

Қостанай мемлекеттік педагогикалық институты

Жаратылыстану – математика факультеті

Информатика және компьютерлік технологиялар кафедрасы

**Айтбенова Аян Алтаевна**

**JAVASCRIPT ТІЛІНДЕ БАҒДАРЛАМАЛАУ**

Қостанай  
2018

ӘОЖ 004.4 (075.8)  
КБЖ 32.973-018.1я73  
А32

**Авторы:**

**А.А. Айтбенова**, Информатика және компьютерлік технологиялар кафедрасының аға оқытушысы, педагогикалық білім беру магистрі

**Пікір жазғандар:**

**А.Т. Байманкулов** - А.Байтурсынов атындағы Қостанай мемлекеттік университетінің «Ақпараттық жүйелер» кафедрасының меңгерушісі, физика-математика ғылымдарының докторы, профессор.

**З.С. Ерсұлтанова** - Қостанай мемлекеттік педагогикалық институтының «Информатика және КТ» кафедрасының аға оқытушысы, техника ғылымдарының кандидаты.

Айтбенова А.А.

А32 JavaScript тілінде бағдарламалау: жоғарғы оқу орындарының студенттеріне арналған оқу құралы. –Қостанай: ҚМПИ, 2018.-100б.

ISBN 978-601-7934-36-1

Ұсынылып отырған оқу құралы қазіргі кезде web–беттер құруға арналған кең қолданыстағы JavaScript тілін оқып–үйренуге арналған. Оқу құралында JavaScript тілі бойынша теориялық және практикалық сабақтарға арналған материалдар қамтылған, көптеген мысалдар келтіріліп, оларды орындау жолдары қарастырылған. Әр практикалық жұмыста өзіндік жұмыстар үшін тапсырмалар және бекіту сұрақтарымен толықтырылған.

Ұсынылған оқу құралын жоғарғы оқу орындарының студенттері, орта мектеп оқушылары мен мұғалімдер және осы тілді өздігінен үйренушілер де пайдалана алады.

ӘОЖ 002 (075.8)

КБЖ 32.973-018.1я73

Қостанай мемлекеттік педагогикалық институтының Ғылыми кеңесінің шешімімен ұсынылған.

ISBN 978-601-7934-36-1

©Айтбенова А.А., 2018

©ҚМПИ, 2018

## МАЗМҰНЫ

Кіріспе.....	4
<b>1. JavaScript тілі туралы жалпы мәліметтер</b>	
1.1 JavaScript тілінің негіздері.....	5
1.2 JavaScript тілінің құрылымы.....	9
1.3 Мәліметтер типі. Операторлар, өрнектер.....	15
1.4 Циклдік операторлар.....	17
1.5 Жолдық амалдар. Функциялар.....	19
1.6 JavaScript тілінің объектілері.....	21
1.7 JavaScript тілінің оқиғаларды өңдеу мүмкіндіктерін пайдалану.....	28
<b>2. Практикалық жұмыстар</b>	
2.1 Практикалық жұмыс №1. Негізгі операторлар.....	31
2.2 Практикалық жұмыс №2. JavaScript тілінің шартты операторлары.....	39
2.3 Практикалық жұмыс №3. JavaScript тіліндегі циклдік операторлар.....	41
2.4 Практикалық жұмыс №4. JavaScript оқиғалары.....	45
2.5 Практикалық жұмыс №5. JavaScript функциялары.....	48
2.6 Практикалық жұмыс №6. Math объектісі.....	52
2.7 Практикалық жұмыс №7. Array объектісі.....	54
2.8 Практикалық жұмыс №8. String объектісі.....	60
2.9 Практикалық жұмыс №9. Date объектісі.....	62
2.10 Практикалық жұмыс №10. Window объектісі.....	67
2.11 Практикалық жұмыс №11. Форма элементтері – жалаушалар, радиобатырмалар, тізімдер.....	70
2.12 Практикалық жұмыс №12. Image объектісі.....	75
Қолданылған әдебиеттер.....	79
Құжаттағы түстерді анықтау.....	80
Тестілік тапсырмалар.....	81
HTML тілінің тегтері туралы анықтамалық сөздік.....	97

## КІРІСПЕ

Қазіргі білім беру мекемелерінің даму болашағы қоғамның даму үрдісімен үнемі өсіп отыратын ақпарат көлемінің әртүрлілігі, студенттерге білім беруде жаңа оқыту технологияларын қолдану, инновациялық бағытта жұмыс жасау заман талабына сай талап етілуде. Компьютерлік техниканың бағдарламалау мүмкіндіктерін педагогикалық мақсаттарға қолдану, білім мазмұнын анықтауда, оқыту формалары мен әдістерін жетілдіруде жақсы әсерін тигізеді.

