сыртқы цикл мен ішкі цикл қиылыспайды, сыртқы циклдің әр бір қадамында ішкі цикл толық орындалады.

Мысалы:

```
for ( i = 1; i <= 10; i++ )
{
for ( j = 1; j <= 10; j++ )
s = s + i + j;
}
```

Мысалы: Қабаттасқан циклда continue операторын қолданайық.

```
outer:
for ( i = 1; i <= 10; i++ )
{
  inner:
  for ( j = 1; j <= 10; j++ )
  {
    if ( j = =2 ) continue outer;
    else
    s = s + i + j;
  }
}
```

Көрсетілген мысалда, j айнымалысының мәні 2-ге тең болса, басқару қайтадан сыртқы циклге жіберіледі. Егер continue операторында белгіні қолданбасақ, онда басқару қайтадан ішкі циклге жіберіледі.

6 JavaScript-дің алгоритмдік мәселелері

6.1 Web-құжатқа скриптті орналыстыру әдістері

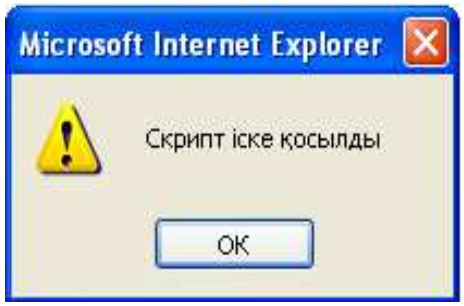
1. *Скрипт коды Web-құжаттың ішінде орналасқан.* Web-құжаттың тақырып бөліміне коды орналасқан қарапайым скриптті жүктеуді қарастырайық. Бағдарлама листингі мен скрипт коды:

```
<html>
<head>
<script language='JavaScript'>
function f1()
     {
      alert("Скрипт іске қосылды");
      }
</script>
</head>
<body>
       <input type="button"  value="Скрипт"  onclick="f1()">
</body>
</html>
```

Скрипт жазылған Web-құжатта төмендегі батырма пайда болады.



Батырмасы бассақ төмендегі хабарлама шығады (сурет 22).



Сурет 22 – Скрипт іске қосылғаны туралы хабарлама

Берілген скриптті талдайық. Скрипт коды мен оқиғаны өңдеу батырмасы бір құжатта берілген. Скриптті іске қосу үшін батырма (button) объекті және батырманы басу (onclick) оқиғасы пайдаланылған. Батырманы басқанда f1() функциясы жұмысқа қосылады. Бұл функция өз кезегінде alert() функциясын шақырып, терезеге мәтінді («Скрипт іске қосылды») шығарады. Скрипт коды Web-құжат бағдарламасынан <script> және </script> тэгтері арқылы бөлектенген.

Сипатталған скрипт кодын Web-құжаттың кез келген жеріне орналыстыруға болады. Скриптті ашу тэгінде скрипт тілі (language='JavaScript') көрсетілуі қажет.

2. *Скрипт коды бөлек файлда орналасқан.* Скрипт коды file.js файлына жазылған болсын:

```
function f1()
{
alert("Скрипт іске қосылды");
}
```

Скриптке баратын жол мен файл атын scr атрибутында көрсетеміз. Скрипт жазылған файлды Web-құжатқа төмендегіше байланыстырамыз.

```
<script language='JavaScript' scr=”file.js”> </script>
```

Скриптті жұмысқа қосу үшін батырмадан пайдаланамыз.

```
<input type="button"  value="Скрипт"  onclick="f1()">
```

Сонымен Web-құжат коды төмендегідей жазылады:

```
<html>
<head>
   <script language='JavaScript' src='file.js'> </script>
</head>
<body>
   <input type="button"  value="Скрипт"  onclick="f1()">
</body>
</html>
```

Скрипт жазылған Web-құжатта төмендегі батырма пайда болады.



Батырмасы бассақ жоғарыда көрсетілген хабарлама шығады (сурет 22).

6.2 Интерактивтік интерфейстің қалып скрипттері

Әдетте тұтынушының Web-құжатпен интерактивтік байланысын әртүрлі қалыптар арқылы, яғни батырма, радиоперне, қосқыш (байрақша), тізім арқылы ұйымдастырады [5]. Мұндай қалыптар, мысалы, студенттердің білімін тест арқылы тексеруде ыңғайлы. Осындай

қалыптарға мысалдар қарастырайық. Мынадай қарапайым мысалға скрипт құрайық [4].

Сұрақ: «Гамлет» трагедиясын кім жазған?
Жауаптары: Монтень, Шекспир, Гомер, Рабле, Софокл
Дұрыс жауабы: Шекспир

Скрипт кодын құру үшін Web-құжаттың негізгі бөліміне сұрақты және <p> тэгтері арқылы жазамыз:

```
<center><p> «Гамлет» трагедиясын кім жазган? </p></center>
```

Енді Web-құжаттың негізгі бөліміне жауаптарды жазамыз. Мысалы дұрыс жауапты

```
<input type=Button value="Шекспир" onclick="javascript:dyris();">
```

тэгімен жазамыз. Мұндағы <input> тэгі скриптты қалай іске қосуды анықтайды. Тзгтің type="button" атрибуты батырманы, value="Шекспир" атрибуты шығатын жауапты сипаттайды. Ал onclick="javascript:dyris();" атрибуты тышқан көрсеткішін шерткенде dyris() функциясын шақырады.

Енді қате жауапты

```
<input type=Button   value="Гомер"   onclick="javascript:kate();">
```

тэгімен анықтаймыз. Мұндағы onclick="javascript:kate();" атрибуты тышқан көрсеткішін шерткенде kate() функциясын шақырады. Сол сияқты қалған қате жауаптарды ды анықтаймыз. Ол үшін value="Гомер" атрибутының мәндерін Монтень, Рабле, Софокл деп кезекпе-кезек өзгертіп отырамыз. Бізде мынадай программа коды пайда болды:

```
<input type=Button value="Шекспир" onclick="javascript:dyris();">
<input type=Button   value="Гомер"   onclick="javascript:kate();">
<input type=Button   value="Монтень"   onclick="javascript:kate();">
<input type=Button   value="Рабле"   onclick="javascript:kate();">
<input type=Button   value="Софокл"   onclick="javascript:kate();">
```

Енді жауаптар браузер экранында бір тегіс шығу үшін, осы кодты кестеге енгіземіз. Ол үшін <table> тэгінен пайдаланамыз:

```
<table align="center" cellspacing="3" cellpadding="3" border="2">
```

Мұндағы align="center" атрибуты ұяшық мәтінін браузер терезесіне туралайды, cellspacing="3" атрибуты кестенің жалпы рамкасы мен әр бір ұяшығының рамкасы арасындағы қашықтықты пиксельде анықтайды,

cellpadding="2" атрибуты кестенің әр бір ұяшығындағы мәтіннің ұяшық рамкасына дейінгі қашықтықты пиксельде анықтайды, border="2" атрибуты кесте рамкасының қалындығын пиксельде анықтайды.

Сол сияқты <tr> және <td> тэгтерінен пайдаланымыз. Мұндағы <tr> тэгі кестенің ұяшықтар қатарын анықтайды, ал <td> тэгі кестенің ағымдағы қатарында ұяшық құрады. Осы жоғарыда сипатталған программа кодын <form> тэгінің ішіне жазамыз.

Сонымен программа кодымыз төмендегі түрде жазылады.

```
<form name="Test1">
<table align="center" cellspacing="3" cellpadding="3" border="2">
<tr> <td> Жауаптар </td>
<td></td>
<td><input type=Button   value="Монтень"
onclick="javascript:kate();"></td>
<td></td>
<td><input type=Button value="Шекспир"
onclick="javascript:dyris();"></td>
<td></td>
<td><input type=Button   value="Гомер"
onclick="javascript:kate();"></td>
<td></td>
<td><input type=Button   value="Рабле"
onclick="javascript:kate();"></td>
<td></td>
<td><input type=Button   value="Софокл"
onclick="javascript:kate();"></td>
</tr> </table>
</form>
```

Әрі қарай dyris() функциясы кодын жазамыз.

```
function dyris()
{
alert("дурыс");
}
```

Бұл функция келесі alert() функциясын шақырып, «дұрыс» деген мәтінді браузер терезесіне шығарады. Сол сияқты kate() функциясының кодын жазамыз.

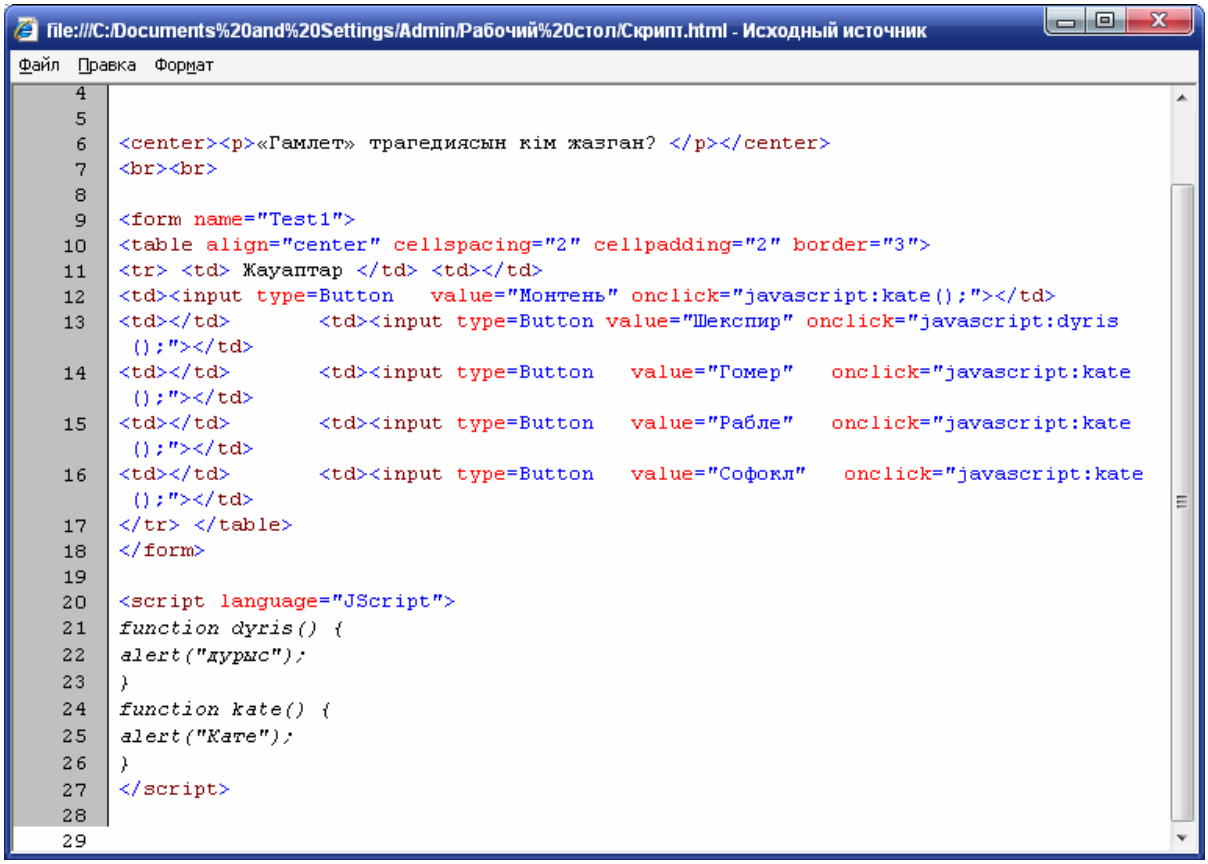
```
function kate()
{
alert("Қате");
}
```

Бұл функция да alert() функциясын шақырып, «Қате» деген мәтінді браузер терезесіне шығарады.

Сонымен скрипттің толық коды мына түрге келеді.

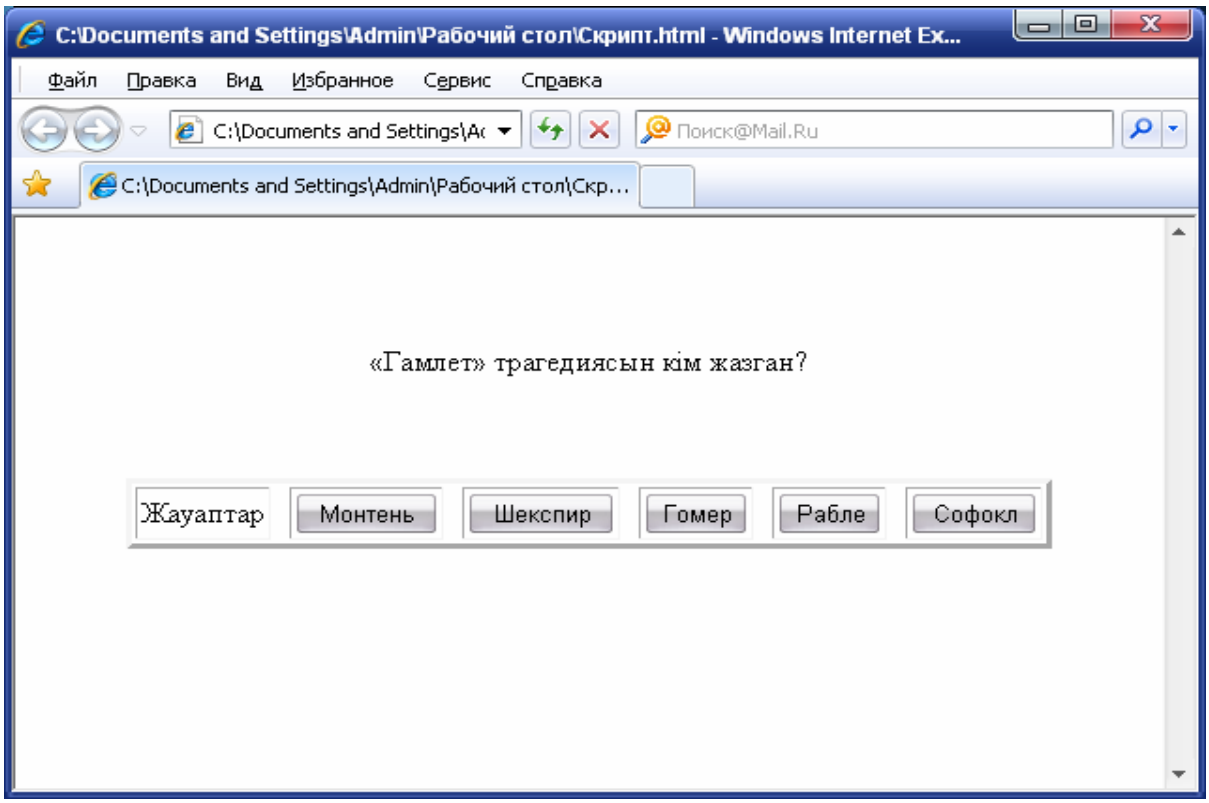
```
<script language="JavaScript">
function dyris()
{
alert("дурыс");
}

function kate()
{
alert("Кате");
}
</script>
```

Жоғарыда сипатталған программа кодтарын жинақтап, Test1.html атауымен сақтаймыз (сурет 23).



Сурет 23 – Test1.html файлының коды

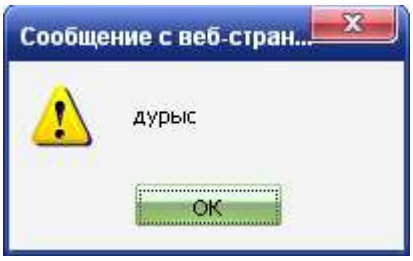
Осы Test1.html файлы браузер терезесінде төмендегі түрде бейнеленеді (сурет 24).



Сурет 24 – Test1.html файлының браузер терезесіндегі бейнесі

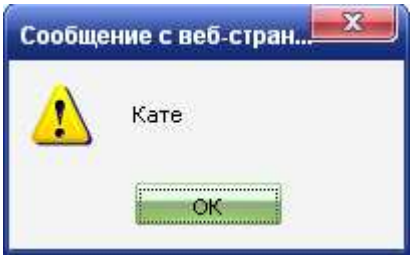
Егер «Шекспир» батырмасын тышқан көрсеткішімен бір рет шертсек, программа dyris() функциясын шақырады. Браузер терезесінде «дұрыс» сөзі жазылған қосымша терезе пайда болады. Терезені жабу үшін «ok» пернесін шертеміз (сурет 25).

Егер «Монтень», «Гомер», «Рабле», «Софокл» батырмасының біреуін тышқан көрсеткішімен бір рет шертсек, программа kate() функциясын шақырады.



Сурет 25 – dyris() функциясының нәтижесі

Браузер терезесінде «қате» сөзі жазылған қосымша терезе пайда болады (сурет 26).



Сурет 26 – kate() функциясының нәтижесі

Әрі қарай, жоғарыда қаралған мысалды радиоперне қалыбы арқылы құру алгоритмін сипаттайық. Қалыптың атын «Test2» деп атаймыз:

<form name="Test2">

Скрипт кодын құру үшін Web-құжаттың негізгі бөліміне сұрақты жоғарыда сияқты анықтаймыз.

 «Гамлет» трагедиясын кім жазган?

Енді Web-құжаттың негізгі бөліміне жауаптарды жазамыз.

<input type=radio value="1" name="jayap"

Мұндағы <input> тэгі қалыптың өрісін анықтайды. Тзгтің type=radio атрибуты радиопернелер тізімі өрісін, value="1" атрибуты радиоперне атауын, name="jayap" атрибуты өрістің өзіндік атын анықтайды. Тізімдегі барлық радиопернелер (radio buttons) бірдей аталады, олардың тек қана біреуі белгіленеді. Таңдалған өрістің name/value мәні анықталады.

Сол сияқты басқа жауаптарды да дәл осылай <input> тэгі арқалы сипатталады. Оларда тек қана value атрибутының мәні ғана өзгеріп отырады.

Әрі қарай скриптты іске қосуды

<input type=button value="Тексеру" name="Go" onclick="javascript:Rbut();">

тэгімен жазамыз. Тзгтің type="button" атрибуты батырманы, value="Тексеру" атрибуты өрістің атын, батырма үстіндегі жазуды, ал name="Go" атрибуты өрістің өзіндік атын анықтайды анықтайды. Ал onclick="javascript:Rbut();" атрибуты тышқан көрсеткішін шерткенде Rbut() функциясын шақырады.

Енді жауаптар браузер экранында бір тегіс шығу үшін, осы кодты кестеге енгіземіз. Ол үшін <table> тэгінен пайдаланамыз.

Сонымен, біздің қалыптың коды мынадай түрге келеді:

<form name="Test2">

```
<table align="center" cellspacing="2" cellpadding="2" border="2">
<tr>
<td>ауыстыргыш</td>
<td>жауаптар</td>
</tr>
<tr align="center"> <td><input type=radio  value="1" name="jayap"></td>
<td> Монтень </td></tr>
<tr align="center"> <td><input type=radio value="2" name="jayap"></td> <td>
Шекспир </td> </tr>
<tr align="center"> <td><input type=radio value="3" name="jayap"></td> <td>
Гомер </td> </tr>
<tr align="center"> <td><input type=radio value="4" name="jayap"></td> <td>
Рабле </td> </tr>
<tr align="center"> <td><input type=radio value="5" name="jayap"></td> <td>
Софокл </td> </tr>
 <tr align="center"> <td colspan="2"> <input type=button value="Тексеру"
name="Go" onclick="javascript:Rbut();"></td> </tr>
</table>
</form>
```

Әрі қарай Rbut() функциясының кодын жазамыз.

```
function Rbut()
{
  jayap=document.Test2.jayap[1].checked;
  if (jayap)
{
  alert("дұрыс жауап");
}
  else
{
  alert("қате жауап ");
}
}
```

Мұндағы jayap=document.Test2.jayap[1].checked қатары "Test2" қалыбынан jayap[1] мәнін радиопернелерден алып, оны jayap айнымалысына меншіктейді. Айта кетейік, jayap[1] айнымалысының аргументін номерлеу 0-ден басталады. Ал if операторы jayap айнымалысының мәні true болса, alert("дұрыс") функциясын, ал кері жағдайда, яғни jayap айнымалысының мәні false болса, онда alert("Қате") функциясын шақырады.

Жоғарыда сипатталған программа кодтарын жинақтап, Test2.html атауымен сақтаймыз (сурет 27).

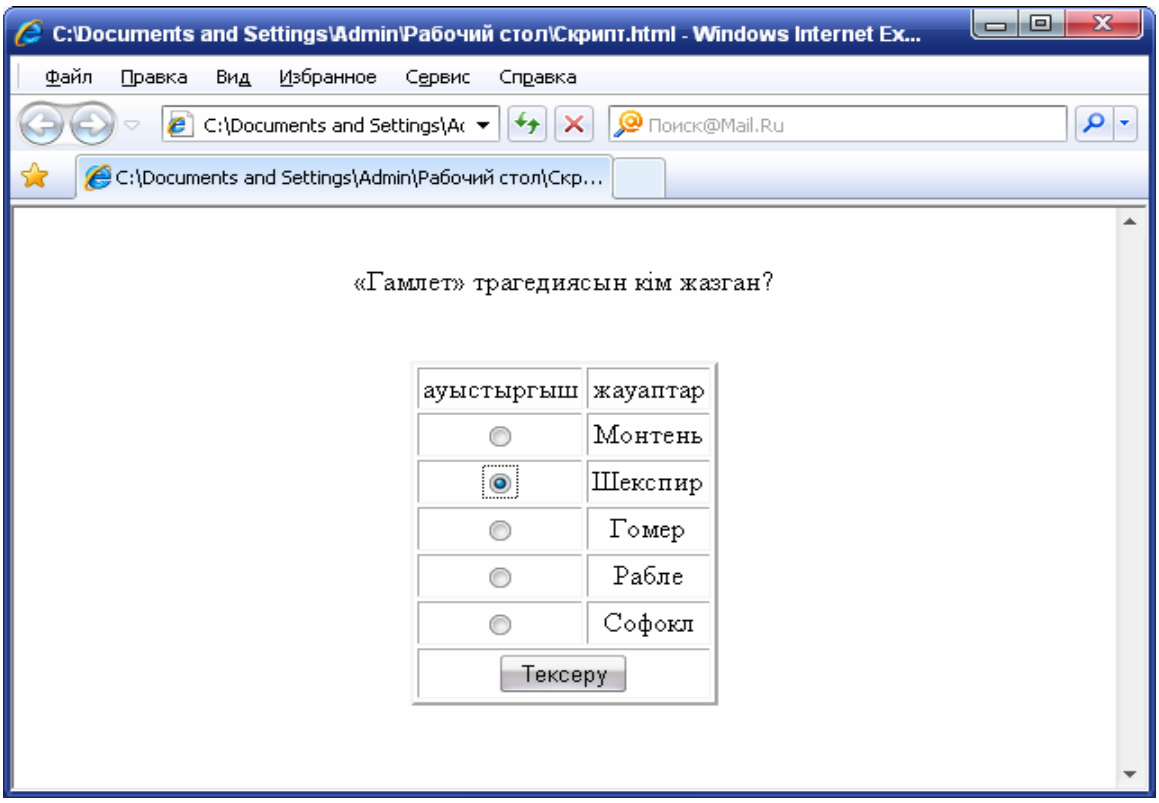


Сурет 27 – Test2.html файлының коды

Сонымен скрипттің толық коды мыны түрге келеді:

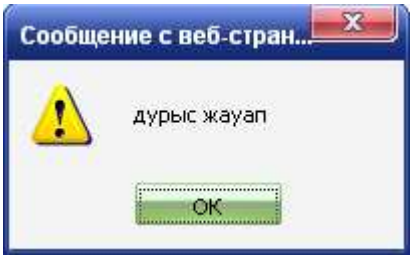
```
<script language="JavaScript">
function Rbut()
{
 jayap=document.Test2.jayap[1].checked;
 if (jayap)
{
 alert("дұрыс жауап");
}
 else
{
 alert("қате жауап ");
}
}
</script>
```

Осы Test2.html файлы браузер терезесінде төмендегі түрде бейнеленеді (сурет 28).



Сурет 28 – Test2.html файлының браузер терезесіндегі бейнесі

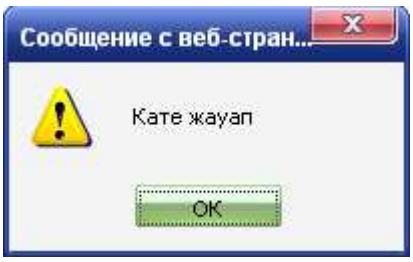
Егер «Шекспир» радиопернесін белгілеп, «Тексеру» батырмасын тышқан көрсеткішімен бір рет шертсек, программа alert("дұрыс жауап ") функциясын шақырады. Браузер терезесінде «дұрыс жауап» сөзі жазылған қосымша терезе пайда болады (сурет 29).



Сурет 29 – alert("дұрыс жауап ") функциясының нәтижесі

Егер «Монтень», «Гомер», «Рабле», «Софокл» радиопернелерінің бірін белгілеп, «Тексеру» батырмасын тышқан көрсеткішімен бір рет шертсек, программа alert("қате жауап") функциясын шақырады. Браузер терезесінде «қате жауап» сөзі жазылған қосымша терезе пайда болады (сурет 30).

Көрсетілген мысалда, тек қана бір сұрақ үшін алгоритм құрылды. Осы алгоритм арқалы сұрақтар санын көбейтуге болады.



Сурет 30 – alert("қате жауап") функциясының нәтижесі

Жоғарыда айтылғандарды қорытындылайық. Сонымен бұл бөлімде тұтынушының интерактивтік интерфейсін, скрипттерді Web-құжатқа енгізу әдістері және интерактивтік интерфейстің қалыптарын құру мәселесін қарастырдық. Сондай-ақ, тұтынушының Web-құжатпен интерактивтік байланысын батырма, радиоперне арқылы ұйымдастыру алгоритмдерін сипаттадық.

6.3 Web-құжатқа ақпаратты енгізу-шығару скрипттері

JavaScript тілінде бағдарламалау барысында ақпаратты енгізу және шығару мәселесі көп туындайды. JavaScript бағдарламалау тілінде ақпаратты енгізу және шығару басқа алгоритмдік тілдерге қарағанда ерекше. Себебі скрипттер Web-құжатта әрдайым орындала бермейді. Олар тек қана тұтынушы шақырғанда ғана жұмыс істейді. Сондықтан ақпаратты енгізу және шығару интерактивті түрде амалға асырылады. Ақпаратты енгізуде және шығаруда әр түрлі әдістер қолданылады. Осы әдістердің жұмыс үрдістерін қарастырайық [13-16].

1. Ақпаратты confirm() әдісімен енгізу. Егер Web-құжатта тек қана екі мәнді, мысалы «ия» және «жоқ» деген, енгізу қажет болса, онда JavaScript бағдарламалау тілінде confirm() әдісін пайдаланамыз. Бұл әдісте ақпарат сұхбат терезесі арқылы енгізіледі. Сұхбат терезесінде екі батырма орналасады. Осы батырма көмегімен екі логикалық нәтиже «true» және «false» мәндері пайда болады. Тұтынушы осы мәндерді қалауы бойынша пайдалана алады. Осы әдістің жұмыс үрдісін мысалмен сипаттайық.

Мысалы:

у айнымалысына логикалық мәнді меншіктейік.

Скрипт кодының ішіне conf() функциясын енгіземіз. Функцияның ішіндегі confirm() әдісі арқылы енгізілген логикалық мән у айнымалысына меншіктеледі. Скрипт коды төмендегі түрде жазылады.

```
<script language="JavaScript">
function conf()
```

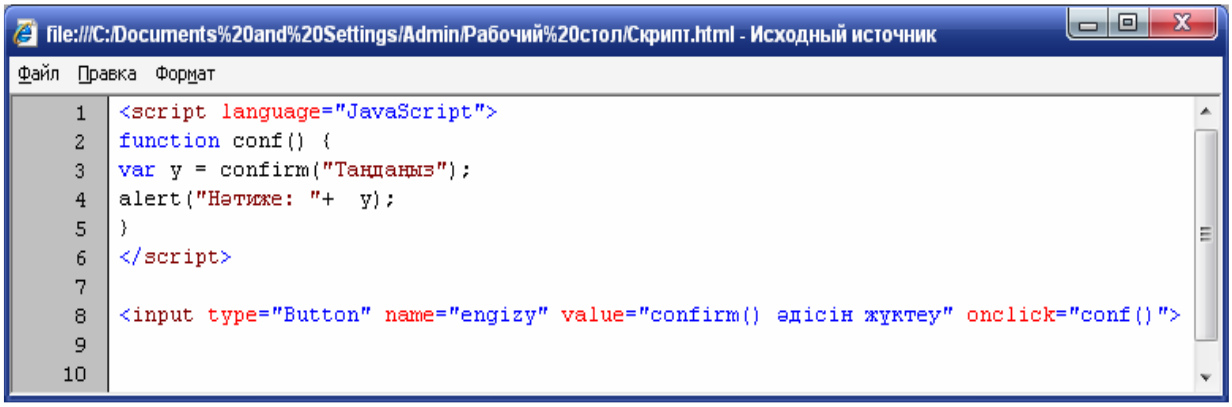
```
{
   var y = confirm("Таңдаңыз");
   alert("Нәтиже: "+  y);
}
</script>
```

Мұндағы alert() функциясы y айнымалысының мәнін браузер терезесіне шығарады. Енді Web-құжатта осы скриптті шақыру қалыбын құрамыз. Қалыпты

```
<input type="Button" name="engizy" value="confirm() әдісін жүктеу"
onclick="conf()">
```

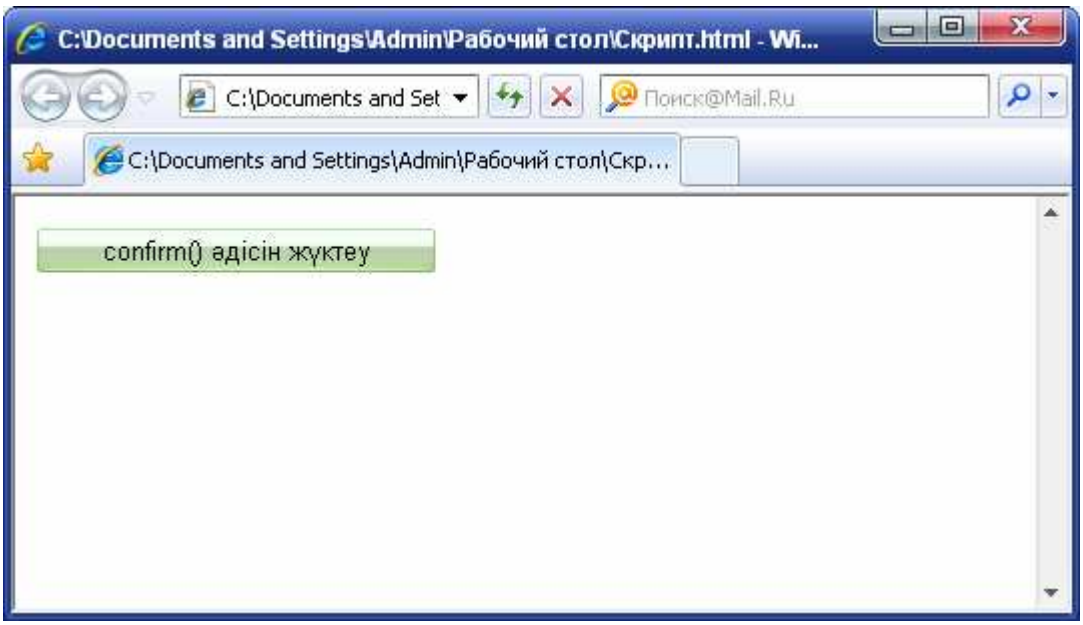
тэгімен жазамыз. Мұндағы <input> тэгі скриптты қалай іске қосуды анықтайды. Тзгтің type="button" атрибуты батырманы, name="engizy" атрибуты батырманың өзіндік атын, value="confirm() әдісін жүктеу" атрибуты шығатын жазуды сипаттайды. Ал onclick="conf()" атрибуты тышқан көрсеткішін шерткенде conf() функциясын шақырады.

