



Цифрлық әдіскер

ақпараттық-білім беру ортасы

ДӘРІСТЕР ЖИНАҒЫ

Web технологиялар

HTML, CSS және JavaScript негізінде веб-беттер құру — World Wide Web-тің жұмыс принциптерінен бастап интерактивті беттерге дейін.

HTML және гипермәтін

CSS стиль кестелері

Кестелер, фреймдер, формалар

JavaScript негіздері

КЕШЕН

Дәріс сабағының құрылымы: HYPERLINK "../Мои документы/Downloads/HTML2.doc"World Wide Web Интернет-технология туралы түсінік Компьютерлік желілер

HYPERLINK "../Мои документы/Downloads/HTML2.doc"World Wide Web

Интернеттің қазіргі дамуы 90-жылдар басында компьютерлер арасында мәлімет алмасудың жаңа хаттамасы (protocol) пайда болғаннан кейін басталды. Бұл хаттама HTTP (Hyper Text Transfer Protocol- гипермәтінді тасымалдау хаттамасы) деп аталған болатын. Осы хаттамамен қатар HTTP серверлерінің кеңейтілген желілері болып табылатын, Интернет арқылы файлдар тасымалдай алатын World Wide Web қызмет бабы (WWW немесе тек Web) пайда болды. Бұл файлдардың басым көпшілігі Web- парақтар түрінде HTML (Hyper Text Markup Language – гипермәтінді белгілеу тілі) тілінде жазылған арнаулы файлдар түрінде болады. Осы файлдарды HTTP серверлерінде (Web-топтарында) орналастыру жолымен Web-парақтар қалың көпшілік пайдаланатындай түрде Интернетте жарияланады. Web- парақтар мазмұны әртүрлі бола береді және олар әртүрлі тақырыптарды қамтиды, бірақ олардың бәрінің де негізгі жариялану яғни жазылу тілі HTML болып табылады. Осындай HTML құжаттарының бәрінің де файл аттарының кеңейтілуі (тіркеуі) HTM немесе HTML болуы тиіс. HTML тілі World Wide Web қызмет бабымен бірге дами отырып, Web – парақтарының ең жақсы деген мүмкіндіктерін жүзеге асырып, оны кең пайдалану жолдарымен толықтырылып отырады. Ол World Wide Web жүйесінің негізі бола отырып, оның өте кең тарауына себепші болды. World Wide Web сөзі қазақ тіліне кеңейтілген бүкіләлемдік өрмек болып аударылады. HTML тілінің мағынасы мен атқаратын қызметін оның атынан анықтауға болады.

Интернет-технология туралы түсінік

Қазіргі кезде Internet те телефон, факс, сияқты адам өмірінің ажырамас бөлігі болып табылады. Адамдардың барлығы қандай да бір ақпарат алу, бір-бірімен сөйлесу мақсатында Internet деп аталынатын дүниежүзілік өрмекте саяхат жасайды. Internet дегеніміз глобальді желі. Интернет физикалық мағынасында бір-бірімен байланыс линиялары арқылы байланысқан бірнеше миллиондаған компьютерлер деп қарастыруға болады. Бірақ Интернетті физикалық мағынада емес, оны ақпараттық кеңістік деп қарастырған дұрыс. Интернетті ішінде мәліметтердің үздіксіз циркуляциясы жүретін кеңістік ретінде қарастыруға болады. Интернеттің теле және радиоэфирден айырмашылығы – эфирде ешқандай ақпарат сақталынбайды, ал Интернетте ол желінің түйіндерінде болатын компьютерлер арасында орын ауыстырады да, қандай да бір уақыт аралығында қатты дискіде сақталады. 2 немесе одан да көп компьютерлер арасындағы физикалық байланысу компьютерлік желі деп аталынады. Жалпы компьютерлік желілерді құру үшін арнайы аппараттық қамсыздану және арнайы бағдарламалық қамсыздану қажет.

Компьютерлік желілер

Компьютерлік желілер аппаратты ресурстардан басқа қосымша программалық ресурстарды да пайдалануға мүмкіндік береді. Компьютерлік желілер локалді және глобальді болып бөлінеді. Интернет – глобальді желі. Қазіргі кезде әлемде 200-ден астам глобальді желілер бар. Барлық әлемдегі глобальді есептеуіш желілер өзара Интернет көмегімен байланысқан. ҒЕЖ-де жұмыс істеу үшін сәкес аппаратты және программалық қамсыздану болу керек. Барлық қамсыздану 2 топқа бөлінеді:

1. Бағдарлама-серверлер - қолданушының компьютеріне қызмет көрсететін желі түйінінде орналасатын бағдарламалық қамсыздану.
2. Бағдарлама-клиенттер – қолданушының компьютерінде орналасады және сервердің қызметін пайдаланатын бағдарламалық қамсыздану.

Өзін-өзі тексеруге арналаған сұрақтар: Интернет деген не? Интернет – технологиялар деген ұғымға қандай анықтама бере аласыз? Желі деген не? Желінің қандай түрлері болады? Қамсыздану түрлерін атаңыз.

Ұсынылатын әдебиет:

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Учебник.-СПб: Питер, 1999.
2. Комер Д. Принципы функционирования Интернета СПб: Питер,2002

Дәріс 2. Интернеттің ұйымдастырылуы

Дәріс сабағының құрылымы: Интернеттің жұмыс істеу принципі. Интернетпен жұмыс істеу кезінде кездесетін негізгі ұғымдар. Интранет туралы түсінік.

Өзара әсерлесетін және бір-бірімен байланысқан компьютерлер тбы өрмекті құрайды. Интернет – бүкіл әлем бойынша таратылған мыңдаған кіші желілерден тұратын компьютердің үлкен өрмегі. Интернеттегі компьютерлер бір ортақ тілде сөйлеседі, ал бұл тіл протоколдарда бекітілген. Протокол деген желідегі компьютерлердің өзара әрекеттесу механизмін анықтайтын жалпыға бірдей ережелер жиынтығы. Интернеттегі мәліметтермен алмасуға арналған стандартты протоколдар: TCP және IP. Осы екі протоколдың комбинациясы TCP/IP тірінде болады. Интернетке қатысты қолданылатын терминдер: Айна - алыста орналастырылған атақты сервердің көшірмесі болатын сервер. Прокси-сервер - -желідегі жүктемені төмендететін және желідегі іс-әрекетті тездететін компьютер. Шлюз -әр түрлі протоколдармен жұмыс жасайтын әр текті екі желіні біріктіретін бағдарламалық-аппараттық құрал. Брандмауэр – қорғауға алынған желіге санкциясыз енуді тоқтатын аппараттық құрал. Маршрутизатор – бірнеше каналдармен жұмыс істейтін құрылғы. Маршрутизациялау – ақпараттың жіберілу жолын таңдау процедурасы. Интернет көптеген әр түрлі ұйымдардың компьютерлерін біріктіретін болса, Интранет бір ғана ұйымның ішіндегі компьютерлерді біріктіреді. Интранеттегі ақпарат тек қана сол ұйымның адамдары пайдалана алады. Интранет желілері электронды пошта секілді бизнес үшін өте тиімді коммуникациялық құрал болып табылады.

Өзін-өзі тексеруге арналаған сұрақтар: Интернеттің жұмыс істеу принципі қандай? Интернеттегі жұмыс қалай ұйымдастырылады? Интранет деген не? Интранеттің негізгі ерекшеліктері қандай?

Ұсынылатын әдебиет:

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Учебник.-СПб: Питер, 1999.
2. Комер Д. Принципы функционирования Интернета СПб: Питер, 2002

Дәріс 3. Интернетке қосылу

Дәріс сабағының құрылымы: Жүйелердің қосымша бөліктерінің салу принциптері.

1. Трафикты локалдау және желілерді қаптау

1. Бірнеше компьютерлер арасында жобалау және есептерді шығаруды жеңілдету үшін ол жүйені бірнеше бір-бірімен байланысқан топтарға бөлінеді. Жүйе арқылы алмасу жүріп жатқанда екі торап келесі ережелерді сақтауы тиіс: Электрлік сигналдардың түрін және деңгейлерін келістіру, Берілістердің ұзындығын анықтау әдістері, Бақылау әдістері. Келісімдер мәліметтерді интерпретациясын анықтайтын, биттарды беру ең кіші деңгейден үлкен деңгейлерге дейін барлық деңгейлерге бірдей болу керек. Бір деңгейдегі берілістердің форматын және тізбегін анықтайтын ондай формалданған ережелерді протоколдар деп атайды. Иерархиялық жүзеге асырылған протоколдардың жиынтығын коммуникациялық протоколдардың стеки деп атайды. Желілік деңгей біріншіден келесі есептерді шығару үшін құралдармен қамтамасыз ету керек: Еркін топологиясы бар желілерге пакеттерді жеткізу, Желіні сенімді локалды трафиктың арқасында құрылымдау, әртүрлі протоколдардың каналдық деңгейледің келісуі.
1. Желідегі трафик кездейсоқ ретімен орналасады, бірақ онда кейбір заңдылықтар көрсетіледі. Кейбір жалпы есеппен жұмыс істейтін қолданушылар (мысалы, бір бөлімнің қызметкерлері) көбінесе бір-біріне немесе жалпы серверге сұраумен қатынасады. Сол кезде олардың басқа бөлімдердің компьютерлерінің ресурстарымен қолдануы қажет болады. Сондықтан жүйенің құрылымы мәліметтердің ағынының құрылымына сәйкес келу керек. Жүйедегі трафигіне байланысты компьютерлер бірнеше бөлімдерге (сегменттерге) бөлінуі мүмкін. Егер бір-бірімен қатынасатын компьютерлердің сұраулары мен жауаптары осы топтың компьютерлеріне арналса, онда олар бір бөлімге жинақталады. Жүйені сегменттерге бөлу үшін мосттар мен коммутаторлар қолданылады. Олар сегменттің ішіндегі локалды трафикті сыртқа ешқандай кадрларды бермей экрандайды, тек қана басқа сегменттерде орналасқан компьютерлерге арналған кадрлардан басқа. Сонда жүйе бірнеше жүйелерге бөлінеді. Ол бір топтың ішіндегі трафиктың жиілігін және топтардың арасындағы алмасудың активтігін ескере отырып байланыс бөлімдерінің өткізу мүмкіндігін рационалды түрде таңдау мүмкіндігін тудырады. Бірақ трафикты мосттардың және коммутаторлардың көмегімен локалдаудың да Однако локализация трафика средствами мостов и коммутаторов имеет существенные ограничения. Бір жағынан жүйенің мосттардың арасында орналасқан логикалық сегменттері бір-бірінен дұрыс оқшауланбаған, өйткені олар кеңдік толқындарынан қорғалмаған. Егер қандай да бір станция кеңдік толқындары арқылы мәліметтерді жіберсе, онда ол мәліметтер жүйенің станцияларының барлық логикалық сегменттеріне беріледі. Мосттардың көмегімен салынған кеңдік толқындарынан қорғану үшін: администратор тек бір түйінге генерирлеуге рұқсат етілген кеңдік толқындарының пакетын шектейді. Екінші жағынан комутаторда жасалған локалды жүйелердің виртуалды сегменттердің механизмдерін пайдалану трафиктың толық локалдауына әкеліп соғады- ондай сегменттер бір-бірінен толық оқшауланған. Сонда мосттар мен комутаторлардың көмегімен салынған жүйелердегі

әртүрлі виртуалды сегменттерге жататын компьютерлер біркелкі жүйені құрамайды. Мосттармен коммутаторлардың келтірілген кемшіліктері олардың каналдық деңгейлердің протоколдарымен жұмыс істеуіне байланысты. Ол протоколдарда үлкен жүйені құрылымдау кезінде пайдалануға болатын жүйенің бөлігі немесе сегментті анықталмайды. Жүйелік технологияларды жобалаушылар каналдық деңгейлерді жетілдірудің орнына ол есепті жүйелік деңгейлерге жүктеді.

Өзін-өзі тексеруге арналаған сұрақтар: Жүйе арқылы алмасу жүріп жатқанда екі торап қандай ережелерді сақтауы керек? Протокол деген не? Трафик деген не? Жүйені сегменттерге бөлу үшін не қолданылады?

Ұсынылатын әдебиет:

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Учебник.-СПб: Питер, 1999.
2. Комер Д. Принципы функционирования Интернета СПб: Питер, 2002

Дәріс 4. Интернеттің қызмет көрсетуі

Дәріс сабағының құрылымы: Интернет ерекшеліктері Почталық байланыс Интернетпен байланысу

Интернеттің қарапайым желілік нүктелері өкімет мекемелерінде, университеттерде, коммерциялық фирмаларда, жергілікті кітапханаларда, мектептерде де орналасқан. Интернеттің бір ерекшелігі оның құрамындағы көптеген компьютерлер нақты BBS тәрізді жұмыс істейді. Мұның өзі Интернетке қосылу деген – басқа жерлерде тұрған 1000-даған компьютерлік жүйелермен байланысу деген сөз. Интернет ақпарат магистраліне өте ұқсас, институт, мектеп терминалы арқылы оған жеңіл кіруге болады, ол үшін Интернеттегі жүйенің нөмірін теру керек. Мұнан кейін керекті жердегі желі нүктесімен байланысып, өзіңізге қажет материалдарға қол жеткізесіз. Интернетпен байланысқан провайдер компаниясы деп аталатын мекемелер әрбір компьютерді Интернетке қосып бере алады. Желіге қосылудың бірнеше түрі бар: тұрақты қосылып тұрмайтын байланыстар; қосылып тұратын тікелей байланыстар; почталық байланыстар. Интернетпен қосыла латаын бірнеше почталық байланыс түрлері бар. Провайдері CompuServe America Online болып келген компьютерлер бірден Интернетпен почталық байланысқа кіре алады. Бұл ортада әр түрлі тақырыптардағы дискуссияларға қатысу үшін LISTSERV жүйесін пайдаланған дұрыс. Осы сияқты почталық байланыстар желілік шлюздер деп аталады. Бізге кенеттен бір файл керек болып қалды делік. Оның қай жерде екені бізге белгілі болсын. Ол файл тегін берілетін программа, жұмысқа керекті ақпарат, сурет немесе кітап болуы мүмкін. Енді осы файлды өз компьютерімізге қалай әкелу жолын қарастырайық. Мұндай мақсат үшін файлды тасымалдау протоколы деп аталынатын жүйе қолданылады. Каталог немесе почталық хабарда файлдарды алу үшін компьютерге FTP жеткізу деген сөздер жиі кездесуі мүмкін. Ол осы файлды алу үшін FTP жүйесі қолданылатынын білдіреді. FTP арқылы қызмет енетін арнайы программасы бар желіге қосылған компьютер FTP-сервер деп аталады. Көптеген FTP –серверлер барлық адамдар үшін ашық болады, кез келген жан одан администратор рұқсатымен әр түрлі мәліметтер ала алады. Бұл тәсіл анонимдік FTP деп аталады.

Өзін-өзі тексеруге арналаған сұрақтар: Интернетке қандай ерекшеліктер тән? Почталық байланыс қалай жүзеге асады? Провайдер деген не? FTP жүйесі деген не? FTP-сервер деген не?

Ұсынылатын әдебиет:

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Учебник.-СПб: Питер, 1999.
2. Комер Д. Принципы функционирования Интернета СПб: Питер, 2002

Дәріс 5. Электрондық почта

Дәріс сабағының құрылымы: Электронды почта туралы түсінік. Хатты толтыру. Телеконференциялар. Файлдың серверлері.

Электрондық почта қашықта орналасқан абоненттермен компьютер арқылы байланысудың ең ыңғайлы түрі болып табылады. Мәліметті өте жылдам, әрі толық күйінде оны өзгертпей барар жеріне сенімді түрде жеткізетіндіктен, қазіргі почта қызметінің ертеңі электронды почта болады деп күтілуде. Электрондық почта кез келген құжатты – ол мейлі қызмет жайлы жалпы мәлімет бола ма немесе жай хабар түріндегі мәтін бола ма – бәрі бір олардың ақпараттық негізін бірдей түрде нық ұстап, нақты әрі сенімді түрде жеткізе алады. Электрондық почтаны пайдалану кезінде компьютерде әр түрлі арнаулы программалар қолданылады және олардың қажеттілігіне қарай бірінен біріне жеңіл ауысып отыруға болады. Мұндай программалар мынадай жұмыс атқарады: алынған мәліметтерді талдау және жаңа мәліметер дайындау; байланыс торабымен жылдам мәлімет алмасу.