Бұл оқу құралының мақсаты – студенттерді HTML және JavaScript тілдері көмегімен қарапайым веб-беттер, сайттар жасауға үйрету болып табылады.

Берілген оқу құралы «JavaScript тілінде бағдарламалау» курсы бойынша теориялық және практикалық материалдармен танысуға көмектеседі. Осы курс бойынша практикалық жұмыстарды орындау үшін қажетті материалдар қамтылып, WWW ортасында осы тілді пайдалану жолдарынан көптеген мысалдар келтіріліп, бағдарламалық кодтар берілген. Әр практикалық жұмыста өзіндік жұмыстар үшін тапсырмалар және бекіту сұрақтарымен толықтырылған. Мұндағы көптеген мысалдарды кез-келген браузер арқылы орындап тексеруге болады.

# 1 JAVASCRIPT ТІЛІ ТУРАЛЫ ЖАЛПЫ МӘЛІМЕТТЕР

## 1.1 JAVASCRIPT ТІЛІНІҢ НЕГІЗДЕРІ

**Мақсаты:** JavaScript тілінің қысқаша тарихымен, негізгі ұғымдармен және оның мүмкіндіктерімен танысу.

**Жоспары:**

1. JavaScript тілі, оның қысқаша тарихы, скриптілік тілдердің тағайындалуы.
2. Негізгі ұғымдар.
3. JavaScript тілінің мүмкіндіктері.

### 1) JavaScript тілі, оның қысқаша тарихы, скриптілік тілдердің тағайындалуы

Netscape Communications корпорациясы ең басынан бастап World Wide Web динамикалық дамуына айтарлықтай үлес қосты. Осы саладағы жетістікке жету үшін Netscape Navigator браузерін жасау және тегін тарату (үйде пайдалану үшін) арқылы насихатталды.

Содан кейін, Netscape қысқа уақыт ішінде LiveScript деп аталатын сценарий тілін құрды, ол статикалық құжаттарды көп немесе аз интерактивті бағдарламаларға айналдыруға мүмкіндік беретін керемет құрал рөлін атқарады. Netscape Navigator браузерінің алғашқы нұсқаларында қолдау көрсетілетін LiveScript өте танымал және табысты болды. Сонымен қатар, Sun Microsystems-дағы инженерлер әртүрлі интерфейстердің стандарттарына сәйкес бағдарламалық жасақтамаға бейімделіп, Java тілін дамытты.

Netscape Communications және Sun Microsystems арасындағы келісімнің арқасында LiveScript-дің идеяларын Java құрылымымен біріктіре отырып, желілік қосымшаларды әзірлеуге және, ақырында, динамикалық веб-беттерді жасауға арналған «Mocha» деп аталатын орта болды. Ол ашық сипатта және қолданылатын бағдарламалық жасақтамадан тәуелсіз болды.

Жоба екі нұсқада 1995 жылы желтоқсанда 1.0 нұсқасында JavaScript атауымен жарияланған техникалық сипаттамаларды жасақтауға әкелді. Өз өнімдерінде JavaScript тілін қолдануға мүдделі көптеген компаниялар жаңа технологияға қызығушылық танытты. Microsoft корпорациясы JavaScript тілін Internet Explorer браузері арқылы көрсете алады деп жариялады.

Бірақ Microsoft корпорациясы JavaScript-ні пайдалануда кейбір қиындықтарды бастан кешірді, өйткені JScript деп аталған осы тілдің алғашқы іске қосылғандығын Internet Explorer 3.0-де жеткілікті сенімді деп санауға болмайды. Бұл Microsoft-тың мәжбүрлі, стандартты JavaScript нұсқасын пайдалануына мәжбүрледі және Internet Explorer авторлық шешімі болып табылатын VBScript (Visual Basic Script) тіліндегі сценарийлерді іске қосу мүмкіндігін қамтамасыз ете бастады. Оның

синтаксисі мен мүмкіндіктері JavaScript-ке өте ұқсас. Алайда, JavaScript әмбебап және кеңінен қолданылатын технология болып табылады.

HTML тілі Web-беттер авторларына мәтіндік және графикалық ақпаратты бейнелеу үшін үлкен мүмкіндіктерді береді. Бірақ HTML тілі көмегімен құрылатын HTML беттер статистикалық болып қала береді – яғни, пайдаланушы бетте орналасқан ақпаратты өзгерте алмайды және де көптеген интерфейстік элементтерді қолдана алмайды. Бетті интерактивті түрде жасау үшін – скриптілік тіл қажет.