Жоғарыда сипатталған программа кодтарын жинақтап, скрипт.html атауымен сақтаймыз (сурет 31).



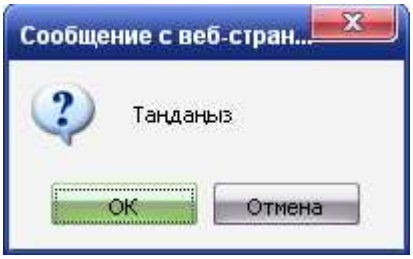
Сурет 31 – скрипт. html файлының коды

Осы скрипт.html файлы браузер терезесінде төмендегі түрде бейнеленеді (сурет 32).



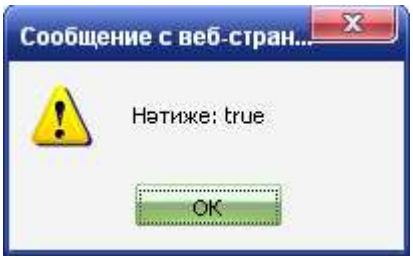
Сурет 32 – скрипт.html файлының браузер терезесіндегі бейнесі

Егер "confirm() әдісін жүктеу" батырмасын тышқан көрсеткішімен бір рет шертсек, программа conf() функциясын шақырады. Браузер терезесінде confirm() әдісінің сұхбат терезе пайда болады (сурет 33).



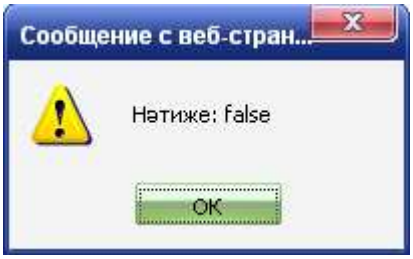
Сурет 33 – confirm() әдісінің сұхбат терезесі

Егер «ok» батырмасын тышқан көрсеткішімен бір рет шертсек, программа y айнымалысының мәнін браузер терезесіне шығарады. Браузер терезесінде «Нәтиже: true» сөзі жазылған терезе пайда болады (сурет 34).



Сурет 34 – alert("Нәтиже: "+ y) функциясының true нәтижесі

Сол сияқты, егер «Отмена» батырмасын тышқан көрсеткішімен бір рет шертсек, браузер терезесінде «Нәтиже: false» сөзі жазылған терезе пайда болады (сурет 35).



Сурет 35 – alert("Нәтиже: "+ y) функциясының false нәтижесі

2. Ақпаратты prompt() әдісімен енгізу. JavaScript бағдарламалау тілінде айнымалының мәнін пернетақта арқылы енгізетін болсақ, онда prompt() әдісінен пайдаланымыз. Осы prompt() әдісінің жұмыс үрдісін төмендегі мысалмен сипаттайық.

Мысалы: Нақты типті x және z айнымалыларына сандық мәндерді клавиатурадан енгіземіз. Сосын $y = x + z$ қосындысын браузер экранына шығарамыз.

Скрипт кодының ішіне sum() функциясын енгіземіз. Функцияның ішіндегі prompt() әдісі арқылы x және z айнымалыларына сандық мән меншіктеледі.

Скрипт коды төмендегі түрде жазылады.

```
<script language="JavaScript">
function sum()
{
  var x = prompt ("x санын енгізіңіз", "");
  var z = prompt ("z санын енгізіңіз", "");
   x =parseFloat( x );
   z =parseFloat( z );
  var y = x + z ;
  alert ("Нәтиже y = "+ y );
}
</script>
```

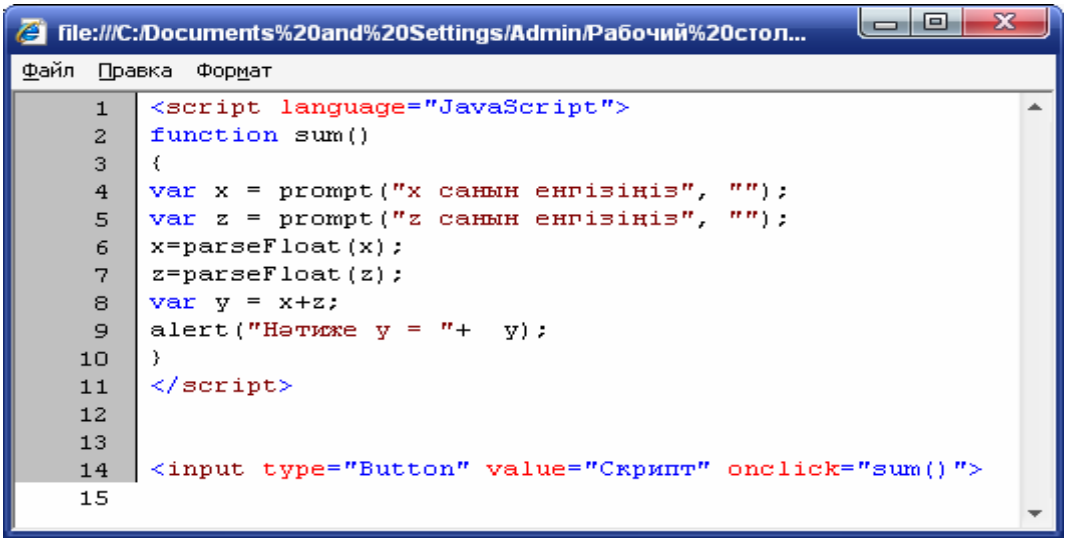
Скрипттегі alert("Нәтиже y = "+ y) функциясы y айнымалысының мәнін браузер терезесіне шығарады. Ал parseFloat() функциясы енгізілген мәнді қатарлық типтен нақты типке өткізеді. Егер бүтін типке өткізу қажет болса, онда parseInt() функциясын қолданамыз. Бұл функцияларды қолданбасақ, онда скрипт айнымалыларды қатарлық тип деп есептеп конкатенация (жалғау) амалын орындайды. Мысалы, мән ретінде 2 және 3 сандарын енгізсек, онда нәтиже 23 болады.

Web-құжатта осы скриптті шақыру қалыбын

<input type="button" value="Скрипт" onclick="sum()">

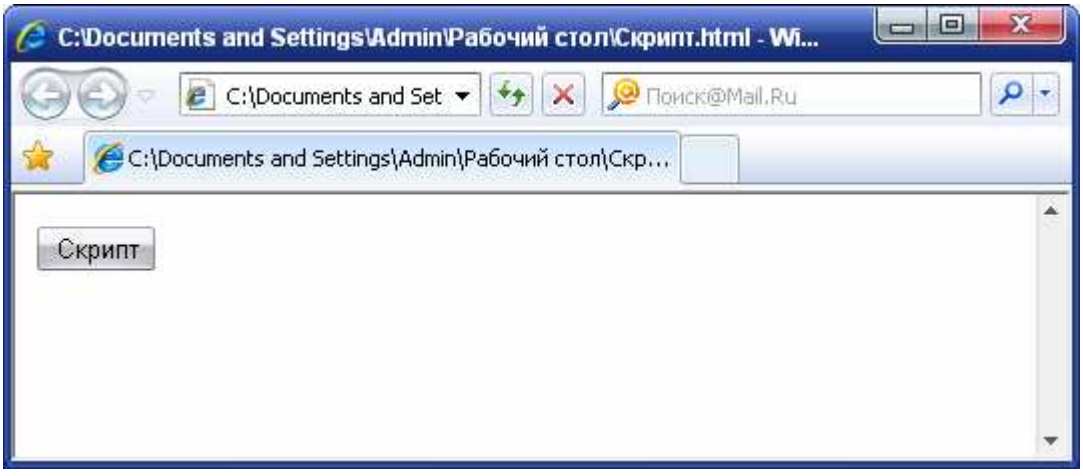
тэгімен жазамыз. Мұндағы <input> тэгі скриптты іске қосады. Тзгтің type="button" атрибуты батырманы, value="Скрипт" атрибуты батырма бетіне шығатын жазуды анықтайды. Ал onclick="sum()" атрибуты тышқан көрсеткішін шерткенде sum() функциясын шақырады.

Жоғарыда сипатталған программа кодтарын жинақтап, скрипт2.html атауымен сақтаймыз (сурет 36).



Сурет 36 – скрипт2. html файлының коды

Онда скрипт2.html файлы браузер терезесінде төмендегі түрде бейнеленеді (сурет 37).



Сурет 37 – скрипт2.html файлының браузер терезесіндегі бейнесі

Егер "Скрипт" батырмасын тышқан көрсеткішімен бір рет шертсек, программа sum() функциясын шақырады. Браузер терезесінде prompt() әдісінің сұхбат терезесі пайда болады (сурет 38).



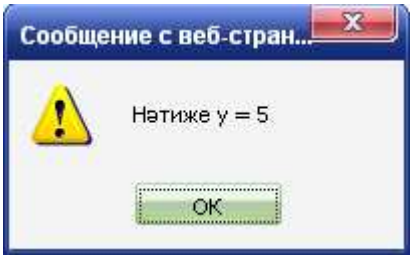
Сурет 38 – prompt() әдісінің сұхбат терезесі (x мәнін енгізу)

Сұхбат терезесіне x айнымалысының мәнін енгізіп, «ok» батырмасын тышқан көрсеткішімен бір рет шертсек келесі сұхбат терезесі пайда болады (сурет 39).



Сурет 39 – prompt() әдісінің сұхбат терезесі (z мәнін енгізу)

Жоғарыдағы сияқты, z айнымалысының мәнін енгізіп, «ok» батырмасын тышқан көрсеткішімен бір рет шертсек, программа y айнымалысының мәнін браузер терезесіне шығарады. Браузер терезесінде «Нәтиже $y = 5$» сөзі жазылған терезе пайда болады (сурет 40).



Сурет 40 – alert ("Нәтиже y = "+ y) функциясының нәтижесі

3. Ақпаратты мәтіндік өріс арқылы енгізу. JavaScript бағдарламалау тілінде клавиатурадан мәтін енгізу үшін мәтіндік өрістен пайдаланады. Мәтіндік өріс қалыптың ішінде <input> тэгімен құрылады. Web-құжатта осы мәтіндік өрісті құрамыз.

```
<form name="form1">
<input type="text" name="text1"  size="30"
value="Өріске мәтін енгізіңіз" onblur="oris()" >
</form>
```

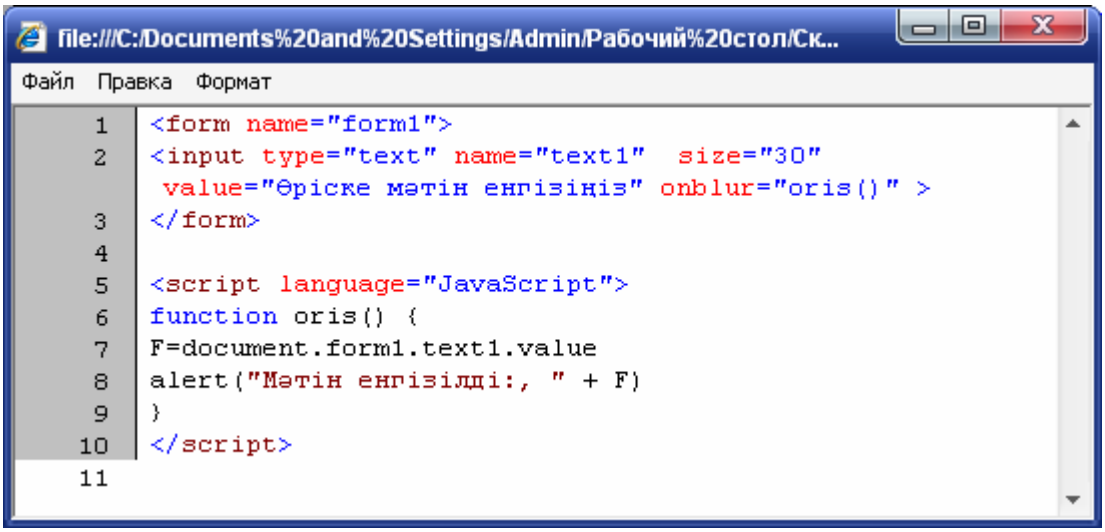
Қалып HTML тілінің <form> тэгімен сипатталған. Қалыптың name="form1" атрибуты қалыптың өзіндік атын анықтайды. Әдетте атрибуттан Web-құжатта бірнеше қалып болғанда пайдаланады. Қалыптың ішіндегі <input> тэгі қалыптың өрісін сипаттайды. Тэгтің type="text" атрибуты атрибуты өрістің типін анықтайды. Біздің мысалда мәтіндік өріс деп берілген. Тэгтің name="text1" атрибуты өрістің өзіндік атын, size="30" атрибуты мәтіндік өріс өлшемін символдар санымен, value="Өріске мәтін енгізіңіз" атрибуты өрістің ішіндегі жазуды анықтайды. Ал onblur="oris()" атрибуты оқиғаны, яғни мәтіндік өрістің ағымдық күйін жоғалқанын көрсетеді, яғни өрістен тыс жерді тышқан көрсеткішімен бір рет шертсек oris() функциясын шақырады.

Осы қалыпты өңдейтін скрипт кодын құрайық. Скрипт кодының ішіне oris() функциясын енгіземіз.

```
<script language="JavaScript">
function oris()
{
   F =document.form1.text1.value
   alert("Мәтін енгізілді: " + F )
}
</script>
```

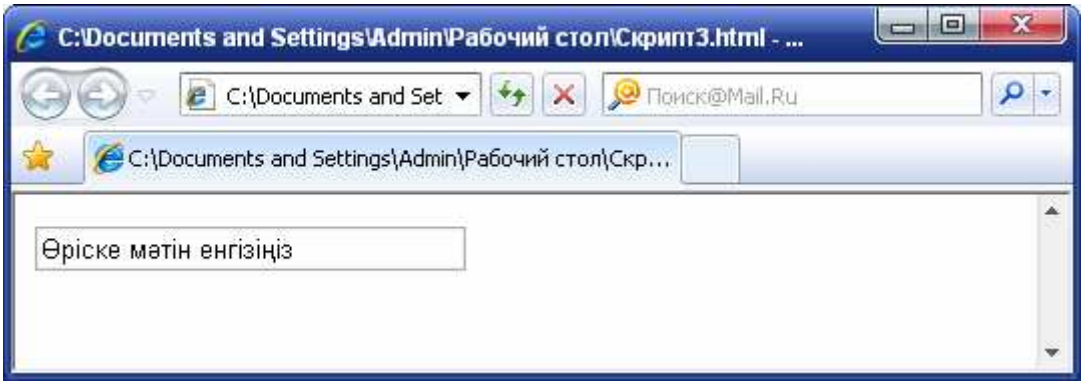
Скрипттегі oris() функциясы скрипттің іс-әрекетін сипаттайды. Функцияның F=document.form1.text1.value қатары мәтіндік өріске енгізілген мәтінді F айнымалысына меншіктейді. Ал alert("Мәтін енгізілді: " + F) функциясы F айнымалысының мәнін браузер терезесіне шығарады.

Осы сипатталған программа кодтарын жинақтап, скрипт3.html атауымен сақтаймыз (сурет 41).



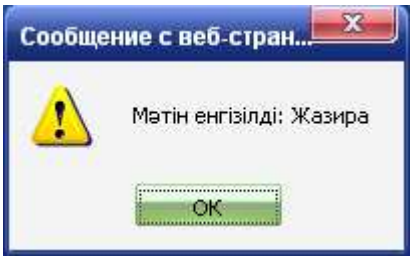
Сурет 41 – скрипт3. html файлының коды

Сонымен скрипт3.html файлы браузер терезесінде төмендегі түрде бейнеленеді (сурет 42).



Сурет 42 – скрипт3.html файлының браузер терезесіндегі бейнесі

Скриптті жұмысқа беру үшін мәтіндік өріске мәтін, мысалы «Жазира» сөзін енгізіп, браузер экранының кез келген жерін тышқан көрсеткішімен бір рет шертсек, скрипт oris() функциясын шақырады. Браузер терезесінде prompt() әдісінің сұхбат терезесі пайда болады (сурет 43).



Сурет 43 – oris() функциясының нәтижесі

4. Ақпаратты мәтіндік аймақ арқылы енгізу. JavaScript бағдарламалау тілінде енгізілетін мәтіннің көлемі үлкен болса, яғни ол бір қатардан артық болса, онда textarea қалыбынан пайдаланамыз. Сонымен, textarea қалыбында енгізілетін немесе монитор экранына шығарылатын мәтін бірнеше қатардан тұруы тиіс. Енді textarea қалыбының жұмыс үрдісін сипаттайтын скрипт алгоритмін қарастырайық.

Мәтіндік аймақ қалыптың ішінде <input> тэгімен құрылады. Web-құжатта осы мәтіндік аймақты құрамыз.

```
<form name="kalip">
<textarea name="aimak" cols="50" rows="6" >
    М.Әуезов атындағы Оңтүстік Қазақстан мемлекеттік университеті,
    «Информатиканы оқытудың теориясы және әдістемесі» кафедрасы
</textarea>
</form>
```

Сонымен, қалып HTML тілінің <form> тэгімен сипатталған. Қалыптың name="kalip" атрибуты қалыптың өзіндік атын анықтайды. Қалыптың ішіндегі <textarea> тэгі қалыптың мәтіндік аймағын сипаттайды.

Тэгтің name="aimak" атрибуты өрістің өзіндік атын, cols="50" атрибуты өріс терезесінде мәтіннің көрінуге тиісті баған санын, ал rows="6" атрибуты өріс терезесінде мәтіннің көрінуге тиісті қатар санын анықтайды

Осы қалыпты өңдейтін скрипт кодын құрайық. Скрипт коды төмендегі түрде құрылады.

```
<script language="JavaScript">
function Matin()
{
  var F =document.kalip.aimak.value;
  alert(F );
}
</script>
```

Скрипт кодының ішіне matin() функциясын енгіземіз. Бұл функция скрипттің іс-әрекетін сипаттайды. Функцияның a=document.kalip.aimak.value қатары мәтіндік өріске енгізілген мәтінді a айнымалысына меншіктейді. Ал alert(F) функциясы F айнымалысының мәнін браузер терезесіне шығарады.

Енді осы скрипт кодын орындау үшін Web-құжаттың негізгі бөліміне

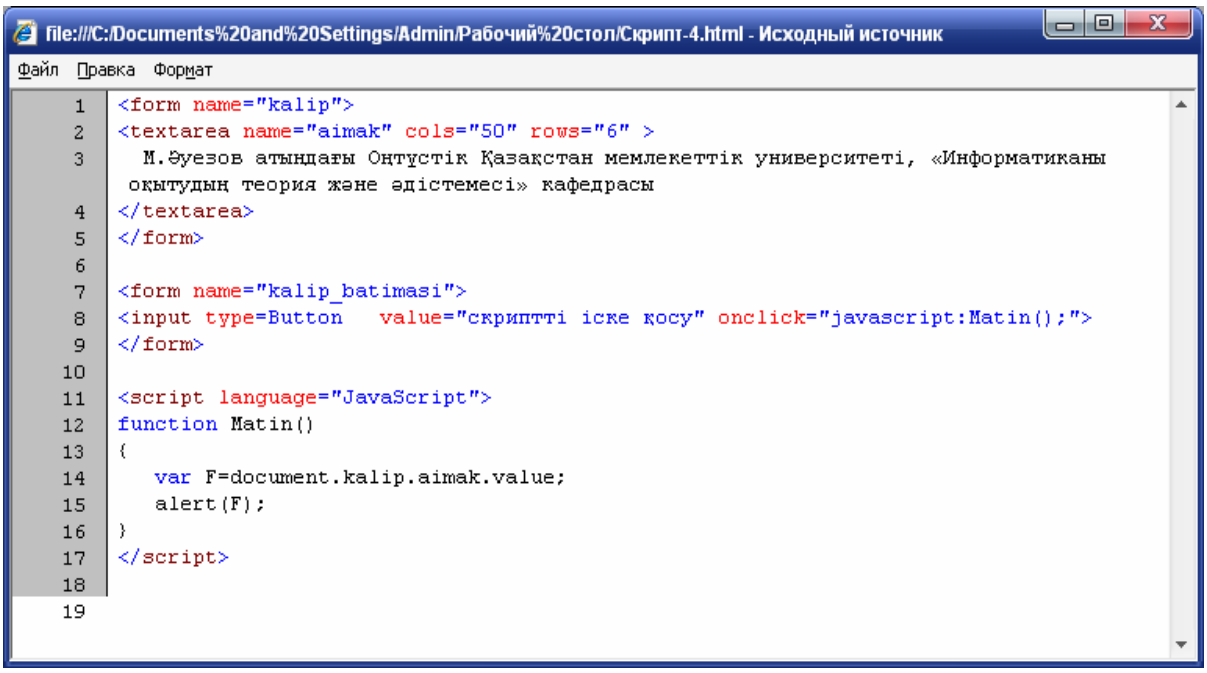
```
<form name="kalip_batimasi">
<input type="Button"   value="скриптті іске қосу"
onclick="javascript: Matin();">
</form>
```

тэгін жазамыз. Мұндағы <form> тэгі қалыпты, ал қалыптың name="kalip_batimasi" атрибуты қалыптың өзіндік атын анықтайды. Қалыптағы <input> тэгі скриптты қалай іске қосуды сипаттайды. Тэгтің type="button" атрибуты батырманы, value="скриптті іске қосу" атрибуты батырманың бетіне шығатын жазуды, ал onclick="javascript:Matin();" атрибуты тышқан көрсеткішін шерткенде орындалатын оқиғаны анықтайды.

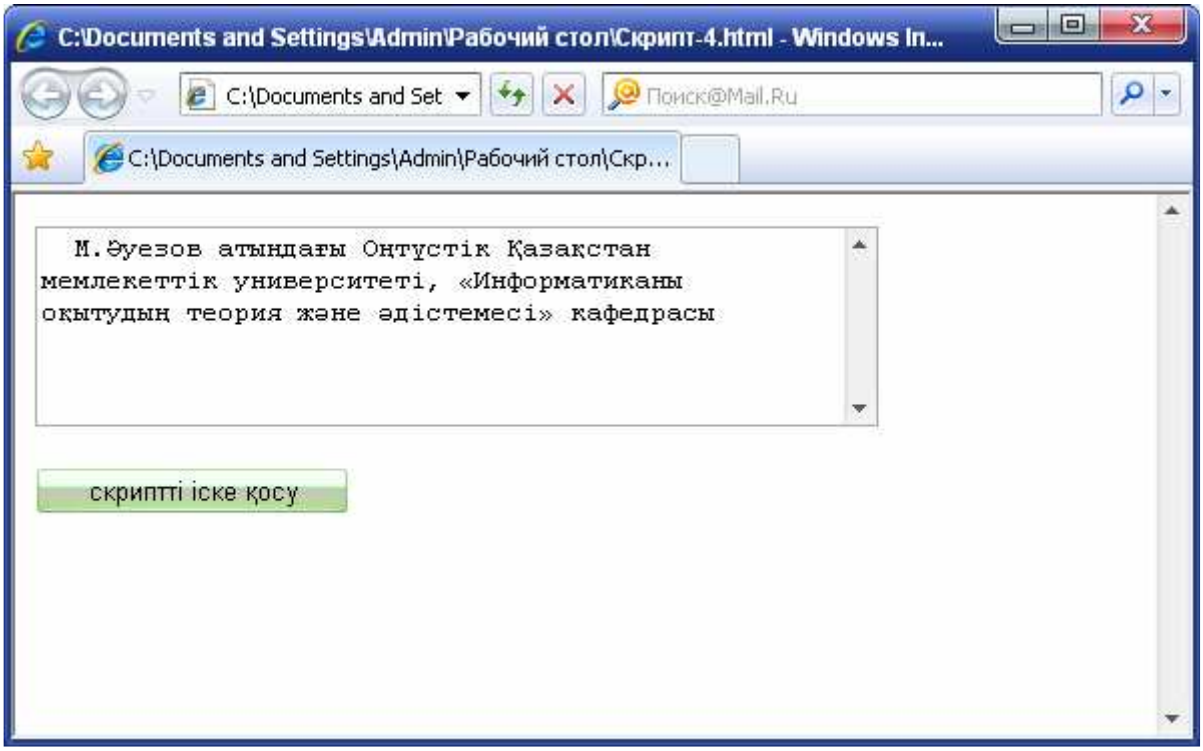
Осы сипатталған программа кодтарын жинақтап, скрипт-4.html атауымен дискіге сақтаймыз (сурет 44).

Жалпы, скрипт-4.html файлы браузер терезесінде төмендегі түрде бейнеленеді (сурет 45).

Скриптті жұмысқа беру үшін мәтіндік өріске мәтін, мысалы «М.Әуезов атындағы Оңтүстік Қазақстан мемлекеттік университеті, «Информатиканы оқытудың теориясы және әдістемесі» кафедрасы» сөйлемін енгізіп, «скриптті іске қосу» батырмасын тышқан көрсеткішімен бір рет шертсек, скрипт Matin() функциясын шақырады.

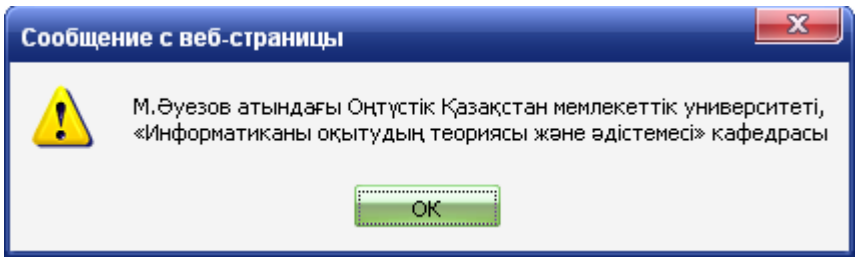


Сурет 44 – скрипт-4. html файлының коды



Сурет 45 – скрипт-4.html файлының браузер терезесіндегі бейнесі

Браузердің терезесінде хабарлама терезесі пайда болып, онда F айнымалысының мәні пайда болады (сурет 46).



Сурет 46 – Matin() функциясының нәтижесі

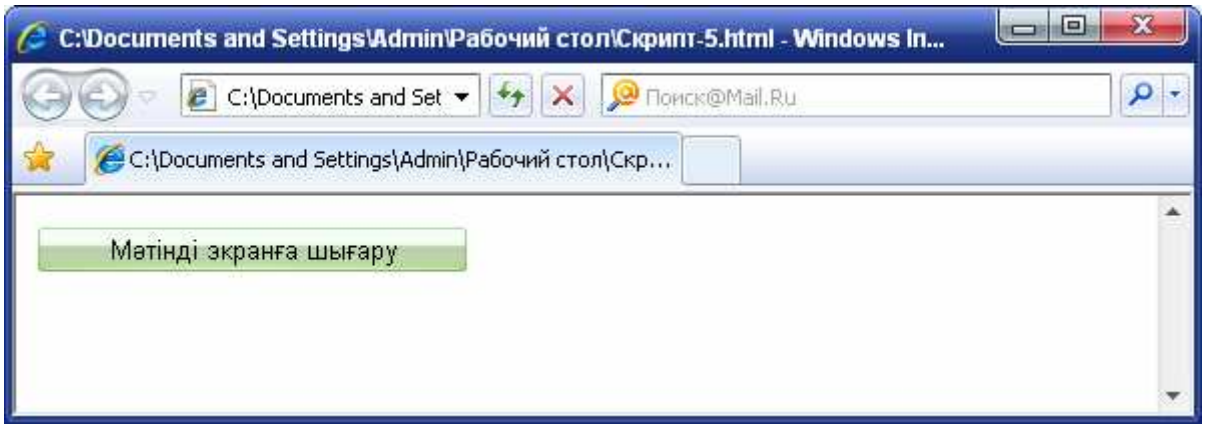
5. Ақпаратты alert() әдісі арқылы шығару. Егер Web-құжатта ақпаратты сұхбат терезесінде хабарлама ретінде шығару қажет болса, онда JavaScript бағдарламалау тілінде alert() әдісін пайдаланамыз. Бұл әдісті жоғарыдағы скрипттерде пайдаландық. Енді осы әдістің жұмыс үрдісін төмендегі мысалда қарастырайық.

Бұл әдісті қолдану үшін Web-құжат кодында

```
<input type="button"  value="Мәтінді экранға шығару"
onclick=’alert("alert  функциясының нәтижесі")’>
```

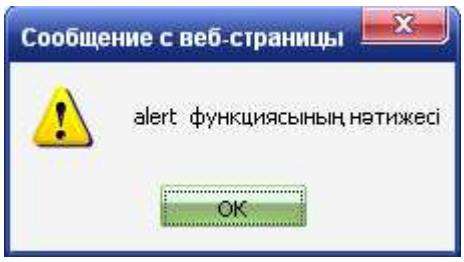
тэгін жазсақ жетерлі. Мұндағы <input> тэгі скриптты қалай іске қосуды сипаттайды. Тэгтің type="button" атрибуты батырманы, value="Мәтінді экранға шығару" атрибуты батырманың бетіне шығатын жазуды, ал onclick=’alert("alert функциясының нәтижесі")’ атрибуты тышқан көрсеткішін шерткенде орындалатын оқиғаны анықтайды.