1. Кәдімгі почта тәрізді электрондық почта да бір абонент пен екінші абонент арасында мәлімет алмасу үшін қолданылады. Жіберілетін мәліметте екі адрес көрсетілуі тиіс: хатты жіберуші және оны алушының аты-жөні мен

оның компьютерінің желідегі адресі көрсетіледі. Хат иесінің адресін компьютер автоматты түрде өзі толтырады. Бұған қоса электрондық почта адрестерінде мынадай мәліметтер жазуға болады:

2. мәлімет тақырыбының қысқаша сипаттамасы;
3. хаттың көшірмесі тағы қандай адрестерге жіберілуі тиіс. Электрондық почта жұмысы барысында хаттың тақырыбына қоса оның жіберілген мерзімі, уақыты т.с.с. керекті мәліметтер тіркеліп қлады.
4. Көптеген желілер алғашқы ретте ғылыми, конструкторлық және оқу орындарын байланыстыратын жүйе ретінде жасалған болатын. Сондықтан ұжымдық пікір-талас жүргізуге арналған желі мүмкіндігі телеконференция деп аталып, ол белгілі бір салаға ғана қолдануға негізделген еді. Бірақ кейінен желілер практикада кең пайдаланылып, оның алғашқы ғылыми семинар, конференцияға арналған мүмкіндігі біршама шетте қала бастады. Телеконференция ұйымдастыру кәдімгі кітапхана немесе архив жұмысына ұқсас жүргізіледі. Мұнда да көпсатылы тақырыптық каталогтар мен олардағы мәліметтерді сақтау ісі ұйымдастырылады.
5. Бірсыпыра байланыс тораптарының файлдық серверлері өз абоненттеріне өздерінде жинақталған ақпараттық материалдарды, яғни файлдар архивін пайдалануды ұйымдастырып жүр. Ол материалдар мәтіндік құжаттардан, кітаптардан, сызбалардан, компьютерлерге арналған программалардан т.б. тұрады. Бұл мәліметтер ұзақ уақыт сақталады және олардың көлемі де үлкен болады.

Өзін-өзі тексеруге арналаған сұрақтар: Электронды почта деген не? Электронды почтамен қалай жұмыс істеу керек? Электронды почтаның кәдімгі почтадан айырмашылығы қандай? Хатты толтыру қалай жүзеге асады? Телеконференциялар деген не? Файлдың серверлері қандай қызмет атқарады?

Ұсынылатын әдебиет:

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Учебник.-СПб: Питер, 1999.
2. Комер Д. Принципы функционирования Интернета СПб: Питер, 2002

Дәріс 6. SDH желілері

Дәріс сабағының құрылымы:

SDH технологиясының пайда болуы SDH хаттамалары

1988 жылы мәліметтер жіберудің синхронды сандық иерархия (SDH) стандартының пайда болуы транспорттық желілердің дамуының жаңа кезеңінің негізін қалады. SDH технологиясы сенімді мәлімет жіберу ұйымымен кең пайдаланылады. Ол төмендегідей мақсаттар үшін қолданылады: Желілік қызметтерді қамтамасыз етуші провайдерлердің қарым-қатынасы үшін стандартты хаттама алу; Американдық, Еуропалық және Жапондық сандық жүйелердің углификациясы; Гега биттік жылдамдықта мультиплексерленген сандық сигналдарды қамтамасыз етеді; Эксплуатация функцияларын қолдауды және OADM техниканың қызмет көрсетуді (Operation, administration and maintenance-функциялану, администрация және техниканың құызмет) қамтамасыз ету; Синхронды жіберу жүйесі алдындағы жүйе (PDH) шектеулерін жеңіп қана қоймай, сонымен қатар ақпаратты жіберудің (накладные) шығыны төмендетті. Уникалды артықшылықтар қатары: барлық ақпаратты мультиплексерлемей – ақ төменгі жылдамдықтағы командирға рұқсаттың болуы, (высокая отказоустойчивость) мониторинг және басқарудың дамыған құрылғылары, тұрақты абоненттік байланыстарды жұмсақ (гибкость) басқару. Бұл жүйелер жаңа кезеңді бірінші желілерінің негізі болады.

SDH хаттамалардың етігі үш негізгі деңгейдегі хаттамалардан тұрады. Байланыс деңгейі – желінің ені қолданушылардың арасындағы мәліметтерді жіберуді бақылайды; Мәліметтерді жіберуді басқару деңгейі – физиканың және административтік бақылау операцияларын қамтиды, желідегі бір элементтің қайтып келуі (отказ) жағдайында әр түрлі реконфигурациялық операцияларды жүзеге асырады.

SDH желісін ұйымдастыру Физикалық деңгей, стандартына фотон (Photonic) деп аталады – свет модуляциясының көмегімен ақпарат биттерін кодтаумен жұмыс істейді. Қазыргы кезде SDH технологиясы перспективаны ғана емес, сонымен қатар транспорттық желі құру үшін де апробиривті табылады. Оны инвестициялық, эксплуатациялық және пайдаланушылық көз қараспен қарағанда бір неше артықшылықтар қатары бар. Ол артықшылықтарға төмендегілер жатады. Монтажда, эксплуатацияға және желінің дамуына, сонымен қатар жаңа тораптарға қосылудың шығындарын төмендететін тұрақты құрылымдық күрделілігі; Мүмкін болатын жылдамдықтың кең диапазоны -155,520 МБит/с – тан 2,488 ТБит/с – қа дейін (STM – 16) және жоғары; Командамен интеграция мүмкіндігі, себебі PDH – ның сандық командалары SDH желісі үшін кіретін командалар болып табылады; Орталықтандырылған мониторинг және басқару арқасында жүйенің жоғарғы сенімділігі, сонымен қатар резервтегі командаларды пайдалану мүмкіндігі; Қызметті динамиканың түрде ұсыну мүмкіндігі – жүйені инфоқұрлысына өзгертулер енгізбей – ақ, абоненттер үшін командалар динамикалық түрде құрылуы және басталуы мүмкін; Жүйе кеңейтілімін және интеграциясын жеңілдететін технологиясын

жоғарғы деңгейдегі стандартизациясы әртүрлі өнеркәсіптер құрылғылардың қолдануға мүмкіндіктер береді; Әлемде SDH стандартының кең таралуы; Айтылған артықшылықтар SDH технологиясына негізделген, инвестиция көз қарасынан рационалды болып табылатын шешім қабылдайды.

Өзін-өзі тексеруге арналаған сұрақтар: SDH технологиясы туралы түсінік. SDH технологиясы қалай пайда болды? SDH хаттамалары деген қандай хаттамалар? SDH технологиясының артықшылықтары қандай?

Ұсынылатын әдебиет:

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Учебник.-СПб: Питер, 1999.
2. Комер Д. Принципы функционирования Интернета СПб: Питер, 2002

Дәріс 7. Интернеттегі қауіпсіздік

Дәріс сабағының құрылымы: Электрондық ақпараттық ресурстар мен ақпараттық жүйелерді қорғаудың мақсаттары. Электрондық ақпараттық ресурстар мен ақпараттық жүйелерді қорғау. Электрондық ақпараттық ресурстарды қорғауды ұйымдастыру.

Электрондық ақпараттық ресурстар мен ақпараттық жүйелерді қорғаудың мақсаттары:

1. Электрондық ақпараттық ресурстар мен ақпараттық жүйелерді қорғау: Электрондық ақпараттық ресурстардың тұтастығын және сақталуын қамтамасыз ету, оларды рұқсатсыз өзгертуге немесе жоюға жол бермеу; Қол жетімділігі шектелген электрондық ақпараттық ресурстардың құпиялығын сақтау; Электрондық ақпараттық ресурстарға қол жеткізу құқығын іске асыру; Электрондық ақпараттық ресурстарды өңдеу және оларды беру құралдарына рұқсатсыз ықпал етуге жол бермеу мақсатында құқықтық ұйымдастырушылық және техникалық (бағдарламалық техникалық) шараларды қабылдауды білдіреді;
 2. Электрондық ақпараттық ресурстар мен ақпараттық жүйелерді қорғау: Электрондық ақпараттық ресурстарды құрсаулау, яғни ақпараттық жүйеге және ол ұсынатын электрондық ақпараттық ресурстарға қолжетімділікті шектеуге немесе жабуға әкелетін іс-қимылдар жасау; Электрондық ақпараттық ресурстарды түрлендіру, яғни бағдарламаға, деректер қорына, материалдық жеткізушідегі мәтіндік ақпараттарға өзгерістер енгізу; Электрондық ақпараттық ресурсты көшіру, яғни ақпаратты басқа материалдық жеткізушіге ауыстыру; Құқық иесінің рұқсатынсыз бағдарламалық өнімдерді пайдалану;
1. Электрондық ақпараттық ресурстарды қорғауды ұйымдастыру
 2. Электрондық ақпараттық ресурстарды қорғауды: Жалпыға бірдей қол жетімді электрондық ақпараттық ресурстарға қатысты – электрондық ақпараттық ресурстарды тарататын тұлға; Қолжетімділігі заңмен шектелген электрондық ақпараттық ресурстары бар ақпараттық жүйенің меншік иесі, иеленушісі немесе ұлттық операторы; Қолжетімділігін олардың меншік иесі немесе иеленушісі шектелген электрондық ақпараттық ресурстарға қатысты – электрондық ақпараттық ресурстардың меншік иесі немесе иеленушісі ұйымдастырады.

Өзін-өзі тексеруге арналаған сұрақтар: Электрондық ақпараттық ресурстар мен ақпараттық жүйелерді қорғаудың мақсаттары. Электрондық ақпараттық ресурстар мен ақпараттық жүйелерді қорғау. Электрондық ақпараттық ресурстарды қорғауды ұйымдастыру.

Ұсынылатын әдебиет:

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Учебник.-СПб: Питер, 1999.
2. Комер Д. Принципы функционирования Интернета СПб: Питер, 2002

Дәріс 8. Гипермәтіндік белгілеулердің принциптері

Дәріс мазмұны: Құжаттардың құрылымы. HTML тегінің топтары.

Гипермәтіндік белгілеулердің принциптері. Құжаттардың құрылымы

HTML – да тегтік модельдер құжаттың белгілеулері болып табылады. Тегтік модель құжатты тегпен басталып және аяқталатын контейнерлердің жиынтығы сияқты сипаттайды. Яғни HTML құжаты қарапайым ASCII-файлы сияқты көрсетіледі. Көбінесе HTML құжаттарының тегтерін түсіну және қолдану оңай, өйткені олар ағылшын тілінің сөздерімен пайдаланылған қысқартуларды және белгіленулерді түсінеді. HTML – тегі қажет емес тегтің атрибуттар тізімінен кейін болатын аттан құралады. Тегтің мәтіні бұрышты жақшалардан тұрады (< және >). Тегтің ең қарапайым оңай варианты – бұрыштық жақшаларға негізделген аты. Мысалға: <HEAD> немесе <i>. Қиын тегтер үшін, функцияның тегтерін видео өзгерту үшін автормен анықталған дәл мағынасы бар болатын атрибуттардың айырмашылығы сипатталады. Тегтердің

атрибутов атымен жүреді және бір немесе бірнеше табуляциялардың белгілерімен біріншісі екіншісінен бөлінеді. Тегте атрибуттардың жазылу реті қажет емес. Атрибуттардың мағынасы, егер ондай бар болса, атрибуттардың атынан кейін тұратын теңдік белгісінен кейін тұрады. Егер атрибуттың мағынасы бір сөз немесе сан болса, онда оны қосымша белгілемей – ақ теңдік белгісінен кейін көрсетуге болады. Барлық қалған белгілеулерді бір (') немесе екі (") тырнақшалармен аяқтау керек, әсіресе егер олар бірнеше пробелдармен бөлінген сөздерден тұрса. Атрибуттың ұзындығының мағынасы 1024 таңбамен шектелген. Атрибуттардың мағынасын айтуға болмайтын, тегтердің және атрибуттардың аттарында таңбалардың регистрі саналмайды. Мысалы, HREF атрибутының мағынасы ретінде басқа құжаттарда URL-ді енгізу кезінде керекті регистрді қолдану қажет. Көбінесе HTML-тегі арасында мәтін және құжаттардың басқа элементтері орналасатын бастапқы және соңғы компанеттерден тұрады. Соңғы тегтің аты бастапқы тегтің атымен бірдей, соңғы тегтің алдына қисық сызық (/) таңбасы қолданылады. Мысалы: шрифт тегінің – курсив түрі үшін <i>, оны жабатын қос былай белгіленеді – </i>, ал тақырып тегі үшін <TITLE>, оны жабатын қос былай болады – </TITLE>. Соңғы тегтер ешқашан атрибуттардан тұрмайды. Тегтер мағынасы әмбебап программалау тіліндегі тырнақшалардағы "begin/end" түсінігіне жақын. Жазылған тегті қолдану кезінде құжатта ерекше назар аударып отыру керек. Соңынан бастап біріншісіне дейін жазылған тегтерді жауып отыру керек. Кейбіреуі автономды элементтер болғандықтан HTML-тегтері соңы элементтерден тұрмайды. Мысалы графикалық бейне құжатына қою үшін, бейне тегі қолданылады. Сондай-ақ автономды тегтерде жолдарды бөлу (
), көлденең сызық (<HR>) және көрініс мазмұнына әсер етпейтін құжат туралы ақпараттан тұратын тегтер, мысалға: <META> және <BASE>. Ең көп таралған тег абзац тегі – <P> өйткені оны құжаттарда өте жиі қолданады, кейде оны әр абзацтың басына қояды. Бір абзац біткен кезде, келесі тег <P> браузерге: «Алдындағы абзацты аяқтап келесіні бастау керек» деп сигнал береді. Авторлардың көбі абзацтың соңғы тегін қолданбайды. Басқада соңғы тегтер бар, браузерлер жақсы жұмыс істейтін. Мысалы HTML-дың соңғы тегі </HTML>. Денген менде құжатты құруда шатаспау және қате кетіrmес үшін соңғы тектерді мүмкіндігінше қолданып отыру керек. Жалпы HTML форматында контейнерлерді құрау схемасының жазылу түрі: "контейнер" := "<тегің аты" "атрибуттардың тізімі" > контейнер мазмұны </"тегің аты">

HTML тегтерінің топтары

Барлық HTML тегтернің тағайындалуы мен әрекеттесу орталарын келесі негізгі түрлерге бөліп көрсетуге болады; Құжаттың құрлымын анықтайтындар;

Гипермәтін блоктарын өңдеу (параграфтар, тізімдер, кестелер, суреттер);

Гипермәтіндік сілтемелер немесе закладки; Диалог ұйымдары үшін формалар; Программаны шақыру.

Гипермәтіндік желінің құрлымы гипермәтіндік сілтемелермен беріледі. Гипермәтіндік сілтеме – бұл сілтемесі анықталған тематикалық, логикалық немесе құжатпен қандайда бір басқа тәсілмен байланысқан HTML құжатының басқа адресі немесе Internet – тің ақпараттық ресурсы. WWW жүйесінде гипермәтіндік сілтемелерді жазу үшін, Universe Resource Locator деп аталатын арнайы форма өңделіп шығарылған. Бұл өңделген форманы қолдануды келесі мысалда көруге болады: Бұл мәтін

```
<A HREF="http://polyn.net.kiae.su/altai/index.html">  
Гипермәтіндік сілтеме </A> – дан тұрады.
```

Жоғарыда көрсетілген мысалда HTML – де якорь (anchor) деп аталатын "A" тегін URL формасында бұл сілтемені жазу үшін, гипермәтіндік сілтемені (Hypertext Reference) білдіретін "HREF" атрибутын қолданады. Берілген сілтеме "polyn.net.kiae.su" желіде "http" протоколымен жүзеге асырылатын рұқсат "altai" директориясында "index.html" атпен жазылған құжатты көрсетеді. HTML – гипермәтіндік сілтеме екі класқа бөлінеді: жалпы және контексті гипермәтіндік сілтеме. Жалпы сілтемелер толықтай барлық құжаттармен байланысқан және құжатың кез келген фрагментің көру кезінде қолдану мүмкін уақытта, жоғардағы мысалда көрсетілгендей, контексті сілтемелер құжат денесінде вмантированный. Стандартты тілде, оның пайда болған кезінен бастап екі класс сілтемелері бар, бірақ алғашқы кезден бастап әйгілілік контекстік сілтемелерді қолданған. Бұл әйгілілік пайдалану механизімі жалпы сілтемелерді түгелдей "атрофирлеуге" әкеп соқты. HTML-құжатының құрлымы бірін-біріне қойған контейнерлерді қолдануға мүмкіндік береді. Құжаттың өзі – бұл <HTML> тегінен басталатын және </HTML> тегімен аяқталатын үлкен бір контейнер:

```
<HTML> құжаттың мазмұны </HTML>
```


HTML контейнері немесе гипермәтіндік құжат екі қойылған контейнерден тұрады: құжаттың тақырыбы (HEAD) құжаттың денесі (BODY): Класикалық құжаттың қарапаймы түрін қарастырайық: Netscape Communication компаниясы фрейм организациясының мүмкіншілігімен құжаттың класикалық формасын кеңейтті, онда жұмыс терезесін бірнеше бағынышсыз фреймдерге болуге болады. Әр фреймда өзінің HTML беті жазылған.