*Скриптілік тілдер* интерактивті беттерді құру үшін қолданылады. Сондай скриптілік тілдің бірі JavaScript.

**JavaScript** – бұл қарапайым объектілі-бейімделген, браузерге түсінікті программалау тілі. Браузер бұл тілдегі командаларды орындай алады, яғни оның ішкі интерпретаторы бар. Ол Internet үшін клиенттік және серверлік қосымшаларды құру үшін арналған.

JavaScript программасы *скрипт* немесе *сценарий* деп аталады. JavaScript тілінде жазылған бағдарламалар HTML-құжаттарының құрамына кіреді және солармен бірге тарайды. Netscape Navigator және Ms Internet Explorer типті көру программалары (яғни, браузерлер) құжаттың мәтініне енгізілген кірістірілген программаларды (Script-кодтар) таниды және оларды орындайды.

JavaScript коды HTML тілін толықтырып, оны көрнекі етеді. JavaScript тілі арқылы гипермәтіндер жаңа мүмкіндіктерге ие болады.

JavaScript тілінде программаларды құру үшін JavaScript-ті тілін қолдайтын сәйкес версиялы браузер және HTML-құжаттарын құруға мүмкіндік беретін мәтіндік редактор қажет. JavaScript программасы HTML-құжатының мәтініне кірістірілген болғандықтан, сіз өз жұмыстарыңыздың нәтижесін, құжатты браузермен көрген уақытта көре аласыз және қажет жағдайда өзгертулер енгізе аласыз.

*Скриптілік тілдер көбінесе мыналар үшін қолданылады:*

- браузердің статустық жолындағы сұхбаттық тақталарды және хабарламаларды бейнелеу үшін;
- жүктелу уақытында беттің құрамын динамикалық түрде құру;
- беттің құрамын өзгерту;
- пайдаланушы көретін ақпаратты өзгерту;
- басқа беттер бойынша навигация;
- Web-бетке Java-апплет және Active X элементтері сияқты кірістірілген объектілерді басқару.

*JavaScript көмегімен келесіні жасауға болады:*

- веб-беттердің мазмұнын динамикалық түрде өзгерту;
- оқиғаларды өңдеушілерді элементтерге байланыстыру (кодты белгілі бір әрекеттер орындалғаннан кейін ғана атқаратын функциялар);
- кодты белгілі бір уақыт аралығында орындау;

- браузердің әрекетін басқару (жаңа терезелерді ашу, көрсетілген құжаттарды жүктеу және т.б.);
- cookie файлдарын жасау және оқу;
- пайдаланушы қай браузерді қолданатындығын анықтау (ож-ні, экранның кеңейтпесін, пайдаланушының қолданған беттерін және т.б. анықтау);
- серверге жіберместен бұрын форманың деректерін және тағы басқаларын тексеру.

## 2) Негізгі ұғымдар

JavaScript программалау тілін Netscape Communications-тегі Брендан Эйк ойлап тапқан. Бұл клиент тарапынан жұмыс істейтін сценарилерді жазуға арналған объектілі-бағдарланған тіл.

JavaScript тілі объектілерге негізделген. JavaScript объектісі – реттелмеген қасиеттер жиынтығы. Олардың әрқайсысының осы қасиеттің қалай пайдаланатындығын анықтайтын атрибуттары болады. JavaScript-тің бірге орнатылған ішкі Global, Object, Error, Function, Array, String, Boolean, Number, Math, Date, RegExp сияқты объектілері бар. Сонымен қатар JavaScript-тің ішкі амалдары болады, олардың функция не әдіс болулары міндетті емес. Оларға қоса оның программаның орындалу логикасын басқаратын ішкі операторлар жиынтығы бар.

JavaScript тілінің негізгі ұғымдарына тоқталайық:

*Типі.* Деректер мәндерінің жиынтығы.

*Қарапайым мәні.* Типтерінің бірі Undefined–тің элементі (анықталмаған), Null (нөлдік), Boolean (логикалық - true немесе false), Number (сандық) немесе String (жолдық). Қарапайым мәндер – бұл тікелей меншіктелетін және тілді ең төменгі деңгейде жүзеге асыратын деректер.

*Объект.* Object; әрқайсысы қарапайым мән, объект немесе функция бола алатын реттелмеген қасиеттер жиынтығы типінің элементі болып табылады. Функция болып табылатын қасиет **әдіс** деп аталады.

*Құрастырушы/Конструктор.* Объектілерді құратын және оларды инициализациялаушы функция. Әр құрастырушының қасиеттерді мұраға қалдыруы және бөлектеуі үшін пайдаланатын сәйкес прототипі болады.