Осы код браузер терезесінде төмендегі түрде бейнеленеді (сурет 47).



Сурет 47 – <input> тэгінің браузер терезесіндегі бейнесі

Егер «Мәтінді экранға шығару» батырмасын тышқан көрсеткішімен бір рет шертсек, скрипт alert() функциясын шақырады. Браузер терезесінде хабарламада «alert функциясының нәтижесі» деген жазу пайда болады (сурет 48).



Сурет 48 – alert() функциясының нәтижесі

6. Ақпаратты write() әдісі арқылы шығару. Егер Web-құжатта ақпаратты сұхбат терезесінде емес, жай ғана браузер терезесіне тікелей шығару қажет болса, онда JavaScript бағдарламалау тілінде write() әдісін пайдаланамыз. Осы әдістің жұмыс үрдісін төмендегі қарапайым мысалда қарастырайық.

Мысалы:

$x = 2$

$y = 3$

$z = x + y$

Осы мысалды өңдейтін скрипт кодын құрайық. Скрипт коды төмендегі түрде құрылады. Скрипт кодының ішіне shigaru() функциясын енгіземіз.

```
<script language="JavaScript">
function shigaru()
{
x =2;
y =3;
z = x + y ;
document.write("Нәтиже:" + z );
}
</script>
```

Бұл shigaru() функциясы скрипттің іс-әрекетін сипаттайды. Функция *x* және *y* айнымалыларының мәндерін қосып, қосындыны *z* айнымалысына меншіктейді. Ал document.write("Нәтиже:" + z) қатары *z* айнымалысының мәнін браузер терезесіне тікелей шығарады.

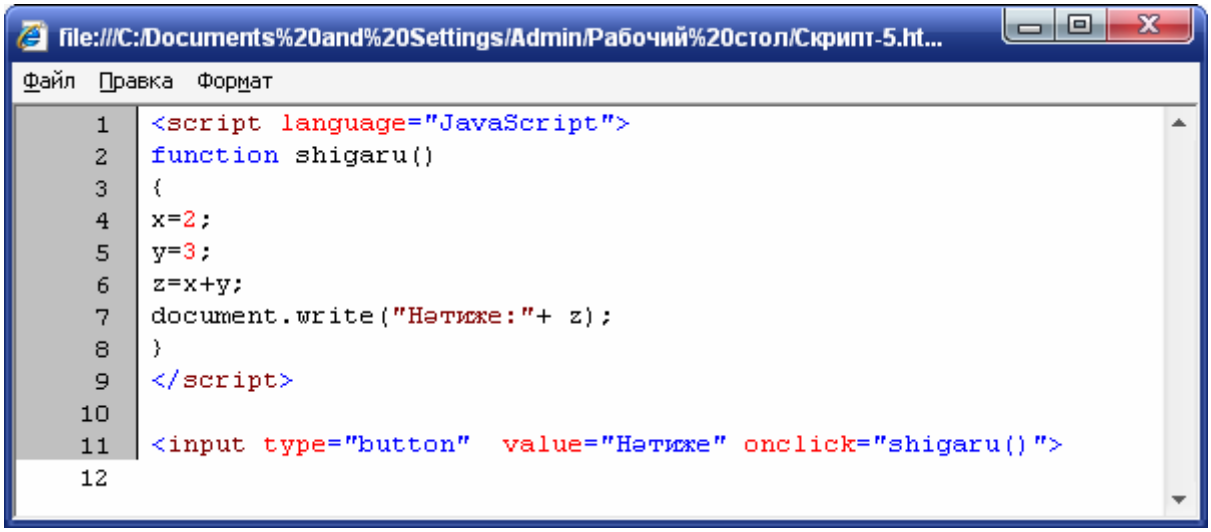
Бұл скриптті орындау үшін Web-құжат кодында

```
<input type="button"  value="Нәтиже" onclick="shigaru()">
```

тэгін жазамыз. Мұндағы <input> тэгі скриптты қалай іске қосуды сипаттайды. Тэгтің type="button" атрибуты батырманы, value="Нәтиже" атрибуты батырманың бетіне шығатын жазуды, ал onclick="shigaru()"

атрибуты тышқан көрсеткішін шерткенде орындалатын оқиғаны анықтайды.

Осы код браузер терезесінде төмендегі түрде бейнеленеді (сурет 49).



Сурет 49 - Ақпаратты write() әдісі арқылы шығару

7. Ақпаратты мәтіндік өріс арқылы шығару. JavaScript бағдарламалау тілінде ақпаратты көрнекі түрде енгізу үшін мәтіндік өрістен пайдалану тиімді. мәтіндік өрістен пайдалануды жоғарыда көрсетілген мысалда қарастырайық.

Мысалы: Берілген x және y сандарын мәтіндік өріс арқылы енгізейік. Сосын $z = x + y$ өрнегін есептеп, нәтижені келесі мәтіндік өрісте шығарайық.

Web-құжатта қалыптың ішінде x санын енгізетін мәтіндік өрісті анықтаймыз.

```
<form name="kalip">
<input type="text" name="x"  size="5" value="2"
</form>
```

Қалып HTML тілінің <form> тэгімен сипатталады. Қалыптың <name="kalip"> атрибуты қалыптың өзіндік атын анықтайды. Қалыптың ішіндегі <input> тэгі мәтіндік өрісті сипаттайды. Тэгтің type="text" атрибуты атрибуты өрістің типін анықтайды. Тэгтің name="x" атрибуты өрістің өзіндік атын, value="2" атрибуты өрістің ішіндегі бастапқы жазуды анықтайды.

Сол сияқты, y және z сандарын енгізетін мәтіндік өрістерді анықтаймыз.

```
<input type="text" name="y"  size="5" value="3" ><br><br>
<input type="text" name="z"  size="5" value="" >
```

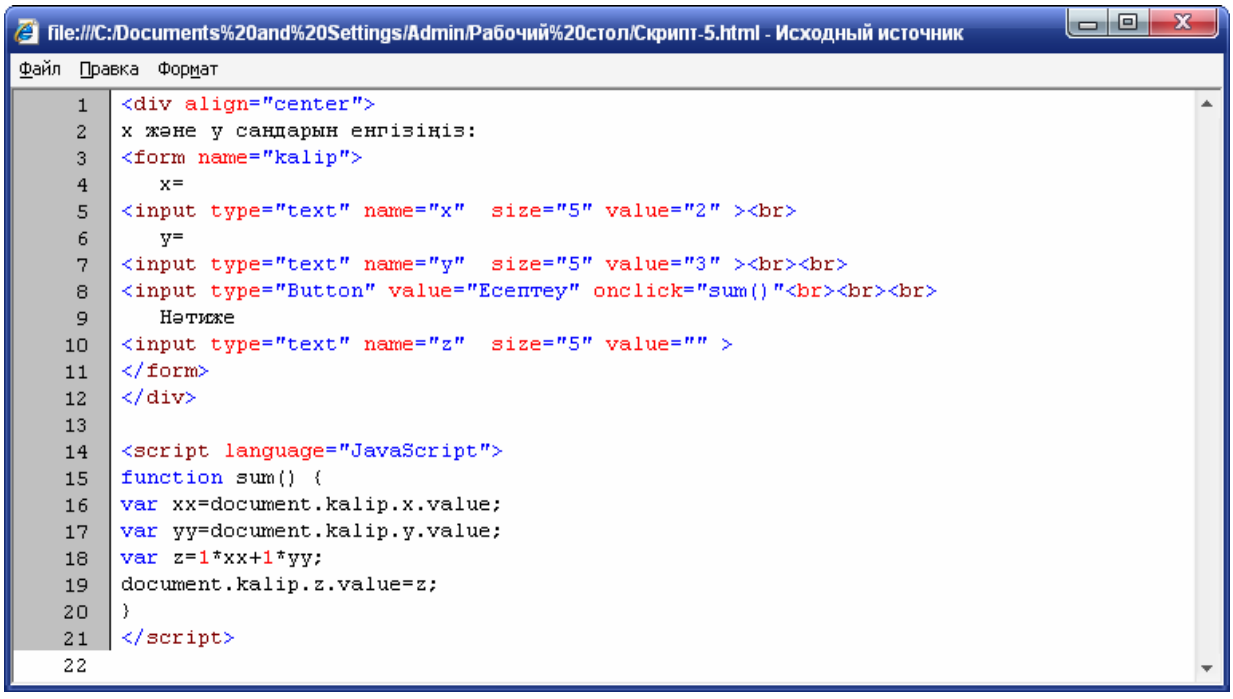
Бұл скриптті орындау үшін

```
<input type="Button" value="Есептеу" onclick="sum()"
```

тэгін жазамыз. Мұндағы <input> тэгінің onclick="sum()" атрибуты тышқан көрсеткішін шерткенде орындалатын оқиғаны, яғни sum() функциясын шақырады. Енді осы қалыпты өңдейтін скрипт кодын құрайық.

```
<script language="JavaScript">
function sum()
{
  var xx=document.kalip.x.value;
  var yy=document.kalip.y.value;
  var z=1*xx+1*yy;
  document.kalip.z.value=z;
}
</script>
```

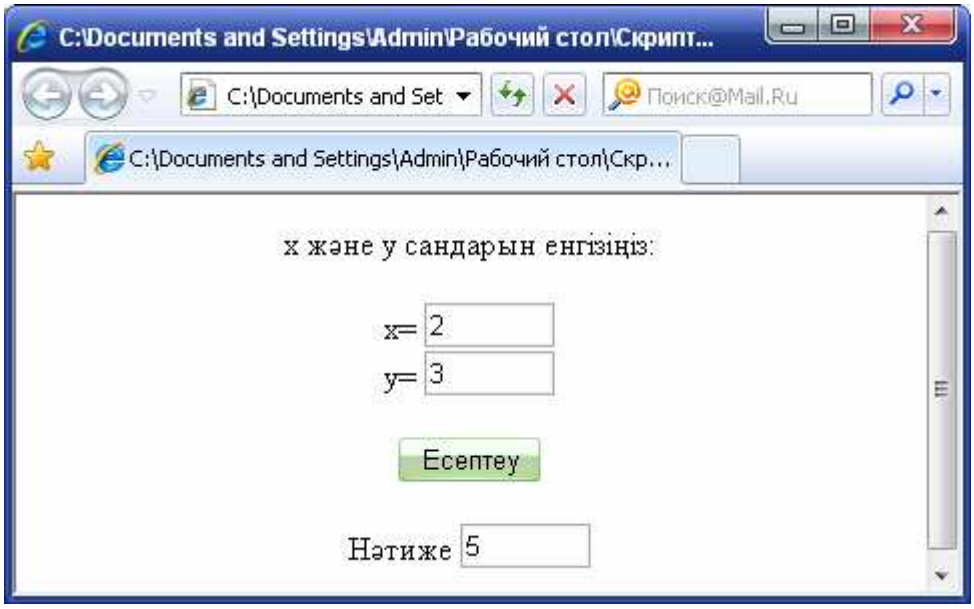
Скрипттегі sum() функциясы скрипттің іс-әрекетін сипаттайды. Функцияның «var xx=document.kalip.x.value» қатары мәтіндік өріске енгізілген мәтінді xx айнымалысына меншіктейді. Сол сияқты «var yy=document.kalip.y.value» қатары мәтіндік өріске енгізілген мәтінді yy айнымалысына меншіктейді. Ал var $z=1*xx+1*yy$ өрнегі енгізілген қатарды сандық деп қабылдап, $z=x+y$ өрнегін есептейді. Сонымен қатар, «document.kalip.z.value=z» қатары нәтижені мәтіндік өріске шығарады.



Сурет 50 – скрипт-5. html файлының коды

Осы сипатталған программа кодтарын жинақтап, скрипт-5.html атауымен дискіге сақтаймыз (сурет 50).

Сонымен, скрипт-5.html файлы браузер терезесінде төмендегі түрде бейнеленеді (сурет 51).



Сурет 51 – скрипт3.html файлының браузер терезесіндегі бейнесі

6.4 JavaScript-ке мысалдар

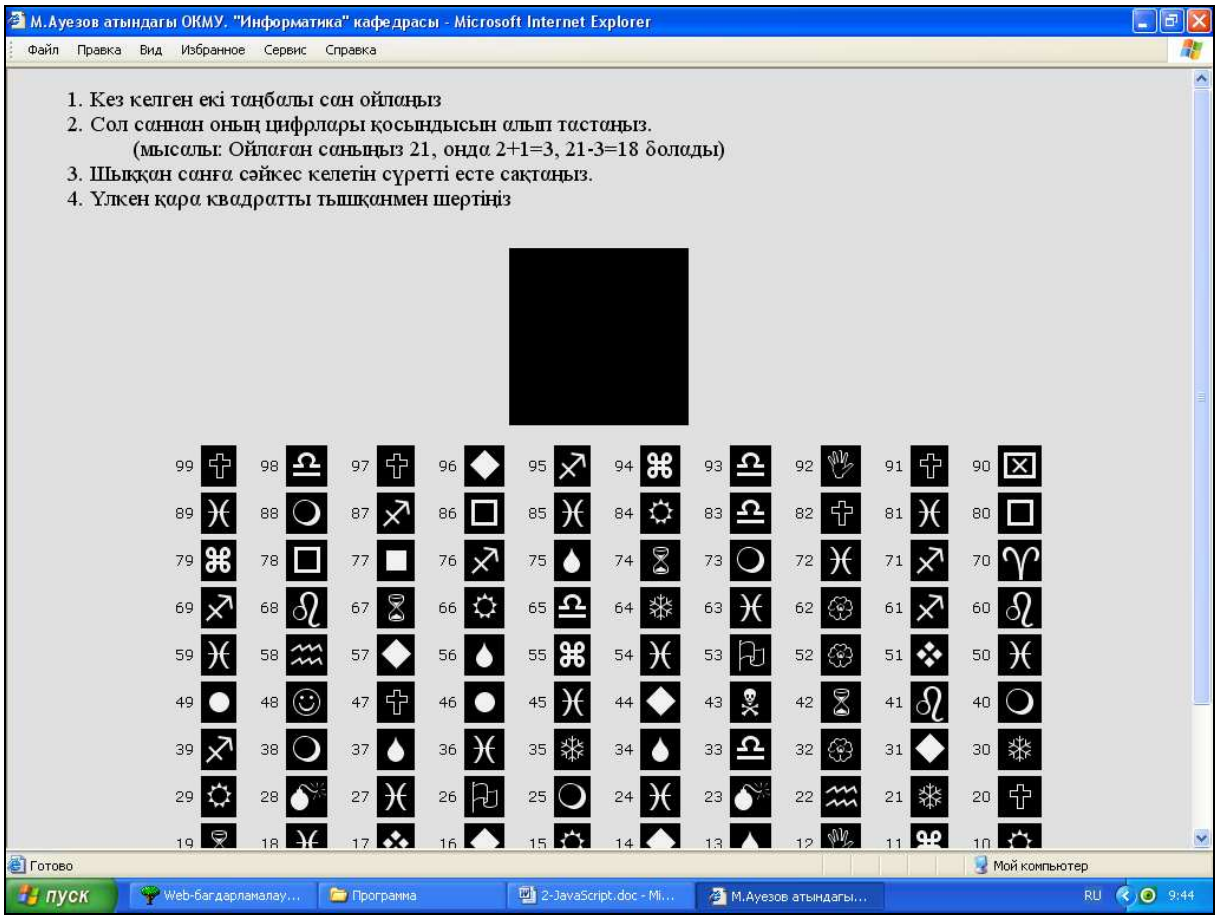
1. «Қара квадрат» скрипті. Скрипт орындауға дайын түрде Web-құжатқа жазылған. Файлды орындау үшін оны html кеңеймесімен сақтау керек. Кейде браузер JavaScript-ті бірден орындамайды, яғни оны орындауға сұраныс жасау керек болады.

```
<!DOCTYPE HTML PUBLIC "-//W3C//Dtd HTML 4.0 Transitional//EN">
<HTML><head><title>М.Ауезов атындағы ОҚМУ.  "Информатика"
кафедрасы</title>
<meta http-equiv=Content-Type content="text/html; charset=windows-1251">
</head>
<body bgcolor="e0e0e0"> <P style="font-size: 13px">
<style type=text/css>.num {
font-size: 12px; width: 30px; color: #000; font-family: verdana; text-align:
right}
.symbols {
font-size: 30px; background: #000; color: #f1f1f1; font-family: wingdings; text-
align: center}
.viewsymbol {
```

```
font-size: 100px; background: #000; width: 150px; color: #f1f1f1; font-family:
wingdings; height: 150px}
</style>
<script language=javascript type=text/javascript>
<!--//
ax=0;
function viewtable() {
ax=Math.round(Math.random()*26);
alphaArray=new Array("a", "n", "b", "d", "f", "h", "{", "i", "l", "v", "x", "z", "I",
"J", "M", "N", "o", "O", "R", "S", "T", "U", "m", "6", "^", "u", "_", "[", "]");
table="<table align=\"center\" border=\"0\" cellspacing=\"5\"
cellpadding=\"1\"><tr>";
j=1;
for ( i = 99 ; i >= 0 ; i-- ) {
a=Math.round(Math.random()*26);
if(i%9 == 0 &&  i < 89)
a=ax;
table+="<td class=\"num\">"+i+"</td><td
class=\"symbols\">"+alphaArray[a]+"</td>";
if(j%10 == 0) table+="</tr><tr>"; j++;
}
table+="</table>";
sym.innerHTML=table
sh.innerHTML=""
}
function show() {
sh.innerHTML=alphaArray[ax]
sym.innerHTML="<center>&#927;сы сурет пе -? <a
href=\"javascript:viewtable()\"><b>&#954;&#945;йт&#945;л&#945;y</b></a
></center>";
}
//-->
</script>
<table cellSpacing=0 width="100%" border=0>
<tbody>  <tr>
<td vAlign=top>
<dd>1. &#922;ез &#954;елген е&#954;i т&#945;&#1187;б&#945;лы
с&#945;н &#959;йл&#945;&#1187;ыз
<dd>2. С&#959;л с&#945;нн&#945;н &#959;ны&#1187;
циф&#961;л&#945;&#961;ы &#1179;&#959;сындысын &#945;лып
т&#945;ст&#945;&#1187;ыз.
  <dd>         (мыс&#945;лы:
&#927;йл&#945;ғ&#945;н с&#945;ны&#1187;ыз 21,
&#959;нд&#945; 2+1=3, 21-3=18 &#948;&#959;л&#945;ды)
```

<dd>3. Шыққαн сαнғα сәйκес
κелетін сүρетті есте сақтαңыз.
<dd>4. Үлκен қαρα
κвαдραтты тышқαнмен
шеρтіңіз
<dd>
<table style="cursor: hand" onclick=show() cellSpacing=10 align=center
border=0>
<Tbody> <tr> <td class=viewsymbol id=sh
align=middle> </td></tr></Tbody></table></dd></td></tr>
<tr>
<td id=sym vAlign=top height=300></td></tr></Tbody></table>
<script language=javascript type=text/javascript>
<!--//
viewtable();
//-->
</script>
</P>
<P style="font-size: 12px" align=right></P>
<HR align=left width="100%" color=#000000 noShade SIZE=1>
<script language=javascript><!--
bmQ='<img src=http://c.bigmir.net/?s75479&t5'
bmD=document
bmD.cookie="b=b"
if(bmD.cookie)bmQ+='&c1'
//--></script> Εсте бοлсын: καмеρα
жұмыс ìстеп тұρ, миκραфοн
қοсулы ...
      Οйын αлгορитмìн
тαпқαнғα Семδиев
Ορдαδαй
αғαй 100$ δеρедì!
<script language=javascript1.2><!--
bmS=screen;bmQ+='&d'+(bmS.colorDepth?bmS.colorDepth:bmS.pixelDepth)+
"&r"+bmS.width;
//--></script>
<script language=JavaScript1.1 type=text/javascript><!--
a+=';j='+navigator.javaEnabled()
js=11//--></script>
<script language=JavaScript1.2 type=text/javascript><!--
s=screen;a+=';s='+s.width+'*'+s.height
a+=';d='+(s.colorDepth?s.colorDepth:s.pixelDepth)
js=12//--></script>
<script language=JavaScript1.3 type=text/javascript><!--
js=13//--></script>

</body></HTML>



Сурет 52 – Скрипт терезесі

7 PHP тіліне кіріспе

7.1 PHP тілінің қарапайым элементтері

PHP (Hypertext Preprocessor – гипермәтінді препроцессор) тілі серверде орындалатын сценарий жазуға арналған тіл. Браузер URL-адресте (Uniform Resource Locator – ықшамдалған ресурстар көрсеткіші) көрсетілген PHP тілінде жазылған Web-құжатты орындау үшін Web-серверге сұраныс жасайды. Web-сервер PHP-процессорын жүктеп, PHP-кодты орындайды ды, нәтижені Web-серверге қайтарады. Ал Web-сервер өз кезегінде нәтижені браузерге терезеде бейнелеу үшін жібереді.

PHP тілін 1994 жылы Расмус Лердорф (Rasmus Lerdorf) жасап шығарды. Оның негізгі мақсаты жеке сайттарда интерактивтік әдістер қолдану болатын. Расмус тілдің негізгі синтаксисі мен интерпретаторын жазып, оны PHP – Personal Home Page атады.

1995 жылы Расмус интерпретаторға қалыпты өңдеу FI (Form Interpretator) бөлімін қосып, интерпретаторды PHP/FI 2.0 деп атайды. Ал, 1998 жылы Зива Суракси мен Энди Гутманс (Zeev Suraski and Andi Cutmans) интерпретатор коды ашық болғандықтан (open source), оны қайта өңдеп, PHP 3.0 (Hypertext Preprocessor) версиясын шығарды. Осы кезден бастап PHP технологиясын дамытумен айналысатын PHP Group пайда болады. Сонымен, PHP тілі қолдануға оңай, түсінуге жеңіл, мүмкіндіктері көп болғандықтан жылдам тарап кетті. Қазіргі кезде тілдің 2000 жылы шыққан PHP 4.0 және 2004 жылы пайда болған PHP 5.0 версиялары белгілі. Төменде тілдің объектке бағытталған PHP 5.0 версиясын қарастырамыз [17-20].

PHP тілі Web-сервер мен пайдаланушы арасында деректер алмасуда, яғни Web-құжатта форум, қонақ кітабы, тіркелу анкетасы, хабарландыру тақтасы, чат құруда қолданылады.

PHP тілінің қарапайым элементтеріне түсініктемелер, қызметші сөздер, тұрақтылар, айнымалылер және функциялар жатады. Енді осыларды қысқаша қарастырайық.

Түсініктемелер. PHP тілі бізге түсініктеме қоюдың бірнеше түрін ұсынады. Олардың ішінен екі қисық сызықтан (//) кейін түсініктеме жазу ең қарапайым болып есептеледі. Осы екі сызықтан кейін PHP процессоры осы қатардың соңына дейін өңдемейді.

Түсініктемені UNIX тілінің скрипт түсініктемелері сияқты нөмір (#) символының көмегімен де беруге болады. Сонымен қатар, C++ тілі стиліндегі (/*..*/) түсініктемелерді де қолдануға болады. Түсініктеме берудің түрлі жолдары төмендегі мысалда келтірілген.

```
<?
echo (“Hello”); // бұл түсініктеме
echo (“Hello”); # UNIX тілі стиліндегі түсініктеме
```

```
/* C++ тілі стиліндегі түсініктеме */
?>
```

PHP түсініктемелері PHP-кодының қысқартылған тэгтерінде ғана жұмыс істейді. Егер түсініктеме символдары қысқа тэг құрамында орналаспаса, онда олар әр түрлі мәтін сияқты браузерде көрсетілуі мүмкін.

Мысалы:

```
<?
echo (“Hello”); //
?>
```

Түсініктемені тек қана оператордан кейін емес, мына түрде де қоюға болады.

```
<?
$a = “Hello, world”;
echo strstr (// бұл функцияны кейінірек қарастырамыз
$a,  “H”);
?>
```

Қызметші сөздер деп JavaScript тілінің жұмысына қажетті нұсқауларды айтады. Әрбір сөздің өз мағынасы, атқаратын қызметі бар. Қызметші сөздерді жұмыс барысында біртіндеп мағынасы қарай түсіндіріп отырамыз.

Тұрақтылар. Тұрақтылар PHP тілінде define (): функциясы көмегімен жазылады. Функцияның жалпы форматы:

```
define (constant, value)
```

Бұл функцияның бірінші параметрі – тұрақтының атауы, ал екіншісі - оның мәні. Пайдалану алдында тұрақтының мәні оның атына меншіктеледі:

```
<?
define (name, 15);
define (t, “\x20”); // бос орынның коды
define (data, “Hello);
echo (name);
echo (t);
echo (data);
?>
```

PHP тілінде тұрақтыларға қатысты тағы бір defined () функциясы қолданылады. Бұл функция тұрақты анықталған ба, жоқ па соны тексереді:

```
defined (constant)
```

Бұл функцияның бір ғана параметрі бар, ол анықталатын тұрақтының атауы. Егер тұрақты анықталған болса, онда функция true мәнін қабылдайды, кері жағдайда false мәніне ие болады.

Мысалы:

```
<?
define (constant, “Hello”);
if (defined (“constant”))
{
echo (“constant is defined”);
}
?>
```

Айнымалылар. PHP тілінде айнымалы $ (ақша бірлігі) белгісімен басталатын әріп, цифр және _ (сызықша) белгісінің тізбегін айтады. Бірақта ($) кейінгі символ әріп болуы шарт, яғни сан, не болмаса арнайы белгі болмауы керек.

Мысалы: $a, $x1, $user_Func_1

Қызметші сөздерге қарағанда айнымалылардың аты PHP тілінде регистрге сезімтал болады. PHP процессоры регистрды есепке алады, яғни бас және кіші әріптерді бөлек деп қабылдайды.

Мысалы:

```
$user_func
$User_func
$USER_FUNC
```

Бұлардың жазылуы бір болғанымен регистрінде айырмашылық бар. Оларды PHP процессоры әр түрлі айнымалылар деп қабылдайды.

PHP тілінің басқа тілдерден тағы бір ерекшілігі, айнымалыны пайдаланудан алдын оның типін көрсету шарт емес. Сонымен бірге, айнымалы скрипт орындалу барысында түрлі типтер қабылдау мүмкін.

7.2 Деректердің типтері және амалдар

PHP тілінде қолданылатын деректер типтерін қарастырайық:

- Integer
- String
- Float (double, real)
- Array
- Boolean
- Object

PHP тілінде айнымалының типін сипаттау үшін settype функциясы қолданылады. Ол айнымалының типін типін белгілейді.

Мысалы:

```
<?
$var = “15”;
echo(gettype($var));
settype ($var, integer);
echo “<br>”;
echo(gettype($var));
?>
```

PHP тілінде айнамалының типін анықтау үшін gettype функциясы қолданылады. Бұл функцияның бір ғана параметрі бар, ол типі анықталатын тұрақтының атауы.

Мысалы:

```
<?
$var_str = “name”;
$var_num = 15;
echo (gettype ($var_str));
echo “<br>”;
echo (gettype ($var_num));
?>
```

Бағдарлама коды орындалғаннан соң, мына нәтижені аламыз:

```
String
Integer
```

Сондай-ақ, PHP тілінде айнымалының типін өзгерту үшін gettype функциясымен қатар, типті сипаттау операторы да қолданылады. Жаңа тип жақшаның ішінде көрсетіледі:

```
$var = (int) $var;
```

Мысалы:

```
<?
$var = “5”; // string типі
$var = (int) $var; //  int
echo (gettype ($var));
?>
```

PHP тілінде амалдар (операторлар) төмендегі түрлерге бөлінеді:

- меншіктеу
- арифметикалық
- биттік
- салыстыру

- логикалық

Меншіктеу амалы немесе меншіктеу операторы = (теңдік) белгісімен беріледі. Әдетте меншіктеу амалы тек қана бір айнымалы мәнін меншіктеу керек болса пайдаланылады. Айнымалының аты сол жақта, ал оның мәні оң жақта жазылады.

Мысалы:

```
<?
$var = 5;
echo ($var);
?>
```

Арифметикалық амалдар. Сандық деректер үстінде төмендегі арифметикалық амалдар орындалады:

+	қосу
-	алу
/	бөлу
*	көбейту
%	арифметикалық модуль – бүтін бөлу амалының қалдығы
++	инкремент – операнд мәнін бір бірлікке арттыру
--	декремент - операнд мәнін бір бірлікке азайту

Арифметикалық амалдағы нәтиженің типі, амалда қатысатын деректердің типімен анықталады.

Мысалы:

```
$c = 4*$a-$b;   - бірнеше арифметикалық амал қолданылған;
$c = $c++;      - инкремент амалы қолданылған;
$c = $c--,      - декремент амалы қолданылған;
```

PHP тілінде кейде амалдарды қысқартып жазуға болады. Мысалы, $a = $a + $b өрнегін қысқартылған түрде былай жазуға болады, $a + = $b. Мұндай әрекеттер жоғарыда көрсетілген барлық амалдарға рұқсат етілген.

Арифметикалық амалдарды операндтардың санына қарай екіге бөледі, унарлы және бинарлы. Унарлы амал бір операндпен жұмыс істейді, ал бинарлы екі операндпен жұмыс істейді. Инкремент және декремент амалдарынан басқа барлық арифметикалық амалдар бинарлы болып есептеледі.

Енді инкремент және декремент амалдарын қарастырайық. Бұл амалдарды префиксті және постфиксті деп екіге бөледі. Инкременттің префикс амалында операнд мәнінің өсуі, оны қолданғанға дейін орындалады. Ал постфиксті инкремент амалында операнд мәнінің өсуі, оны қолданғаннан кейін орындалады. Осылай инкремент және декремент амалдарын екі түрде қолдануға болады.

Мысалы:

```
<?
// префиксті инкремент амалы
$var = 1;
++$var;
// постфиксті инкремент амалы
$var = 1;
$var++;
?>
```

Келтірілген мысалда префикстік және постфикстік жазулар арасындағы айырмашылық үлкен емес, екеуінің де нәтижесі бірдей. Бірақта бұл амалдармен жұмыс барысында мұқият болған жөн. Инкремент амалының дұрыс қолданбағанына мысал келтірейік.