Өзін тексеруге арналған сұрақтар:

1. Интернетте адресация: ip-адресі және URL.
2. HTML тілі. Таңбалар, тегтер, элементтер, атрибуттар.

Ұсынылатын әдебиеттер:

1. Веб-дизайн/Дмитрий Кирсанов - СПб: Символ-Плюс, 2004-376 стр.
2. HTML 4.0 в подлиннике/HYPERLINK ".Матросов, HYPERLINK ".Сергеев, HYPERLINK "http://www.books.ru/shop/authors/235" Чаунин – СПб: BHV, 2000 – 672 стр.

Дәріс 9. HTML-құжатының контейнері

Дәріс мазмұны: Құжат бетінің тегі Белгілеуді басқару тегі Символдарды бейнелеуді басқару тегі Бейнелеу формасын басқару тегі Табуляция Тізім Гипермәтіндік сілтеме

HTML-құжатының контейнері

Әрбір құжаттың құрама бөліктерінде оның ішінде қолданылатын өзінің контейнер жинағы болады. Құжат денесінің контейнерлері тақырыпта немесе FRAMSET контейнерінде қолданылмайды. Контейнерлердің әрбір тобын толықтай қарастырайық. HTML-HEAD құжатының тақырыбының контейнерлері Құжат тақырыбы атрибут қажет етпейді. Тақырып тегінің басты тағайындалуы – бұа барлық құжаттардың бейнелеу параметрлерін түгелдей сипаттау. Бұндай параметрлерге құжаттарды бейнелеу стилін, гипермәтіндік сілтеменің жалпы базалық адресін, жалпы гипермәтіндік сілтеме, идентификатор және құжат атын және т.б. жатқызуға болады. Біз тек қана жиі кездесетін контейнерлерді қарастырамыз. TITLE Тақырып тегінде жиі қолданылатын құжат аты болып табылады. TITLE келесі синтаксисті алады

```
<TITLE> Құжат аты </TITLE>
```

TITLE тегінің мазмұны құжат атының алаңында бейнеленеді. BASE BASE тегі URL формасында гипермәтіндік сілтеме көрсету формасымен байланысты. URL спецификациясы құжат адресатының екі формасын анықтайды: толық және толық емес. HTML URL адресінің толықтай формасын қолдануға рұқсат етеді. Сонымен спецификацияның екінші формасын қолдану үшін, оны бірдемемен негіздеу керек, базалық адресі сурау толық еместен URL-дің толық формасын қалыптастыруда қолдануға болады. BASE тегі осы базаны анықтауға мүмкіндік береді. ISINDEX HTML-құжатын кілттік сөз бойынша іздеу мүмкіндігі құжат тақырыбының ISINDEX тегі арқылы анықталады. Тілдің алғашқы версиясында берілген тег қосымша атрибут қабылдамады. Егер сервер кілттік сөз бойынша сұраныс жасайтын болса, онда ол автоматты түрде тақырыпқа ISINDEX тегін қояды. Кілттік сөз тізімін клиент құжат адресіне "?" символдан кейін жазады. Түсінікті, сұранысты сервер атқарады, "?" символдар түгел болған кезде іздеу машинасына айналды. HTML-құжаты "жазғы" программасымен, сонда кілттік сөз "?"-кейін осы программаның адресіне жазылады. Тілдің алғашқы версиясында өңдеу программасын көрсету және "SEARCH ISINDEX" стандартының орнына сөйлемді сұрау мүмкіндігі туды.

```
<ISINDEX HREF=
```

HYPERLINK "http://polyn.net.kiae.su/cgi-bin/search" http://polyn.net.kiae.su/cgi-bin/search

```
PROMPT="Enter Keywords:">
```

Көрсетілген мысалда HREF атрибуты сұранысты өңдеу программасының адресін анықтайды, ал атрибут PROMPT – шақыру мазмұнын анықтайды. META META тегі HTML спецификациясында жоқ құжат тақырыбының конструкциясын анықтау үшін қажет. Ол үш атрибут қабылдайды: NAME, CONTENT, HTTP-EQUIV. Берілген текті қолдану қиындығы, осы тег арқылы еңгізілетін конструкцияны интерпретациялау үшін, сервис немесе интерфейс қолданушылар осы конструкцияны

кеңейтуі және қолдануы керек. Осындай жұмыс түрі үшін программа SGML конструкциясын түсіндіріп беру(интерпретировать) керек. Берілген тегті тәжірибеде қолданудың бірден-бір түрі тақырыпқа анықталған HTTP-EQUIV атрибуты арқылы протокол бойынша HTTP ақпаратын қосу.

```
< META HTTP-EQUIV="Keywords"
CONTENT="Plasma, Nuclear Physics">
```

Осындай қолдану кезінде почтаны жіберуге ыңғайлы болу үшін, HTTP-пакет тақырыбында мынадай жолдар қосылады: Keywords: Plasma, Nuclear Physics. Құжат денесінің тегі. Құжат денесінің тегі интерфейстік қолданушы программасында ақпаратты көрсетуді басқарады. Олар мәтінде тізілген контекстік гипермәтіндік сілтеме көмегімен деректер базасының гипермәтіндік құрылымын бейнелейді. Құжат денесі тұрады: -иерархиялық контейнер және заставка;

- тақырыптар (H1-P6 дейін);
- блоктар(параграфтар, тізімдер, формалар, кестелер, суреттер және т.б.);

-көлденеңінен белгілеп алу және адрестер;

- мәтіндер, стильдерді жүргізу облыстарына бөлінген(сызылған, белгіленген, курсивті);

-математикалық бейнелеу, графиктер және гипермәтіндік сілтеме; BODY Құжат денесінің тегін бейнелеу BODY тегі арқылы бастап жазылады. HEAD тегінен айырмашылығы, BODY тегі атрибут қабылдайды: ID-тег идентификаторы. Тегті атау үшін, сол сияқты гипермәтіндік сілтеме бойынша нүктелік ауысу түрінде қолданылады. Берілген атрибут құжат бетінің барлық тегінде бар. LANG-екісандық ISO-639 код түрінде құжат тілін анықтайды, нүктеден кейін міндетті емес ISO-3166 форматында қала коды алынады. Тіл стандартын құраушылардың ниеті бойынша берілген атрибут түсіндіру программасын тану және көптілді мәтінді бейнелеуді басқару керек. Сонымен қатар Arena да, арнайы HTML 3.0 иллюстрациялау үшін арналған, ол осы мүмкіндікті іске асырмайды. CLASS-"ADDITION.FIRST" типті иерархиялық құрылған ат. Мәтін тегін, анықталған бейнелеу стилімен байланыстыруға арналған. Қазіргі кезде қолданылмайды. Жоғарыда көрсетілген үш атрибут дене тегіне жалпы болып табылады және алдағы уақытта оларды қарастырмаймыз. Егер атрибут құжат мәтіннің дене тегінде кеңейтілмесе, онда ол арнайы көрсетіледі. BACKGROUND-құжат мәтінін бейнелейтін фонды анықтайды, мысалы, HTML-құжатында фон ретінде үлкен емес графикалық кескін "bgr.gif" қолданылады.

```
< BODY BACKGROUND ="bgr.gif">
```

Осы мысалда көрсетілгендей, берілген атрибуттың мәні ретінде URL формасында қысқартылған адрес қолданылады. Бұл локальдық файл адресі. Белгілеу тегін басқару. Тақырыптар. Тақырыптар құжаттың бастапқы бөлімін қарастырады. Стандартта тақырыптың алты деңгейі анықталған: H1-ден H6-ға дейін. Мәтін, <H1></H1> тегімен қоршалған, үлкен көлемде болады-бұл басты тақырып. Егер мәтін <H2></H2> тегімен қоршалса, онда ол бірнеше есе кішірек көрінеді; ішкі мәтін <H3></H3> одан да кішірек және ары қарай <H6></H6> дейін. Кейбір программалар тақырыптың көп санын қолдануға мүмкіндік береді, бірақ та үш деңгейден көбі сирек кездеседі, ал бестен көбі тіпті сирек кездеседі. ALIGN атрибуты ALIGN атрибуты мәтінді сол жақ шетке, оң жақ шетке, ортаға және ені бойынша теңестіруге арналған. Үнсіздік бойынша мәтін сол жақ шетке қарай теңестіріледі. Берілген атрибутты сол сияқты кестеге және графикке қолданамыз.

Келесі кесте ALIGN атрибутының мүмкін мәнін анықтайды. Мән Қолданылуын сипаттау Left Right Justify Center Сол жақ шетке теңестіру Оң жақ шетке теңестіру Сол және оң жақ шетке теңестіру Ені бойынша теңестіру

Justify мәні интерпретацияның барлық программаларында іске асырылмаған.

Айналдыру (оборачивание)

ALIGN атрибутының көмегімен сіз мәтінді графикалық объектінің айналасында <орап қою> болады. Ол үшін тегін графикалық объект тұратын жерге апарып қою керек және ALIGN=LEFT, ALIGN=RIGHT және ALIGN=CENTER атрибутын қосу қажет. Одан басқа HSPACE= және VSPACE= атрибутының (олар төменде сипатталады)

көмегімен бейнені мәтіннен айыратын тік және көлденең алаң ұзындығы сұралады. Сонымен қатар бейненің айналасына рамка құруға және кестені мәтінмен жиектеуге болады. Жалпы айтқанда, CLEAR барлық блоктық тегтер үшін жалпы болып табылады, бірақ та барлығына да іске асырылған жоқ.. CLEAR келесі мәндерді қабылдайды:

Мәні Тағайындалуы Left Беттің сол жақ шетінде орналасқан суретті өткізу Right Беттің оң жақ шетінде орналасқан суретті немесе кестені өткізу All Суреттің немесе кестенің алдында тұрған мәтінді ұзу және төменде жалғастыру

CLEAR атрибутында сандық мәнді де көрсетуге болады:

```
<P CLEAR="100 pix">  
<BR> тегі
```

Жолдың аударылымы мәтінді бейнелеудің стандарттар ретін бұзу үшін қолданылады. Интерпретацияның қарапайым режимінде қолданушының интерфейс программасы мәтінді автоматты түрде жолдарға бөлу арқылы жұмыс терезесінде көрсетеді. Бұл режимде мәтінде бар болған соңғы жол ескерілмейді. Кей-кезде үлкен айқындық үшін басуды жаңа жолдан бастау керек. Ол мақсатта
 тегі қолданылады.
 тегіндегі CLEAR атрибуты көрсетілген нүктеде объектіні мәтінмен сүйірлеу(обтекание) үшін қолданылады және содан кейін мәтінді объектінің ар жағында бос областа жалғастыру үшін. Объектінің ар жағында жалғасып жатқан мәтін CLEAR атрибутында LEFT, RIGHT және ALL мәндеріне сәйкес тегістеледі

```
<BR CLEAR =LEFT>
```

Мәтін сол жақтағы жақын арадағы бос алаңнан бастап жалғастырылады

```
<BR CLEAR =RIGHT>
```

Мәтін оң жақтағы жақын арадағы бос алаңнан бастап жалғастырылады

```
<BR CLEAR =ALL>
```

Мәтін сол және оң жақтағы алаң бос болғанда жалғастырылады

```
<NOBR> тегі
```

<NOBR> тегі (No Break, үзіліссіз) браузерге барлық мәтінді бір жолда үзіліссіз бейнелеуге нұсқау береді. Егер <NOBR> тегінде бекітілген мәтін экранға сыймаса, браузер құжат терезесінің төменгі бөлігіне горизонтальды айналдыру жолағын қосады. Егер сіз айқын жерден жолды алып тастағыңыз келсе, сол жерге
 тегін қойыңыз. Символдарды бейнелеуді басқару тегі Бұл тегтерді екі классқа бөлуге болады: бейнелеу формасын басқару тегі (font style) және ақпарат типін сипаттайтын тегтер (information type). Өртүрлі тегтер бейнелеу кезінде бірдей нәтиже көрсетеді. Ол басты жағдайда програманы түсіндірушінің жөндеуіне және қолданушының ұнатуына байланысты. Бейнелеу формасын басқару тегі Курсив, күшейту, асты сызылған, жоғарғы индекс, төменгі индекс, үлкен шрифт, кіші шрифт, қызыл, көк, өртүрлі комбинациялар – бетті көркемдеуге және функциональдауға мүмкіндік береді. Internet Explorer және Netscape Navigator Face= атрибутының көмегімен шрифты анықтауға болады. Енді бір бетке бірнеше шрифт түрлерін біріктіруге болады.

```
SIZE=атрибуты
```

 тегіндегі SIZE=атрибуты берілген областа мәтін мөлшерін сұрауға мүмкіндік береді. Егер сіз барлық беттің шрифт мөлшерін беру үшін <BACKGROUND SIZE =n> тегін қолданбасаңыз, онда үніздік бойынша 3 алынады.

```
<BIG> және <SMALL> тегтері  
<BIG></BIG> және <SMALL></SMALL> тегтері арасында орналасқан мәтін сәйкес келеді.  
COLOR=xx атрибуты
```

Егер сіз өзіңіздің бетіңізді одан әрі әдемілегіңіз келсе, онда тегіндегі COLOR= атрибутын қолдануыңызға болады және басты шектеу компьютердегі қолданушылардың түстер палитрасы болады. Өзірше тек қана танымал браузерлер шрифт түсін бейнелейді.

 тегін шрифт түсін ауыстыруға қолданады.

Бейнелеу формасын басқару тегтері Тегтер Мәні

<I>...</I>
Курсив (Italic)
...
Усиление (BOLD)
<TT>...</TT>

Телетайп

<U>...</U>

Асты сызылған

<S>...</S>

Сызылған текст

<BIG>...</BIG>

Үлкейтілген фонт

<SMALL>...</SMALL>

Кішірейтілген фонт

_{...}

Жолма-жол символдар

^{...}

Жол үсті символдар

Ақпарат типін сипаттайтын тегтер Тегтер Мәні

...

Типографиялық күшейту

<CITE>...</CITE>

Цитирование

```
<STRONG>...</STRONG>
```

Усиление

```
<CODE>...</CODE>
```

Мысал кодын көрсетеді (мысалы, программа коды)

```
<SAMP>...</SAMP>
```

Литералықтардың жүйелілігі

```
<KBD>...</KBD>
```

Клавиатура арқылы символдарды еңгізу мысалы

```
<VAR>...</VAR>
```

Айнымалылар

```
<DFN>...</DFN>
```

Анықтау

```
<Q>...</Q>
```

Жақшаға алынған мәтін

Осы тегтерді қолдану кезінде, олардың бейнеленуі программа-интерфейсін қолданушылардың жөнге келтіруіне байланысты екенін ұмытпаған жөн, олар гипермәтіндік программа құрушының жөнге келтіруімен сәйкес келмеуі де мүмкін. Интерфейс жөнге келтірулерін айырбастау стильдерін қолдану, казіргі кезде проблемалық болып келеді.

```
<DL> тегі
```

(Definition List:<DL>) тізіміндегі тегті қолдану қарапайым мәтіндік редакторда шегіністі қолдануды еске түсіреді. <DL> тегі мәтінді форматтау үшін құрылған, оңға қарай үлкен шегініс арқылы анықталады. <DL> тегі номерсіз шегінетін жеке абзацтар және маркерлер құруға арналған. Шегініс ол жақтан басталады. Егер сіздің бетіңізде бірнеше <DL> тегі бар болса, онда мәтін ақырын оңға қарай жылжи бастайды. Анықтау соңында жабылған </DL> тегін қойыңыз. Ұмытпаңыз, <DL> тегі тек қана абзацтың сол жақ шекарасын хылжытады.

```
<BLOCKQUOTE> тегі
```

Тег мәтіннің сол және оң жағынан орын қосады. Бұл қолайлы тег, ол мәтінді беттің ортасына орналастыруға мүмкіндік береді. <BLOCKQUOTE> тегін бірнеше рет қолданған кезде мәтін ортаға бірнеше рет қысыла береді. Microsoft Internet Explorer және Netscape Navigator <BODY> тегінде LEFTMARGIN=n және TOPMARGIN=n атрибуттарын қолдануға рұқсат етеді. LEFTMARGIN= атрибуты барлық бет үшін сол жақ орынды береді. TOPMARGIN= жоғарғы орнын анықтайды. n саны орынның енін пиксель бойынша көрсетеді. Мысалы, <BODY LEFTMARGIN="40"> тегі барлық бет үшін сол жақ енін 40 пиксельде құрады. n арқылы, 0 тең, сол жақ орны шегінеді. Табуляция Бұл құралдардың қажеттілігі бұрын пайда болды, тек жақында ғана құрастырушылар ұсынды. Табуляцияны бірнеше тәсілмен беруге болады. Ең қарапайым, <TAB IDENT=n> тегін жазу, онда n жаңа абзацтың алдындғы еп-бос орындар санын анықтайды. еп-бос орын – өлшеудің

типографиялық бірлігі, сіз қолданатын шрифтағы n әріпінің енімен бірдей. Соған сәйкес <TAB IDENT=4> тегі ені 4 ен-бос орын болатын табуляция символын алады. Егер сіз бірнеше орынға берілген мөлшерде табуляция символын қолданғыңыз келсе, онда оның мөлшерін беретін орынға <TAB> тегін ID= атрибутымен бірге қойыңыз, мысалы, соған сәйкес:

```
<TAB ID="tabone">
```

Енді беттің кез-келген жеріне <TAB TO="TABONE"> жазса жеткілікті, және табуляция символы TABONE тең болады. Сәйкесінше TABTWO, TABTHREE, TABFOUR ұқсас тәліммен құруға болады.