*Прототип.* JavaScript-те құрылымды, жағдайды және өзін-өзі ұстауды мұраға қалдыруды жүзеге асыруға арналған объект. Құрастырушы объект құрғанда осы объектінің қасиетіне сілтеме жасауға мүмкіндік беретін құрастырушы прототипіне көрінбейтін сілтеме құрады. Прототип қасиеттері соның негізінде жасалған барлық объектілерге бөлінеді.

*Тіл объектісі.* JavaScript тілін жүзеге асыруды қолдайтын, бірақ сценарилердің орындалу ортасы болып табылмайтын кезкелген объект. Тіл объектісінің бір бөлігі өзімен бірге орнатылады, басқалары сценарилерді орындау барысында жасалады.

*Бірге орнатылатын объект.* Қандай ортада орындалуына қарамастан javascript тілін жүзеге асыруды қолдайтын және сценаридің орындалуының бастапқы сәтінде бар кезкелген объект. Барлық бірге орнатылған объектілер тіл объектісі болып табылады.

*Орта объектісі.* Тіл объектісі болып табылмайтын, сценарилер орындалатын орта қолдайтын кезкелген объект.

*Анықталмаған мән.* Айнымалыға ешбір мән меншіктелмегенін білдіретін қарапайым мән.

*Undefined типі.* Бұл тип анықталмаған мән болып табылатын undefined-тің жалғыз мәнінен тұрады.

*Нөлдік мән.* Нөлдік, бос немесе жоқ сілтемені білдіретін қарапайым мән.

*Null типі.* Бұл тип нөлдік болып табылатын жалғыз null мәнінен тұрады.

*Логикалық мән.* Ақиқат немесе жалған екендігін білдіретін Boolean типті қарапайым мән.

*Boolean типі.* Мына: true (ақиқат) және false (жалған) мәндерінен тұратын тип.

*Логикалық объект.* Бірге орнатылған логикалық объект экзепляры болып табылатын Object типті элемент. Басқаша айтқанда, логикалық объект *new Boolean(value)* өрнегімен жасалады, мұнда value – логикалық мән. Нәтижелік объектінің Boolean типті көрінбейтін (аты жоқ) қасиеті болады.

*Жолдық мән.* String типті элемент. Символдар жолын, яғни нөлдерден тұратын реттелген жиым немесе Unicode (16-биттік таңбасыз бүтін сандар) символдарын құрайды.

*String типі.* Бұл тип бүкіл жолдық мәндерден тұрады.

*Жолдық объект.* Бірге орнатылған жолдық объект экзепляры болып табылатын Object типті элемент. Басқаша айтқанда, жолдық объект *new String (value)* өрнегімен жасалады, мұнда value – жолдық мән. Нәтижелік объектінің String типті көрінбейтін (аты жоқ) қасиеті болады.

*Сандық мән.* Number типті элемент. Санның тікелей бейнесі болып табылады.

*Number типі.* Бүкіл сандық мәндерден тұратын тип.

*Сандық объект.* Бірге орнатылған сандық объект экзепляры болып табылатын Object типті элемент. Басқаша айтқанда, сандық объект *new Number (value)* өрнегімен жасалады, мұнда value – жолдық мән. Нәтижелік объектінің Number типті көрінбейтін (аты жоқ) қасиеті болады.

Енді бұдан әрі осыларды қолдануды қарастырайық.

*Объектілік модель.* Енді біз теориядан нақты іске көшейік. Біздер оқиға, объект дегендердің не екенін білдік. JavaScript тілінің объектілік моделін қарастыруға кірісейік.

Сонымен объектілік модель мыналардан құралады:



- объектілерден;
- қасиеттерден;
- әдістерден;
- оқиғалардан;
- коллекциялардан.

### 3) Java Script тілінің мүмкіндіктері

JavaScript көмегімен HTML құжаттарының бейнеленуін және құрамын басқаруға болады. Экранда бейнеленетін құжатқа кез-келген HTML-кодты жазуға болады.

JavaScript браузердің жұмысын бақылауға мүмкіндік береді. Мысалы: WINDOWS объектісі экранға шығарып сұхбаттық территорияларды шығаруға, браузердің жаңа терезесін құруға, ашуға және жабуға терезе өлшемдерін беруге және тағы басқа мүмкіндік беретін әдістерді қолдайды.

JavaScript құжаттардың құрамымен өзара әрекет жасауға мүмкіндік береді. Document объектісі және онда қамтылатын объектілер программаларға HTML құжатының бөлігін оқуға, кейде олармен өзара әрекет жасауға мүмкіндік береді. Мәтіннің өзін оқуға болмайды, бірақ, мысалы гипермәтіндік сілтемелердің тізімін алуға болады. Қазіргі таңда құжаттардың құрамымен өзара әрекет жасайтын Form объектісі және ол қамтитын объектілер кең мүмкіндіктер береді.