Мысал:

```
<?
// 1 - жағдай – префиксті инкремент амалы
$var = 1;
Echo (+ + $var);
// 2 – жағдай – постфиксті инкремент амалы
$var = 1;
echo ($var + +);
?>
```

Бірінші жағдайда біз 2 санын аламыз. Себебі инкременттің префикстік амалы бірінші амалды орындап, одан кейін алынған мәнді шығарады. Екінші жағдайда нәтиже ретінде біз 1 санын аламыз.

Инкремент және декремент амалдарында операнд тұрақты болмауы қажет. Мысалы, ++1 жазуы дұрыс емес, сол сияқты, ++($a + $b) жазуы да қате, яғни операнд ретінде екі айнымалылы өрнекті қолдануға болмайды. Инкремент және декремент амалдарында операнд тек қана біреу болуы тиіс.

PHP тілінде инкремент және декремент амалдарын тек қана integer типті бүтін айнымалыларға ғана емес, басқа типтерге де қолдануға болады. Инкремент амалын string типіндегі айнымалыға қолданып көрейік.

Мысалы:

```
<?
$var = “aaa”;
echo (+ + $var);
?>
```

Нәтижеде біз aab қатарын аламыз, себебі алфавитте b әрпі a әрпінен кейін келеді.

Биттік амалдар арифметикалық амалда сияқты сандық мәндерді өңдейді.. Бұл жерде бүтін сандар биттер қатары болып көрсетіледі.

~	инверсия – теріске шығару (not)
&	конъюнкция – логикалық көбейту (and)
|	дизъюнкция – логикалық қосу (or)
^	логикалық алу (xor)
<<	солға жылжыту
>>	оңға жылжыту

Биттік амалдардың мәндері:

A	b	~a	~b	a & b	a | b	a ^ b
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	1	0

Мысалы:

```
<?
echo ( 4 << 2 ); // нәтиже 16-ға тең болады
echo “<br>”;
echo ( 5>>1 ); // нәтиже 2-ге тең болады
echo “<br>”;
echo ( 6 & 5 ); // нәтиже 4-ге тең болады
echo “<br>”;
echo ( 6^5 ); // нәтиже 3-ке тең болады
echo “<br>”;
?>
```

Осы мысалдарды талдайық, 4 саны үшін екілік код 100-ге тең, екі разряд солға қарай жылжытсақ, 100 коды 10 000 болады, бұл ондық кодта 16 санына тең. Әдетте, n позиция солға қарай жылжыту 2^n –ге көбейткенге тең, ал оңға қарай жылжыту осы мәнді 2^n рет кішірейтеді және нәтиженің бөлшек бөлігін тастайды (сондықтан 5>>1 амалы 2-ге тең). Қалған мысалдар осы сияқты логикалық жолмен қарастырылады.

Салыстыру амалдары. Деректер үстінде төмендегі салыстыру амалдары орындалады:

>	үлкен
>=	үлкен немесе тең
<	кіші
<=	кіші немесе тең
= =	тең
!=	тең емес

Салыстыру амалының нәтижесі *true* немесе *false* мәндеріне ие. Бұл амалдарда деректер кез келген типке, ал нәтиже тек қана логикалық типке тиісті.

Логикалық амалдар. Логикалық амалдар тек қана логикалық *true* және *false* мәндері үстінде орындалады. Логикалық теріске шығару амалы бір аргументті, ал қалған амалдар екі аргументті.

!	теріске шығару (not)
&&	конъюнкция (логикалық көбейту) (and)
||	дизъюнкция (логикалық қосу) (or)

A	b	!a	!b	a && b	a || b
False	false	true	true	false	false
False	true	true	false	false	true
True	false	false	true	false	true
True	true	false	false	true	true

Өрнектердегі амалдардың орындалу тәртібі амалдар приоритетімен анықталады. Амалдар приоритетін жақшалармен өзгертуге болады. Өйткені жақша ішіндегі амалдар бірінші орындалады. Өрнектегі приоритеті бірдей амалдар солдан оңға қарай орындалады.

Приоритет	*аталуы*	*амалдар*
I	унарлық	-, ++, --, ~, !
II	мультипликатив	*, /, %, &&
III	аддитив	+, -, ||
IV	салыстыру	>=, >, <, >=, = =, !=

7.3 Операторлар

Қарапайым және құрама операторлар. Жоғарыдада көрсетілген Echo (5 + 5) функциясы қарапайым операторға мысал болады. PHP-де барлық оператор ; (нүктелі үтір) белгісімен аяқталады.

Қарапайым оператор бірнеше қатарда жазылуы да мүмкін. Мысалы, мына екі кодтың мағынасы бірдей.

```
<?
echo (5 + 5);
echo (5 - 2);
echo (“Hello”);
?>

<?
echo (5   +
5); echo (5 –
```

```
2); echo (“Hello”
);
?>
```

Қарапайым операторлардан басқа тағы құрама оператор да бар, олармен операторлар тізбегін анықтауға да болады. Кейде құрама операторларды көп қатарлы деп те атайды. Құрама операторлардың қарапайым операторлардан айырмашылығы, ол нүктелі үтірмен аяқталмайды, оның аяқталғанын фигуралық жақша білдіреді.

If шартты операторы. Шартты оператор деректерді өңдемейді, өңдеуді басқарумен айналысады. Операторлардың жалпы форматы:

If (шарт) {оператор1} else {оператор2}

Егер шарт орындалса, яғни логикалық шарттың мәні true болса, онда оператор1 орындалады. Ал керi жағдайда, яғни логикалық шарттың мәні false болса, онда оператор2 орындалады.

Мысалы:

```
<?
$flag = true;
if ($flag)
{
echo “<p>” flag айнымалысы ақиқат мәнді қабылдады. “</p>”
}
else
{echo “<p> flag айнымалысы жалған мәнді қабылдады “</p>”
}
?>
```

Көрсетілген мысалда шартты оператор алдымен flag айнымалысын тексереді. Егер шарт қанағаттанарлық болса, онда бірінші фигуралық жақшаның ішіндегі оператор орындалады, ал егер шартты қанағаттандырмаса, онда екінші фигуралық жақшаның ішіндегі оператор орындалады.

Шартты операторды қысқартылған түрде де жазуға болады:

If (шарт) {оператор}

Мысалы:

```
<?
$flag = true;
if ($flag)
{
echo “<p>” flag айнымалысы ақиқат мәнді қабылдады. “</p>”
```

```
}
?>
```

Егер шартты операторда тек қана жалғыз шарт қарастырылса, онда логикалық шарт онша күрделі болмайды. Ал егерде екі немесе одан да көп шарттар қарастырылса, онда шарт құрауда логикалық амалдар қолданылады.

Мысалы:

```
<?
$flag1=true
$flag2=true
if ($flag1 && $falg2)
{
echo “<p>” Екі шарттың мәні ақиқат “</p>”;
}
else
{
echo “<p>” Екі шарттың кемінде біреуі жалған“</p>”;
}
?>
```

Switch таңдау операторы. Таңдау операторы айнымалыға берілген бірнеше мәннің ішінен біреуін таңдауда қолданылатын оператор. Осы таңдалған мәнде жазылған операторлар тізбегі орындалады. Switch операторының жалпы форматы:

```
Switch (expression)
{
case value:
statements;
break;
case value:
statements;
break;
default;
statements;
}
```

Таңдау операторы алдымен expression айнымалысының мәнін қабылдайды, содан кейін case операторындағы тұрақтылардың бірін таңдайды да, сол блоктағы операторларды break операторы кездескенше орындайды.

Цикл операторлары. Цикл операторы бағдарламаның бір бөлігінің бірнеше рет қайта-қайта орындалуын қамтамасыз етеді. PHP тілінде цикл операторының төрт түрі бар:

- while

- do…while
- for
- forreach

Алдыңғы үш цикл операторлары C++ тілінен алынған болса, ал соңғы оператор Perl тілінен алынған.

While цикл операторы қайталаулар саны логикалық шартқа байланысты болған жағдайда қолданылады. While операторында басында логикалық шарт тексеріліп, содан кейін операторлар тізбегі орындалады. While операторының жалпы форматы:

```
While (шарт)
{
операторлар тізбегі;
}
```

Циклдың орындалу тәртібі мынадай, алдымен логикалық шарт тексеріледі. Егер шарт ақиқат болса, онда операторлар тізбегі орындалады.

Мысал:

```
<?
$var = 5;
$i = 0;
while (++$i <= $var)
{
echo($i);
echo('<br>');
}
?>
```

Циклдан шығу үшін break операторы қолданылады. Бұл оператор орындалу барысында кездессе, онда циклдің орындалуы тоқтайды.

Мысал:

```
<?
$var=7;
$i=0;
while (++$i<=$var)
echo($i);
echo('<br>');
if ($i==3) break;
}
?>
```

Егер тек қана орындалып жатқан оператордан шығу керек болады, онда continue операторы қолданылады.

Мысалы:

```
<?
$var = 7
$i = 0;
```

```
while (++$i <=$var)
{
if ($i==5)
{
continue;
}
echo($i);
echo('<br>');
}
?>
```

Do…while операторы. Оператордың жалпы форматы:

```
Do
{
операторлар тізбегі;
}
While (шарт);
```

Бұл оператор while операторына ұқсас, тек қана логикалық шарт цикл соңында тексеріледі. Сондықтан операторлар тізбегі кемінде бір рет орындалады. Цикл соңында шарт тексеріледі, егерде шарт қанағаттанарлық болса, онда цикл жалғасады.

Мысал:

```
<?
$var = 5;
$i = 0;
do
{
echo($i);
echo('<br>');
}
while(++$i<=$var)
?>
```

for цикл операторы қайталаулар саны белгілі болғанда қолданылады. Оператордың жалпы форматы:

```
For (мән1, мән2, мән3)
{
операторлар тізбегі;
}
```

мұндағы: мән1 - циклдің бастапқы мәні; мән2 – логикалық шарт; мән3 – операторлар тізбегі.

Мысал:

```
<?
$var = 5;
$i = 0;
```

```
for ($i=0; $i<=$var; $i++)
{
echo($i);
echo('<br>');
}
?>
```

Бағдарлама кодында циклдар кейде бір-бірінің ішінде орналасуы мүмкін. Мұндай жағдайда ішінде орналасқан циклды ішкі цикл деп, ал сыртында орналасқан циклді сыртқы цикл деп айтады.

Мысал:

```
<?
For ($i=2; $i<100; $i++)
{
for ($j=2; $j<$i; $j++)
{
if (($i%$j)!=0)
continue;
else
{
$flag=true;
break;
}
}
if (!$flag)
{
echo($i);
echo(" ");
}
$flag=false;
}
?>
```

7.4 Функциялар

PHP тілінде функция бағдарламаның өз алдына орындалатын бөлігі ретінде қарастырылады. Функцияны бағдарламаның қажет жерінде, қалағанша шақырып орындатады. Қарапайым түрде функция бірнеше айнымалылардан тұрады. Функция осы берілген айнымалыларды пайдаланып қандайда бір іс-әрекетті орындайды және нәтижені бағдарламаға қайтарады.

PHP тілінде функцияны сипаттаудың жалпы форматы:

```
Function Myfunction (айнымалы)
{
операторлар;
```

```
return айнымалы;
}
```
мұндағы, Function - қызметші сөз; Myfunction – функцияның атауы.

Егер функцияда айнымалылар бірнешеу болса, онда олар үтірлермен бөлектеніп жазылады.

Мысалы:

```
Function Myfunction ($var, $var1, $var2)
```

Егерде функцияда нәтиже негізгібағдарламаға қайтарылатын болса, онда функцияда міндетті түрде return операторы болуы керек.

Мысалы:

```
<?
Function get sum()
{
$var=5;
$var1=10;
$sum=$var+$var1;
return $sum;
}
echo (get sum());
?>
```

Көрсетілген мысалда екі санның қосындысы шығады. Бұл функция айнымалылардың мәнін қабылдайды да, осы екі айнымалының қосындысын нәтиже ретінде негізгі бағдарламаға қайтарады. Сонан соң echo операторын шақырады да, нәтижені браузер терезесіне шығарады.

Осы мысалды жалғастырайық, енді алынған нәтиже негізгі бағдарламаға қайтарылмай, тікелей браузерге шығарылсын:

```
<?
Function get sum ()
{
$var=5;
$var1=10;
$sum=$var+$var1;
echo $sum;
}
get sum ();
?>
```

Функцияда $var және $var1 айнымалыларын бағдарлама ішінде сипаттау қажет емес.

```
<?
Function get sum ($var ,$var1)
{
$sum =$var+$var1;
```

```
echo $sum;
}
get sum (5,2);
?>
```

Егер функцияда айнымалы аргумент арқылы берілсе, онда ол параметр функциясы деп аталады. Мысалы:

```
<?
Function get sum ($var)
{
$var=$var+5;
returm $var;
}
$new var =20;
echo(get sum($new var)); // 25 шығады
echo ("<br>$new var"); // 20 шығады
?>
```

Мысалда көрсетілгендей 5 және 20 сандары аргумент болады, ал айнымалылар $var және $var1 - функция параметрлері деп аталады.

Сонымен қатар, айнымалыларды локальды және ауқымды деп екіге бөледі. Локалды айнымалылар тек қана осы функция ішінде, ал ауқымды айнымалылар бағдарламаның кез-келген жерінде анықталады.

Локальды айнымалыны ауқымды етуге болады. Ол үшін айнымалыны global қызметші сөзімен сипаттайды, сонда оны кез келген жерден шақыруға болады. Айнымалының өмірі деп бағдарламаның орындалуы уақытын атайды. Мысалы:

```
<?
Function get sum ()
{global $var;
$var=5;
echo $var;
}
$var =10;
echo ("$var<br>");  // 10 шығады
get sum (); // 5 шығады
?>
```

Локальді айнымалы шақырылу кезінде жаңадан сипатталады. Мысалы:

```
Function   counter ()
{
$counter = 0;
retutrm ++ $counter
}
```

Сонымен қатар, локальды айнымалы қайтадан шақыруда өз мәнін сақтап қалу үшін, оны сипаттауда static қызметші сөзінен пайдаланады.

```
Function   counter ()
```

```
{
static $counter = 0;
retutrm ++$counter
}
```

PHP тілінде көптеген стандарт функциялар анықталған. Олар PHP-скрипттерде көп қолданылады. Оларды PHP-скрипт кезекпен орындайды. Енді осы стандарт функиялардың жиі кездесетіндерін қарастырайық.

Мысалы: PHP-скрипт тұтынушы сайтқа кіргенде, оның IP-адресін, хосттағы браузер типін анықтау, мәліметтерді сұрыптау, файлға мәлімет енгізу сияқты іс-әрекеттерді орындау.

```
<?
Function get brows() // тұтынушы браузерінің типін анықтау
{
…
}
Function get  os () // тұтынушы операциялық жүйесін анықтау
{
…
}
Function get ip () // тұтынушының IP-адресін анықтау
{
…
}
Function get host () // хост атын анықтау
{
…
}
Function is search engine ($referer) // іздеу жүйесімен сайтқа кіру
{
…
}
Function displeyDate () // мәліметтерді сұрыптау және
// оларды браузерге шығару
{
…
}
Function main ()   // сипатталған функцияның орындалуы
{
get brows ();
get os ();
get referer ();
get ip ();
…
createDate ();
getdate ();
```

```
displeyDate ();
}
main ();
// main () функциясын шақыру
?>
```

Strcmp функциясы. Жалпы форматы:

```
Int strcmp (string str1, string str2)
```

Strcmp функциясы екі қатарды салыстырады:

- егер қатарлар бір-біріне сәйкес келсе, онда 0 саны шығады;
- егер str1 қатары str2 қатарынан үлкен болса, онда 1 саны шығады;
- егер str1 қатары str2 қатарынан кіші болса, онда -1 саны шығады.

Мысалы:

```
<?
$str1= “ttt”;
$str2= “tttttttt”
echo (“($str1, $str2): -салыстыру нәтижесі”);
echo (strcmp (str1, str2)); echo (“<br>”);
echo (“($str2, $str1): -салыстыру нәтижесі ”);
echo (strcmp (str2, str1)); echo (“<br>”);
echo (“($str1, $str1): -салыстыру нәтижесі ”);
echo (strcmp (str1, str1));
>
```

Нәтиже:

```
(ttt, tttttttt): -1 -салыстыру нәтижесі
(tttttttt, ttt):1 -салыстыру нәтижесі
(ttt, ttt): 0 -салыстыру нәтижесі
```

Strncmp функциясы. Егер қатарды регистрді есептемей салыстыру қажет болса, онда Strcasecmp () функциясын қолдансақ болады. Функцияның жалпы форматы:

```
Int strncmp (string str1, string str2, int len)
```

Бұл функцияның strncnp функциясынан айырмашылығы, функция қатар басын салыстырады. Дәлірек айтсақ, бірінші len байттарын салыстырады. Егер len ең кіші қатар ұзындығынан да кіші болса, онда қатарлар толық салыстырылады.

8 PHP-скрипттерді құру технологиясы

8.1 Apache локальді серверін орнату

Алдыңғы тарауда тұтынушының интерактивтік интерфейсін құру технологиясын қарап шықтық. Динамикалық Web-құжатта CGI технологиясын пайдалану үшін, HTML және PHP тілдерінен қолданамыз. Нақты айтар болсақ, клиенттік бөлімді жазуда HTML тілінен, ал cgi-модульді жазуда PHP тілінен пайдаланамыз. Сондықтан Apache серверін орнату қажеттігі туындайды. Бұл бөлімде осы Apache серверін және PHP препроцессорын орнату технологиясын қарастырамыз.

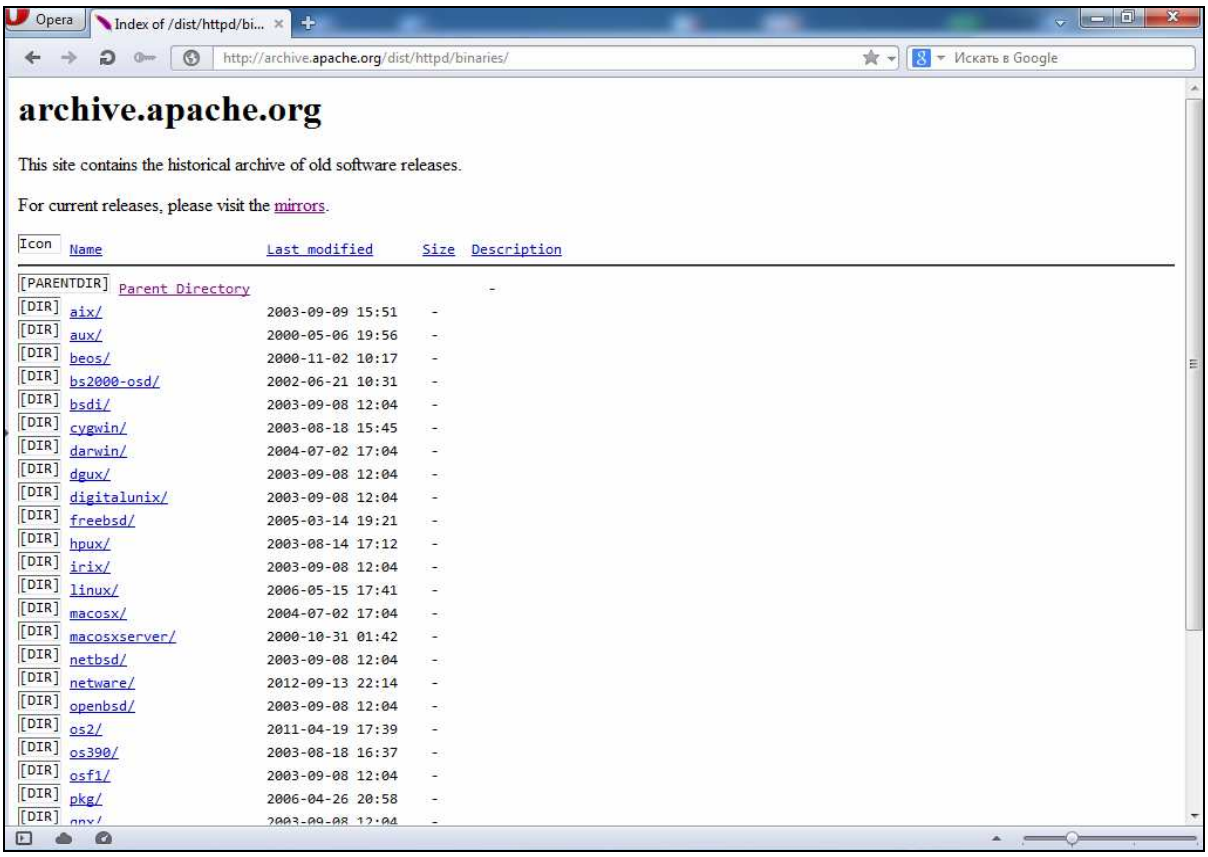
Жалпы, Apache локальды серверін пайдаланудың бірнеше себептері бар. Біріншіден Интернет негізінен Unix немесе Linux амалдық жүйелері негізінде жұмыс жасайды. Ал дербес компьютерлерде MS Windows амалдық жүйесі кең таралған. Сондықтан Linux амалдық жүйесін орнату қосымша қиындықтар туындатады. Екіншіден, cgi-модульді жазу үшін PHP тілінен пайдаланымыз деп айтық, ал PHP препроцессорынан пайдалану үшін Web-сервер қажет. Ал Web-құжатты Интернеттегі хостингде тестілеу қымбатқа түседі.

Apache сервері 1995 жылы жазылған. Сервер алғашқыда NCSA HTTPd 1.3 серверінің қателерін туралау мақсатында құрылған. Оны құрастырушылар Apache Software Foundation қауымдастығына біріккен. Сервердің атауы Солтүстік Американдағы үндіс халқының апачи тайпасының атауынан алынған. 1996 жылдан бастап Интернеттің ең көп тараған HTTP-сервері болып саналады. Сонымен қатар сервер тегін таратылады. Apache сервері туралы соңғы деректерді Apache Software Foundation ресми сайтынан көруге болады [21].

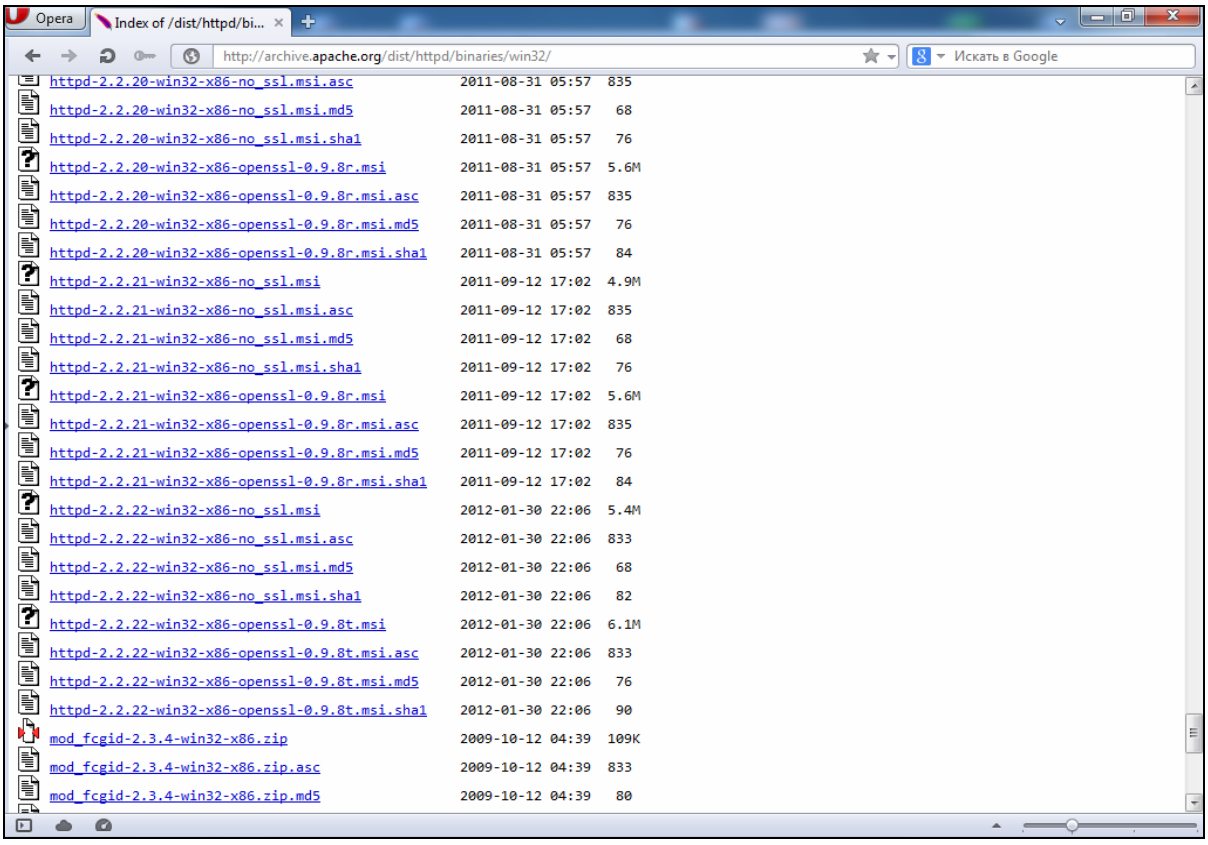
Жұмысты Интернеттен Apache архивін көшіріп алудан бастаймыз [22]. Көшіріп алуда мына жайтты ескерген жөн. Apache сервері Unix амалдық жүйесіндегі атауы бойынша httpd деп аталады. Жалпы, сайт бетінде көптеген гиперсілтемелер бар. Оның себебін жоғарыда айтып кеттік, Apache серверін құрушылар өте көп. Сондықтан көрсетілген тізімнен біреуін таңдау қажет (сурет 53).

MS Windows амалдық жүйесіне арналған версиясы win32 деп аталады. Сондықтан win32/ сілтемені таңдаймыз, терезеде MS Windows амалдық жүйесіне арналған Apache архивтерінің тізімі пайда болады. Apache архивтері тізімінен мына файлдардың біреуін көшіріп аламыз (компьютердегі MS Windows версиясына байланысты) (сурет 54):

apache_2.0.48-win32-x86-no_ssl.exe
apache_2.0.49-win32-x86-no_ssl.exe
apache_2.0.50-win32-x86-no_ssl.exe
apache_2.0.53-win32-x86-no_ssl.exe
apache_2.2.4-win32-x86-no_ssl.msi

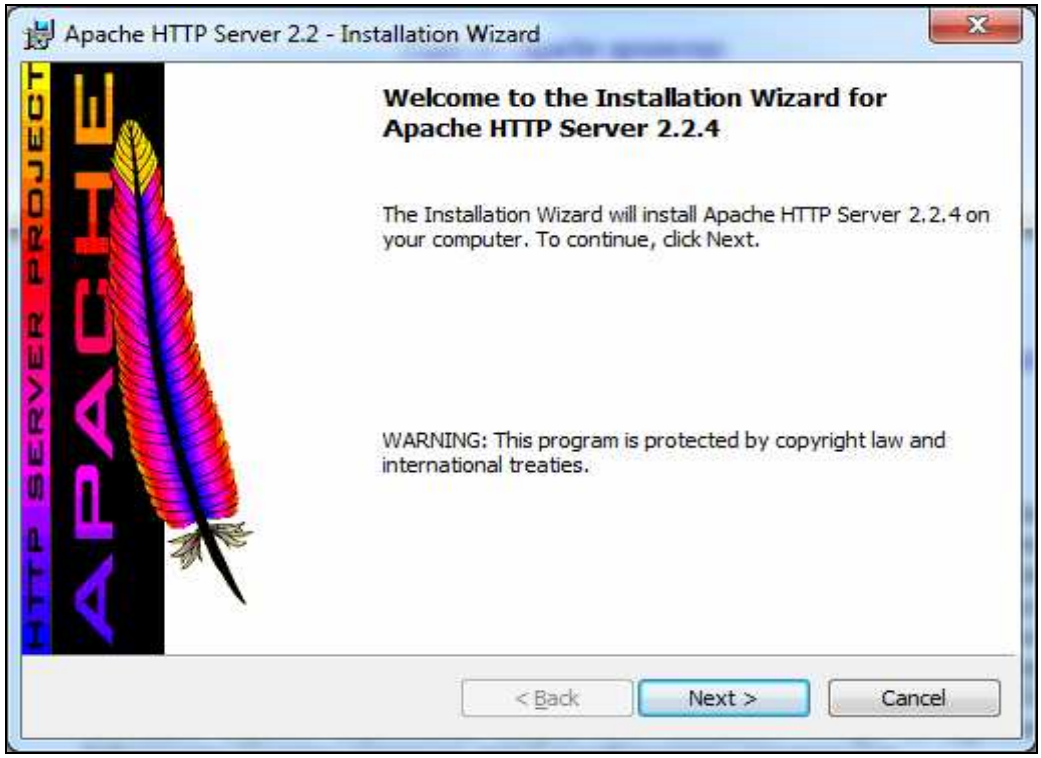


Сурет 53 – http://archive.apache.org/dist/httpd/binaries/ сайты

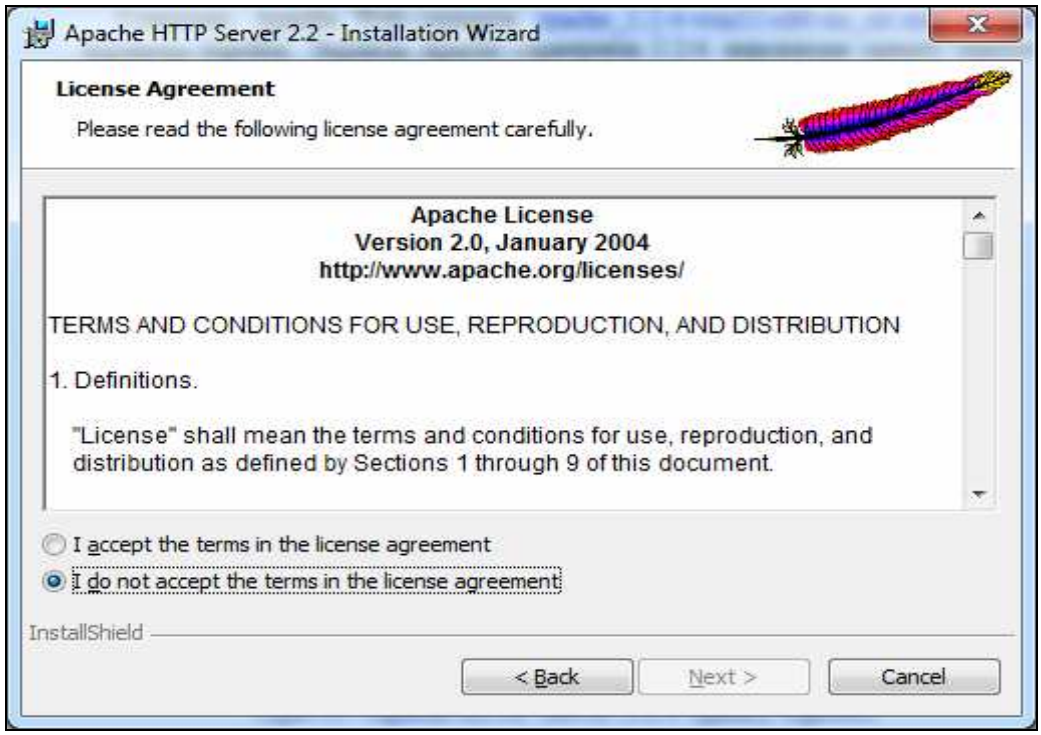


Сурет 54 – Apache архивтері

Сонымен, Apache Web-серверін орнату үшін apache_2.2.4-win32-x86-no_ssl.msi архивін жұмысқа берейік. Экранда Apache серверінің 2.2.4. версиясын орнату терезесі пайда болады (сурет 55). «Next» пернесін басып келесі терезеге өтеміз. Экранда Apache серверінің лицензиялық келісім терезесі пайда болады (сурет 56).