<TAB> тегін мәтінде және графикте орналастыруда қолдануға болады.

Тізімдер Тізімдер мәтін құрылымының негізгі құралдары болып табылады және белгілеулердің барлық тілдерінде қолданылады. HTML-де келесі тізім түрлерінен тұрады: номерленбейтін тізім (реттелмеген), номерленетін тізім (реттелген) және анықтамалдың тізімі. Номерленбейтін тізім тегі (Unordered Lists) және номерленетін тізім тегі (Ordered Lists) – бұл HTML негізі. HTML 3.0 номерленбейтін тізімде маркердің әртүрлі типін таңдауда тізім тегіне бірнеше атрибуттар қосады. Тізім ортасындағы маркер типін ауыстыру үшін мынадай (List Item) тегіне атрибут қосуға болады. Жаңа атрибут пайда болған кезде, тізімде қалған маркерлер де осындай түрді алады.

```
<UL> тегі
```

Номерленбейтін тізім. Номерленбейтін тізім мәтін типін құруға арналған. тізімнің бірінші элементі тізімнің екінші элементі тізімнің үшінші элементі Берілген тізім келесі түрде жазылады:

```
<UL>
<LI> тізімнің бірінші элементі
<LI> тізімнің екінші элементі
<LI> тізімнің үшінші элементі
</UL>
```

 және тегі – бұл номерленбейтін тізімнің басы және соңы, (List Item) тегі тізім элементінің тегін береді. Осы тегке қосымша, тізімді атауға арналған - LH (List Header) бар. Келесі түрдегі номерленбейтін тізімді бейнелейтін мысал келтірейік:

```
<HR> тегі
```

Горизонтальды сызып алу (horizontal rule) құжатты бөлікке бөлу үшін қолданылады. Бір ғана <HR> тегі арқылы бетке мүлдем өзгеше түр беруге болады. <HR> тегін тәжірибеде көріңіз және сонда сіз қолданып жүрген сызықтан өзгеше сызық аласыз.

```
<PRE> тегі
```

Форматтаусыз мәтінді бейнелеу.

```
<BLINK> тегі
<BLINK> тегі оған қосылған мәтіннің жыпылықтауын туғызады.
```

Гипермәтіндік сілтеме Жоғарыда қарастырылған мәтінді бейнелеу құралдары, құжаттың басты тегі - гипермәтіндік сілтемеге – қосымша, сөзсіз маңызды болып келеді. Гипермәтіндік сілтеме жазу үшін, “зәкір (якорь)” (anchor) деп аталатын <A...>..... контейнері қолданылады. Зәкір бірнеше атрибут қабылдайды, ең негізгісі HREF (Hyper Text Reference) болып табылады. Қарапайым сілтемені мына түрде жазуға болады:

```
<A HREF="http://polyn.net.kiae.su/index.shtml">  
Деректер базасының индексі"Жусан"</A>
```

HREF атрибутының мәні болып, "polyn.net.kiae.su" машинасында "index.shtml" құжат адресі табылады, оған рұқсат HTTP протоколы арқылы беріледі. Осы адресің жазу формасы универсалды локааторлық ресурс (Universe Resource Locator) деп аталады және WWW технологиясының құрама бөлімі болып табылады. А контейнерінің мазмұны, тег басы және тег соңы аралығында қосылады, мәтінде контекстік гипермәтіндік сілтемеге арналған түспен бөлінеді. Web-серверді құрудың ерекшелігі, онда көрсетілген ақпаратты жеке бөліктерге бөлуге болады. Жеке бөліктер арасындағы байланыс ұйымдары гипермәтіндік сілтеме арқылы анықталады.

Өзін тексеруге арналған сұрақтар:

1. html-құжатының құрылымы. Беттің құрылымдық элементтері.Элементтер типі.
2. WWW-сервисінің жұмыс тәртібі. Сервер және клиент арасында мәлімет алмасу. Формалар.

Ұсынылатын әдебиеттер:

1. Веб-дизайн/Дмитрий Кирсанов - СПб: Символ-Плюс, 2004-376 стр.
2. HTML 4.0 в подлиннике/HYPERLINK ".Матросов, HYPERLINK ".Серегеев, HYPERLINK "http://www.books.ru/shop/authors/235" Чаунин – СПб: BHV, 2000 – 672 стр.

Дәріс 10. Webтегі графика

Дәріс мазмұны: графика өлшемдерін қалай беруге болады; активті бейнелер; активті бейнені қалай жасауға болады; сервердегі активті бейнелер; клиенттегі активті бейнелер; серверде де, клиентте де жұмыс істейтін активті бейнелер.

Графика Webтегі ең кең таралған графикалық форматтар болып GIF және JPEG форматтары болып табылады. Web – бетке бейне қою үшін ең алдымен оны салып алу қажет немесе дайын бейнені алуға болады. Сурет салудың кез-келген программасында қарапайым бейнені салу және оны қажет етілген форматқа сақтау қиынға түспейді. Егер программа бұл форматты қабылдаса, онда файлды өңдеу қажет. Бір графикалық форматтың басқа форматқа өңделуіне арналған бірнеше программалар бар. Суреттерді әртүрлі программалық пакеттерден немесе Internet Web – беттерінің бірінен алуға етуге болады. Егер браузер Web – бетті бейнемен бірге шығарса, сәйкес графикалық файл компьютер жадысында уақытша сақталады. Браузердің көпшілігінде файлды үнемі локальды дискіде сақтауға рұқсат беретін командалары бар.Графикалық файлды алудың бірнеше басқа да варианттары бар: Бейнелер өзінің ақпараттылығымен пайдалы болуы мүмкін және Web – бетке көз тартар түр береді. Бейнелерді қолдануда кең таралған жағдайларды қарастырайық: іскерлік беттегі компанияның логотипі; жарнамалық хабарландыруға арналған графика; әртүрлі суреттер; диаграммалар және графиктер; шығармашылықтар шрифтер; бет авторының қолы; графикалық жолдың горизонтальды бөлінді сызық ретінде қолданылуы; әдемі маркерленген тізімдерді құруға арналған графикалық маркерлердің қолданылуы. Енді Web – бетке бейнені қалай қою керек екендігін қарастырайық. HTML тегі болып браузер бейнені шығаруды талап ететін келесі ортақ форматы табылады:

```
<IMG SRC ="picture.gif">
```

Тұйықтаушы тег талап етілмейді.Мұндағы SRC – (SouRCe)бастауды білдіреді, ал файлдың аты шығарылған графикалық файлдың атын білдіреді. Web – беттегі бейнелер қарапайым мәтін сияқты гипермәтінді сілтемелер ретінде қолданылуы мүмкін.. Бетті оқушы бейненің үстінен шертеді де, басқа бетке немесе бейнеге ауысады. Гипермәтінді белгі сияқты бейнені белгілеуде, мәтінде де қолданылатын <A> тегі қолданылады, бірақ <A> және арасына бейнесінің тегі қойылады.

```
<A HREF="файлдың немесе бейненің адресі">  
<IMG SRC="picture.gif"> </A>
```

Бұндай бейненің гипермәтінді сілтеме ретінде қолданылуы қосымша рамка ретінде. Шектелген тікбұрыштар және ALT= атрибуттары Графика бетінің жүктелу уақытын азайту үшін HTML – кодында бейненің өлшемін анықтау пайдалы. Егер ол беттің жүктелгенінен бұрын белгілі болса, онда браузер ең басында суретке рамканы апарады, содан кейін бетке мәтінді

жүктей алады. Егер графикалық бейненің өлшемі анықталған болса, онда оған браузер шектелген тікбұрыш түріндегі орын резервтейді. Графика жүктеліп жатқан кезде бетті оқушы мәтінді оқи беруіне болады. Бұндай жұмыстың методикасы - өз оқушыларына деген жұмсақ мінезді қарым – қатынасын білдіреді.

Графика өлшемдерін қалай беруге болады?

Егер сізде қандай да бір графикамен жұмыс істеу программасы бар болса, онда бейненің өлшемін анықтау қиын емес. Ең алдымен файлды графикалық редакторде ашыңыз және суреттің өлшемін пиксельмен анықтаңыз. Бейне тегінде өз суретіңіздің ені мен ұзындығын мына түрде беріңіз:

```
<IMG SRC= "picture.gif " WIDTH=413 HEIGHT=356>
```

Егер сіз графикалық бейненің өлшемін көрсетпесеңіз, онда браузер әрбір суретті толығымен жүктейді және осыдан кейін ғана мәтінді жүктеуге көшуге болады. Ал бұл көп уақытты алады. Егер өлшемдер беріліп тұрса, онда көптеген жүргізуші браузерлер қажетті өлшемдегі шектелген тікбұрышты сыза алады және оны бейнемен төгуге етуді бастайды.

ALT= атрибуты

 тегінің ALT= атрибуты қолданушыларға, тек мәтінді көретін браузерлер (егер браузерде графика режимі өшіп жатса), графиканың бетте орналасуы немесе оның мақсаттары туралы ақпарат алуына мүмкіндік береді. Microsoft Internet Explorer бейне шектелген тікбұрыштағы ALT= атрибутынан мәтінді көрсетеді. Егер Auto Load Image режимі қосылып тұрған жағдайда ғана Netscape Navigator бұл мәтінді шығарады.

 тегі ALT= атрибутымен бірге келесі түрде болады:

Активті бейнелер Активті бейнелер (image maps) немесе бейнелер, тышқан батырмасын сезушілер сіздерге туынды формасындағы графикалық мәзірді өз тармақтарыңызда құруға мүмкінді береді. Мұндай мәзірді қолдану арқылы оқушылар барлық тұйықтарда және сіздің Web – тармақ даңғылыңызда серуендей алады. Активті бейнелер - бұл басқа беттердің URL – на немесе тармақтарына сілтемеленетін активті облысты деп аталатын жай сурет. Келесі түрде жұмыс істейтін бейне бар: қолданушы активті бейнелердің IMG тегіндегі ISMAP атрибуты көмегімен анықталған суретті тышқанмен шерткен кезде, шерту координаттары Web- серверге беріледі. Сервер картада жіберілген координатты құрайтын активті облысты іздейді. Егер мұндай облыс бар болса, онда берілгендер URL картасында активтенеді және қолданушының браузері жаңа бетке көшеді. Активті бейненің көмегімен сіз бірден екі есепті шеше аласыз: өзіңіздің Web – тармағыңыздың мәзірін құра аласыз және онда графика орналастыра аласыз. Активті бейнені қайда қоюға болады: серверде немесе клиентте? Активті бейненің 2 типі бар: серверде және клиентте. Бірінші типтегі бейнені сервер URL активті облыстың сәйкес берілгендерін табу және браузерге қажетті бетті браузерде қолданады. Клиенттік машинада жұмыс істейтін активті бейнелер HTML бетіндегі активті облыстар жайлы ақпарат береді. Сондықтан браузер қандай облыс активті екенін өзі түсіндіреді және серверден қажетті бетті сұрайды.

Активті бейнені қалай жасауға болады? Активті бейненің құрылу процесі 2 сатыдан тұрады. Ең алдымен сіз активті етіп жасайтын суреттің облысын анықтауыңыз қажет, содан кейін оны сілтемелермен басқа URL – апаруыңыз керек. Мұның бәрін активті облыстың координат бұрыштарын анықтау арқылы жасауға болады, бірақ қандай да бір программамен, мысалы, MapEdit – ті қолдану оңайға түседі. Картаны анықтау оңай. MapEdit – та сіз құрғалы отырған активті облыстағы бейнені құрайтын файл ашу қажет, одан кейін тышқанды апарып шерту арқылы облыс шекарасын береміз. Программа файлды активті облыстың белгіленген шекарасын автоматты түрде генерациялайды. Содан кейін осы облыспен URL-ді жазу қажет. Бейненің кез-келген орнында тікбұрыш, көпбұрыш немесе шеңбер формасында активті облыстар салуға болады және әрқайсысына URL-ді анықтау керек. Активті облыстар қиылысуы мүмкін, бірақ бұл жағдайда мәселелер туындайды. Оқушы дұрыс сілтемені таңдап отырғанына сенімді болу үшін облыстар арасына біраз орын қалдыру қажет; Активті облыстың шекаралары тікбұрыш, көпбұрыш немесе шеңбердің центрі мен радиусының координаттар бұрышымен беріледі. Бұл параметрлер картаға (Map-файл) жазылып отырады. Активті бейнені құрып болғаннан кейін, сіз алынған файлды NCSA немесе CERN форматында сақтай аласыз. Егер ол серверде немесе CSIM форматында жұмыс істесе және егер ол клиенттік машинада функционирленсе. Ал қалған барлық жұмысты MapEdit программасы орындайды. Ол картаны серверде құрады немесе картаны клиент жақтағы сіз көрсеткен HTML файлында кірістіреді. Егер сіз активті

бейнені клиентте жасайтын болсаңыз, онда MapEdit <Map> тегтері үшін ғана берілгендерді қояды. USEMAP атрибутымен бірге бейне тегін өзіңіз беруіңізге тура келеді және оны </MAP> тегінен кейін бірақ аласыз. USEMAP атрибутында карта атының алдынан # символын қоюды ұмытпаңыз:

```
<IMG SRC=" mymap.gif " USEMAP="glava1~2.shtml#sitemap">
```

Сервердегі активті бейнелер. Бұрын қолданыста болған активті бейненің құрылуында (html 2.0 үшін) IMG тегіндегі ISMAP атрибутының қолданылуы талап етіледі. IMG тегі бейнеге қатысты, оны бастапқы және соңғы тегі арасына файл – картаның сілтемелеріне орналастыру қажет. Сізге HTML – файлына мынадай жолдарды енгізу керек:

```
<A HREF="path/somemap.map">  
<IMG SRC =" path/somemap.map " ISMAP> </A>
```

ISMAP атрибуты браузерге берілгендер бейнесі активті екенін көрсетеді. Егер оның қандай да бір облысында тышқан шертпесі пайда болса, онда ISMAP атрибуты көмегімен серверге координат шертпесін құрайтын хабарлама жіберіледі. Клиенттегі активті бейнелер. Алдыңғы бөлімде жазылған бейнелерге қарағанда, клиенттегі активті бейнелер сервердің программалық жабдықтарына тәуелсіз жұмыс істейді және сіз өз файлдарыңызда басқа серверге ауыстырсаңыз да функционируін тоқтатпайды. Бұндай бейнелерге 2 нәрсе қажет: HTML 3.0 қолдайтын браузер және HTML файлында жазылған карта туралы ақпарат. Серверде де, клиентте де жұмыс істейтін активті бейнелер. Сіздің серверде де, клиент – машинасында да жұмыс істейтін бейнемен жұмыс істегіңіз келетін шығар. Серверде де, клиентте де активті бейнелер құрғаннан кейін оларды бір HTML файлында біріктіру қиын емес. Ол үшін HTML документіне сервердегі активті бейне үшін жасауға арналған сол жазуды апару қажет. IMG тегіне USEMAP= атрибутын қосуды ұмытпаңыз. USEMAP= атрибуты ISMAP тегіне қарағанда жоғары приоритетті болып табылады. Егер браузер клиентте жұмыс істейтін активті бейнені қолдаса, онда бұл атрибутты ол мойындайды. Бұндай бейнелерді білмейтін браузер USEMAP= атрибутына әсет етеді. Өзін тексеруге арналған сұрақтар:

1. XML және HTML. XML Синтаксисі. XML-дің HTML-ден айырмашылығы. DTD.
2. XML- құжатының визуалды тәсілдері.

Ұсынылатын әдебиеттер:

1. Веб-дизайн/Дмитрий Кирсанов - СПб: Символ-Плюс, 2004-376 стр.
2. HTML 4.0 в подлиннике/HYPERLINK ".Матросов, HYPERLINK ".Сергеев, HYPERLINK "http://www.books.ru/shop/authors/235" Чаунин – СПб: BHV, 2000 – 672 стр.