JavaScript пайдаланушымен өзара әрекет жасауын қамтамасыздандырады. JavaScript кез-келген математикалық есептеулерді орындауға мүмкіндік береді.

#### **Бақылау сұрақтары:**

1. JavaScript тілі қандай тіл?
2. Скриптілік тілдер не үшін қолданылады?
3. Java Script тілінің мүмкіндіктері қандай?
4. JavaScript тілін кім ойлап тапқан?
5. Скрипт немесе сценарий деген не?
6. JavaScript тілінің қандай объектілері бар?
7. Объектілік модель неден құралады?

## 1.2 JAVASCRIPT ТІЛІНІҢ ҚҰРЫЛЫМЫ

**Мақсаты:** JavaScript тілінің мәліметтер типімен, айнымалылармен, құжаттың құрылымымен және ақпаратты шығару тәсілдерімен танысу.

#### **Жоспары:**

1. Негізгі мәліметтер типі. Айнымалылар.
2. Құжаттың құрылымы.
3. Ақпаратты шығару.

## 1) Негізгі мәліметтер типі. Айнымалылар

Айнымалылардың, функциялардың және өрнектердің мәндері мынадай типті болады:

1. Бүтін сандық;
2. Нақты сандық;
3. Логикалық;
4. Жолдық;
5. Null-мәнінің «бос жиынын» белгілеу үшін арналған арнайы мән.

Айнымалылар идентификаторлармен белгіленеді. Идентификатор – міндетті түрде әріптен басталатын латын әріптері мен цифрлар тізбегі.

Айнымалы деп - алгоритмнің орындалу процесінде мәні өзгертін шаманы айтады. Айнымалыны анықтау үшін **var** кілттік сөзі қолданылады.

Мысалы:

```
var x=1 // x-ке 1 мәні меншіктелген.
```

```
var y=2 // y-ке 2 мәні меншіктелген.
```

```
var z=3 // z -ке 3 мәні меншіктелген.
```

x, y, z айнымалылар болғандықтан, олардың мәндері кез келген уақытта өзгере алады. Мысалы:  $x=y+z$  // $x=5$ , бұл 2 және 3 сандарының қосындысының нәтижесі.

## 2) Құжаттың құрылымы

JavaScript скриптінің коды HTML-бетте орналасады. Скрипт екі — `<SCRIPT>` . . `</SCRIPT>` тегтерінің арасында орналасады. Оның жалпы жазылу түрі мынандай:

```
<SCRIPT LANGUAGE="JavaScript">
```

```
...
```

```
JavaScript бағдарламасы
```

```
...
```

```
</SCRIPT>
```

Мұндағы: Language атрибуты берілген бағдарлама қай тілде жазылғанын көрсетеді. Скриптік бағдарламаларды қолдамайтын браузерлер, оларды тастап кетуі үшін, оларды комментарийлер блоктарының ішінде орналастыру керек:

```
<SCRIPT LANGUAGE="JavaScript">
```

```
<!--
```

```
...
```

```
JavaScript бағдарламасы
```

```
...
```

```
//-->
```

```
</SCRIPT>
```

Комментарийлер `<!-- ... //-->` жақшаларға алынады, HTML-комментарийдің тегі жабылуы үшін `"//"` екі қиғаш сызықшалар қолданылады.

Құжаттың жалпы құрылымы:

JavaScript тілінде жазылған бағдарлама құрылымының функциялары HTML-құжатының <HEAD> бөлімінде орналасады. HTML-құжатының мысалы:

```
<HTML>
<HEAD>
  <TITLE> JavaScript тіліндегі бағдарлама мысалы</TITLE>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      ...
      JavaScript бағдарламасы
      ...
    -->
  </SCRIPT>
</HEAD>
<BODY>
  ...
  HTML-құжатының мәтіні және JavaScript-те функцияны шақыру
  ...
</BODY>
</HTML>
```

#### **<NOSCRIPT>...</NOSCRIPT> тегтері.**

<NOSCRIPT>. . </NOSCRIPT> командасы міндетті емес болып табылады. Бұл командалардың арасында браузер JavaScript тілін көрмейтін пайдаланушылар үшін мәтін қамтылады. Бұл команданың арасында HTML тегтері орналасады.

```
<HTML>
<HEAD>
<TITLE> JavaScript тіліндегі бағдарлама мысалы /TITLE>
<SCRIPT LANGUAGE="JavaScript">
  <!--
    ...
    Java Script бағдарламасы
    ...
  ->
</SCRIPT>
  <NOSCRIPT>
  <!--
    Мұнда браузер JavaScript тілін көрмейтін пайдаланушылар
    үшін мәтін қамтылады
  ->
  </NOSCRIPT>
</HEAD>
```

```
<BODY>
```

```
...
```