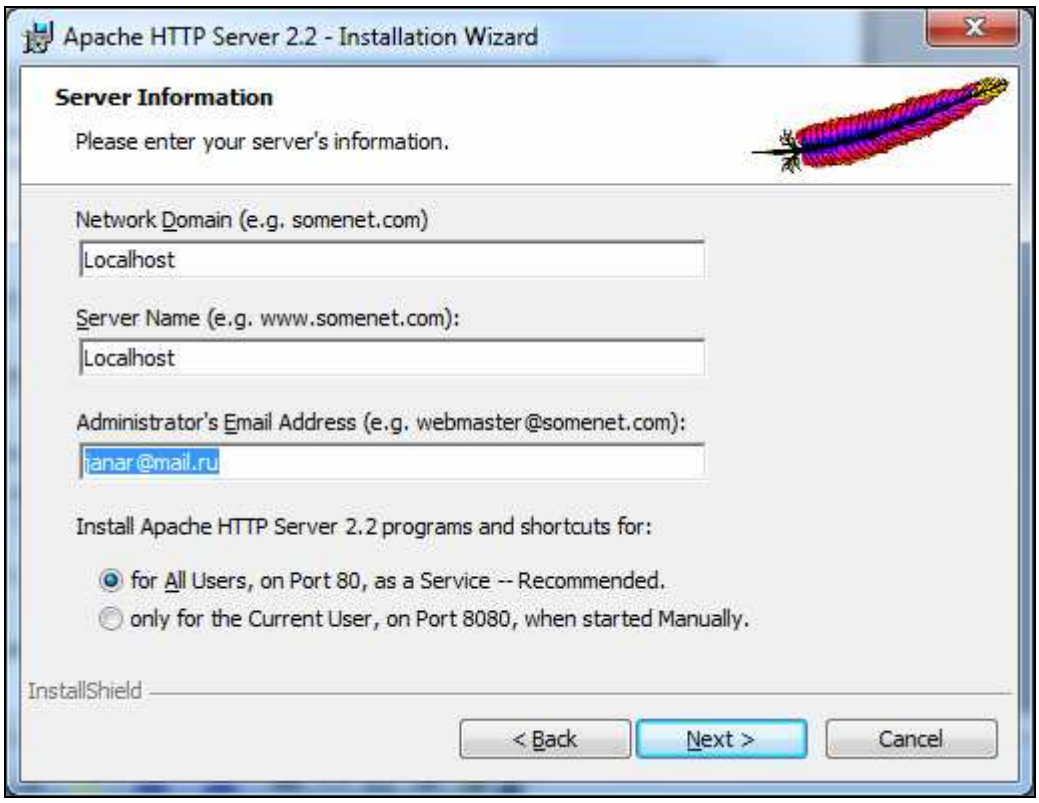


Сурет 55 – Apache HTTP Server 2.2.4. орнату терезесі



Сурет 56 – Apache серверінің лицензиялық келісім терезесі

Осылай «Next» пернесін баса отырып, Apache серверін орнатамыз. Келесі терезе сервер туралы ақпарат енгізуге сұрайды, олар: сервердің домендік аты, сервердің аты, тұтынушының электрондық поштасы. Егер орнату локальды машинада жүретін болса, домендік атау мен сервер атауына өріске localhost жазуын енгізу керек болады. Localhost бұл IP-адреспен байланысты локалды машинада серверді пайдалануға арналған атау. Терезенің төменгі бөлігінде сервер сұранысты қабылдайтын порт нөмерін таңдау керек болады (8080 порт). Ол төмендегі суретте көрсетілген (сурет 57).

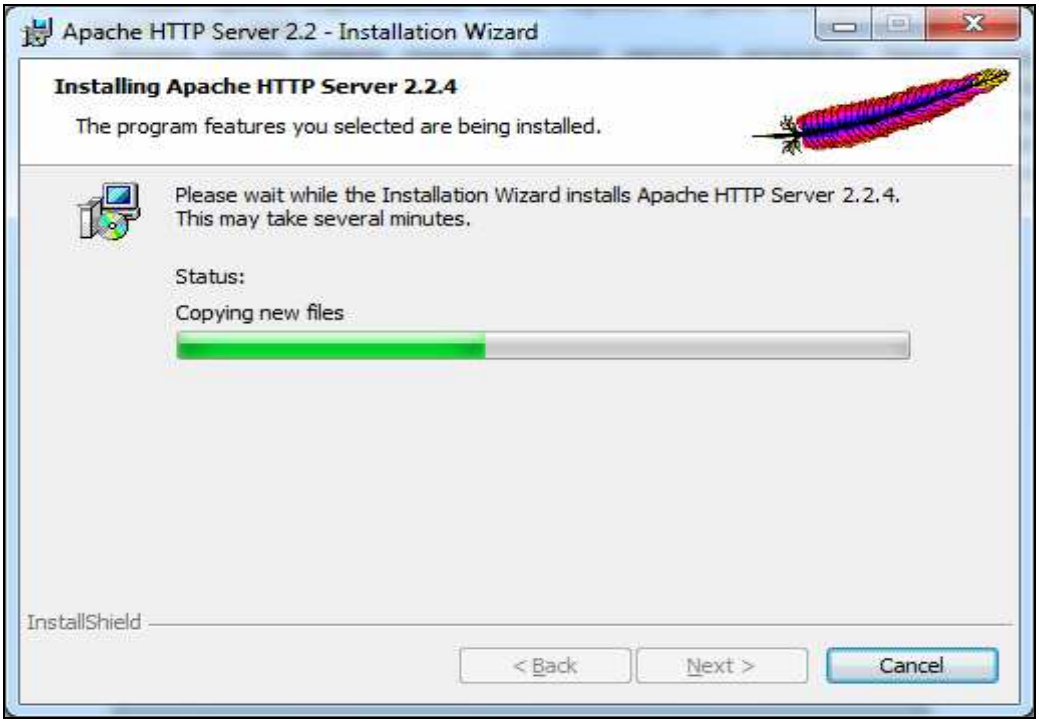


Сурет 57 – Apache параметрлерін енгізу

Әрі қарай Apache серверін орнату әдістері ұсынылады. Typical (Стандартты) немесе Custom (таңдаушы). Бұл әдістер сервер компонентін таңдауға мүмкіндік береді. Typical әдісінде сервер компоненттері үндемеу қағидасы бойынша таңдалады, ал Custom әдісінде тұтынушы таңдайды. Әдетте Typical әдісі таңдалады. Apache серверін орнатудың функциялары автоматтандырылған. Тек қана шебердің кейбір сұрақтарына, мысалы, қай бумаға орнату қажеттігі сияқты, жауап берсек жетерлік.

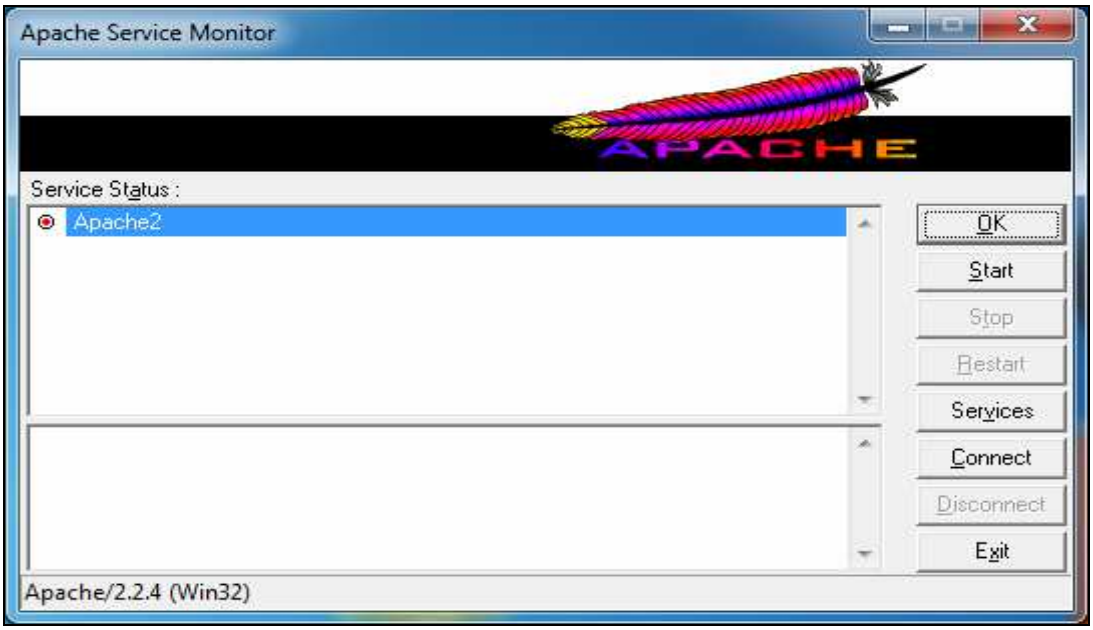
Келесі терезе серверді орнату бумасын таңдауға арналған. Әдетте серверді C: /Program Files /Apache Software Foundation бумасына орнатады, дегенмен біз басқа буманы таңдауды ұсынамыз. Мысалы, C:/LocalWeb. Осыдан соң орнату шебері орнату үрдісіне дайын екендігін хабарлайды. Терезедегі «Install» пернесін шертсек, Apache сервері орнатылады (сурет 58).

Егер Apache сервері сәтті орнатылса, Windows есептер панелінде сервердің логотипі шығып тұрады.



Сурет 58 – Apache серверін орнату

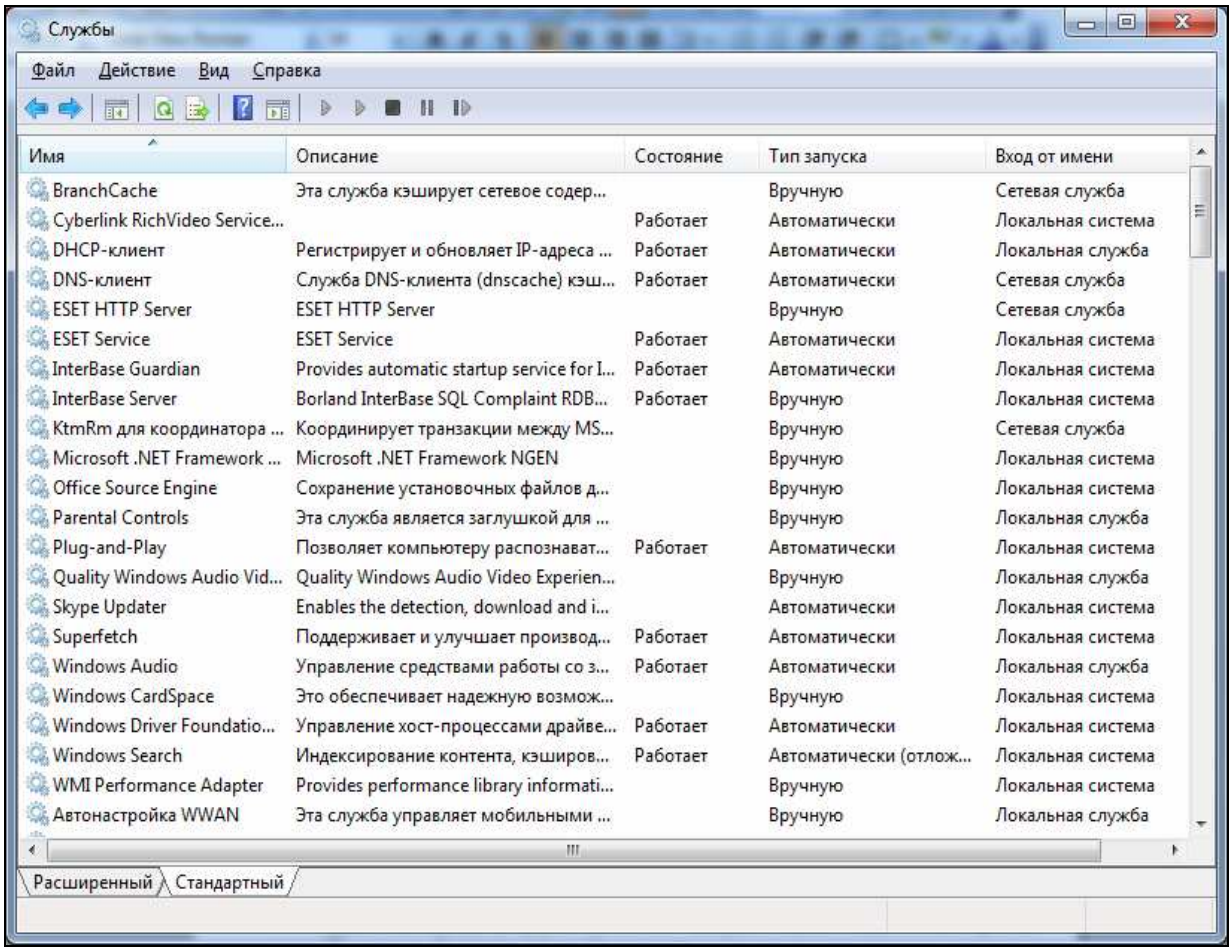
Әрі қарай Apache серверін басқару қажет. Атап айтсақ серверді іске қосуды, жұмыс істеуді және тоқтатуды үйрену керек. Бұл іс-әрекеттерді орындаудың үшін Apache серверінің «Apache Service Monitor» терезесін шақырамыз (сурет 59).



Сурет 59 – Apache Service Monitor терезесі

Терезенің бетінде, 59-суретте көрініп тұрғандай, серверді іске қосу (Start), тоқтату (Stop), қайта қосу (Restart) нұсқауларын орындауға болады.

Ал Apache серверін баптау үшін «Sevices» нұсқауын таңдау қажет (сурет 60).



Сурет 60 – Apache серверінің «Службы» терезесі

Әрі қарай C:/LocalWeb/Apach2/conf бумасында орналасқан Apache–http.conf Web-серверінің конфигурация файлын баптауға кірісу қажет. Айта кетейік, конфигурацияда амалға асырылған өзгерістер серверді қайта жүктегеннен кейін ғана орындалады.

Жалпы, Apache сервері ISO кодында жұмыс істейді, яғни сервер осы кодта жазылған файлдарды ғана өңдейді. Ал біздің компьютерде MS Windows амалдық жүйесі орнатылған. Сондықтан ISO кодын WinCode кодына өзгерту керек. Ескерте кетейік, Қазақстанда WinCode кодының 1251 парағы (Ресей парағы) қолданылады. Сонымен, кодты өзгерту үшін, http.conf файлын ашып, мына қатарды табу керек:

AddDefaultCharset ISO-8859-1

Осы қатарды мынадай етіп өзгерту керек:

AddDefaultCharset –Windows-1251

8.2 PHP препроцессорын орнату

Сонымен, Apache сервері құрылды. Әрі қарай PHP-ді құруға кірісу керек. Жұмысты Apache сервері мен PHP 5.4 препроцесорын байланыстырудан бастаймыз. PHP 5.4 препроцессорын PHP Group тобының ресми сайтынан алуға болады [23]. Сайтта PHP препроцессоры екі формада берілген, яғни алғашқы кодта (Complete Source Code) және компилицияланған нұсқада (Windows Binsries). Бізді екі нұсқада таралатын компиляцияланған версия қызықтырады: орнату түрінде (php-5.4.11-installer.exe) және zip архив түрінде (php-5.4.11-nts-Win32-VC9-x86.zip). Алғашқы файлды орнату ыңғайлы, бірақта бұл PHP версиясының ауқымы шектеулі. Сонымен қатар инсталляторды пайдалану үшін Apache серверінің конфигурация файлын құру қажеттілігі туындайды. Сондықтан да zip-архивті жүктеуді ұсынынамыз. Сол сияқты PHP Group сайтынан PHP 5.4 тіліне арналған құжаттарды да алуға болады. Мұндайда MS Windows амалдық жүйесін пайдаланушылар үшін chm-формат ыңғайлы.

Сонымен аталған сайттан zip-архивті көшіріп аламыз. Ол үшін D:/PHP бумасын құрамыз және осы бумаға zip-архивін ашамыз. Ал, PHP-ді орнату үшін бөлек бума құру қажет. Ол үшін, мысалы: C:/PHP бумасын ашамыз. Сонымен қатар Windows мәзірінен Пуск (Start) пернесін басып, нұсқау қатарына subst E: C:\PHP нұсқауын береміз. Нәтижеде бізде виртуалды E дискісі пайда болады. Енді сол дискте usr бумасын құрып, содан кейін usr бумасының ішіне php бумасын құрамыз. Сосын exe файлды жүктеп, пайда болған сұраныс терезелерінде Next батырмасын басып отыру қажет. Сұраныста Standart типін орнатқан жөн. Орнату жолына e:\usr\php қатарын көрсету қажет. Әрі қарай SMTP серверін және электронды пошта адрестерін енгіземіз. Содан кейін бізге PHP жұмыс істейтін серверді таңдау қажет болады. Ол сервер ретінде біз жоғарыда сипатталған Apache серверін аламыз.

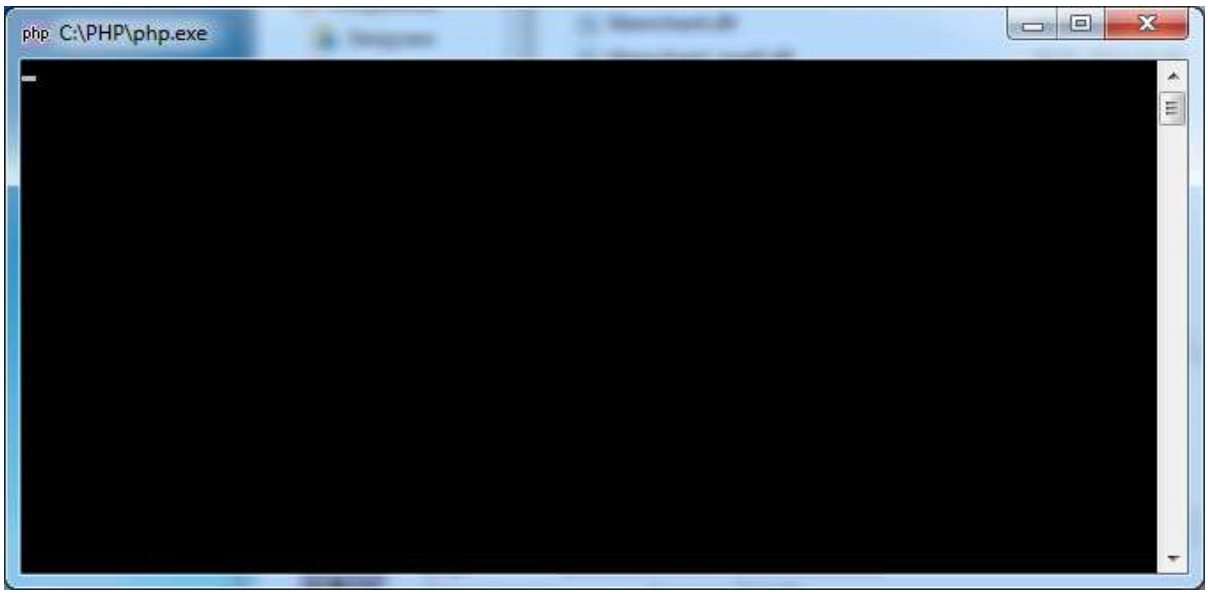
Енді бізге қосымша модульдерді орнату мен Apache серверіне PHP кодтарын байланыстыру қажет. PHP-ді Apache серверімен байланыстыру үшін E:/usr/apache/conf бумасынан Apache-дің httpd.conf файлын кез-келген мәтіндік редактормен ашамыз. Файл мәтінінен

```
# AddType application/x-httdp-php .php
```

қатарын іздейміз. Осы қатардан (#) түсініктеме белгісін алып тастау қажет. Әрі қарай, осы қатардан кейін

```
ScriptAlias /_php/ “PHP қатары”
Action application /x-httpd-php “/_php/php.exe”
```

қатарларын жазамыз. Мұндағы, «PHP қатары» қатары біздің мысалда e:/sr/php/. Бұдан кейін файлды әуелі сақтаймыз, сосын жабамыз.



Сурет 61 – PHP-дің консолдық терезесі

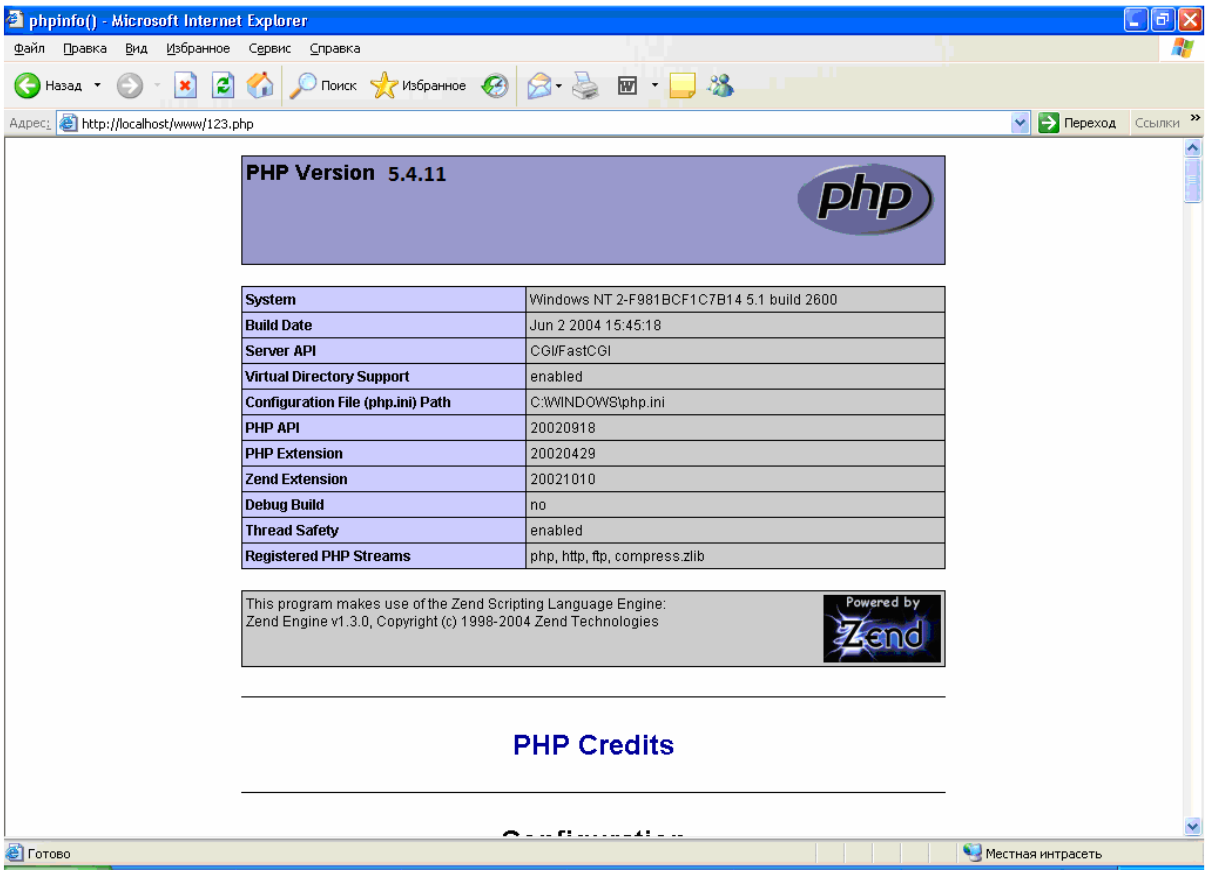
Apache серверінде бір компьютерде бірнеше виртуалды хосттар немесе жүздеген виртуалды серверлер болуы мүмкін. Сондықтан өзімізге сондай бір виртуалды сервер жасау қажет. Ол үшін тағы да сол httpd.conf файлын ашамыз және файлдың соңына келесі қатарларды жазамыз:

```
// <VirtualHost 127.0.0.1 >
// ServerAdmin admin@server
// rname.com
// Server
// erName www.servername.com
// DirectoryIndex index.php5 index.phtml index.php
// Index.htm Index.html Index.shtml Index.shtm
// DocumentRoot “e:/usr/public_html/host/virtual”
// ScriptAlias /cgi-bin/ “e:/usr/public_html/host/virtual/cgi-bin/”
// ErrorLog e:/usr/php
// ublic_html/host/virtual/logs/error.log
// CustomLog e:/usr/public_html/host/virtual/logs/access.log common
// </VirtualHost>
```

PHP скрипттері жұмысқа жарамды екендігін тексеру үшін, алдымен e:/usr /public_html/host /virtual бумасына test.php файлын құрып, ішіне келесі кодты жазамыз:

```
<? phpinfo(); ?>
```

Браузердің адрестік қатарына http://127.0.0.1 /test.php деп жазамыз. Егер экранда суретте көрсетілгендей PHP туралы ақпарат пайда болса, онда PHP препроцессоры жұмыс істеп тұр (сурет 62).



Сурет 62 – PHP процессоры жұмысын тексеру

Браузер Web-құжатта PHP-скриптті көргесін, Apache серверіне PHP препроцессор арқылы өңдеуге сұраныс береді. PHP-скрипттер HTML-құжатқа енгізілетін PHP-кодтарымен беріледі. HTML-құжатқа PHP-скрипт нұсқауларын енгізу үшін, оларды арнайы тэгтермен байланыстыру қажет. Оларды төмендегі төрт түрлі әдіспен береді.

1. // 1 – түрі – тэг XML стилімен
   <? php
   . . .
   ?>

2. // 2 – түрі – тэг SGML стилімен
   <?
   . . .
   ?>

3. // 3 – түрі – тэг ASP стилімен
   <%
   . . .
   %>

4. // 4 – түрі – тэг HTML стилімен

<scrip language = “ php ”>
. . .
</ script>

Сонымен PHP тілінде жазылған құжатты PHP-скрипт деп атайды. Скрипттер .php кеңеймесі бар мәтіндік файлдар. Оларды құруда қарапайым мәтіндік редакторлардан пайдалануға болады. Бірақта PHP-скрипттерді арнайы HTML-редакторларда құру ыңғайлы. Оларға мысал ретінде 1st Page 2000 [24], PHPEdit [25], Web Development Studio [26] редакторларын келтіруге болады.

8.3 PHP-скрипттерге мысалдар

Web-сайттарда жиі қолданылатын PHP-скрипттерді қарастырайық.

1. «Санақ» скрипті тұтынушының сайтқа қанша рет кіргені туралы ақпаратты сақтау үшін қолданылады.