Дәріс 11. HTML-да кестенің жазылу тәсілдері. Беттің дизайнында кестелерді қолдану

Дәріс мазмұны:

```
<TABLE>, <TR>, <TD>, <TH>, <CAPTION>.
```

NOWRAP, COLSPAN=, ROWSPAN=, WIDTH=, UNIT=, COLSPEC=, DP=, CELLPADDING=, ALIGN= және VALIGN=, BORDER=, CELSPACING= атрибуттары. Көптүсті кестелер құру. Netscape Navigator-дағы түрлі-түсті шекаралар.

HTML-да кестенің жазылу тәсілдері

```
<TABLE> тегі.
```

Кестенің жазылуы үшін <TABLE> тегі қолданылады. <TABLE> тегі басқалар сияқты жолды кестеге дейін және кестеден кейін автоматты түрде аударады.

```
<TR> тегі.
```

<TR> тегі (Table – Row сөзінен қысқартылған) – кестенің жолын құрады. Егер кесте екі жиын тегінен <TR> </TR> құралса, онда ол екі жолдан да тұрады. Сіз бір жолға орналастыратын бүкіл мәтін, басқа тег және атрибуттар <TR> </TR> тегтерінің арасына орналасуы қажет.

<TD> тегі.

Кесте жолының ішінде әдетте берілгендер ұяшығы орналасады. Мәтін немесе бейне құрайтын әрбір ұяшық <TD> </TD> тегтерімен қоршалуы керек. Жолдағы <TD> </TD> тегтерінің саны ұяшық санын анықтайды. <TD> </TD> тегтерінің бес сыңарлы жолы бес ұяшықтан тұрады.

<TH> тегі.

Кестенің баған және жолының тақырыпшасын беру кезінде <TH> </TH> тақырып тегі қолданылады. Бұл <TD> </TD> тегімен аналогты болып келеді. <TH> </TH> тегтерінің арасындағы мәтін автоматты түрде қою шрифтпен жазылады және үнсіздік бойынша ұяшықтың ортасында орналасады. Орталықтандыруды алып тастауға болады және мәтінді сол жақ немесе оң жақ жиек бойынша тегістеуге болады. Егер <TD> </TD> ні тегімен және <ALIGN=CENTER> атрибутымен бірге қолданса, онда мәтін де тақырып сияқты көрінеді. Бірақта, барлық браузерлердің кестеде қою шрифті қабылдамайтынын ескерген жөн, сондықтан кестенің тақырыбын <TH> көмегімен берген дұрыс.

<CAPTION> тегі.

<CAPTION> кестенің тақырыбын құруға рұқсат береді. Үнсіздік бойынша тақырып орталықтанады және (<CAPTION ALIGN=TOP>) кестесінің үстінде немесе (<CAPTION ALIGN= BOTTOM>) кестенің астында орналасады. Тақырып кез-келген мәтіннен және бейнеден тұруы мүмкін. Кей кезде <CAPTION> тегі сурет астындағы жазуға қолданылады. NOWRAP атрибуты . Кестедегі ұяшықтың бір жолына сыймайтын кез-келген мәтін келесі жолға көшеді. Бірақта, NOWRAP атрибутын <TH> және <TD> тегтерімен бірге қолданғанда ұяшықтың ұзындығы мәтін бір жолға сыятындай етіп кеңейтіледі.

COLSPAN= атрибуты.

<TH> және <TD> тегтері COLSPAN= атрибутының көмегімен модифицирленеді (Column Span, бағандардың бірігуі). Егер сіз қандай да бір ұяшықты кеңейткіңіз келсе, онда осы COLSPAN= атрибутын қолдануға болады.

ROWSPAN= атрибуты.

<TH> және <TD> тегтерінде қолданылатын ROWSPAN= атрибуты COLSPAN= атрибутына аналогты болып келеді және ол жолдардың санын береді. Егер сіз ROWSPAN= атрибутында санды көрсетсеңіз, онда жолдың сәйкес саны кеңейтілген ұяшықтың үстінде болуы қажет. Оны кестенің төменгі жағына орналастырмау керек.

WIDTH= атрибуты.

WIDTH= атрибуты екі жағдайда қолданылады. Оны барлық кестенің енін беруде <TABLE> тегіне қоюға болады және ұяшық тобының немесе ұяшық енін беруде <TR> немесе <TH> тегтерінде қолдануға болады. Енін пиксельмен немесе пайызбен көрсетуге болады. Мысалы, егер сіз <TABLE> тегінде WIDTH=250 деп берсеңіз, онда сіз монитордағы бет өлшеміне тәуелсіз ені 250 пиксель кестені аласыз. <TABLE> тегінде WIDTH=50% беру кезінде, кесте экрандағы кез келген өлшемдегі бейнеде беттің жарты енін алып тұрады.

UNIT=атрибуты.

<TABLE> тегінің UNIT=атрибуты барлық кесте мен оның жеке бағандарының өлшемдерін көрсетуде қолданылатын өлшем бірлікті анықтайды. UNIT=атрибуты үш мәнді қабылдауы мүмкін: UNIT=EN - үнсіздік бойынша қосылатын мән және ен - бос орынға тең өлшем бірлікті береді. Ен - бос орын - <n> әріпінің еніне тең типографты өлшем бірлік. Бос орынның нақты өлшемі таңдалған шрифтке тәуелді: ұсақ шрифтке қарағанда ірі шрифтке ен - бос орын көбірек беріледі. Әдетте ен- бос орын шрифт өлшемінің жартысына тең. Мысалы, 12 - пунктті шрифт үшін ен - бос орын 4 - пунктті алады. UNIT=RELATIVE барлық кесте еніндегі баға енін процентпен беру үшін қолданылады. Бұл тәсілді мүмкіндігінше проценттегі енді көрсетудің орнына қолдануға болады. Қолданылатын бірлікті беру кезінде енгізілген сандар проценттердегі бағандар ені болып қабылданады. UNIT= PIXELS – бұл мән экрандағы бағанның енін нақты білуде қолданылады. Бұл жағдайда оны пиксельмен беру оңай. Мысалы, <TABLE UNIT= PIXELS WIDTH=340> кестенің енін 340 пиксельмен формалайды.

COLSPEC=атрибуты.

UNIT=атрибутымен бірге қолданылатын COLSPEC=атрибуты кестенің әрбір бағаны қанша орын алатынын және онда берілгендер қалай тегістелетінін анықтайды. Тек <TABLE> тегінде ғана қолданылады. COLSPEC= барлық бағанды санайды және әрқайсысы үшін тегістеу мен өлшем береді. Баған үшін (немесе ұяшық үшін) тегістеудің 5 тәсілі бар: L – сол жақ жиек бойынша, C – центр бойынша, R – оң жақ жиек бойынша, J – оң және сол жақ жиек бойынша және D – ондық үтір бойынша тегістеу. Егер сізде 5 баған болса, онда оның енімен әрқайсысын тегістеуді келесідей анықтауыңызға болады.

<TABLE UNIT= PIXELS COLSPEC = "L10 C15 J25D30">
DP=атрибуты.

DP=атрибуты (Decimal Point, ондық белгі) ондық бөлгіштің толық бөлімін бөлуде қолданылатын символды анықтайды. DP= " "(үнсіздік бойынша) ондық символ ретінде нүктені көрсетеді. DP= " " үтірді береді. Бос ұяшықтар. Егер ұяшық берілгендерді қамтымаса, онда оған шекарада қажет етілмейді. Егер сіз ұяшықтың берілгендері болмай, шекарасы болғанын қаласаңыз сіз оған көрсетілім кезінде көрінбейтін бір нәрсе қоюыңыз қажет.
 бос жолын қолдануға болады. Алынған ұяшыққа ешқандай берілгендерді енгізбей-ақ, олардың ендерін пиксельмен анықтап бос бағандарды да беруге болады. Бұл тәсіл бетке мәтін мен графиканы қою кезінде пайдалы болуы мүмкін.

CELLPADDING=атрибуты.

CELLPADDING=атрибуты - ұяшық берілгендерімен оның шекарасы арасындағы бос кеңістіктің енін анықтайды, яғни ұяшық ішінен алаң береді.

ALIGN= және VALIGN= атрибуттары.

<TR>, <TH> және <TD> тегтерін ALIGN= және VALIGN= атрибуттарының көмегімен модифицирлеуге болады. ALIGN=атрибуты мәтін және графиканың горизонталь бойынша тегістелуін анықтайды, яғни оң және сол жақ жиек бойынша, центрі бойынша тегістеу. Горизонталь тегістеу бірнеше тәсілмен берілуі мүмкін: ALIGN=BLEEDLEFT – ұяшықтың берілгендерін сол жаққа тығыздап қысады. ALIGN= LEFT - CELLPADDING=атрибутымен берілген ұяшықтың берілгендерін жылжыту есебімен сол жаққа тегістейді. ALIGN=CENTER – ұяшықтың берілгендерін центр бойынша орналастырады. ALIGN=RIGHT - CELLPADDING=атрибутымен берілген ұяшықтың берілгендерін жылжыту есебімен оң жаққа тегістейді. VALIGN= атрибуты – мәтін және графиканы ұяшық ішінен вертикаль бойынша тегістейді. Вертикаль тегістеу келесі тәсілдермен берілуі мүмкін: VALIGN=TOP – ұяшықтың берілгендерін оның жоғарғы шекарасында тегістейді. VALIGN=MIDDLE – ұяшықтың берілгендерін вертикаль бойынша орталықтандырады. VALIGN=BOTTOM - ұяшықтың берілгендерін оның төменгі шекарасында тегістейді.

BORDER=атрибуты.

<TABLE> тегінде үнемі кестенің өзі және кестенің ұяшықтарын қоршап тұратын рамкалардың, яғни сызықтардың қалай көрінетіндігін анықтайды. Егер сіз рамкасын бермесеңіз, онда сызығы жоқ кесте аласыз.

CELLSPACING= атрибуты.

CELLSPACING= атрибуты – ұяшықтар арасындағы аралық енді пиксельмен анықтайды. Егер бұл атрибут берілмесе, онда үнсіздік бойынша 2 пиксельге тең көлем беріледі. CELLSPACING= атрибутын мәтін мен графиканы өзіңізге қажетті жерге орналастыру үшін қолданылады. Егер сіз бос орын қалдырғыңыз келсе, онда ұяшыққа бос орынды жазуға болады.

Беттің дизайнында кестелерді қолдану. Кестелердің жақсы бір қасиеті – ол құрып отырған кестенің шекараларын көрінбейтін етіп жасауға болады. Ол <TABLE> тегінің көмегімен бетте текст пен графиканы кірістіре отырып әдемі орналастыруға болады. Осы күнге дейін <TABLE> тегі HTML-де өңдеудің жалғыз қуатты құралы болып отыр. Web-беттердің дизайнерлері қазір практикалық түрде баспа беттерін құрушылар сияқты <бос кеңістікті > қолдануда. Кең көлемді кестелер басқаларға қарағанда текстердің Web –беттерде иерархиялық орналасудан алшақтатады. Егер браузер кестені қабылдаса, онда кестенің көмегімен алынған эффектілерді дұрыс көрсететін болады. Түрлі-түсті кестелер құру. Жетекші браузерлер бірнеше түстерді бейнелейді. Кестені бояудың бірнеше әдістері бар. Ол әдетте қолданылатын браузерге байланысты. Netscape Navigator –тағы түрлі-түсті шекаралар. Сіз кестені жақтаумен әсемдеп қана қаймай, оған фон мен текстің түсінен әлдеқайда жақсы түс беруіңізге болады. Қарапайым GIF (немесе өзіңіз фон ретінде қойғыңыз келген кез-келген GIF,) құрыңыз және оны <BODY> –тегінде фон ретінде анықтаңыз. Сосын бет фонының түсін беріңіз. Нәтижесінде сіздің <BODY> –тегіңіз мына түрде болады:

```
<BODY BACKGROUND="coolbg.gif" BGCOLOR=" # FF0000">
```

Сіз екі қабаттан тұратын фон құрдыңыз- GIF және берілген түс. Нәтижесінде фондық түс кестенің барлық шекараларында және көлденең (<HR>) жолақтарда көрінетін болады. Сіздің фондық GIF –іңіз сұр түсті болуына тәуелсіз, кестенің түрлі-түсті сызықтары мен шекаралары ерекшеленіп тұрады. Егер фондық GIF күрделі емес болып құрылса, бетті жүктеу уақыты кішкене өседі. Төмендегі мысалда –дегі түстер әсіресе кестелерде кең көлемде қолдануға мүмкіндік беретінін көрсетеді.

Өзін тексеруге арналған сұрақтар:

1. Форма мен мазмұнның ортогональдық принципі
2. HTML –дің ерекшеліктері
3. XML –дің ерекшеліктері
4. Статикалық және динамикалық құжаттарды өңдеудегі ерекшеліктер.
5. Белгі тілдердің жалпы принциптері.

Ұсынылатын әдебиеттер:

1. HTML және CSS мысалдары/Мержевич – СПб: BHV, 2005-448 б.
2. HTML және CSS. Бірігіп қолдану/ДеБольш – СПб: HT-Пресс, 2006-504 б.

Дәріс 12. Фреймдер мен формаларлармен жұмыс

Дәріс мазмұны: Фрейм дегеніміз не? Фреймдер не үшін қолданылады? Фреймдер қалай жұмыс істейді? Фреймдік құрылым жасау Берілгендері бар фреймді дайындау Сіздің формаңыз жақсы көріну үшін не істеу керек? Формаларды жұмыс істеуге мәжбүр ету.

Фреймдер Фрейм дегеніміз не? Фрейм дегеніміз – сурет айналасындағы жақтау, терезе немесе бет деген сөздермен мағыналас. <FRAME>-деген тегті енгізеді, HTML-беттің дизайнері экран браузерін бөліктерге бөледі. Нәтижесінде бетті қарап отырған адам оның бөліктерін басқа бөліктеріне тәуелсіз қарай береді. Фреймдерді қабылдайтын браузер әр түрлі беттерді түрлі секцияларға , фреймдарға жүктеп тастайды. Мысалы, сіз бетті өзіңіздің фирмалық белгіңізді экранның жоғарғы бөлігіне орналастырып, қалған бөліктерін қолданушы өзі ары қарай қарапайым тәсілмен ашып отыратындай етіп құруыңызға болады. Жанынан навигациялық батырмалар қоюға болады. Оның көмегімен оқырман осы батырманы тышқанмен шерткенде орын ауыстырылмай, эканның белгілі бір бөлігі ғана қозғалады.

Фреймдер не үшін қолданылады? Навигация құралдарының немесе фирмалық белгінің фиксациясы – фреймдерді қолданудың көрнекі тәсілі болғаны оларды толық қолдану деген сөз емес. Фреймдерді қолдануда дизайнерлердің ойына ең бірінші саналып өткен шешімдер келеді. Бірақ браузер экранына фрейм қойылған сайын сіз берілгендерді орналастыруға арналған территорияны қысқартып отырасыз. Сіз фреймдерді неғұрлым көбірек кіргізсеңіз, пайдалы

кеңістік соғұрлым азаяды және сіз осы әрекетіңізбен фреймдері бар Web-беттердегі навигациялармен таныс емес өз оқырмандарыңызды шатастырасыз. Мысалы, фреймде ең қажетті беттердің алғашқы бірнеше деңгейлерінде компанияның логотипін сақтап, ал негізгі берілгендерді фреймдерсіз көрсеткен дұрыс.

Фреймдер қалай жұмыс істейді? Фреймдер күрделі болғанымен оларды түсіну оңай, егер кесте ұяшықтарына аналогия жүргізсе. Фреймдердің экранда орналасуы ұяшықтардың кестеде орналасуы сияқты: тегтер мен атрибуттар кестедегі сияқты жұмыс атқарады. Бірақ, беттегі бірлік фреймдер мен кесте ұяшықтары арасындағы аналогияның маңыздылығымен қоса оның айырмашылығын ажырата білген жөн. Ұяшық берілгендері кестесі бар HTML-бетінің кодында берілген. Кестенің берілгендерін құрайтын мәтін немесе графика, HTML-бетіндегі кестені бейнелейтін тег немесе атрибут сияқты енгізіледі. Керісінше, фреймдері бар экран HTML-бетте (frameset) деп аталатын сөзбен бейнеленеді. Фреймдердегі берілгендер кез - келген жерде – басқа каталогта, локальді серверде немесе желінің жойылған түйіндерінің бірінде орналасқан бөлек HTML-бет. Фреймдік құрылым фреймі бар экранды ұйымдастыру тәсілін және әрбір фреймнің бастапқы берілгендері қайда орналасқанын ғана анықтайды. Барлық фреймдер үшін олардың берілгендерінің орнын бейнелейтін URL болады. Ереже бойынша фреймдік құрылымы бар бетте олардың берілгендері болмайды. Мұндай бет үлкен емес – ол экранны ңтек кадрлық құрылымын ғана анықтайды. Құжат фреймдерге жүктелген уақытта сіз осы құжаттағы сілтемеге тышқанмен шертіп, басқа кадрлардағы фреймдік құрылымда жасалған бір - бірімен байланысқан құжаттарды шақыруыңызға болады.