HTML-құжатының мәтіні және и JavaScript-те функцияны шақыру

```
...
```

```
</BODY>
```

```
</HTML>
```

<NOSCRIPT>...</NOSCRIPT> командасы келесі түрде көрсетілуі де мүмкін:

```
<NOSCRIPT>
```

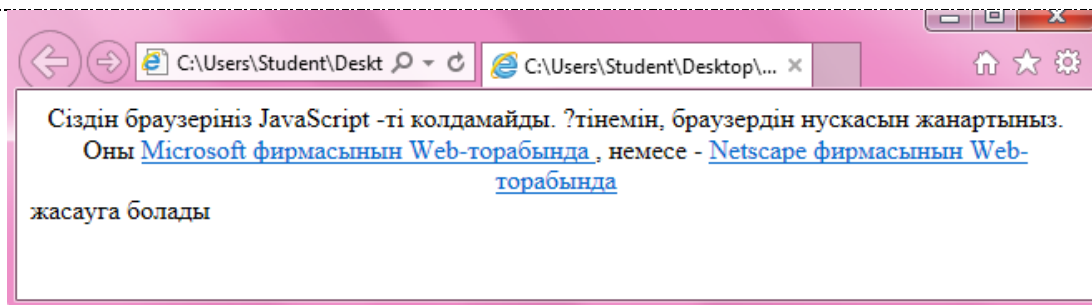
```
<center>
```

Сіздің браузеріңіз JavaScript -ті қолдамайды. Өтінемін, браузердің нұсқасын жаңартыңыз.

Оны [Microsoft](http://www.microsoft.com/) фирмасының Web-торабында [Netscape](http://www.netscape.com/) фирмасының Web-торабында жасауға болады

```
</center>
```

```
</NOSCRIPT>
```



Сурет 1. Мысалдың нәтижесі

### 3) Ақпаратты шығару

JavaScript тілі ақпаратты шығарудың екі тәсілін ұсынады: alert әдісі және write әдісі.

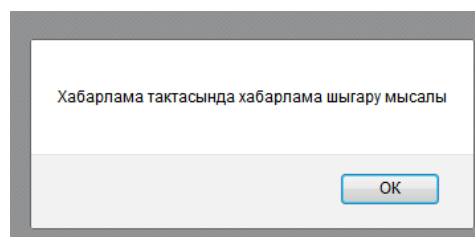
**1-тәсіл: alert әдісі.** Бұл әдіс көмегімен браузер экранында кішігірім хабарламалар шығаруға болады. Олар бөлек сұхбаттық терезеде пайда болады. Ок батырмасын шерту арқылы оны жабамыз.

*Мысалы:*

```
<script language="JavaScript">
```

```
    alert('Хабарлама тақтасында хабарлама шығару мысалы')
```

```
</script>
```



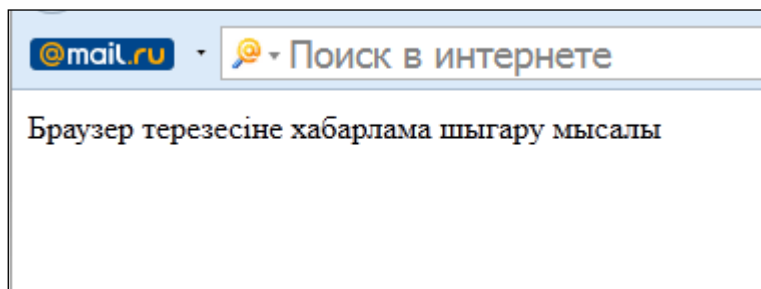
Сурет 2. Хабарлама шығару

## 2-тәсіл: write әдісі

Бұл әдіс *document* объектісімен ұсынылады. Анықталған параметрлері көрсетілген *document.write* әдісін шақырғанда, браузер терезесінде мәтін пайда болады. Бұл әдісті шақырғанда параметр ретінде, біз экранда көргіміз келетін жолды көрсетеміз.