Негізгі Web-құжатта орналасатын PHP-скрипт коды:

```
<center><a href="info.php" target='_blank'><img src="counter.php"
width="88" height="31" border=0> </a></center>
```

Web-сайтқа кіргендер санын көрсететін PHP-скрипт коды:

```
<?
#error_reporting (E_ALL);
include "config_counter.php";
$ldate="0";
if (is_file("$data/last.dat")) {$lline=file("$data/last.dat"); $li=count($lline); if
($li>0) {$ldate=$lline[0];}}
$datescribe=$ldate+7*86400;
$today=mktime();
if ($today>$datescribe) {include("infomail.php");}
function addSpace($num) {$strlen=17-strlen($num); $space=null; while($strlen)
{$space.=" "; $strlen--;} return $space.$num;}
function read_file($path)
{if(!is_file($path))return false;
elseif(!filesize($path))return array();
elseif($array=file($path))return $array;
else while(!$array=file($path))sleep(1);
return $array;}
function normal_numeric($number)
{if(!isset($number))return false;
else{$strlen=strlen($number);
```

```
$new=null;
for ($i=$strlen-1;$i>-1;$i--)
{$n = $i;$n++;if(strstr($n/3,."))$new.=$number[$strlen-1-$i];
else if($n!=$strlen)$new.=    " ".$number[$strlen-1-$i];
else$new.=$number[$strlen-1-$i];}
return $new;}}
if(!extension_loaded("gd")) {error("",date("Дата: d.m.Y. ya&#1179;ыт:
H:i:s",time())); exit;}
$ip=(isset($_SERVER['REMOTE_ADDR']))?$_SERVER['REMOTE_ADDR']:
0;
$a=$b=$c=null;
$today=date('d.m.Y',time());
$times=mktime();
if (isset($_SERVER["HTTP_REFERER"]))
{$from=$_SERVER["HTTP_REFERER"];} else {$from="";}
$who="";
if (isset($_SERVER["SCRIPT_NAME"]))
{$gde=$_SERVER["SCRIPT_NAME"];} else {$gde="";}
if (!is_file("$data/$today.dat")) {$OpenToday=fopen("$data/$today.dat","a");
fclose($OpenToday);}
$itogo=read_file("$data/all.dat"); if(!isset($itogo[0])) $itogo[0]=0;
$fp=fopen("$data/all.dat","w"); flock ($fp,LOCK_EX);
fwrite($fp,($a=$itogo[0]+1)); flock ($fp,LOCK_UN); fclose($fp);
@chmod("$data/all.dat", 0644);
$ft=fopen("$data/$today.dat","a"); flock ($ft,LOCK_EX);
fwrite($ft,"$ip|$times|$gde|$who|$from|\r\n"); flock ($ft,LOCK_UN);
fclose($ft); @chmod("$data/$today.dat", 0644);
$newlines=read_file("$data/$today.dat");
for ($i=0;$i < count($newlines); $i++) {$dt=explode("|", $newlines[$i]);
$lines[$i]=$dt[0];}
$b=count($lines); $c=count(array_unique($lines));
if(strlen($a)>9||!isset($a)) $a="?"; if(strlen($b)>9||!isset($b)) $b="?";
if(strlen($c)>9||!isset($c)) $c="?";
header("Expires: Mon, 01 Apr 2012 05:00:00 GMT");
header("Last-Modified: ".gmdate("D, d M Y H:i:s")." GMT");
header("Cache-Control: no-store, no-cache, must-revalidate");
header("Cache-Control: post-check=0, pre-check=0", false);
header("Pragma: no-cache");
header("Content-type: image/png".chr(10).chr(10));
$image=ImageCreateFromPNG("./images/".$image);
$color1=ImageColorAllocate($image,$s1r,$s1g,$s1b);
$color2=ImageColorAllocate($image,$s2r,$s2g,$s2b);
$color3=ImageColorAllocate($image,$s3r,$s3g,$s3b);
ImageString($image,1,0,2, addSpace(normal_numeric("$a")),$color1);
ImageString($image,1,0,12, addSpace(normal_numeric("$b")),$color2);
```

```
ImageString($image,1,0,21, addSpace(normal_numeric("$c")),$color3);
ImagePNG($image);
?>
```

Web-сайтқа кірген тұтынушылыр туралы жалпы ақпаратты монитор экранында диаграмма түрінде көрсететін PHP-скрипт

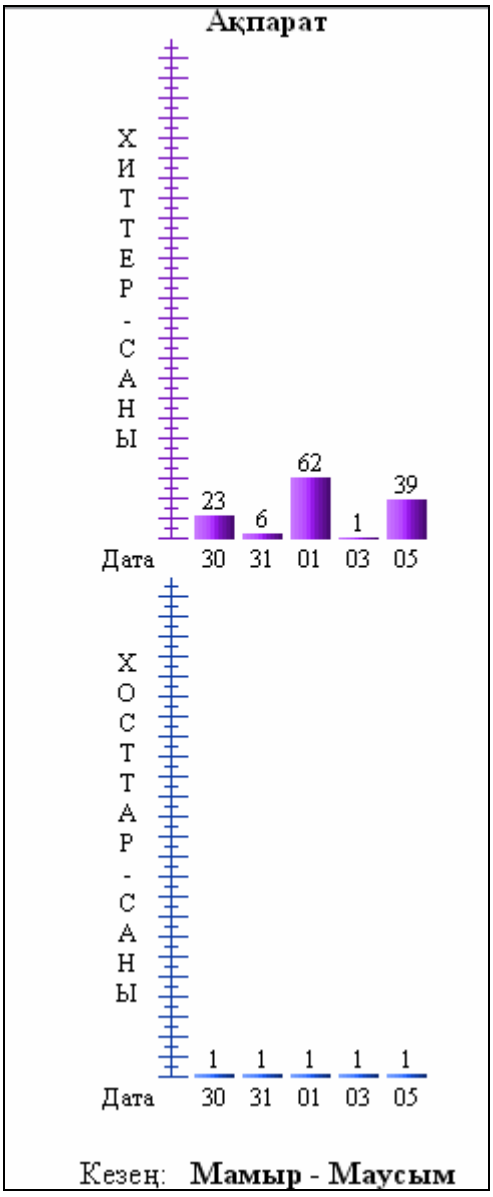
```
<?
#error_reporting (E_ALL);
include "config_counter.php";
function prcmp ($a, $b) {if ($a==$b) return 0; if ($a<$b) return -1; return 1;}
$months=array("&#1178;а&#1187;тар","А&#1179;пан","Наурыз","С&#1241;
у&#1110;р","Мамыр","Маусым","Ш&#1110;лде","Тамыз","&#1178;ырк&#1
199;йек","&#1178;азан","&#1178;араша","Желто&#1179;сан");
$deldt=mktime()-$days*86400;
$i=0; if ($handle = opendir($data)) {
while (($file = readdir($handle)) !== false)
if (!is_dir($file)) {$lines[$i]=$file; $i++;}
closedir($handle);
} else {print 'Бума бос!';}
$itogo=count($lines); $k=0; $text=null;
do {
$fline=file("$data/$lines[$k]");
$fitogo=count($fline);
if ($fitogo!=0 and $lines[$k]!="all.dat" and $lines[$k]!="last.dat" and
$lines[$k]!="mainbase.dat" and $lines[$k]!=".htaccess")
{
$thendayx=str_replace(".dat","",$lines[$k]);
$dt=explode(".",$thendayx);
@$then=mktime(0,0,0,$dt[1],$dt[0],$dt[2]);
$tekdate=date("d.m.Y",$then);
if ($deldt>$then) {unlink ("$data/$lines[$k]");}
usort($fline,"prcmp");
$numip="0"; $numsys="0"; $hi=0; $ab="0"; $ac="0";
do {
$dt=explode("|",$fline[$hi]);
if ($ab!=$dt[0]) {$ab=$dt[0]; $numip++;}
if (isset($dt[2])) {if ($ac!=$dt[2]) {$ac=$dt[2]; $numsys++;}}
$hi++;
} while ($hi<$fitogo);
$text.="$then|$fitogo|$numip|$numsys|\r\n";
}
unset($fitogo);
unset($fline);
$k++;
```

```
} while ($k<$itogo);
$fp=fopen("$data/mainbase.dat","w");
flock ($fp,LOCK_EX);
fputs($fp,"$text");
fflush ($fp);
flock ($fp,LOCK_UN);
fclose($fp);
@chmod("$data/mainbase.dat", 0644);
$lines=file("$data/mainbase.dat");
$maxi=count($lines);
usort($lines,"prcmp");
print"<HTML><head><META content='text/html; charset=windows-1251'
http-equiv=Content-Type></head>
<body text=#000000 leftMargin=0 topMargin=0 rightMargin=0
bottomMargin=0 marginheight=0 marginwidth=0><center>
<B>A&#1179;парат</B>
<table border=0 cellSpacing=0 cellPadding=0 width=\"98%\"><tr><td
valign=top>";
if (sizeof($lines)>=1) {
if ($gtype=="1") {
$xdaym="</tr><tr><td align=center><small>Дата</td><td> </td>\r\n ";
$graph1=""; $graph2="";
$g1shapka="<table border=0 cellSpacing=0 cellPadding=0 align=center><tr
align=center valign=bottom><td
valign=middle><small>X<BR>И<BR>T<BR>T<BR>E<BR>P<BR>-
<BR>C<BR>A<BR>H<BR>ЫI</td><td><img src='images/v1scale.gif'
border=0></td>\r\n";
$g2shapka="<table border=0 cellSpacing=0 cellPadding=0 align=center><tr
align=center valign=bottom><td
valign=middle><small>X<BR>O<BR>C<BR>T<BR>T<BR>A<BR>P<BR>-
<BR>C<BR>A<BR>H<BR>ЫI</td><td><img src='images/v2scale.gif'
border=0></td>\r\n";
for ($i=0; $i<$maxi; $i++)  {
$dtt=explode("|",$lines[$i]);
$dttn1=round($dtt[1]*$scale1);
$dttn2=round($dtt[2]*$scale2);
$graph1.="<td><table cellSpacing=0 cellPadding=0><tr><td width=24
align=center><small>$dtt[1]</small></td></tr><tr><td align=center><img
src='images/v1.gif' height=$dttn1 width=20></td></tr></table></td>\r\n";
$graph2.="<td><table cellSpacing=0 cellPadding=0><tr><td width=24
align=center><small>$dtt[2]</small></td></tr><tr><td align=center><img
src='images/v2.gif' height=$dttn2 width=20></td></tr></table></td>\r\n";
if (!isset($m0)) {$m0=date("m",$dtt[0])-1;}
if ($i==$maxi-1) {$m1=date("m",$dtt[0])-1;}
$xday=date("d",$dtt[0]);
```

```
$xdaym.="<td align=center><small>$xday</small></td>\r\n";
}
$mm1=$months[$m0]; $mm2=$months[$m1]; if ($mm1==$mm2) {$mm1="";}
else {$mm1.=" - ";}
$msdat="</tr></table><BR><center>Кезе&#1187;:   <B>$mm1
$mm2</B></center>\r\n";
print"$g1shapka $graph1 $xdaym </tr></table>";
print"$g2shapka $graph2 $xdaym $msdat <BR>";
} else {
$g1shapka="<table cellSpacing=0 cellPadding=0 align=center><tr align=center
valign=bottom><td valign=middle><small>Дата</td><td>ХИТТЕР-
САНЫ</td></tr><tr><td> </td><td><img src='images/g1scale.gif'
border=0></td>\r\n";
$g2shapka="<table cellSpacing=0 cellPadding=0 align=center><tr align=center
valign=bottom><td valign=middle><small>Дата</td><td>ХОСТТАР-
САНЫ</td></tr><tr><td> </td><td><img src='images/g2scale.gif'
border=0></td>\r\n";
$xdaym=""; $graph1=""; $graph2="";
for ($i=0; $i<$maxi; $i++) {
$dtt=explode("|",$lines[$i]);
//$dtt[0]=date("d.m.y",$dtt[0]);
if (!isset($m0)) {$m0=date("m",$dtt[0])-1;}
if ($i==$maxi-1) {$m1=date("m",$dtt[0])-1;}
$xday=date("d",$dtt[0]);
$dttn1=round($dtt[1]*$scale1);
$dttn2=round($dtt[2]*$scale2);
$graph1.="<tr><td align=center><small>$xday</small></td><td><table
cellSpacing=0 cellPadding=0><tr><td><img src='images/g1.gif' height=16
width=$dttn1></td><td align=center> 
<small>$dtt[1]</small></td></tr></table></td></tr>";
$graph2.="<tr><td align=center><small>$xday</small></td><td><table
cellSpacing=0 cellPadding=0><tr><td><img src='images/g2.gif' height=16
width=$dttn2></td><td align=center> 
<small>$dtt[2]</small></td></tr></table></td></tr>";
}
$mm1=$months[$m0]; $mm2=$months[$m1]; if ($mm1==$mm2) {$mm1="";}
else {$mm1.=" - ";}
$msdat="</tr><tr><td colspan=2><center><small>Кезе&#1187;:</small>
<B>$mm1 $mm2</B></center>\r\n";
print"<BR>$g1shapka $graph1 $xdaym </tr></table></td><td width='50%'
valign=top>";
print"<BR>$g2shapka $graph2 $xdaym </tr></table></td>$msdat<BR>";
} // else ($gtype)
print"</center>P.S. <B>Хиттер</B> - санауыш орнатыл&#1171;ан веб-
беттерге кіру саны;<BR>
```

```
       <B>Хосттар</B> - IP-адрестер
саны.<BR>";
}
else {print"<center><br><br>Бос<br> К&#1110;рулер орын
алмады<br><br>";}
?>
```

Скрипт нәтижесі келесі суретте көрсетілген (сурет 63).



Сурет 63 – «Санақ» скриптінің нәтижесі

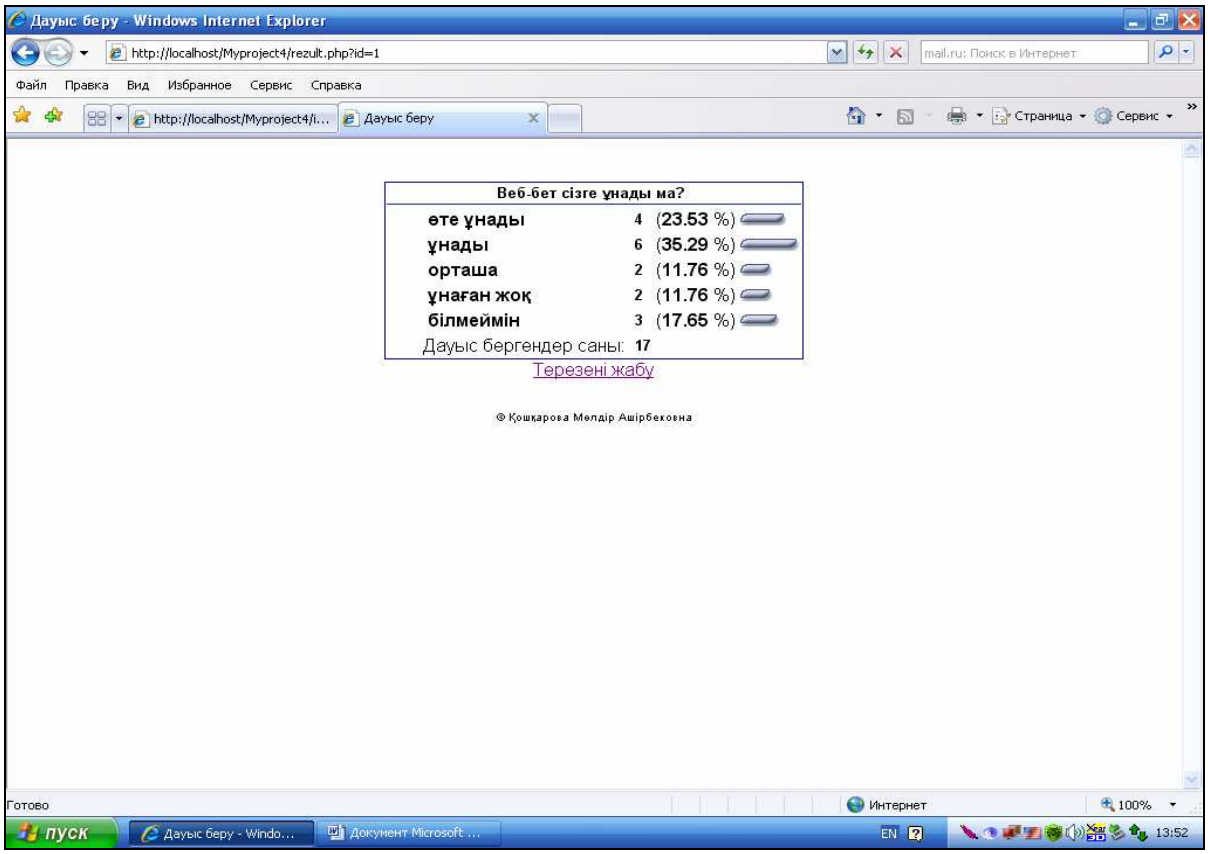
2. Web-сайтта «Дауыс беру» PHP-скрипті. Негізгі беттегі PHP-скриптті іске қосу коды төмендегіше жазылады:

```
<script language=JavaScript><!--
```

```
function gosub()
{WRSub=window.open('vote.php','WRGolos','width=450,height=350,left=250,top=100'); WRSub.focus();}
function gorez()
{WRSub=window.open('rezult.php','WRRezultGolos','width=450,height=350,left=250,top=100'); WRSub.focus();}//-->
</script>
<table cellPadding=3 align=center border=0><tbody><tr><td vAlign=top align=middle>
<table bgcolor=navy cellSpacing=1 cellPadding=0 align=center border=0><tr><td>
<table bgcolor=#ffffff  style="BACKGROUND-IMAGE: url(images/fonlogo3.jpg); BACKGROUND-REPEAT: no-repeat" cellSpacing=0 cellPadding=1 align=center border=0>
<form name=wrvote action='vote.php?id=1' method=post target='WRGolos'>
<tr><td colspan=3 align=middle bgColor=#FFFFFF><font face=arial size=2><B> Веб-бет сізге &#1201;нады ма?  </B></font></td></tr>
<tr><td colspan=3><img src='images/l1.gif' width=100% height=1></td></tr>
<tr><td><table border=0 cellSpacing=0 cellPadding=2 width=100%><tbody><tr>
<td width=15> </td><td><input name='votec' type=radio value='1'></td>
<td><font face=arial size=2><B> &#1257;те &#1201;нады</B></font></td>
</tr><tr>
<td width=15> </td><td><input name='votec' type=radio value='2'></td>
<td><font face=arial size=2><B> &#1201;нады</B></font></td>
</tr><tr>
<td width=15> </td><td><input name='votec' type=radio value='3'></td>
<td><font face=arial size=2><B> орташа</B></font></td>
</tr><tr>
<td width=15> </td><td><input name='votec' type=radio value='4'></td>
<td><font face=arial size=2><B> &#1201;на&#1171;ан жо&#1179;</B></font></td>
</tr><tr>
<td width=15> </td><td><input name='votec' type=radio value='5'></td>
<td><font face=arial size=2><B> білмеймін</B></font></td>
</tr><tr><td colspan=3 align=center>
<input type=image alt='Дауыс беру' src='images/vote.gif'  onClick='gosub();' border=0></td></tr>
</form></tbody></table>
</tbody></table>
</tbody></table>
<A href='rezult.php?id=1' onClick='gorez();' target='WRRezultGolos'>H&#1241;тижелер</A>
```

</td></tr></table>
<!—Дауыс беру скриптінің соңы -->

Скрипт нәтижесі келесі суретте көрсетілген (сурет 64).



Сурет 64 – «Дауыс беру» скриптінің нәтижесі

PHP-скрипттің негізгі коды:

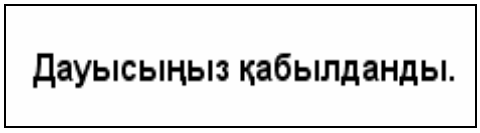
```
<?
error_reporting (E_ALL);
include "config.php";
$shapka="<html><head><title>Дауыс беру</title><META HTTP-
EQUIV='Pragma' CONTENT='no-cache'><META HTTP-EQUIV='Cache-
Control' CONTENT='no-cache'><META content='text/html; charset=windows-
1251' http-equiv=Content-Type></head><body>";
if (isset($_GET['id'])) {$id=$_GET['id']; if ((!ctype_digit($id)) or
(strlen($id)>2)) {exit("<B>Жауаптар саны 1-9 дейін!!!</B>");}} else {$id=1;}
$lines = file("$datadir/vote$id.dat");
$itogo=count($lines); $i=1; $glmax=0;
// Дауыстарды санау
do {$dt=explode("|",$lines[$i]); $glmax=$glmax+$dt[1]; $i++; }
while($i<$itogo); $i=1; $all=$glmax;
$vdt = explode("|",$lines[0]);
print"$shapka
```

```
<center><table cellPadding=3 align=center border=0><tbody><tr><td
vAlign=top align=middle>
<table border=0 bgcolor=navy cellSpacing=1 cellPadding=0
align=center><tr><td>
<table border=0 bgcolor=#ffffff  style=\"BACKGROUND-COLOR: white;
BACKGROUND-REPEAT: no-repeat\" cellSpacing=0 cellPadding=1
align=center border=0>
<form name=wrvote action='submit.php' method=post>
<tr><td colspan=3 align=middle bgColor=#FFFFFF><font face=arial
size=2><B> $vdt[0] </B></font></td></tr>
<tr><td colspan=3><img src='images/l1.gif' width=100% height=1></td></tr>
<tr><td><table border=0 cellSpacing=0 cellPadding=2 width=100%><tbody>";
do {$dt = explode("|",$lines[$i]);
if ($glmax==0) {$glmax=0.1;}
$glpercent=round(10000*$dt[1]/$glmax)/100;
$hcg=round($glpercent);
if ($glpercent<1) {$hcg=1;} if ($glpercent>100) {$hcg=100;}
print"<tr>
<td width=25> </td><td><B> $dt[0]</B></td>
<td><font face=arial size=2><B> $dt[1]</B></font></td>
<td>(<B>$glpercent</B> %)</td>
<td><img src='g1/lg.gif' border=0><img src='g1/cg.gif' width=$hcg height=11
border=0><img src='g1/rg.gif' border=0></td>
</tr>";
$i++;
} while($i<$itogo);
print"<tr>
<td> </td><td>Дауыс бергендер саны:</B></td>
<td><font face=arial size=2><B> $all</B></font></td>
<td> </td>
<td> </td>
</tr>
</form></tbody></table>
</tbody></table>
</tbody></table>
<a href='rezult.php' onClick='self.close()'>Терезені жабу</b></a>
</td></tr></table>";
?>
```

Дауыс берілгендігін, оның қабылданғандығын көрсететін скрипт. Нәтижеде, экранға



немесе



дауыс қабылданғаны туралы хабарламасы шығады.

```
<?
error_reporting (E_ALL);
include "config.php";
if (isset($_POST["votec"])) {$numv=$_POST["votec"];} else {print
"<BR><BR><BR><center> Сіз <B>  ешбір жауапты
та&#1187;дамады&#1187;ыз</B> </center>"; exit;}
$ip=(isset($_SERVER['REMOTE_ADDR']))?$_SERVER['REMOTE_ADDR']:
0;
if (isset($_GET['id'])) {$id=$_GET['id']; if ((!ctype_digit($id)) or
(strlen($id)>2)) {exit("<B>Жауаптар саны 1-9 дейін!!!</B>");}} else {$id=1;}
$lines = file("$datadir/vote$id.dat");
if (is_file("$datadir/ip$id.dat")) {
$iplines=file("$datadir/ip$id.dat"); $sizef=count($iplines);
if ($sizef > 1) { $itip=$sizef;
do {$itip--; $idt=explode("|",$iplines[$itip]);
if ($ip==$idt[0]) { $dayx=date("d.m.Y в H:i:s",$idt[1]); $stime=$idt[1];
$today=mktime();
}
$allredy="<B>Сіз дауыс берді&#1187;із  $dayx!</B></center>";}
 while ($itip>0); } }
if ($ipblok!="1") {$allredy=""; unset($allredy);}
if (!isset($allredy)) {$allredy="<B>Дауысы&#1187;ыз
&#1179;абылданды.</B>";
$mkdate=mktime();
$lines=file("$datadir/vote$id.dat");
$itogo=count($lines); $i=$itogo;
do {$i--; if ($numv==$i) {$vote=$i;}} while ($i>0);
$i=$itogo;
do {$i--; $dt=explode("|",$lines[$i]);
if ($vote==$i) {$dt[1]++;}
$lines[$i]="$dt[0]|$dt[1]|\r\n";
} while ($i>0);
$fp=fopen("$datadir/vote$id.dat","a+");
flock ($fp,LOCK_EX);
ftruncate ($fp,0);
for ($i=0; $i<$itogo; $i++) {fputs($fp,$lines[$i]);}
fflush ($fp);
```

```
flock ($fp,LOCK_UN);
@chmod("$fp", 0644);
$fp=fopen("$datadir/ip$id.dat","a+");
flock ($fp,LOCK_EX);
fputs($fp,"$ip|$mkdate|\r\n");
fflush ($fp);
flock ($fp,LOCK_UN);
@chmod("$fp", 0644);
}
print "<center><script language='Javascript'>function reload() {location =
'rezult.php?id=$id'}; setTimeout('reload()', 1500);</script>
<BR><BR><BR> $allredy";
?>
```

8.4 «Web-дизайн» электрондық оқулығын құру технологиясы

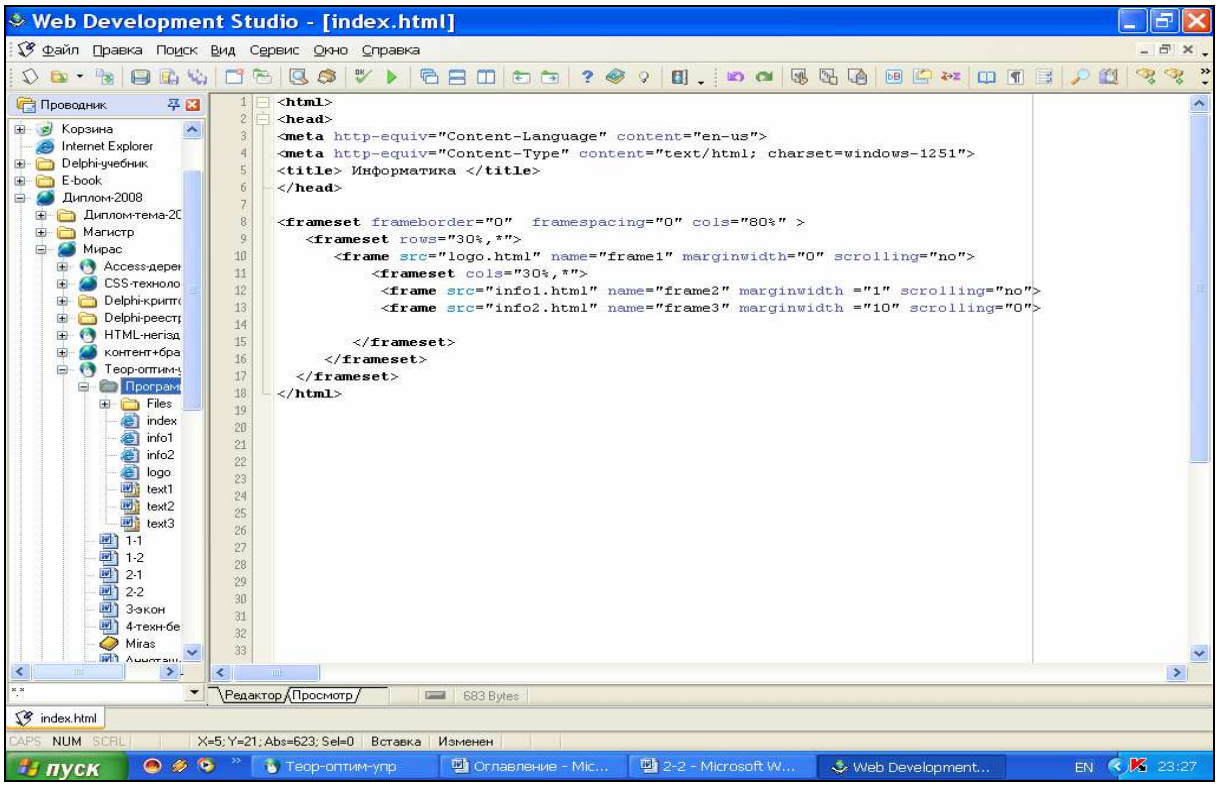
Электрондық оқулықты кез келген мәтіндік редакторда немесе арнайы HTML-редакторларында және конвертерлерде жасауға болады. Электрондық оқулықты дайындайтын редакторларды таңдау автордың жеке пікіріне байланысты. Сондықтан электрондық оқулықты құруда Web develoment Studio редакторынан пайдаланамыз. Редакторды Web develoment Studio сайтынан көшіріп аламыз [26].

Web develoment Studio редакторының басқа HTML-редакторлардан айырмашылығы HTML тілі тэгтерін басқа түспен белгілеп көрсетумен қатар, құжатты ішкі браузерде көруге болады. Сонымен қатар бағдарламалар тілдері туралы ішкі анықтамалар, әр түрлі сервис функциялары, ыңғайлы интерфейс, тэг атрибуттарын ұсыну сияқты қызметтері бар. Редакторды жүктегенде экранда негізгі терезе пайда болады (сурет 65).

Web develoment Studio редакторының мүмкіндіктерін қарастырайық. Олар төменде берілген:

- HTML, PHP, Perl, CSS, JS, VBS, ASP, SSI, XML технологияларын қолдайды;
- аталған технологиялар бойынша анықтамалар енгізілген;
- HTML, XML, PHP, Perl тілдерінің синтаксисін тексереді;
- функциялар туралы анықтама береді (Shift+Ctrl+Бос орын);
- кодты талдау және зерттеу севисі енгізілген;
- алмасу буферін пайдалануға болады;
- нәтіжені көруге ішкі және сыртқы браузер енгізілген;
- файлдық менеджер енгізілген;
- іздеу және файл бойынша ауыстыру функциясы орнатылған;

- MDI интерфейсі орнатылған, яғни бір уақытта бірнеше Web-құжатпен жұмыс істеуге болады;
- дайын кодтың шаблондары орнатылған (Ctrl+J);
- енгізуді түзету функциясы орнатылған:
- файлдарды салыстыру функциясы енгізілген;
- код бойынша жылдам өту функциясы енгізілген;
- дайын Web-құжаттың шаблондары орнатылған;
- макростармен жұмыс істеу функциясы енгізілген;
- синтаксисті әр түрлі түспен көрсету функциясы енгізілген;
- әр түрлі стильдерден, бағдарлама интерфейсінен пайдалануға болады.



Сурет 65 - Web develoment Studio редакторының негізгі терезісі

Редактордың мәзірінде төмендегі бөлімдер бар:

- Файл
- Правка
- Поиск
- Вид
- Сервис
- Окно
- Справка

Файл бөлімі – барлық қосымшалар сияқты файлдарды құруға, ашуға, сақтауға, алдын ала көруге, баспаға шығаруға және редактордан шығуға арналған.

Правка бөлімінде файлды түзету нұсқаулары орналасқан. Оларға көшіру, қиып алу, орналастыру, жұмыс режимдерін орнату, іс-әрекетті алып тастау немесе қалпына келтіру, мәтінді бөлектеу нұсқаулары жатады. Осы бөлім режимдері арқалы мәтіндегі HTML тілі тэгтерін басқа түспен белгілеп көрсетуге болады.

Поиск бөлімі – файлды және мәтінді іздеу, оларды алмастыру, қажетті белгіге өту, белгіні алып тастау нұсқауларын орындауға арналған.

Вид бөлімі редактор терезесін баптауға арналған. Осы бөлім арқалы инструменттер панеліне, интерфейске, редактор сілтеуішіне, оқулық терезесіне қажетті параметрлер меншіктеледі. Осы бөлімде XML, ASP, CSS, HTML, PERL, PHP, JavaScript және VBScript тілдері туралы қысқаша анықтаманы алуға болады.

Сервис бөлімі HTML кодын браузер экранында көруге, синтаксис және орфографияны тексеруге, шаблон және макростан пайдалануға арналған.

Окно бөлімі редакторда бір уақытта бірнеше файлды өңдеу мүмкіндігін береді. Терезеде оларды әр түрлі әдіспен орналастыруға болады. Бұл терезелерден пайдалану электрондық оқулықтың бірнеше тарауларын өзара байланыстырғанда ыңғайлы.

Справка бөлімінде программа, авторлар, программа версиясы, сайты туралы анықтамалар орналасқан.

Электрондық анықтама құруды HTML тілінде жазылған кодын теруден бастаймыз. Жұмысты анықтаманың негізгі бетін сипаттаудан бастаймыз. Негізіг бетті бірнеше фреймге бөлеміз. Анықтама құруда фреймдерден пайдалану, құжатты рәсімдеу мүмкіндіктерін арттырады. Фреймдер браузер экранын бірнеше терезеге бөледі. Сонда әр бір терезеде Web-құжаттың жеке бір тарауы бейнеленеді. Осы технология негізінде браузерге электрондық анықтаманың бірнеше бірнеше тарауын бір уақытта жүктеуге болады. Сондықтан HTML тілінің фреймдермен жұмыс істейтін тэгтеріне <frameset>, <frame> және <noframes> тэгтері жатады.