Фреймдері бар қарапайым беттерді құру. фреймі бар HTML беттерді құру және негізгі тегтер мен атрибуттардың қажеттілігін жақсылап түсіну үшін фреймдері бар қарапайым жолдар жұбын құрайық. Екі фреймнен тұратын бет құрайық. Сол жақтан басылымдар мазмұнының тақырыбын бейнелейтін, ал оң жақтан басылымдардың өздерін бейнелейтін фрейм құрайық. Қолданушы мазмұн орналасқан экран бөлігіндегі сілемені тышқанмен шерткенде, сол басылым оң жақта пайда болатындай етіп жасайық. Бұл фреймдерді қолдануда кеңінен таралған әдіс.

Фреймдік құрылым жасау. Алдымен біз фреймдердің қайда орналасуын және олардың қандай мөлшерде болатын беттің жалпы түрін ойластыруымыз қажет. Содан кейін олардың мазмұны жайлы қарастырамыз. Төменде <FRAMESET> тегінің көмегімен құрылған қарапайым фреймдік құрылымның коды көрсетілген. Мынаған назар аударыңыз: фреймдік құрылымы бар бетте <BODY> тегі жоқ.

Берілгендері бар фреймді дайындау. Енді берілгендері бар фрейм жасайық. Сол жақ фреймге оң жақ фреймдегі екі беттің арасында ауысып тышқанмен шертетін menu.html бетін құрайық. menu.html файлы бұл – мазмұн ретінде құрылған қарапайым HTML-бет. Негізінде біз мазмұны бар дайын бетті алып оны қолдана беруімізге болады. Бұл фреймнің жіңішке жіне биік екенін ұмытпаңыз, сондықтан оған жүктелетін бет соған сәйкес жобалануы қажет. Енді сілтемені тышқанмен шерткенде басқа беттердің қайда пайда болатынын анықтауымыз қажет. Біз олардың оң жақта пайда болуын қалағандықтан сілтеме тегіне TARGET= (TARGET="main") атрибутын енгізейік. Бұл қолданушы сілтемені басқан кезде шақырылып отырған бет main фреймінде пайда болады дегенді білдіреді. Біз main фрейміндегі барлық беттерді бейнелегендіктен, мазмұндағы барлық сілтемелердің тегіне TARGET="main" атрибутын қосамыз, сонда сол жақтағы фреймде қай жерде тышқанмен шертсек, сол жерде бет пайда болады, ал ол бізге керек емес, бірақ басқа жағдайда олай жасаған керек те болып қалады. Мысалы, сіз жай ғана келесі сілтемелерді шығарып беретін < Мазмұнның басқа пункттері > сілтемесін қосуыңызға болады. Оқырмандар сілтемелердің көптігін байқау үшін мазмұнды ұзынырақ етіп жасаған дұрыс. main фреймін дайындау. Оң жақтағы main фреймінде HTML-беттер болады. Сіз кішірек болып көрінетіндей терезеде жобалағаныңыз дұрыс, себебі экранның белгілі бір бөлігінде мазмұнның сол жақ кадрлары орналасады. Но больше эти страницы ничем не примечательны.

<NOFRAMES> тегін пайдалану.

Сіздің көптеген қолданушыларыңызда әлі фреймдермен жұмыс істей алмайтын браузерлер бар. Сол себепті сіздің негізгі беттеріңізді фреймсіз ұсынған жөн. Егер сіздің фреймдік құрылымды бетіңізде ескірген браузерлі оқырман болса, онда осы бетте орналасқан < NOFRAMES >және </ NOFRAMES >тегтерінің арасындағылардың барлығы өте жақсы көрініп тұратын болады – браузер жай ғана фреймдерді елемейтін болады. Сондықтан міндетті түрде <BODY></BODY> тегтерін пайдалану керек. Әйтпесе, сіз экранды қайта фреймдерсіз құруыңызға тура келеді. Фреймдік бетке No Frames (Фреймсіз) батырмасын кірістіруге болады. Оның бағыты белгілі. Мұндай әдіс ойға қонымды және тез жүзеге асады.

Формалар. Интерактивті формалардың көмегімен оқырмандар Web-бетті құрушылармен оңай хабарласа алады. Форма мен <mailto:> тегінің қарапайымдылығының арқасында Web-түйіндер мен олардың оқырмандарының арасында екіжақты диалог мүмкін болды. Бұл Web- навигаторларға коммерциялық мүмкіндіктер мен қызықты болашаққа жол ашады. Өкінішке орай, көптеген ескірген браузерлер <mailto:> тегін қабылдамайды. Сіздің формаңыз жақсы көріну үшін неістеу керек? Барлық қиындықтарды жеңудің бір тәсілі ол – енгізу алаңы мен мәзір жолын беттің сол жағына, ал

бағыттаушы мәтінді оң жағына орналастыру болып табылады. Енгізу аумағының енін SIZE= атрибутының көмегімен сіз жолдарды сол жақтан туралайсыз, ал оң жақтағы мәтін көптеген құжаттардағы сияқты тураланбаған болады. Мұндай форма әлдеқайда жинақы көрінеді. Мәтін мен форманы енгізу алаңының арасын ашуды ұмытпаңыз. Бұдан басқа күрделірек тәсіл – форма элементтерін кесте ұяшықтарына орналастыру. Бұл бағыттаушы мәтінді сол жақтан, ал енгізу алаңын оң жақтан орналастыруға көмектеседі.

Формаларды жұмыс істеуге қалай мәжбүр етуге болады? Сіз күтіп отырған бөлім осы шығар – енгізілген берілгендерді формамен серверге қалай сілтеуге болады. Шындығында олай жасау өте оңай. Ең бірінші мәселе – ол берілгендерді қайда жіберу, соны түсіну. Сіз <FORM> тегінде ACTION= атрибутын жазасыз және енгізілген берілгендерді өңдеп, олармен пайдалы бір нәрсе істейтін URL программаға сілтеме жасайсыз.

Өзін тексеруге арналған сұрақтар:

1. Қолданушыдан қайтымды ақпаратты алудың мақсатқа лайықтылығы
2. Қонақ кітап/форумды құру.
3. Қолданушының ақпаратты енгізуі және өзгертуі
4. Форма – қолданушы мен қарым қатынас жасаудың негізгі элементі.

Ұсынылатын әдебиеттер:

1. HTML и CSS на примерах/Мержевич – СПб: BHV, 2005-448 стр.
2. HTML и CSS. Совместное использование/ДеБолт – СПб: HT-Пресс, 2006-504 стр.
3. HTML, CSS, Скриптер: сайт құруда практика/Ломов – СПб: BHV, 2005-416 б.
4. HTML/XML/CSS/Штайнер – СПб: Базалық білімдер Лабораториясы, 2001, 512 б.

Дәріс 13. Апплеттер. Стильдердің каскадты кестелері (CSS)

Дәріс мазмұны: APPLET ұғымы. APPLET контейнерінің түрі PARAM контейнерін қолдану Жаңа HTML – контейнерлер. HTML контейнерлерінің қасиеттері. Контейнерлердің қасиеті.

Апплеттер

Желі ақылы қолданушы компьютеріне кішкене ұялы телефондардың программасын енгізіп, олармен жұмыс істеу әрине Java көмегімен орындалады. Бұл программалар Java терминологиясында апплеттер (алмалар) деп аталады. HTML-құжаты мәтініне апплеттерді шақыру үшін APPLET контейнері енгізілді, ол <applet> тегімен басталып, </applet>тегімен аяқталады. Апплеттерге сілтемелері бар жалпы құжат түрі төмендегі мысалдағыдай болады.

Бұл мысалда құжат тақырыбы және (H1 тегі) және қысқь сызықтан кейін ені 200 пиксель және биіктігі 100 пиксель болатын апплет өрісі басталады. Берілген өріске hello (файл hello.class) деп аталатын applet жүктеледі. applet контейнері тегтердің басы мен аяғының арасындағы мәтін ны қолданбай, браузер көмегімен бейнеленетін құрылған мәтіндер мен контейнерлерді орналастыру үшін қолданылады.

```
<HTML>
<HEAD>
<TITLE> applet-ке сілтемелері құрылған құжат.
</TITLE>
</HEAD>
<BODY bgcolor=#FFFFFF>
<CENTER>
<H1> hello Java- апплетімен құрылған құжат </H1>
<HR>
<APPLET CODE=hello width=200 height=100>
Егер сіздің браузеріңіз Java қабылдаса, Applet осы орында бейнеленеді
</APPLET>
<HR>
</BODY>
</HTML>
```

Мұндай құжатқа сілтеме нәтижесінде біріншіден құжат мәтіні жүктеледі, сосын applet контейнері табылып, апплет кодының жүктелуі басталады. hello.class файлы HTML-файлы қайда орналасса, ол да сонда орналасуы керек.

Апплетті қабылдаған соң браузер өз жұмыс алаңында оған орын қалдырады, содан кейін барып оны орындауды бастайды.

Жалпы жағдайда APPLET контейнері мына түрде болады: <applet

```
[codebase = codebase url]
code = applet.class
[alt = text]
[name= applet name]
width = number of pixels
height = number of pixels
[align = alignment]
[vspace=number of pixels]
[hspace=number of pixels]
[<param name=param name value=param value>]
[HTML text]
</applet>
```

Параметр codebase параметрі апплет кодын іздеуіне арналған базаға сұрау қояды, code – бұл class кеңейтілуі болуы тиіс апплет файлының аты, alt – альтернативті текст – апплеттің орындауы аяқталған жағдайда бейнеленеді, name – applet контейнерінің аты, контейнерге сілтеме үшін қолданылады, width – апплетаны бейнелеудің облыс ені, height – апплетаны бейнелеудің облыс биіктігі, align – управляет выравниванием области отображения апплета внутри рабочей области браузера, vspace и hspace – указывают на отступ от текста HTML-документа (вертикальный и горизонтальный, соответственно).

PARAM контейнерін қолдану, параметрлерді апплет ішіне және HTML-құжатқа беруді қамтамасыз етеді. Бұл командалық жолдың әр түрлі аргументтерімен команданы шақыру болып табылады. Параметрлерді апплет ішінде алу үшін getParametr () әдісін пайдаланған жөн.

APPLET контейнерінің атрибуттарынан code, width және height ғана міндетті болып табылады. Қалған басқа атрибуттарды (олар тік жақшаға алынған "[]") түсіре беруге болады. Java-программаларды өңдеу жүйелерінің көпшілігі HTML-құжатты, дәлірек айтқанда, Java-апплеттерді тестілеу үшін олардың макеттерін генерациялайды. Бұлай мысалы үшін, IBM-нің ADK (Applet Development Kit) компаниясында жасайды. applet шақырылған URL құжатты алу үшін, getDocumentBase және getCodeBase әдістерін қолдануға болады. Оның біріншісі құжатқа арналған адрес базасын анықтаса, екіншісі сол уақытта апплеттің базасын анықтайды. Мұндай әдістерді applet құжаты орналасқан берілгендермен (мәтін немесе графика) жұмыс істегенде қолданады.

Стильдердің каскадты кестелері (Cascad Style Sheets)

Стильдердің каскадты кестелері HTML 3.0. спецификациясының ішінде өңделген w3c (www Consortium) да ұсынылған. Бірақ, олар 1997 жылы ғана шынайы әсер етуші навигатор болып жасалды. HTML – тілінде оларды қолдану Netscape Navigator 4.0 және Internet Explorer 4.0 версиясымен ғана қолданыла алады. Кесте негізіндегі идея өте қарапайым. 4.0 – ші версияға HTML тілі құжаттарды контейнер түріне келтіруге сүйенетін белгілеу тіліне айналды, яғни құжат дегеніміз – бұл бір біріне салынған көптеген контейнерлер жиынтығы, олардың әрқайсысының ақпаратты көрсету қасиеті бар. Көптеген контейнерлерді біртүрлі контейнерлер класы деп топтауға болады, мысалы, тақырыпша және параграфтар. Контейнер қасиеті – контейнер басындағы tag атрибуты негізінде саналады. Сонымен қатар, көптеген контейнерлерде HTML – дың 3.0 версиясынан бастап осы атрибуттар жинағы типтелген. Көбінесе контейнерлер әмбебап программалау тілдерінде блоктарға кіреді. Сондықтан контейнер қасиеті айнымалы ретінде интерпретациялауға болады. Программалау тілдеріндегі сияқты, мысалы Паскальда, қасиеттің жазылуын құжаттың басында арнайы секцияға жазуға болады. бұндай секциялардың бірі – каскадты стиль секциясы. Әдетте стиль STYLE контейнерінің ішінде жазылады.

```
<style type = "text/css"> <! – стильдің жазылуы -- > < /style>
```

Жалпы айтқанда Netscape тағы бір стиль кестесінің нотациясын қолдайды. Ол Java Script нотациясы.

```
<style type = "text / javascript"> <! – стильдің жазылуы -- > </style>
```


Стиль кестесін суреттегенде біз Netscape құжатын қолданамыз, бұл Internet – тің ең атақты браузері. Internet Explorer w3c спецификасын ғана қолдайды. Бірінші мысалдан бастайық. Құжаттың тақырыбында және параграфында жазылатын мәтін стилін суреттеу керек.

```
<html> <head>
<style type ="text/css">
p {color blue; font – family: times; font – size:10pt;}
h1 {color: black; font – size:12pt; font – style; Arial; text – align: center;}
</style> </head>
<Body>
<h1> Test Style Shits in Communicator </h1>
<p> This is a first part of the document
```

</body.

```
</html>
```

Дәл осы мысалды Java Script - нотацияда жазайық.

```
<html> <head>
<style type ="text / javascript">
Tags. p. color ="blue"; tags. p. fontFamily="times"; tags. p. fontSize="10pt";
Tags. p. color="black";
Tags. h1. fontSize="12pt"; tags. H1. fontStyle="Arial";
Tags. textAlign="center";
</style> </head>
<body>
<h1> Test Style shifts in Communicator </h1>
<p> This is a first part of the document
</body>
</html>
```

Енді осы стильдерді тереңірек қарастырайық. Олардың қолдану облысы және оларды құжатта орналастыру амалдары, Жаңа HTML – контейнерлер. Стильдер кестесінің найда болуымен қатар тілде үш жаңа контейнер пайда болды. Олар: STYLE, LINK, SPAN. Негізінде LINK бұл жаңа таг емес, ол ескі тагтың жаңаша қолданылуы. STYLE контейнері <style type=" ... "> ... </style> стильдерінің жазылу кестесін анықтау үшін қызмет жасайды. CSS спецификациясында STYLE – ды құжаттың қандай контейнерінде қолданылатыны туралы тура айтылмайды, бірақ мысалдарда HEAD контейнерінің ішінде осы контейнері жиі қолданылады. LINK контейнері стильді жазу контекстінде берілген құжатқа стильдердің жазылуымен сыртқы файлды анықтау үшін қолданылады. Мысалы, сыртқы файл келесідей стильдер жазуынан тұрсын: /* CONTENTS OF THE EXTERNAL STYLE SHEETS FILE CSS.HTM*/

```
p {color: blue; font – family: times; font – size: 10pt;}
h1 {color: black; font – size: 12pt; font – style: Arial; text – align: center;}
```

/* the end of style sheets definition */ Осы стильдер жазылуын құжат тақырыбында қолдану үшін келесі тагты қолдану керек.

```
<html> <head>
<link REL=STYLESHEET TYPE ="text/css" HREF://LOCALHOST/CSS.HTM>
</head>
<body> The body of the document should be here.
</body>
</html>
```

CSS.HTM файлындағы стильдің жазылуы STYLE контейнеріндегі жазылуымен бірдей екендігін осы мысалдан көруге болады. SPAN контейнері ағымдағы мәтін фрагментін бейнелейтін стильді анықтау үшін қолданылады және кей жағдайларда FONT контейнеріне сәйкес. Әдетте SPAN типографиялық эффектiге жету үшін қолданылады, мысалы абзацтың бастапқы әрпінен жекелеу.

```
<HTML> <!--Author: Paul Khramtsov    Date: September 19, 1997    URL:
```

[HTTP://POLYN.KIAE.SU/INTERNET/INTRO.HTML](http://POLYN.KIAE.SU/INTERNET/INTRO.HTML) →

```
<HEAD> <style TYPE="text/css">
FS.all {color: red; font-size: 24pt; font-family: times;}
```