Мысалы:

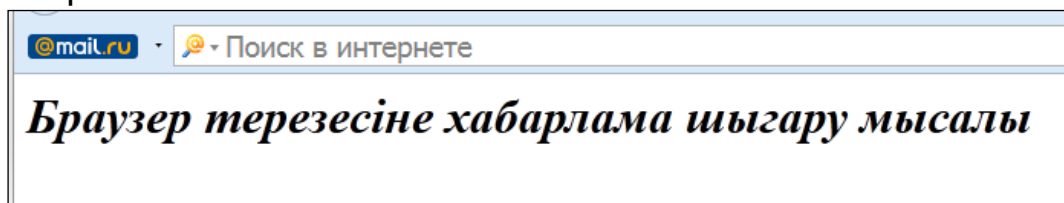
```
<script language="JavaScript">
  document.write('Браузер терезесіне хабарлама шығару мысалы')
</script>
```



Сурет 3. Браузер терезесіне хабарлама шығару

Шығарылатын жол тілінің тегтерін қамти алады. Бұл жағдайда браузер берілген жолды, HTML құжатында орналасқан жол сияқты түрде шығарады. Мысалы,

```
<script language="JavaScript">
  document.write('<H1><B><I>Браузер терезесіне хабарлама шығару
  мысалы</I></B></H1>')
</script>
```



Сурет 4. Браузер терезесіне хабарлама шығару

### Prompt функциясы.

Prompt функциясы ақпарат енгізу үшін қажет. Оның жазылуы:

```
prompt(">жазу", "үндеу бойынша енгізілетін мән");
```

мұнда экранға ішінде екі батырмасы бар терезе шығады. Біз жазба мәліметті енгізу жолына жазамыз да, ОК басамыз. Сонда терезе жоқ болады да, терезеге енгізілген мәліметтер шығады.

Ол мәнді мысалы, айнымалыға меншіктеуге, артынан басқа командаларда пайдалануға болады. Егер Cancel батырмасын шертетін болсақ, онда prompt функциясы арнайы null мәнін қайтарады (бұл "null" сөз тіркесі емес, яғни бос жол (" ") емес, ол мәліметтің арнайы мәні).

### Confirm ("жол").

**Confirm (Расмай)** әдісі пайдаланушының таңдауы қажет болғанда қолданылады.

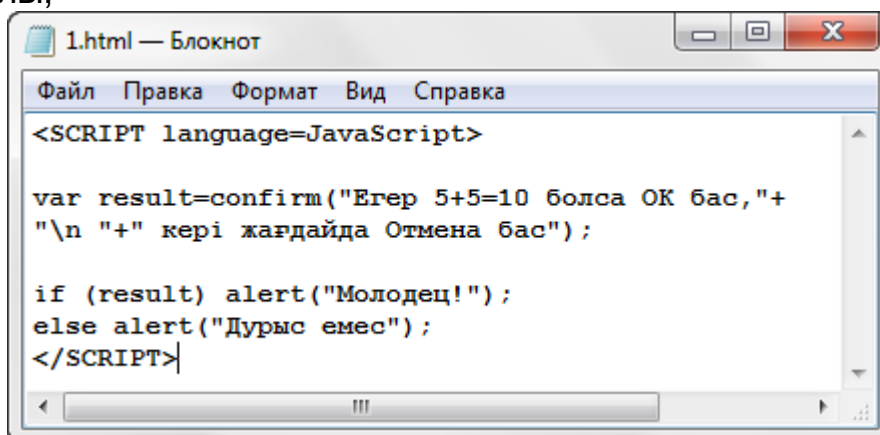
Confirm әдісі пайдаланушыға сұрақ мәтінін және «OK» және «Болдырмау» ларын қамтитын сұхбаттық терезесін көрсетуге мүмкіндік береді.

Confirm функциясы пайдаланушы басқан батырмаға байланысты логикалық мәнді қайтарады:

- «OK» true;
- «Болдырмау» false деп орнатылды.

Әдетте, функцияның нәтижесі мысалда көрсетілгендей әрі талдау үшін логикалық айнымалыға тағайындалады.