HTML тілінің <frameset> тэгі Web-құжаттың фреймдік құрылымын анықтайды. Тэг фреймдердің геометриялық өлшемін, олардың экрандағы орнын сипаттайды. <frameset> тэгі Web-құжатта <body> тэгінің орнына қолданылады. Егер құжатта <frameset> тэгі пайдаланылса, онда <body> тэгі қолданылмайды. Тэгтің ішінде <frame> және <noframe> тэгтері жазылады.

Анықтамада 3 фреймдік терезеден пайдаланамыз. Мысалы:

```
<frameset frameborder="1" framespacing="0" border="1" cols="250,*">
    <frame src="info_1.html" name="page1">
      <frameset rows="160,*">
        <frame src="info_2.html" name="page2" marginwidth="0">
```

```
        <frame src="info_3.html" name="page3" marginwidth ="0">
       </frameset>
      <noframes> фреймдер </noframes>
  </frameset>
```

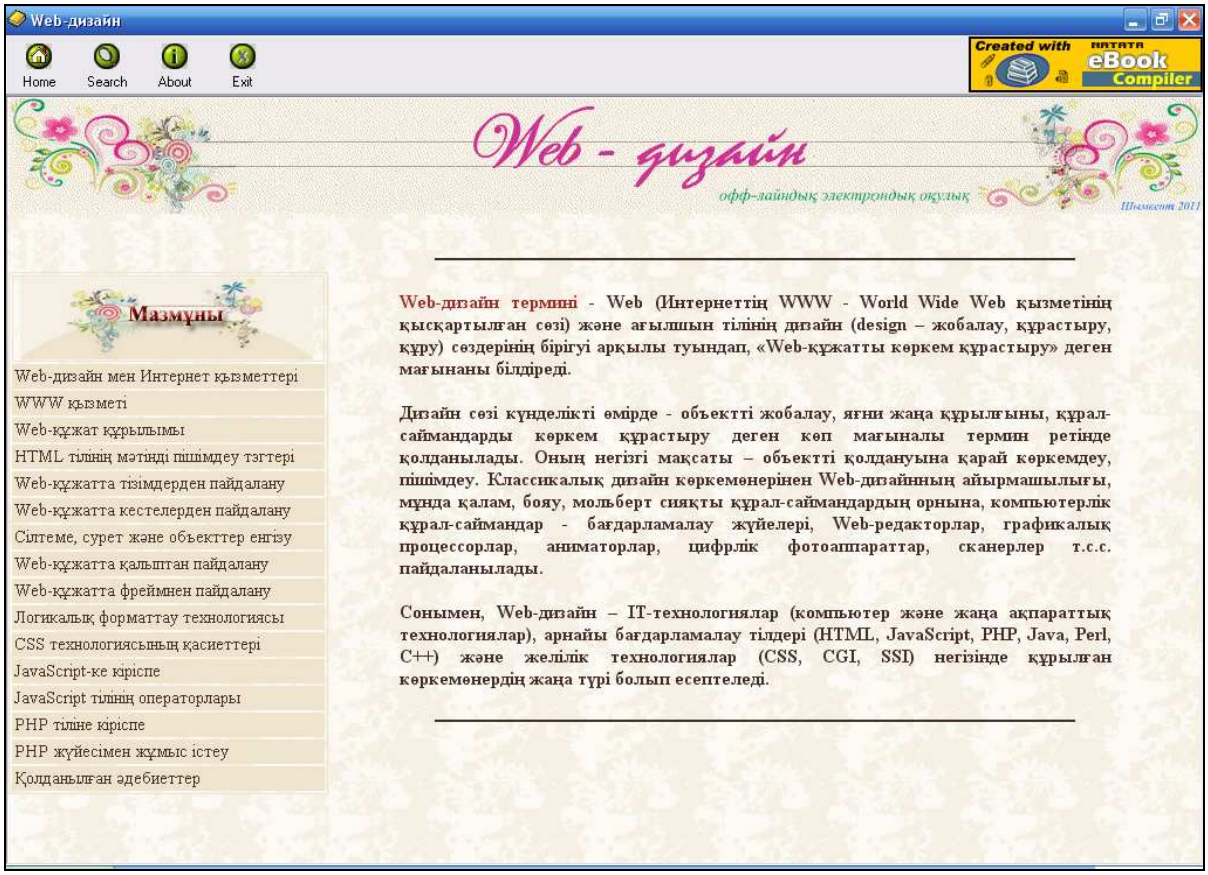
Сонымен оқулықтың негізгі терезесінің коды төмендегіше анықталады:

```
 <html>
   <head>
    <meta http-equiv="Content-Language" content="en-us">
      <meta http-equiv="Content-Type" content="text/html;
       charset=windows-1251">
       <title> Web-дизайн</title>
        </head>
        <frameset frameborder="0"  framespacing="0" cols="80%" >
        <frameset rows="30%,*">
       <frame src="logo.html" name="frame1" marginwidth="0"
       scrolling="no">
      <frameset cols="30%,*">
     <frame src="info1.html" name="frame2" marginwidth ="1"
    scrolling="no">
   <frame src="info2.html" name="frame3" marginwidth ="10"
  scrolling="0">
 </frameset>
</frameset>
</frameset>
</html>
```

Электрондық оқулықтың негізгі терезесі төмендегі суретте көрсетілген (сурет 66). Жасалған оқулықты «Web-дизайн» бумасына сақтап қоямыз. Электрондық оқулықты іске қосу үшін алдымен негізгі бетті іске қосу керек. Негізгі бет Index.html файлында сақталған.

Негізгі терезе үш фреймнен тұрады. Бірінші фреймде оқулықтың логотипі көрініп тұрады. Фреймді төмендегіше анықтаймыз.

```
   <frameset frameborder="0"  framespacing="0" cols="80%" >
      <frameset rows="30%,*">
       <frame src="logo.html" name="frame1" marginwidth="0"
       scrolling="no">
     </frameset>
  </frameset>
```



Сурет 66 – Электрондық оқулықтың негізгі терезесі

Фреймге оқулықтың логотипін сурет түрінде енгізілген. Сурет Adobe PhotoShop программасында құрылып logo.html атымен сақталған. Web-құжатқа суретті <img> тэгі енгізуде қолданылады. Сурет jpg, gif немесе png форматтарының бірінде болуы тиіс. Тэг жабылу тэгісіз жазылады.

Мысалы оқулықта <img border="1" src="files/graf10.gif" width="700" height="79" align="left"> қолданылды, яғни баузер терезесінің сол жақ шетіне files бумасындағы graf10.gif суреті шығады.

Екінші фреймде оқулықтың тарауларына сілтемелер берілген. Гиперсілтемелерді құруда және пайдалануда **<a>** тэгі қолданылады. Тэг тэгімен жабылады. Тэг төмендегі атрибуттарға ие:

- href – атрибуты бастапқы және жабылу тэгтері арасындағы мәтінді немесе суретті гиперсілтеме деп анықтайды.
- name – атрибуты бастапқы және жабылу тэгтері арасындағы мәтінді сілтеме жасау объекті деп анықтайды. Мән ретінде қайталанбайтын кез-келген сөзді латын әріптерімен жазуға болады.

Егер байланыс түрі мен URL-адрес көрсетілмесе бастапқы адрес ретінде ағымдағы бума алынады. Бұл салыстырмалы сілтеме жасауға мүмкіндік береді: Мысалы:

 Web-дизайн .

Бұл жерде ағымдағы буманың files ішкі бумасындағы text10.html файлына сілтеме жасалған.

Мысалы: соңына

тэгін құжат басына, ал

 соңы

тэгін құжат соңына жазайық. Браузер терезесінде құжат басында асты сызылған соңы сөзі басқа түспен шығып тұрады және ол құжаттың соңына сілтеме жасайды.

Мысалы: құжаттың кез-келген жеріне

 Web-дизайн

тэгі жазылсын, онда

 Web-дизайн

тэгі сол «Web-дизайн» сөзі жазылған бөлімге сілтеме жасайды. Браузер терезесіндегі басқа түспен бөлектенген Web-дизайн сөзін тышқан көрсеткішімен басу арқалы, сол бөлімге келеміз.

Оқулықта қолданылған target – атрибуты сілтемеде көрсетілген құжаттың бейнелену фрейм терезесін анықтайды. Қабылдайтын мәндері:

- "_self" – href атрибутында анықталған құжат ағымдағы фреймде бейнеленеді;
- "_parent" – href атрибутында анықталған құжаттың ағымдағы фреймнің аналық фреймінде бейнеленуі қажет екенін көрсетеді. Басқаша айтқанда ағымдағы фреймді қосушы frameset тегі сипатталған терезені көретеді;
- "_top" – href атрибутында анықталған құжаттың бастапқы фреймде бейнеленуі қажет екенін көрсетеді;
- "_blank" – href атрибутында анықталған құжаттың жаңа терезеде бейнеленуі қажет екенін көрсетеді.

Мысалы: Web-дизайн

Бұл жерде ағымдағы буманың files ішкі бумасындағы text1.html файлына сілтеме жасалған. Аталған файл жаңа терезеде бейнеленеді.

Оқулықтың тарауларына сілтемелер төмендегіше берілген.

```
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
</head>
<body bgcolor="#eaeaea">
<table width="100%" border="0"> <tr > <td >
<a href=text1.html target="frame3"> <li>Web-құжат құру әдістері
</li></a><br>
<a href=text2.html target="frame3"> <li>Web-құжат құрылымы
</li></a><br>
<a href=text3.html target="frame3"> <li>HTML тілінің мәтінді
форматтау тэгтері </li></a><br>
<a href=text4.html target="frame3"> <li>HTML тілінде тізімдерден
пайдалану</li></a><br>
</td></tr></table>
</body>
</html>
```

Оқулық кодында <table> тэгінен пайдаландық. Web-құжатқа <table> тэгі кесте құру үшін қажет. Кестенің ұяшықтары <caption>, <tr>, <td> және <th> тэгтерімен құрылады.

Үшінші фрейм оқулық тарауларын бейнелеуге арналған. Фреймді төмендегіше анықтаймыз.

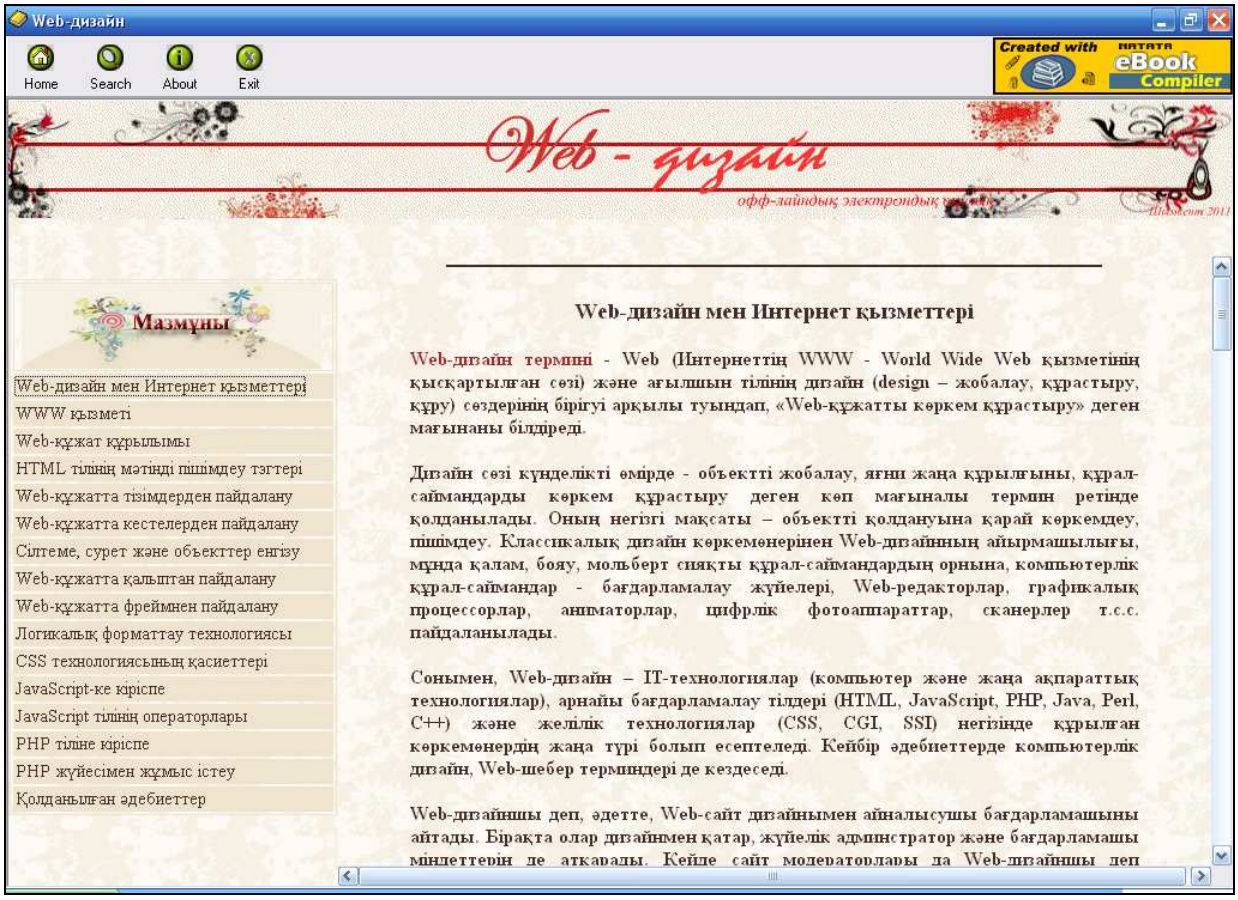
```
<frameset cols="30%,*">
<frame src="info1.html" name="frame2" marginwidth ="1"
scrolling="no">
<frame src="info2.html" name="frame3" marginwidth ="10"
scrolling="0">
</frameset>
</frameset>
```

Терезеге оқулық тарауларын жоғарыда сипатталған тэгтер көмегінде бейнелейміз.

Мысалы бірінші тарау төмендегіше шақырылады.

```
<a href=text1.html target="frame3"> <li>Web-құжат құру әдістері</li>
</a><br>
```

Браузер терезесінде оқулық тарауларының мәтіні бейнеленеді (сурет 67).



Сурет 67 – Электрондық оқулық мәтіні

Терезедегі оқулық тарауларын жоғарыда сипатталған тэгтер көмегінде сипаттаймыз.

```
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
</head>
<body bgcolor="#eaeaea">
<h2 align=center>Web-құжат құру әдістері</h2>
<p align="justify"> Web-құжат файлдары .html немесе .htm кеңеймесі бар мәтіндік файлдар. Оларды құруда қарапайым мәтіндік редакторлардан пайдалануға болады. Мысалы: MS Windows ортасында Notepad (Блокнот) редакторы. Бірақта Web-құжаттарды арнайы HTML-редакторларда құру ыңғайлы. HTML-редакторлар екі топқа бөлінеді:
<li>ASCII-редакторлар (ASCII-editors, Source editors);</li>
<li>WYSIWYG-редакторлар (WYSIWYG-editors).</li> <br></p>
<p align="justify"><b>ASCII-редакторлар </b>қарапайым мәтіндік редакторлар. Тек олар HTML-құжаттарды құруға арналған қосымша мүмкіндіктерге ие. Оларға Webber, Macromedia HomeSite, DarekWare HTML Author, WebEdit редакторлары жатады.<p>
```

WYSIWYG-редакторлар HTML-құжаттарды дайындалу барысында терезеде көру, құжатқа алдын-ала дайындалған шаблондарды қосу мүмкіндігін береді. Редакторлар қосымша терезе - браузерлермен жабдықталған. Оларға Microsoft FrontPage, Netscape Navigator Gold редакторлары жатады.

Web-браузерлер HTML тілінде Internet желісіне арнап жазылған Web-сайттар мен Web-құжаттарды парақтап көру, оқу мүмкіндігін береді. Web-браузерлер ішінде ең көп таралғандары Netscape Navigator (Netscape Communications) және Internet Explorer (Microsoft).

HTML тілінің мәтіндерді белгілеу нұсқаулары тэг (tag) деп аталады. Тэг символдар тізбегінен тұрады. Барлық тэг < (кіші) символынан басталып, > (үлкен) символымен аяқталады. Осы символдар ішіне қызметші сөз – тэг орналасады. Әрбір тэг өзіне тән бір арнаулы қызмет атқарады. Оларды жазуда бас әріпті де, кіші әріпті де қатар қолдануға болады, әріптер бірдей қабылданады. Браузер мәтінді монитор терезесінде көрсеткенде тэгтер бейнеленбейді, тек олардың құжат мәтініне тигізетін әсері ғана білініп тұрады. HTML тілінің әрбір тэгі әдетте құжаттың белгілі бір бөлігіне ғана әсер етеді. Осыған орай екі тэг қатар қолданылады: бірі - тэгті ашады, екіншісі - жабады. Ашатын тэг белгілі бір нұсқауды бастайды, > ал жабатын тэг - оны аяқтайды. Жабу тэгінің алдына / (қиғаш сызық) символы жазылуы тиіс. Кейбір тэгтер мағынасына орай жабылмауы да мүмкін.

Тэгтердің мәтінге тигізетін әсерлерін түрлендіруде олардың атрибуттарынан пайдаланады. Әдетте атрибуттар ашылу тэгтерінде кездесіп, қосымша қызметші сөздерден тұрады. Көбінесе атрибуттар " (қос тырнақша) символдары ішіне жазылған мәнімен беріледі. Атрибуттың мәні оның қызметші сөзінен = (теңдік) символы арқылы бөлініп жазылады.

Web-құжатта түсініктеме жазу үшін <!-- comment --> тэгінен пайдаланамыз. Түсініктемеде жазылған мәтін немесе тэг өңделмейді, яғни браузер терезесіне шықпайды. Түсініктемені құжаттың кез келген жеріне жазуға болады.

```
 &nbu; &nbu; &nbu; &nbu;
<center><a href=index.html target="_top"> top </a></center>
</td></tr></table><hr color="808080" size="3" width="80%"  ></p>
</body>
</html>

<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<link type="text/css" rel="stylesheet" href="../1/style.css">
</head>
```

```
<body class="fon1">
<hr color="ff0000" size="3" width="80%"  >
<h1 align=center><font color="#0000ff">JavaScript тілінің қарапайым элементтері</font></h1><h2>
<p align="justify"><i>Алфавит.</i> JavaScript тілінің алфавиті төмендегі белгілерден тұрады:</p>
<ul><li>26 латын алфавиті әріптері::  A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z.</li>
<li>10 араб цифрлары:  0, 1, 2, 3, 4, 5, 6, 7, 8, 9.</li>
<li>арнайы белгілер: </li></ul> </p>
  <table align="center" width="80%" border="1">
    <tr>
      <td width="10%" align="center"> + </td>
      <td width="45%" align="left"> қосу </td>
      <td width="10%" align="center"> ~ </td>
      <td width="45%" align="left"> тильда </td>

    </tr>
    <tr>
      <td width="10%" align="center"> - </td>
      <td width="45%" align="left"> алу </td>
      <td width="10%" align="center"> ! </td>
      <td width="45%" align="left"> леп белгісі </td>
    </tr>
    <tr>
      <td width="10%" align="center"> * </td>
      <td width="45%" align="left"> жұлдызша </td>
      <td width="10%" align="center"> @ </td>
      <td width="45%" align="left"> коммерциялық эт </td>
    </tr>
    <tr>
      <td width="10%" align="center"> / </td>
      <td width="45%" align="left"> қисық сызық </td>
      <td width="10%" align="center"> # </td>
      <td width="45%" align="center"> нөмір белгісі </td>
    </tr>
    <tr>
      <td width="10%" align="center"> = </td>
      <td width="45%" align="left"> тең </td>
      <td width="10%" align="center"> $ </td>
      <td width="45%" align="left"> ақша бірлігі  </td>
    </tr>
    <tr>
      <td width="10%" align="center"> , </td>
      <td width="45%" align="left"> үтір </td>
```

```
  <td width="10%" align="center"> % </td>
  <td width="45%" align="left">  пайыз </td>
</tr>
<tr>
  <td width="10%" align="center"> . </td>
  <td width="45%" align="left"> нүкте </td>
  <td width="10%" align="center"> ^ </td>
  <td width="45%" align="left">  көрсеткіш </td>
</tr>
<tr>
  <td width="10%" align="center"> : </td>
  <td width="45%" align="left"> қос нүкте </td>
  <td width="10%" align="center"> & </td>
  <td width="45%" align="left"> амперсанд  </td>
</tr>
<tr>
  <td width="10%" align="center"> ; </td>
  <td width="45%" align="left"> нүктелі үтір </td>
  <td width="10%" align="center"> | </td>
  <td width="45%" align="left"> тік таяқша  </td>
</tr>
<tr>
  <td width="10%" align="center"> > </td>
  <td width="45%" align="left"> үлкен </td>
  <td width="10%" align="center"> \ </td>
  <td width="45%" align="left"> кері қисық сызық </td>
</tr>
<tr>
  <td width="10%" align="center"> < </td>
  <td width="45%" align="left"> кіші </td>
  <td width="10%" align="center"> () </td>
  <td width="45%" align="left"> жақша </td>
</tr>
<tr>
  <td width="10%" align="center"> ? </td>
  <td width="45%" align="left"> сұрақ белгісі </td>
  <td width="10%" align="center"> [] </td>
  <td width="45%" align="left"> тік жақша </td>
</tr>
<tr>
  <td width="10%" align="center"> _ </td>
  <td width="45%" align="left"> сызықша </td>
  <td width="10%" align="center"> {} </td>
  <td width="45%" align="left"> фигуралық жақша </td>
</tr>
```

```
    <tr>
      <td width="10%" align="center">  </td>
      <td width="45%" align="left"> бос орын </td>
      <td width="10%" align="center"> ‘ </td>
      <td width="45%" align="left"> апостроф </td>
    </tr>
    <tr>
      <td width="10%" align="center"> “ </td>
      <td width="45%" align="left"> тырнақша </td>
      <td width="10%" align="center">  </td>
      <td width="45%" align="center">  </td>
    </tr>
  </table>
```

JavaScript интерпретаторы регистрды есепке алады, яғни бас және кіші әріптерді бөлек деп қабылдайды. Сонымен қатар орыс немесе қазақ алфавиті әріптерін тек қана мәтін қатарында қолдануға болады.

Алфавит белгілерінен қызметші сөз, идентификатор, литерал, оператор т.с.с. тіл элементтері құрылады.

Қызметші сөздер деп JavaScript тілінің жұмысына қажетті нұсқауларды айтады. Әрбір сөздің өз мағынасы, атқаратын қызметі бар. Қызметші сөздерді жұмыс барысында біртіндеп мағынасы қарай түсіндіріп отырамыз.

Идентификатор деп цифр немесе _(сызықша) белгісінен басталатын әріп, цифр және _ белгілерінің тізбегін айтады. Идентификатор - литерал, оператор, функция сияқты қарапайым элементтерді атауға қажет. Идентификаторды жазуда бас және кіші әріптер әртүрлі деп есептеледі. Сонымен қатар идентификатор ретінде қызметші сөздерді қолдануға болмайды. Мысалы:

Дұрыс жазылған идентификаторлар:

Java_Script – сызықша орынды қолданылған;

_a1 – сызықшадан басталады;

X – латын әрібі қолданылған.

Қате жазылған идентификаторлар:

Java-Script –басқа белгі қолданылған;

For – қызметші сөз;

1a – цифрден басталған;

Скрипт –орыс әріптері қолданылған.

Литерал деп әдетте JavaScript тілінде қолданылатын деректерді айтады. Литерал тұрақты және айнымалы деп екіге бөлінеді. Тұрақты литерал деп скриптте мәні өзгермейтін деректі айтамыз. Ал мәні өзгеретін деректі айнымалы литерал дейміз. Литералдың атауы идентификатормен беріледі.

Деректердің типтері. JavaScript тілінде қолданылатын деректер типтерін қарастырайық:

бүтін;
нақты;
логикалық;
қатарлық;
null.
Бүтін типті деректерге бүтін сандар жатады. Сандар ондық, сегіздік және он алтылық санақ жүйесінде берілуі мүмкін. Сегіздік санақ жүйесіндегі сандардың алдына 0 (нөл) таңбасы, ал он алтылық сандардың алдына 0x таңбасы жазылады. Мысалы: -100, 0175, 0xA1.
Нақты типті сандар екі түрде жазылады:
тұрақты нүктелі;
жылжымалы нүктелі.
Тұрақты нүктелі сандар, нүктемен бөлектенген бүтін және бөлшек түрінде жазылады. Санда бөлшек бөлім болмаса да, нүкте қойылады. Мысалы: 3.1415, -100.

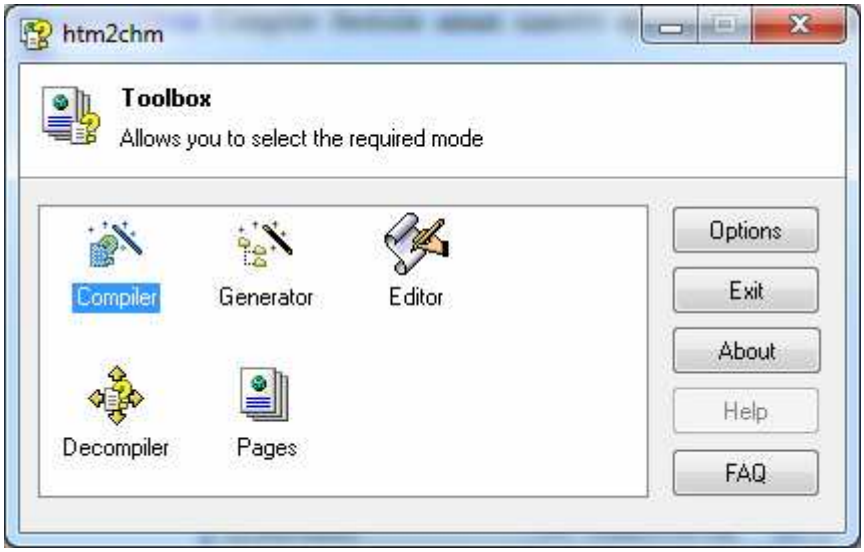
Жылжымалы нүктелі сандар, E (үлкен әріп) символымен бөлектенген мантисса және дәреже түрінде жазылады. Мысалы: 1E-5, 3.14E3.
Логикалық типті деректер тек қана true (ақиқат, 1) false (жалған, 0) мәндерін ғана қабылдайды.
Қатарлық типке мәтін жатады. Мәтін апостроф (‘) немесе қос тырнақша (“) ішіне жазылады. Мысалы: “TurboPascal тіліне ұқсас”.
JavaScript тілінде қатарлық типке арнайы қызметші және баспаға шықпайтын символдарды енгізуге болады.

<table align="center" width="50%" border="1">
<tr>
<td width="20%" align="center"> \b </td>
<td width="80%" align="left"> бос орын (backspace) </td>
</tr>
<tr>
<td width="20%" align="center"> \f </td>
<td width="80%" align="left"> жаңа бетке өту (PageDown) </td>
</tr>
<tr>
<td width="20%" align="center"> \n </td>
<td width="80%" align="left"> жаңа қатарға өту </td>
</tr>
<tr>
<td width="20%" align="center"> \r </td>
<td width="80%" align="left"> енгізу (enter) </td>
</tr>
<tr>
<td width="20%" align="center"> \t </td>

```
          <td width="80%" align="left"> табуляция (tab)  </td>
        </tr>
        <tr>
          <td width="20%" align="center"> \’ </td>
          <td width="80%" align="left"> апостроф  </td>
        </tr>
        <tr>
          <td width="20%" align="center"> \” </td>
          <td width="80%" align="left"> тырнақша   </td>
        </tr>
        <tr>
          <td width="20%" align="center"> \\ </td>
          <td width="80%" align="left"> кері қисық сызық (backslash)    </td>
        </tr>
      </table><br>
    <table  width="100%" border="0">
         <tr>
          <td width="30" align="center"> Мысалы: </td>
          <td width="50" align="left"> my_file </td>
          <td width="300" align="left"> -идентификатор; </td>
        </tr>
        <tr>
          <td width="30" align="center">  </td>
          <td width="50" align="left"> 12345 </td>
          <td width="300" align="left"> -бүтін сан; </td>
        </tr>
        <tr>
          <td width="30" align="center">  </td>
          <td width="50" align="left"> 1.2E-4 </td>
          <td  width="300"> -жылжымалы нүктелі нақты сан (1,2*10<sup>-
4</sup>);</td>
        </tr>
        <tr>
          <td width="30" align="center">  </td>
          <td width="50" align="left"> “Сауле” </td>
          <td width="300"> -қатар.</td>
        </tr>
    </table>
    </h2><center><a  href=index.html  target="_top">  Бастапқы  бетке
</a></center>
    <hr color="ff0000" size="3" width="80%"  >
    </body>
    </html>
```

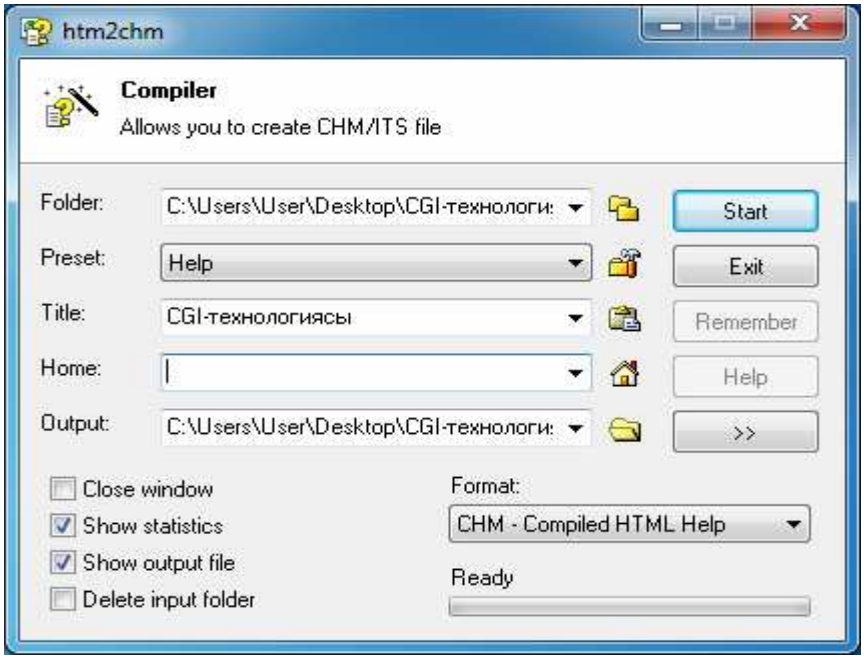
Байқасаңыз электрондық оқулық бірнеше файлдардан тұрады. Мұндай

электрондық анықтаманы офф-лайндық түрде пайдалану қолайсыз, электрондық оқулықты бір файлға жинақтау қажет. Электрондық оқулықты жинақтауды htm2chm-3.0.9.3-en бағдарламасы арқылы амалға асырамыз. Бағдарлама опцияларын да осы терезеде анықтаймыз (сурет 68).