H1 {color: navy; text-transform: uppercase; font-size: 18pt; font-weight: bold; font-family: times;} H2 {color: navy; font-size: 14pt; font-weight: bold; font-style: italic; font-family: times;}

```
    H3 {color: navy; font-size: 10pt; font-weight: bold; font-family: times;}
P {color: navy; font-size: 12pt; font-family: times; text-align: justify;}
P.HELP {color: dackgreen; font-size: 10pt; font-family: times; text-align: justify;}
P.LEFT {color: navy; font-size: 12pt; font-family: times; text-align: right;}
</style>
</HEAD>
<BODY bgcolor = lightyellow>    <center>
<h3> Ақпаратты технологиялар орталығы </h3>
<h1> Internet ақпараттық ресурсы </h1>
<h3> Професионалдар емес үшін оқу әдістемесі </h3>
<h3> Храмов П.Б</h2> <h3> Москва, 1997</h2>
<hr> </center>
<p><span class = FS> С </span> 1995 жылы Internet – ке 25 жыл болды. Қазіргі кезде – бұл бүкіл әлемдегі өте
үлкен ақпараттық ресурс. Сонымен қатар Internet – бұл жер шарындағы ең атақты желі. 1997 жылы Internet
(Internic – Internet Information Center) рамкасында халықаралық желі координация орталығының берген бағасы
бойынша желіде бірнеше ондаған миллион компьютер – серверлер үнемі қосылып тұратындығы анықталған. Осы үлкен
ақпараттық ресурс толығымен орталықтандырылмаған және бір де бір мемлекетке және дүние жүзінің ірі финанстық
компанияларына тәуелді емес.
</Body> </HTML>
```

Берілген мысалда SPAN контейнері <p> параграфының басында тагтан кейін қолданып тұр. Бұл абзацтың бірінші әрпін ерекшелеуге мүмкіндік береді. Стильдердің жазу таблицасының жаңа контейнерінен басқа тағы жаңа атрибуттар жасалып шығарылды. HTML контейнерінің жаңа қасиеттері. HTML тегіндегі жаңа атрибуттарды STYLE атрибутынан бастаған дұрыс. Бұл атрибут контейнер басындағы таг ішіндегі контейнер бейнеленген стильді анықтау үшін қолданылады.

```
<h3 style = "line-height: 24pt; font-weight: bold; color: blue"> The blue text
<h3 style = "lineHeight = '24pt'; fontWeight='bold'; color='blue' "> The blue text
```

Сонымен қатар стиль классын анықтауға болады және CLASS атрибутының көмегімен қолдануға болады.

```
<style type = "text/css">
H3.class1 {font-size: 12pt; color = blue} </style>    ...
<h3.class = "class1"> This is a blue text
```

Осы жағдайда біз үшінші дәрежелі тақырып классын анықтадық, сонымен қатар кез келген контейнерге қолданылатын класты анықтауға болады.

```
<style type = "text/css"> all.class1 {font-size: 12pt; color = blue} </style>
```

Класты анықтағаннан басқа ат берілетін стильді құруға болады. Аты берілген стиль берілген бір класты анықтау негізінде құрылады. <style type = "text/css"> all.class1 {font-size: 12pt; color = blue} #C1 {font-size:20;} </style>

```
<h3.class = "class1"> This is a blue text <h3.class = "class1" id = "C1"> This is a blue text
```

Стильдерді жазуды басқаратын контейнерлер қасиеті. Бірінші топқа шрифт қасиеті жатады Font-size, font-family, font-weight, font-style Екінші топқа мәтін қасиеті жатады Line-height, text-decoration, text-transform, text-align, text-indent Үшінші топқа мәтін блогының қасиеті жатады margin-left, margin-right, margin-top, margin-bottom, margin, padding-top, padding-right, padding-bottom, padding-left, paddings, border-top-width, border-bottom-width, border-left-width, border-right-width, border-width, border-style, border-color Төртінші топқа фон түсін және текст түсін бейнелеуіштер құрастырады: color, background-image, background-color. Сонымен қатар, тізім элементтерін пулькалар типін және HTML-белгі элементтерінің басқа қасиеттер қатарын анықтайтын қасиеттері бар болады.

Өзін тексеруге арналған сұрақтар:

1. www- беттердегі көріну мен позициялауды басқару.
2. html-беттердегі апплеттер және басқа объектілер.
3. Www-беттерді ұсыну және ұстануды бөлу технологиясы.CSS.
4. Стильдің күрделі кестелері. Селекторлар, қасиеттер, қасиеттің мәні. Псевдоклассы.

Ұсынылатын әдебиеттер:

1. HTML, CSS, Скриптер: сайт құруда практика/Ломов – СПб: BHV, 2005-416 б.
2. HTML/XML/CSS/Штайнер – СПб: Базалық білімдер Лабораториясы, 2001, 512 б.
3. HTML, CSS, Скрипты: практика создания сайтов/Ломов – СПб: BHV, 2005-416 стр.
4. HTML/XML/CSS/Штайнер – СПб: Лаборатория Базовых Знаний, 2001, 512 стр.

Дәріс 14. Web –тармақ беттерін көруді басқару. JavaScript

Дәріс мазмұны: JavaScript объектітері - Navigator'a объектітері модельдері Кластар иерархиясын жазу Объектілер әдісі және объектілер қасиеті Оқиғалар Массивтер Графика

Қазіргі заманғы гипермәтіндік ақпараттық жүйелерді бірнеше компоненттердің бірігуі түрінде елестетуге болады: гипермәтіндік объектілерді сақтау жүйесі, гипермәтіндік объектілерді бейнелеу жүйесі, гипермәтіндік объектіні дайындау жүйесі және гипермәтіндік объектіні көруді программалау жүйесі. Осы көзқарастың арқасында World Wide Web 1996 жылы ғана аяқталған және функционалды толық түрін алды. Ең бірінші болып сақтау және көру жүйесі жасалды (1989-1991 жж) және бұлар қазіргі кезде дамуын жалғастырып келеді. 1990 жылдың аяғында бірінші рет құжатты дайындау жүйесі пайда болды. Соңында 1995 жылы көру сценарийін басқару тілдері бірінші рет ұсынылды. Деректер базасының гипермәтінді көру процедураларының программалануы Netscape, Microsoft немесе Sun – ң жаңалықтары емес. Бірнеше дәрежедегі барлық локальды гипермәтіндік жүйелер практика жүзінде гипермәтіндік объектіні манипуляция жасау ететін программалау құралын қамтиды. Кей жағдайларда барлық гипермәтіндік деректер базасы үлкен бір программа ретінде ұсынылуы мүмкін, бұндағы гипермәтіндік тармақ ол – программалық модульдер, ал олардың арасындағы байланыс – бір модульден екіншісіне беруді басқару, Осындай жақын келу жағдайы традициялық статистикалық белгінің алдында белгілі гипермәтіндік желіні орнатудың ыңғайлағы, деректер базасының фрагментін айналдыру программасын ойлап табу қабілеті, белгілі қарапайым компоненттерден гипермәтіндік объектіні генерациялау. Динамикалық объектіні статистикалықтан алу өте оңай, көру рограммасының бар болу жағдайына байланысты жүйеде гипермәтіндік деректер базасын интерактивті режимде көру пакеттік режимге ауысуы мүмкін. Бұнда оператор қызметі программа командаларымен ауысады. Гипермәтіндік беттерді көру программасын скрипт деп атайды. Sh типті командалық интерпретатор үшін жазылған. Бұрындағыдай локальді жүйеде World Wide Web гипермәтіндік құжатты көру программалауынжа екі қадам болады: скриптті көруді программасын жасау немесе байт – код компиляциясы. Біріші қадам World Wide Web дәстүрін жалғастырады, гипермәтіндік бетті өңдеу үшін қарапайым текстік редактор керек және гипермәтіндік құжаттың өзі оператор адамға оңай оқылуы керек. Екінші қадам программаны орындау эффектісін арттырады және оның қауіпсіздігін қамтамасыз етеді. Бірінші қадам сияқты, екінші қадам да программалауға объектілі – бағытталған қадамға жүгінеді. Байт – код немесе оны мобильді – код деп те атайды. Олар Java программалау технологиясының контекстінде көп жазылған, сондықтан барлық ойымызды скриптке аударайық, нақтырақ айтқанда JavaScript тілінде жазылған скрипт туралы.

Navigator объектісінің – JavaScript объектісінің моделі

JavaScript – ң идеясы өте қарапайым. JavaScript программасында орындалатын операциялар жақсы танымал және түсінікті объектілер әрекетін бейнелейді, Олар Netscape Navigator прораммасымен жұмыс істеу облысы және HTML тілі контейнерінің элементі болып табылады. JavaScript – ң объектілі – бағытталуы осымен тоқталмайды. Қасиеттер жиынтығы бар және объектілер функциясының жиынтығынан тұратын объектілер ғана бар. Ақырғылары әдіс деп аталады. Әдістерден басқа функциялар да бар, олар дәстүрлі программалау тілінің функцияларына өте ұқсас және ол стандартты математикалық типпен жұмыс істеуге немесе программаны орындау процесін басқаруға мүмкіндік береді. Сонымен қатар JavaScript – те программалық үзілу аналогі деген оқиға бар. Осы оқиғалар World Wide Web – те жұмыс істеуге бағытталған, мысалы, Navigator жұмыс істеу облысына бетті жүктеу немесе гипермәтіндік сілтемені таңдау. Оқиғаларды қолдана отырып гипермәтіндік бет және программа авторы динамикалық объектіні көруді ұйымдастыра алады, мысалы, жүгіртпелі жол немесе көптерезелі интерфейсін басқару. Кластар иерархиясын жазу Барлық JavaScript объектісінің құрылуы Navigator жұмыс істеу облысынан бастау алады, оларды келесідей иерархия түрінде көруге болады, Қолдануға осы объект класынан басқа өзінің жеке объектісін құруға болады. Бірақ әдетте көптеген программа осы класс жүйесін қолданады және жаңасын құрмайды.

Объектілер әдісі және объектілер қасиеті. Есептеу ағымын басқару

Осы кластардың әрқайсысы әдістер кластарының объектісін басқару функциясынан тұрады. Осы әдістердің ең бастысы объектілердің мәндерін тағайындайтындар болып табылады. Әдетте бұлар меншіктеу операциясы арқылы бөлінеді. Бірақ қарапайым программалау тіліндегі операторлардың типтері JavaScript – те де қолданылады (+, -, *, /, %, >, <, +=, -=, ...). Сонымен қатар "+" қосу операторы жолмен жұмыс істегенде жолдың соңына тағы жаңа жолды қосуды білдіреді.

```
s = "string1"+"string2"
```

JavaScript – те сандар және стандартты класстарды жазу операцияларынан басқа есептеу ағымын басқару командалары да бар. break – циклдан шығу;

```
while(i < 6)
{   if(i==3) break;   }
```

continue –цикл соңына көшу;

```
while(i < 6)
{   if(i==3) continue;   }
```

for – цикл;

```
for(i=0; i<9; i++)
{   ...   }
```

for – объектіні меншіктеу операторы;

```
for(i in obj)
{   str = obj[i]   }
```

if..else - шартты оператор;

```
if(i>0)
{   ...   }
```

else

```
{ ... }
```

while - шартты цикл;

```
while(j==k)
{ j++; k--; }
var – айнымалыларды хабарлайтын оператор.
var kuku = "kuku"
```

Айнымалылар типі оған меншіктелетін мәнмен анықталады. Осы көрсетілген операторлар JavaScript – ң операторларының толық жиынтығы емес, бірақ осы операторлар арқылы практикалық жұмыстарды орындауға болады.

Оқиғалар Тілдің маңызды элементі оқиға болып табылады. Программист оқиғаны скриптің программалық кодының белгілі бір бөлігін орындау үшін қолданады. Ең жиі қолданылатын амалдардың бірі – Navigator беттерін жүктеу кезінде анықталған іс - әрекетті орындау. Барлық оқиғаларды толық санамай – ақ олардың ішіндегі көп қолданылатынын ғана қарастырайық. Netscape Navigator 3.0 пайда болуы және JavaScript 1.1 версиясының пайда болуы Website – ы көру сценарийін басқару мүмкіндіктерін бақылауды жалғастыруға мүмкіндік берді. Тілдің жаңа версияларында мыналар қарастырылған: JavaScript және Java арасындағы байланыс мүмкіндігі, plug-ins орнатылғанын анықтау, объектінің (Area, Function, Image) жаңа типтері анықталған және басқа да ерекшеліктер. Құрастырушылардың ойынша осылар JavaScript программасының қызметін арттырады.

Массивтер

```
Біз қарастыратын жаңа объектілердің бірінші типі массивтер болып табылады. "Array" типі JavaScript 1.1- де Navigator мен бейнеленетін әртүрлі объектілерді манипуляция жасау ету үшін келтірілген. Бұл берілген Website бетінің барлық гипермәтіндік сілтемелерінің тізімі, берілген беттің барлық суретінің тізімі, берілген беттің барлық applet – ң тізімі, форманың барлық элементтерінің тізімі. Array() конструкторын қолдану арқылы қолданушы өзінің жеке массивін құра алады. Бұл келесідей әрекетпен іске асады.
new_array = new Array()
new_array5 = new Array(5)
colors = new Array ("red","white","blue")
```

Массивтің өлшемі динамикалық өзгеруі мүмкін. Ең бірінші массивті анықтау керек, содан барып оның бір элементіне мән меншіктеуге болады, Осы мән меншіктелгеннен соң барып, массивтің өлшемі де өзгереді.

```
colors = new Array()
colors[5] = "red"
```

Осы жағдайда массив 6 элементтен тұрады, яғни массивтің бірінші элементі 0 индексінен басталады. Массив үшін үш әдіс анықталған: join, reverse, sort. Join символдар жолында массив элементтерін біріктіреді. Бұл әдісте аргумент ретінде белгіш көрсетіледі.

```
colors = new Array("red","white","blue")
string = colors.join("+")
```

Символдар жолындағы string меншіктеу мәнін қабылдағанда біз келесі жолды аламыз.

```
string = "red + white + blue"
```

Басқа әдіс reverse – бұл массив элементтерінің ретін өзгертеді, ал sort әдісі элементтерді өсуі бойынша реттейді. Массивтерде екі қасиет бар: length және prototype. Length – массивтің элементтерінің санын анықтайды. Егер массивтің элементтеріне бірнеше ескішіл операция орындау керек болса, онда мына цикл типтерін қолдануға болады.

```
color = new Array("red","white","blue")
  n = 0
  while(n != colors.length)
  {... цикл денесінің операторы ...}
```

Prototype қасиеті массив объектісіне қасиет қосуға мүмкіндік береді.

Графиктер Navigator 3.0 – ге дейін JavaScript – те Image типіндегі құрылған объектілер қолданып келеді. Тілдің жаңа версиясында осы типті объектілер үшін конструкторлар пайда болады.

```
new_image = new Image()
  new_image = new Image (width,height)
```

Әдетте мультипликация жасау мақсаты үшін графикалық объектілер массивін жасайды, одан кейін олар бірінен кейін бірі орындалады.

```
img_array = new Array()
img_array[0] = new Image(50,100)
img_array[1] = new Image(50,100)
```

....

```
img_array[99] = new Image(50,100)
```

Image объектісінің 10 қасиеті бар, оның ішінде ең маңыздысы src.img_array массив элементіне нақты бір суретті меншіктеу үшін келесі командалар тізбегін орындау керек.

```
img_array[0].src = "image1.gif"
img_array[1].src = "image2.gif"
```

...

```
img_array[99].src = "image100.gif"
```

Осы жағдайда атты меншіктеу цикілінде қолдануға болар еді, онда олар константадан және индекстік айнымалының мәнінен тұруы мүмкін. Тілдің жаңа версиясында Image типі объектіні HTML тегінде IMG деп атын өзгертуге болады. Сонымен қатар егер Image форманың ішінде қолданылса, онда ол осы форманың қасиеті болатынын ескеру керек. Бұл келесі графикалық объектілер үшін әртүрлі құрама аттар қолдану керек дегенді білдіреді.

```
<img name=car src=car.gif>
```

← Құжатқа қоса салынған объект document.car.src = "car1.gif"

```
<form name=kuku>
  <img name=car src=car.gif>
```

← Құжатқа қоса салынған объект

```
</form>
```

document.kuku.car.src = "car1.gif" Бірақ скрипттерді мысалдарда қолданғанда берілген бетте барлық графикалық объектілердің массивінде индекс бойынша Image – ге көп көңіл бөлетіні жиі кездеседі. Мысалы біздің объект беттегі екінші Image болсын, ол форманың ішінде болмаса, оның сыртында болса да бәрібір, оған әрқашан индекс бойынша қарауға болады.

```
document.images[1].src = "car1.gif"
```

Енді осы мысалды Image массивімен кеңейте отырып, біз мультипликациясы бар құжат құрамыз.

```
<HTML>
  <HEAD>
    <SCRIPT>
      function multi_pulti()
```

```
{
```

```
  img_array = new Array()
  img_array[0] = new Image(50,100)
```

```
....
```

```
  img_array[99] = new Image(50,100)
  img_array[0].src = "image1.gif"
```

```
...
```

```
  img_array[99].src = "image100.gif"
  n=0
  while(n==0)
```

```
{
```

```
  document.images[0].src = img_array[0].src
```

```
... } }
```

```
</SCRIPT>
</head>
<body onLoad="multi_pulti()">
<img src=image1.gif>
</body>
</html>
```

Өзін тексеруге арналған сұрақтар:

1. Web-сервер және браузердің өзара байланыс принциптері. 2. Клиент – сервер қатынасының негізі.
2. Жоғары деңгейлі протоколдар.
3. JavaScript тілі. Синтаксисі. Функциялары, объектер.
4. JavaScript тілі. Құжаттың объекті моделі

Ұсынылатын әдебиеттер:

1. Бранденбау Дж. JavaScript: сборник рецептов. СПб.: Питер, 2000. — 416 с.

Дәріс 15. JavaScript тілінің элементтері және оны HTML құжатында орналастыру

Дәріс мазмұны: Гипермәтінді сілтеме стектері Фреймдер және терезелер Өртүрлі беттермен скриптер кодының мұрағатталуы Java, JavaScript және Plug-ins SCRIPT контейнері LANGUAGE атрибуты Көптірліліктегі бірлік

Гипермәтінді сілтеме стектері Тілге объектінің жаңа типі – Area енгізілген. Area - бұл client-site imagemap-ты анықтайтын MAP контейнерінің элементі. Нақты айтқанда, AREA-да анықталған сондай объектінің ең басты қасиеті, гипермәтінді сілтемелер, қайта анықтау үшін ғана рұқсат етілген. Олар беттің қарапайым сілтемелер массивінде пайда болады. AREA объектісіне аты бойынша қатысуға болмайды. Құжаттың гипермәтінді сілтеме массивінің индексін ғана қолдануға болады. Гипермәтіндік сілтеменің стек контекстінде, гипермәтіндік сілтеме бойынша жалпы көшумен байланысты JavaScript-тің тағы бір мүмкіндігін қарастыру қызықты. Қарапайым жағдайда A контейнерінің HREF параметрі қандайда бір мәнге ие болуы керек. Егер, мысалы, onClick оқиғасында жаңа терезе ашу және ескіде бейнеленген құжатты сақтау керек болса, онда оның URL-н HREF мәні ретінде көрсету қажет. Ал, кері жағдайда, егер HREF=" " болса, ескі терезеге бос бет жүктеледі. JavaScript жаңа нұсқасында void функциясы енгізілген. Анығырақ айтқанда қандайда бір мәннің жоқтығын білдіретін void типі. Егер бізге гипермәтінді сілтеме тандау кезінде қандайда бір әрекет орындау керек болса, бірақ осыған қоса ағымдағы беттерді асыра тиету етпеу керек болса, онда HREF параметрінде конструкция көрсетуге болады:

```
<A HREF="javascript:void(0)">kuku</A>
```

Ондай тәсілдерді гипермәтіндік сілтеме алаңы арқылы тышқан манипуляторын өткізумен байланысты оқиғаларды программалау барысында жиі пайдаланады. Фреймдер және терезелер Фреймдермен және терезелермен жұмыс істеген кезде JavaScript-тің алдыңғы нұсқаларда үнемі оларды дұрыс жабу үшін, терезелер мен фреймдердің кезекті ашылуларын мұқият бақылауға тура келетін. Кейбір терезелермен жұмыстың ақаулары mail-bombs деп аталатынға негізделген. Егер қолданушы пошта арқылы құрамы бір ғана командадан тұратын құжат қабылдаса:

```
window.close()
```

онда жүйе қолданушыдан сұрамай-ақ кезекті терезелерді жабады, ал бұл сәтте электронды пошта терезесі осындай терезе болып табылады. Енді, бір нәрсені жаппас бұрын, жүйе рұқсат сұрайды. Шындығына келсек, үнемі емес. Егер терезені жабу туралы команда алынған сәтте экранда жалғыз терезе Navigator болса, онда жүйе оны ешқандай сұхбатсыз жабады. Бірақта, терезелермен және фреймдермен жұмыс осымен не исчерпывается. Біріншіден жүйеге кезекті терезе немесе фрейм үшін анықталған орепер жаңа қасиеті енгізілген, және blur және focus әдістері терезелермен жұмыс істеу үшін көп таралған. Орепер қасиеті кезекті құжат терезесін ашуды шақырған, құжат терезесін анықтайды. Бұл қасиет кез-келген фрейм және терезе үшін анықталған. Берілген терезені ашқан, терезеге қатысты кейбір функцияларды орындау керек, және мына түрдегі өрнекті орындауға болады:

```
window.opener.[method]
```

Мысалы, егер негізін салушы – терезені жабу керек болса, онда жай ғана close әдісін орындауға болады:

```
window.opener.close()
```

Тура сондай тәсілмен write немесе writeln әдістерінің көмегімен осы терезенің құрамын өзгертуге болады. Негізін салушы терезесінде объектілердің басқа да қасиеттерін ауыстыруға болады. Келесі мысал JavaScript спецификациясына қосымшадан алынған: window.opener.document.bgColor='cyan' Тап осы жағдайда негізін салушы терезесі үшін фон түсі ретінде ашық көк түс анықталған. Бірақ, ең кереметі негізін салушыны ауыстыруға болатындығы. Бұл дегеніміз, автор терезелерді өз бетімен ашып жабуға мүмкіндік алады. Бұл жағдайда көптерезелі интерфейсті басқару анағұрлым икемді болады: window.opener= new_window window.opener = null Бірінші мысал ағымдағы терезе негізін салушы – терезе үшін арналған, сол уақытта екіншісі негізін салушыны қандай да бір әрекеттерден қорғайды. Терезелер туралы айтылғандардың барлығы фреймдерге де таралады. Фреймдермен жұмыс кезінде, фрейм көбінесе басқа бір фреймнің бөліктерге бөліну жолымен туындайды. Егер осыған қаса негізін салушы – фреймнің негізін салушы – терезесіне қатынасу

керек, онда бұл жағдайда оренер қасиеті алмаспайды. Тілдің өңдеушілері авторлардың тілектеріне Websit'тарға және тілді өңдеудің алғашқы сатыларында айқындалмаған JavaScript қолдану практикасына сүйенеді. Терезелер мен фреймдердің әртүрлі қасиеттеріне назар аударғаннан басқа, өңдеушілер blur және focus әдістерінің әрекетін фреймнен бастап терезеге дейін кеңейтті. Енді фрейм ғана емес терезе де ағымдағы focus әдісін пайдалану арқылы жасалады немесе керісінше blur әдісінің көмегімен фонға көшіріледі. Жағдайлар қатарында бірнеше беттер жасалған кезде, мысалы, бұл функцияларға назар аудару пайдалы болады.

Скриптер кодының әртүрлі беттермен мұрағатталуы Әртүрлі Website беттер арасындағы қандай да бір мұрағаттаудың жоқтығынан, өңдеушілер бір беттен екінші бетке жиі қолданылатын функциялар мен айнымалылардың жетерліктей үлкен бөлігін тасымалдауларына тура келді. Глобальды ресурстарға рұқсат алу немесе сондай глобальды ресурстарды анықтау мүмкіндігін алу жаман болмас деген әңгіме JavaScript пайда болған сәттен бастап келе жатыр. Өкінішке орай, функциялар және фреймдер өлшемдерін бір терезеден немесе фреймнен басқасына жіберудің логикалық механизмі JavaScript-теде жоқ. Әйткенмен, бұл бағытта бүліну болады. SCRIPT контейнерінде SRC атрибуты пайда болды. Бұл авторларға функциялардың өзіне тән библиотекасын құруға мүмкіндік береді, кез-келген беттен қатыса алатын. (Это дает возможность авторам строить своеобразную библиотеку функций, к которым можно обращаться из любой страницы, в которой будет ссылка на такую библиотеку.) Сонымен қатар библиотеканың өзін сол Website бетінің гипермәтіні орналасқан серверде орналастыру міндетті емес. (Басқа бір адамдардың да функцияларын пайдалануға болады Можно использовать и чужие функции, написанные кем-либо из ветеранов программирования на JavaScript на другом конце Земли.) SRC атрибутында қарапайым URL қолданылады. Скриптер файлының ішінде SCRIPT тектері қолданылмайды. Бұл белгілі-бір функция және айнымаларды қолданатын қарапайым файл. Әрине, бөтен скриптерді пайдалану пайдалы туынды ғана емес, сонымен қатар күтпеген мәселеге айналуы мүмкін. Для обычных пользователей страниц Website беттерінің күнделікті қолданушылары үшін скриптердің көп мөлшерін көшіріп алу шығынға әкеледі. Библиотекадан екі функцияның біреуін бір бетте пайдалану мүмкін жағдай, ал көшіруге келгенде бүкіл библиотеканы алуға тура келеді. Функциямен жұмыс істеудің басқа бір жаңа мүмкіндіктері жаңа Function объектісін енгізілуі болды. Function объекті Function конструкторынан туындайды:

```
new_Function = new Function(arg1,arg2,...,argn,
```

```
function_body)
```

 Функцияның әдеттегі декларирленуінен басты айырмашылығы сонда, оның нақ осы жағдайда кез-келген объектімен жұмыс жүргізуге болатын new_Function айнымалысы туындайды. Функцияны кәдәмгә бөлгеннен мұндай айнымалылар тумайды. Кез-келген объект сияқты Function-ның өзінің қасиеттері бар,бірақ әдістері жоқ. Функция қасиеттері ретінде аргументтер орындалады және prototype арқылы жаңа қасиеттер ұсыну мүмкіндігі. Функциялар және мұрағаттау туралы әңгімені аяқтай отырып оренер терезелерінің қасиетіне және фреймдерге назар аудару қажет. Бұл қасиетті ата-аналар – беттеріне объектілерге қатынасқанда қолдануға болады.

Java, JavaScript және Plug-ins Тілдің жаңа нұсқасында Java-applet'тер және JavaScript-скриптер арасында өзара қарымқатынас ұйымдастыру мүмкіндігі бар. APPLET контейнерінде MAYSCRIPT атрибутын қолдану есебінен жүзеге асады. JavaScript-те applet'-тер массивінде индекс бойынша немесе атауы бойынша қатыса алатын APPLET типті объект анықталған. Бұл объектіде бір ғана қасиет бар - атауы. Ешқандай қасиеттер немесе әдістер объектілердің берілген типтеріне белгіленбеген. Applet'тер мен скриптер өзара арақатынас бөлшектері Java applet'-тер программалау шеңберінде талқылаған дұрыс, сондықтан біз осы жерде бұл ерекшеліктерді босатамыз. поэтому здесь мы эти особенности опустим. Applet'терден басқа JavaScript Plug-ins-термен де жұмыс істеуге мүмкіндік береді. Соңғылары өздерін қасиеттер қатары анықталған сәйкес типтегі массив ретінде жариялайды. Осы қасиеттерді пайдалана отырып орнатылған plug-ins-ті және оның MIME-типтеріне сәйкестігін анықтауға болады. Plug-ins –ті тағайындау немесе атауларды манипуляциялауға болмайды.

HTML-құжатына орналастыру Скриптерді HTML-құжатының денесіне орналастыру үшін SCRIPT контейнері пайдаланылады. Көру программаларының барлығы тануға(распознавать) және скриптерді орындауға қабілетті емес, сондықтан скриптің денесінің өзі комментари контейнерінде орналасады. Анықтылық үшін мысал қарастырайық:

```
<HTML>
```

```
<!-- Author: Pavel Khramtsov Date: March 5, 1996 -->
```

```
<HEAD>
<TITLE>Registration</TITLE>
<SCRIPT LANGUAGE="JavaScript">
```

<!-- The JavaScript Functions Definition

```
function help_win()
```

```
{
```

```
    Help = window.open("reg_help.htm","Help",
    "scrollbars=yes,resizable=yes")
```

```
}
```

```
function main_menu()
```

```
{
```

```
    Main_menu = window.open
    ("main_m.htm","Main_menu")
```

} // The end of JavaScript Functions Definition →

```
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<TABLE>
<TR><TH><IMG SRC="radleg3.gif"></TH><TH><H2>ISTC.
Project RADLEG(#245)</H2>
RRC "Kurchatov Institute"<BR>
<I>Dec, 25, 1995</I>
</TH></TR>
</TABLE>
</CENTER>
<HR>
<CENTER>
<h2>Registration Form</h2>
<hr>
```

..... Беттің мәтіні.

```
<hr>
<FORM NAME=help_call>
<TABLE BORDER=0>
<TR>
<TD><INPUT NAME=next TYPE=button VALUE=Next
onClick="main_menu()"></TD>
<TD><INPUT NAME=help TYPE=button VALUE=Help
onClick="help_win()"></TD>
</TR>
</TABLE>
</FORM>
<hr>
</CENTER>
```

</BODY>

</HTML>

Берілген мысалда құжат тақырыпшасына (HEAD контейнер) SCRIPT контейнері қосылған. Осы уақытқа дейін LANGUAGE атрибуты бұл контейнерде міндетті болып табылған жоқ. Бірақ, VBSCRIPT пайда болған сәттен бастап тілдің типін көрсеткіші ретінде пайда болған - Navigator скриптерді басқа тілдерде түсінбейді. Атрибуттың жоқтығы гипермәтінді беттерді бейнелеу кезінде қателіктерге әкелуі мүмкін. Далее в тексте страницы определен комментарий, в который включен текст скрипта. Комментарий жолдан басталады: <!-- The JavaScript Functions Definition Және жолдан бітеді: // The end of JavaScript Functions Definition → Скриптің ішкі контейнерінде пайдаланушының екі функциясы айқын: help_win() және main_menu(). Бұл функциялардың мәні жеткілікті ашық. Help_win() help тәуелді контексті ұйымдастыру үшін арналған, ал main_menu() қолданушыны Website бас мәзірге кіру рұқсатын жүзеге асырады. Екі функция да Netscape Navigator жаңа терезесін ашуға және оған гипертекстік құжаттарды жүктеуге бір орен әдісін қолданады (window объектісімен анықталған JavaScript –те құрылған функция). main_menu функциясы стандартты терезе болып саналады (батырмалармен, икондермен және т.б.), ал help_win() функция терезесі стандартты емес атрибуттар болып саналады. Тек осы терезеде ғана бұрылғыш сызғыштары болады (scroll bars). Функцияны шақыру HTML-формсында анықталған help_call сәйкес батырманы басқанда ғана жүзеге асады. Функцияның орындалуы мынадай жағдайда іске асады, егер оқиға сәйкес форма өрісінде INPUT контейнерінде onClick атрибутымен жазылған болса. Егер нақтырақ болсақ, онда келтірілген мысал – бұл фрагмент, қолданушының тіректуін іске асыратын. Оның орындалу мезетінде бас мәзір терезесі ашық болады, сондықтан бұл терезені құру кезінде "NEXT" батырмасын таңдаған кезде орындалмайды, ал текст ашылған терезеде жүктеледі.

Көптірліліктегі біртүрлілік Қорытындылап айта кетсек, JavaScript – бұл құжаттарды қарау сценарийін басқаруда жалғыз тіл емес. Microsoft-тің өзінің версиясы Visual Basic негізінде VBScript аналогтық тілі бар. Сонымен қатар, Java-applet'-ден де қарау сценариінде басқаруға болады, бірақ бұл күрделірек, тиімдірек және қауіпсіз болады. Бұдан программист барлық мұрагерлік артықшылықтарды және объекті-бағдарламалау қалған атрибуттарын алады. Ақыр соңында беттің алғашқы түрін қалпына келтіруге және объектілерді қайта жабуға жұмыс облысының экранын фрагменттерге бөлуге мүмкіндік беретін, фреймдер мен терезелерді құру үшін HTML контейнерлерінің сәйкес атрибуттарын қолдануға болады. Сайып келгенде, қазірге кезде Web гипертекстік мәліметтер қорын қарау сценарилерін басқарудың еңболмағанда үш тәсілі бар, олардың әрқайсысы өзіндік жақы болады. Оның қайсысын таңдау Website авторының жұмысы.

Өзін тексеруге арналған сұрақтар:

1. JavaScript — объектіге-бағытталған программалау тілі сияқты. 2. JavaScript-те сценари (программа) — өзара қарым-қатынасқан объектілер жиынтығы сияқты.
2. Анықтымысын айтыңыз: тип, берілгендер мәнінің жиыны.
3. Анықтымысын беріңіз; объект, конструктор, тұпнұсқа.

Ұсынылатын әдебиеттер:

1. Бранденбау Дж. JavaScript: сборник рецептов. СПб.: Питер, 2000. — 416 с.
2. Вейнер П. Языки программирования Java и JavaScript. М.: Лори, 1998. — 242
3. Дарнелл Р. JavaScript: справочник. СПб.: Питер, 1999. — 192 с.
4. Фролов А. В., Фролов Г. В. Сценарии JavaScript в активных страницах Web. М.: Диалог-Мифи, 1998. — 284 с.