Сурет 68 – htm2chm-3.0.9.3-en бастапқы терезесі

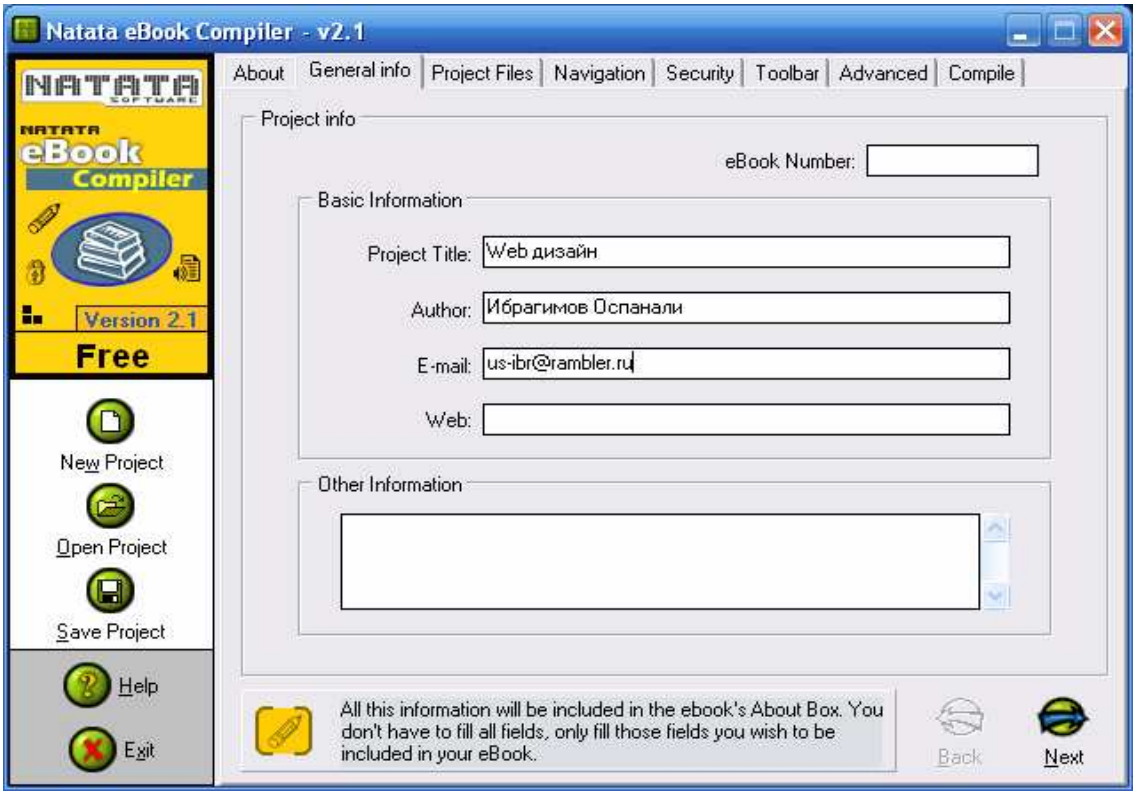
Бағдарламада Compiler, Generator, Editor, Decompiler, Pages бөлімдері орналасқан. Бұлардың ішінде бізге қажеттісі Compiler опциясы. Осы опция көмегінде электрондық оқулықты компиляциялайды. Бағдарламаның Compiler бөлімін ашып қажетті өрістерді толтырамыз (сурет 69).



Сурет 69 – Compiler терезесі

Өрістеді толтыруда қажетті буманы, анықтама тақырыбын, бастапқы бет адресін көрсету қажет. Жинақталған файл кеңеймесі .chm болады.

Сонымен қатар электрондық оқулықты компиляциялауда Natata Software программасынан да пайдалануға болады (сурет 70).



Сурет 70 – Natata Software терезесі

Программаның New Project опциясын ашып қажетті өрістерді толтырамыз. Өрістеді толтыруда қажетті буманы, электрондық оқулық тақырыбын, бастапқы бет адресін, автор туралы мәліметтерді көрсету қажет. Копиляцияланған файл кеңеймесі .exe болады. Пайда болған электрондық оқулық браузерді қажет етпейді.

Құрылған электрондық оқулық келесі талаптарды қанағаттандырады:

- тұтынушы интерфейсінің ыңғайлы болуы;
- электрондық оқулық мазмұнын шапшаң өзгерту мүмкіндігі;
- оқулық мазмұнын импорттау және экспорттау функциясының дамығандығы;
- оқулық дайындауда Интернет-технологиялардың қолданылуы.

Электрондық оқулықтың компьютерге қоятын талаптары:

- IBM PC платформасындағы компьютері;
- MS Windows XP/7 операциялық жүйесі;
- 64 Mb жедел жады;
- MS Internet Explorer 5.0 браузері және жоғары версиялары;
- SVGA-тақша (800x600, 65536 түстер).

Жеке жұмысқа тапсырмалар

Тапсырма №1. Төменде берілген тақырыптарға сипаттама беріп, талдап, мысалдармен түсіндіріңіз. Тапсырманы орындауда график, кесте, диаграмма т.с.с. пайдалануға болады. Тапсырма презентация түрінде қабылданады (вариант бойынша):

1. Интернет компоненттері;
2. Интернет қызметтері (WWW, FTP, Usenet, Telnet, E-mail);
3. Интернет хаттамалары (TCP/IP, HTTP, FTP);
4. WWW қызметі;
5. Домендік адрестер;
6. Іздеу серверлері;
7. Интернетке қосылу әдістері (Қазақстан провайдерлері);
8. HTML тілі тэгтері;
9. ASCII-редакторлар;
10. WYSIWYG-редакторлар;
11. Web–браузерлер;
12. IP-адрес және URL-адрес;
13. Web-құжат құрылымы;
14. RGB (red, green, blue) моделі;
15. CSS –стильдер кестесі.

Тапсырма №2. HTML тілін пайдаланып өзіңізге резюме жазыңыз (Блокнот редакторын пайдалану қажет). Тапсырманы орындаған кезде әртүрлі шрифттерді қолдана отырып, олардың түсін, өлшемін және шрифт түрін өзгертіңіз, фотосуретті пайдаланыңыз.

Резюме төмендегі бөлімдерден тұрады:

- тегі;
- аты;
- әкесінің аты;
- туылған жері;
- ұлты;
- білімі;
- кәсіби біліктілігі;
- білетін тілдері;
- мекен-жай;
- байланысу телефондары.

Тапсырма №3. HTML тілін пайдаланып браузер экранына келесі тізім шығатын Web-құжат құрыңыз (вариант бойынша):

1. Топ студенттерінің тізімі;
2. Қалааралық телефон кодтарының тізімі;
3. Оңтүстік Қазақстан облысындағы аудандар тізімі;
4. Оқылатын пәндер тізімі;

5. Қазақстан университеттерінің тізімі;
6. Курстық жұмысты жазу үшін қолданылған әдебиеттер тізімі;
7. Компьютерлік бағдарламалар тізімі;
8. Браузерлер тізімі;
9. Антивирустар тізімі;
10. Қазақстан қалаларының тізімі;
11. Бағдарламалау тілдерінің тізімі;
12. Футбол бойынша Қазақстан командаларының тізімі;
13. Көп қолданылатын сайттар тізімі;
14. Ұялы телефон шығаратын корпорациялар тізімі;
15. Іздеу серверлерінің тізімі.

Тапсырма №4. HTML тілін пайдаланып кесте орналасқан Web-құжат құрыңыз. Тапсырманы орындаған кезде әртүрлі шрифттерді қолдана отырып, олардың түсін, өлшемін және шрифт түрін өзгертіңіз, суреттерден пайдаланыңыз. Кестенің ішіне берілген тақырып бойынша деректер шығатын болсын (вариант бойынша):

1. Сабақ кестесі;
2. Футбол чемпионаты ойындарының кестесі;
3. Шахмат турнирі ойындарының кестесі;
4. Информатика пәні оқытушысының сабақ кестесі;
5. Компьютерлік сыныпта өтетін сабақ кестесі;
6. Қала маңындағы жүретін автобустар қозғалысының кестесі;
7. Қазақстан бойынша ұшақтар қозғалысының кестесі;
8. Темір жол вокзалынан жүретін пойыздар қозғалысының кестесі;
9. Календарь;
10. Компьютер дүкенінің прайс-листі;
11. Университет оқу үрдісінің кестесі;
12. Мекеменің жұмыс кестесі;
13. Студенттердің рейтинг кестесі;
14. Сессия нәтижелері кестесі;
15. Студент анкетасы.

Тапсырма №5. HTML тілін пайдаланып браузер экранына берілген тақырып бойынша жарнама мәтіні және баннер шығаратын Web-құжат құрыңыз (вариант бойынша):

1. Жоғары оқу орнына түсуге шақыратын жарнама (мамандықтар тізімі, оқу бөлімдері – күндізгі және сырттай оқу, жеңілдіктер, қажетті құжаттар, қабылдау мерзімі, мекен-жайы, байланыс телефоны және т.б.);
2. Атақты әншінің концертіне шақыратын жарнама (концерт өтетін уақыты мен орны, билет бағасы, өлеңдер тізімі мен орындаушының фотосуреті және т.б.);
3. Компьютер сататын дүкеннен тауар сатып алуға шақыратын жарнама (дүкенде сатылатын компьютерлер, олардың

модельдері, техникалық сипаттамасы, түрлері, бағалары, шығарған фирманың аты, мекен-жайы, байланыс телефоны және т.б.);

4. Кинотеатрға кино көруге шақыратын жарнама (көрсетілетін кинофильмдер тізімі, сеанстар уақыты, билет бағасы, танымал киноактерлардың фотосуреті және т.б.);
5. Театрға спектакль көруге шақыратын жарнама (спектакльдер тізімі, уақыты мен орны, билет бағасы, актерлар фотосуреті және т.б.);
6. Супермаркеттен тауар сатып алуға шақыратын жарнама (тауарлар тізімі, бағалары, жаңа тауарлар, жеңілдіктер мерзімі, тауар шығаратын атақты фирмалар, супермаркет бөлімдері және т.б.);
7. Спорт мектебіне шақыратын жарнама (спорттық секциялар тізімі, тренердің аты-жөні, спорт мектебінің атақты түлектері, мекен-жайы, байланыс телефоны және т.б.);
8. Спорт жарысына шақыратын жарнама (мысалы – футбол, қатысатын командалар тізімі, жарыстың өтетін уақыты мен орны, демеушілер тізімі, атақты ойыншылардың фотосуреті және т.б.);
9. Ұялы телефон сататын дүкеннен тауар сатып алуға шақыратын жарнама (дүкенде сатылатын телефон түрлері, олардың бағалары, модельдері, шығарған фирманың аты, мекен-жайы, байланыс телефоны және т.б.);
10. Ән байқауына қатысуға шақыратын жарнама (байқауға қатысатын атақты әншілердің аты-жәні, жюри мүшелері, алдыңғы байқау жеңімпаздары, байқаудың өтетін уақыты мен орны, демеушілер тізімі, байланыс телефоны және т.б.);
11. Айтысқа шақыратын жарнама (қатысатын ақындар тізімі, жеңімпаздарға арналған сыйлықтар, уақыты мен орны, билет бағасы, демеушілер тізімі және т.б.);
12. Әсемдік салонына шақыратын жарнама (көрсетілетін қызметтер тізімі, бағалары, мекен-жайы, байланыс телефоны және т.б.);
13. Әйелдер киімін сатып алуға шақыратын жарнама (тауарлар тізімі, бағалары, жеңілдіктер мерзімі, тауар шығаратын атақты фирмалар, дүкен бөлімдері, мекен-жайы, байланыс телефоны және т.б.);
14. Құрылыс материалдарын сатып алуға шақыратын жарнама (тауарлар тізімі, бағалары, тауар шығаратын атақты фирмалар, дүкен бөлімдері, мекен-жайы, байланыс телефоны және т.б.);
15. Банк қызметіне арналған жарнама (қызметтер тізімі, кредит және депозит түрлері, пайызы, жеңілдіктер мерзімі, банк филиалдары, мекен-жайы, байланыс телефоны және т.б.).

Тапсырма №6. HTML тілін пайдаланып берілген тақырып бойынша Web-сайт құрыңыз. Тапсырманы орындаған кезде түрлі-түсті шрифттерден, қалыптардан, кестелерден, фреймдерден, баннерлерден, суреттерден пайдаланыңыз (вариант бойынша):

1. АҚ «Қазақтелеком»;
2. АҚ «Шымкент шина»
3. АҚ «Су ресурстары - Маркетинг»;
4. АҚ «Қазпошта»;
5. «Астана моторс» автосалоны;
6. ЖШС «Шымкент жарық»;
7. «Рамстор» дүкендер желісі;
8. «Қала әкімшілігі»;
9. «Интернет-дүкен»;
10. №1 облыстық дарынды балалар лицейі;
11. №69 экономика және құқық лицейі;
12. «Информатика» кафедрасы;
13. «Жиһаз дүкені»;
14. «Әсемдік салоны»;
15. «Рахат» кондитер фабрикасы;
16. Еркін тақырып (студент сұранысы бойынша).

Тапсырма №7. Құрылған Web-сайтқа арнап, Блокнот редакторында my.css каскадты стильдер кестесі файлын құрыңыз. Құрылған my.css файлын гипермәтіндік құжатқа енгізіңіз (вариант бойынша).

Тапсырма №8. Құрылған Web-сайтқа объекттер (Java-скрипттер, клиптер, апплеттер) енгізіп, сайтты безендіріңіз (вариант бойынша).

Тапсырма №9. Apache жүйесін және PHP процессорын орнатыңыз (Denwer немесе Joomla жүйелерін де қолдануға болады). Құрылған Web-сайтқа PHP-скриптті енгізіп, браузерде жұмысын қадағалаңыз (вариант бойынша).

Тест сұрақтары

1. Web-құжатының файл кеңеймесі (extension) қандай?
   a) .ppt
   b) .xls
   c) .bmp
   d) .doc
   e) .html

2. TCP/IP деген не?
   a) деректер алмасуға арналған стандарт хаттама
   b) техникалық, программалық және информациялық компонент
   c) физикалық адрестерге сәйкес домендік адрестер жүйесі
   d) гипермәтіндік белгілеу тілі
   e) Internet ресурстарын қарау браузері

3. Internet қызметтері:
   a) www, ftp, telnet, E-mail
   b) us, ru, kz, uk
   c) http, ftp, imap, pop3
   d) com, edu, gov, net
   e) Internet Explorer, Netscape Navigator

4. URL-адресті көрсет:
   a) www.ukgu.kz
   b) 192.168.32.1.
   c) nurlan@reiting.ball.kos
   d) http://www.
   e) esbol@agai.kele.jatir

5. IP-адресті көрсет:
   a) 192.168.32.1.
   b) www.ukgu.kz
   c) nurlan@ bes.koyip.ber
   d) http://www.
   e) esbol@shpargalkani.jasir

6. Аймақтық домен аттары:
   a) us, ru, kz
   b) com, edu, gov
   c) http, ftp, imap
   d) www, ftp, telnet
   e) E-mail, Usenet, ISQ

7. Қызмет бабына негізделген домен аттары:
   a) com, edu, gov, net
   b) us, ru, kz, uk
   c) http, ftp, imap, pop3
   d) www, ftp, telnet, E-mail
   e) Internet Explorer, Netscape Navigator

8. Іздеу серверлері:
   a) www.rambler.ru www.yandex.ru
   b) www.referat.ru www.bazar.net
   c) www.nursat.kz www.telecom.kz
   d) www.mail.ru www.ok.kz
   e) www.khabar.kz www.president.kz

9. Браузерлер:
   a) Internet Explorer, Opera, Mozilla FireFox
   b) us, ru, kz, uk
   c) http, ftp, imap, pop3
   d) www, ftp, telnet, E-mail
   e) com, edu, gov, net

10. Электрондық пошта адресі:
   a) nurlan@mail.ru
   b) www. kazatk.kz
   c) 192.168.32.1.
   d) http://www.
   e) esbol@.jayabi.kaisi

11. Графикалық файлдың форматын көрсет:
   a) jpeg, gif, bmp
   b) doc, rtf, dot
   c) rar, zip, arj
   d) exe, com, bat
   e) xls, txt, tmp

12. Пиксел деген не?
   a) графикалық режимдегі экрандағы ең кіші элемент
   b) кішкене төртбұрышты элемент
   c) растрлық элемент
   d) графикалық режимдегі нүкте
   e) графикалық режимдегі сәуле

13. Растр деген не?
a) пикселдер матрицасы
b) монитордағы жарық шығарушы нүктелер
c) бейненің форматы
d) суреттің форматы
e) бейне сапасы

14. Интернет желісінде электрондық поштаның адресі берілген: user_name@mail.ru Тұтынушының логині қандай?
a) user_name
b) ru
c) @user_name
d) mail.ru
e) user_name@ mail.ru

15. HTML тегі деген не?
a) HTML тілі нұсқаулары
b) HTML -құжат беттері
c) HTML -гипермәтінді белгілеу тілі
d) HTML -құжаттар іздеу сервері
e) HTML -деректер алмасуға арналған стандарт хаттама

16. Қандай атрибут түсті анықтайды?
a) color
b) size
c) width
d) height
e) title

17. <html>…</html> тэгі не үшін қолданылады?
a) HTML-құжатты ашып жабу үшін
b) ені бойынша түзету
c) тақырыпты бейнелеу үшін
d) абзацты бейнелеу үшін
e) шрифттармен жұмыс істеу үшін

18.
 тэгі нені білдіреді?
a) жаңа жолға өту
b) абзацтың бейнеленуі
c) тақырыптың бейнеленуі
d) шрифт өлшемі
e) көлденең түзу сызықты салады

19. Web-құжатқа сурет қандай тэг арқылы қойылады?
a) <img>
b) <html>
c) <body>
d) <head>
e)

20. Web-құжатқа кесте қандай тэг арқылы қойылады?
a) <table>
b) <head>
c) <title>
d) <body>
e) <img>

21. … - тэгтері нені анықтайды?
a) абзац
b) сол жақтан бос орын қалдыру
c) құжат тақырыбыі
d) тэг атрибуты
e) шрифт өлшемі

22. JavaScript-те (!=) амалы нені білдіреді?
a) тең емес
b) кіші
c) үлкен
d) қатаң түрде тең
e) false

23.PHP-скрипт нәтижесін көрсетіңіз

```
<?
Function get sum ($var)
{
$var=$var+5;
return $var;
}
$new var =20;
echo(get sum($new var));
?>
```

a) 25
b) 20
c) 0
d) 5
e) false

24. <h1 align = center>…</h1>. Бұл жазу нені білідіреді?
a) центрге түзету
b) еніне түзету
c) оң жаққа түзету
d) сол жаққа түзету
e) жалпы түзету

25. <body>…</body> тэгі не үшін қолданылады?
a) құжат денесінің басы мен аяғын анықтайды
b) ені бойынша түзету үшін
c) абзацтарды сипаттау үшін
d) суреттерді орналастыру үшін
e) құжат тақырыбын орналастыыру үшін

26. <hi>…</hi> тегі не үшін қолданылады?
a) тақырыпты бейнелеу үшін
b) абзацті бейнелеу үшін
c) шрифттармен жұмыс істеу үшін
d) суреттерді орналастыру үшін
e) ені бойынша түзету

27. Бас шрифтті анықтау үшін қандай тэг қолданылады?
a) <basefont>
b) <html>
c) <body>
d) <head>
e) <title>

28. Түсті анықтау қандай символдан басталады?
a) #
b) ?
c) *
d) $
e) @

29. bgcolor атрибуты нені береді?
a) фон түсін анықтау
b) шрифтті орнату
c) абзацты анықтау
d) тақырыпты бейнелеу
e) графикалық объекттерді орналастыру

30. Гиперсілтеме қандай тэг арқылы қойылады?
a) <a>
b) <html>
c) <body>
d) <head>
e) <img>

31. RGB моделіне 0,255,0 параметрлері қойылса, экранда қандай түс пайда болады?
a) жасыл
b) қызыл
c) ақ
d) көк
e) сары

32. RGB моделінде негізгі түстер ретінде:
a) қызыл, жасыл, көк
b) көгілдір, күлгін, сары
c) қызыл, көгілдір, сары
d) күлгін, сары, қара
e) қызыл, сары, қара

33. Модемнің ең маңызды параметрі:
a) жылдамдық
b) өнімділігік
c) биттер саны
d) такт жиілігі
e) разрядтылық

34. Модем жылдамдығының өлшем бірлігі:
a) бит/сек
b) файл/сек
c) символ/сек
d) МГц
e) пиксел/дйюм

35. Web-құжатты көруге арналған бағдарламаның жалпы атауы:
a) Browser
b) Usenet
c) E-mail
d) WWW
e) FTP

Пайдаланылған әдебиеттер

1. Снелл Нед. Интернет. Пер. с англ. –М.: «Вильямс», 2006. –384с.
2. Комягин В.Б., Печников В.Н. Создание Web-страниц и Web-сайтов. Учебное пособие. –М.: Триумф, 2003. –496 с.:ил.
3. Ибрагимов У.М., Абдиева З.А. HTML & CSS. Электронный учебник / Свидетельство о гос. регистрации объекта интеллект. собст. №288 от 14.04.2009
4. Ибрагимов О.М. Web-бағдарламалауға кіріспе. Оқу құралы. - Шымкент: ОҚМУ, 2012. -128 б.
5. Алексеев А.П. Введение в Web-дизайн. Учебное пособие. –М.: Солон-Пресс, 2008. –192 с.:ил.
6. https://ru.wikipedia.org/wiki/Всемирная_паутина
7. http://4stud.asoiu/web-programming/lectures.php
8. Таненбаум Э., Уэзеролл. Д. Компьютерные сети. 5-е издание. -СПб.: Питер, 2012. -960 с.: ил.
9. https://kk.wikipedia.org/wiki/OSI_моделі
10. http://www.firststeps.ru/cgi/cgi1.html
11. Питер Лабберс, Брайан Олберс, Фрэнк Салим. HTML 5 для профессионалов: мощные инструменты для разработки современных веб-приложений. -М.:«Вильямс», 2011. -272 с.
12. Стивен Шафер. HTML, XHTML и CSS. Библия пользователя, 5-е издание. -М.:«Диалектика», 2010. -656 с.
13. http://javascript.internet.com/
14. Гарнеев А.Ю. Web–программирование на Java и JavaScript. –СПб.: БХВ–Петербург, 2002. –1040 с.
15. Макинтош Дж. Java & JavaScript: библиотека программиста.–Спб.: Питер, 2003. –208 с.
16. Ибрагимов О.М., Оразов И. Web-дизайн: HTML+JavaScript // Свидетельство о гос. регистрации объекта в Комитете по правам интеллект. собст. МЮ РК. №679 от 29.04.2011.
17. Мазуркевич А. PHP: настольная книга программиста. –М.: Новое знание, 2004. –479 с. ил.
18. Макинтош Дж. Perl & PHP: библиотека программиста. –СПб.: Питер, 2004. –218 с. ил.
19. Орлов А.А. PHP: Полезные приемы. –2-е изд. перераб. и доп. –М.: Горячая линия – Телеком, 2006. –272 с.: ил.
20. Парижский С.М., Литвиненко Н.А. PHP. Теория и практика. –К.: МК-Пресс, 2006. -384. ил.
21. http://www.apache.org/dyn/closer.egi
22. http:// www.apache.rin.ru/dist/httpd/binaries/win32/
23. http://www.php.het/downloads.php
24. http://www.evrsoft.com/1stpage2.shtml
25. http://www.phpedit.com/en
26. http://soft.sibnet.ru/soft/5733-web-development-studio-2-0/

Қосымшалар

Қосымша 1 – Аймақтық домен аттары

AF Afghanistan	GR Greece	PE Peru
AL Albania	GL Greenland	PH Philippines
DZ Algeria	GD Grenada	PN Pitcairn
AS American Samoa	GP Guadeloupe	PL Poland
AD Andorra	GU Guam	PT Portugal
AO Angola	GT Guatemala	PR Puerto Rico
AI Anguilla	GN Guinea	QA Qatar
AQ Antarctica	GW Guinea-Bissau	RE Re'union
AG Antigua and Barbuda	GY Guyana	RO Romania
AR Argentina	HT Haiti	RU Russian Federation
AM Armenia	HM Heard and McDonald Islands	RW Rwanda
AW Aruba	HN Honduras	SH Saint Helena
AU Australia	HK Hong Kong	KN Saint Kitts and Nevis
AT Austria	HU Hungary	LC Saint Lucia
AZ Azerbaijan	IS Iceland	PM Saint Pierre and Miquelon
BS Bahamas	IN India	VC Saint Vincent and the Grenadines
BH Bahrain	ID Indonesia	WS Samoa
BD Bangladesh	IR Iran	SM San Marino
BB Barbados	IQ Iraq	ST Sao Tome and Principe
BY Belarus	IE Ireland	SA Saudi Arabia
BE Belgium	IL Israel	SN Senegal
BZ Belize	IT Italy	SC Seychelles
BJ Benin	JM Jamaica	SL Sierra Leone
BM Bermuda	JP Japan	SG Singapore
BT Bhutan	JO Jordan	SK Slovakia
BO Bolivia	KZ Kazakhstan	SI Slovenia
BA Bosnia-Hercegovina	KE Kenya	SB Solomon Islands
BW Botswana	KI Kiribati	SO Somalia
BV Bouvet Island	KP Korea	ZA South Africa
BR Brazil	KR Korea (Republic of)	SU Former Soviet Union
IO British Indian Ocean Territory	KW Kuwait	ES Spain
BN Brunei Darussalam	KG Kyrgyzstan	LK Sri Lanka
BG Bulgaria	LA Lao People's	SD Sudan

	Democratic Republic	
BF Burkina Faso	LV Latvia	SR Suriname
BI Burundi	LB Lebanon	SJ Svalbard and Jan Mayen Islands
KH Cambodia	LS Lesotho	SZ Swaziland
CM Cameroon	LR Liberia	SE Sweden
CA Canada	LY Libyan Arab Jamahiriya	CH Switzerland
CV Cape Verde	LI Liechtenstein	SY Syria
KY Cayman Islands	LT Lithuania	TW Taiwan, Province of China
CF Central African Republic	LU Luxembourg	TJ Tajikistan
TD Chad	MO Macau	TZ Tanzania
CL Chile	?? Macedonia	TH Thailand
CN China	MG Madagascar	TG Togo
CX Christmas Island	MW Malawi	TK Tokelau
CC Cocos(Keeling) Islands	MY Malasia	TO Tonga
CO Colombia	MV Maldives	TT Trinidad and Tobago
KM Comoros	ML Mali	TN Tunisia
CG Congo	MT Malta	TR Turkey
CK Cook Islands	MH Marshall Islands	TM Turkmenistan
CR Costa Rica	MQ Martinique	TC Turks and Caicos Islands
CI Cote d'Ivoire	MR Mauritania	TV Tuvalu
HR Croatia	MU Mauritius	UG Uganda
CU Cuba	MX Mexico	UA Ukraine
CY Cyprus	FM Micronesia	AE United Arab Emirates
CZ Czech Republic	MD Moldova	UK United Kingdom
DK Denmark	MC Monaco	US United States
DJ Djibouti	MN Mongolia	UM United States Minor Outlying Islands
DM Dominica	MS Montserrat	UY Uruguay
DO Dominican Republic	MA Morocco	UZ Uzbekistan
TP East Timor	MZ Mozambique	VU Vanuatu
EC Ecuador	MM Myanmar	VA Vatican City State
EG Egypt	NA Namibia	VE Venezuela
SV El Salvador	NR Nauru	VN Vietnam
GQ Equatorial Guinea	NP Nepal	WF Wallis & Futuna Islands
EE Estonia	NL Netherlands	EH Western Sahara

ET Ethiopia	AN Netherlands Antilles	YE Yemen
FK Falkland Islands	NT Neutral Zone (between Saudi Arabia & Iraq	YU Yugoslavia
FO Faroe Islands	NC New Caledonia	ZR Zaire
FJ Fiji	NZ New Zealand	ZM Zambia
FI Finland	NI Nicaragua	ZW Zimbabwe
FR France	NE Niger	
GF French Guiana	NG Nigeria	
PF French Polynesia	NU Niue	
TF French Southern Territories	NF Norfolk Island	
GA Gabon	MP Northern Mariana Islands	
GM Gambia	NO Norway	
GE Georgia	OM Oman	
DE Germany	PK Pakistan	
VG Virgin Islands (British)	PW Palau	
VI Virgin Islands (U.S.)	PA Panama	
GH Ghana	PG Papua New Guinea	
GI Gibraltar	PY Paraguay	

Қосымша 2 – Қазақша әріп кодтары

Бас әріп	код	Кіші әріп	код
Ә	Ә	ә	ә
Ө	Ө	ө	ө
Ң	Ң	ң	ң
Қ	Қ	қ	қ
Ғ	Ғ	ғ	ғ
Ү	Ү	ү	ү
Ұ	Ұ	ұ	ұ

Нысанов Ерназар Айдарұлы
Құрақбаев Жұмағали Салбекұлы
Ибрагимов Оспанәлі Мусақулұлы

WEB-БАҒДАРЛАМАЛАУҒА КІРІСПЕ

Оқу құралы

Басуға _____ қол қойылды
Пішімі $X^{x}Y$ 1/16
Типографиялық қағаз. Офсеттік баспа. Көлемі __ б.т.
Таралымы _____ дана. Тапсырыс _____

© М.Әуезов атындағы Оңтүстік Қазақстан мемлекеттік университетінің баспаханасында шығарылған