Мысалы,

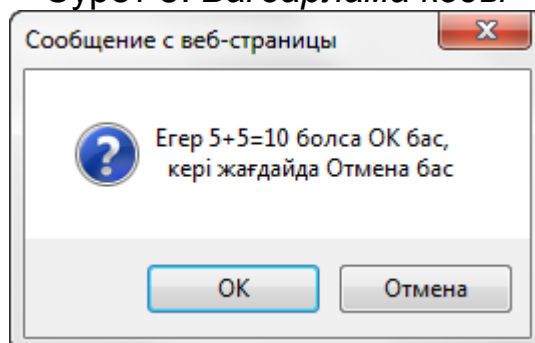


```
1.html — Блокнот
Файл  Правка  Формат  Вид  Справка
<SCRIPT language=JavaScript>

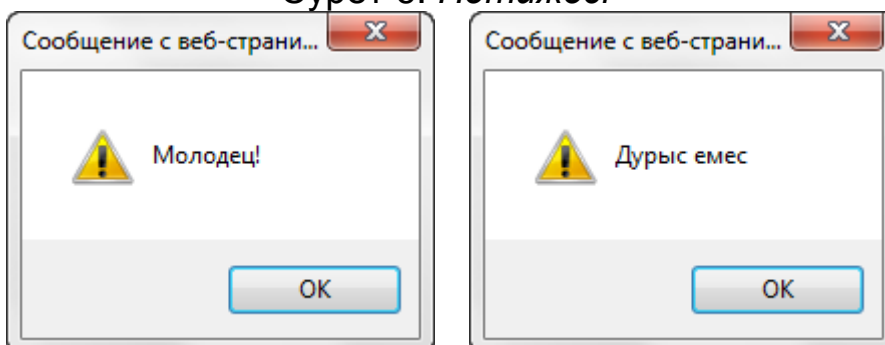
var result=confirm("Егер 5+5=10 болса OK бас, "+
"\n "+" кері жағдайда Отмена бас");

if (result) alert("Молодец!");
else alert("Дурыс емес");
</SCRIPT>
```

Сурет 5. Бағдарлама коды



Сурет 6. Нәтижесі



Сурет 7. Нәтижесі

**Бақылау сұрақтары:**

1. Айнымалылардың қандай типтері бар?
2. JavaScript тегтері қайтегтердің арасында орналасады?
3. <NOSCRIPT> тегінің атқаратын қызметі?
4. Ақпаратты шығарудың қандай тәсілдері бар?

### 1.3 МӘЛІМЕТТЕР ТИПІ. ОПЕРАТОРЛАР, ӨРНЕКТЕР

**Мақсаты:** JavaScript тілінің мәліметтер типімен, операторларымен және тармақталу операторларымен танысу.

**Жоспары:**

1. Мәліметтер типі. Операторлар.
2. Арнайы символдарды кірістіру
3. Тармақталу операторлары.

#### 1) Мәліметтер типі. Операторлар.

Кейбір JavaScript тілін сүйемелдейтін браузерлердің бұрынғы нұсқалары оларды орындамайды. Ондай браузерлер скриптерді түсінбейтән болғандықтан, JavaScript тілінің операторларын комментарий тегтерінің `<!--...-->` ішіне орналастыру қалыптасқан. Скриптердегі қолданылатын тілді анықтайтын параметрлердің бірі `<Script>` тегінің `language` атрибуты болып табылады.

JavaScript-тегі арифметикалық операторлар мынадай:

Кесте 1. Арифметикалық операторлар

Оператор	Атауы
+	Қосу
-	Азайту
*	Көбейту
/	Бөлу
%	бүтінсанды бөліндіден қалған қалдық
Икрементті және декрементті операторлар	
++	Операнд мәнін бірге өсіру
--	Операнд мәнін бірге кеміту

Меншіктеу операторы мынадай:

`"=", "+=", "-=", "*=", "/=", "%="`

**Икрементті және декрементті операторларының жазылуы**

`"x++", "++x", "x--", "--x"`

#### 1. Ақпарат енгізу

Alert функциясы ақпаратты экранға шығару үшін керек. Ал ақпарат енгізу үшін prompt функциясы қажет:

`prompt(">жазу", "келісім бойынша енгізілетін мән");`

мұнда экранға ішінде екі батырмасы бар терезе шығады. Біз жазба мәліметті енгізу жолына жазамыз да, ОК басамыз. Сонда терезе жоқ болады да, терезеге енгізілген мәліметтерметтер шығады.

Ол мәнді мысалы, айнымалыға меншіктеуге, артынан басқа командаларда пайдалануға болады. Егер Cancel батырмасын шертетін болсақ, онда prompt функциясы арнайы null мәнін қайтарады (бұл "null" сөз тіркесі емес, яғни бос жол (" ") емес, ол мәліметтің арнайы мәні).

#### 2. Блок

Жүйелік жақшаға алынған командалар тізбегі {1-команда; 2-команда;..} бір команда секілді орындалады. Мұндай күрделі команда құрама немесе блок деп аталады. Жақшадан «}» кейін «;» символы қойылмайды.

### 3. Балама мән енгізу

Егер екі балама мәннің бірін таңдап алу керек болса, онда prompt функциясын пайдалану керек.

Бірақ бұдан гөрі қарапайым әрі жеңіл тәсіл бар, ол- confirm функциясын пайдалану:

*Confirm*("жазу");

Егер ОК батырмасын шертсек, confirm функциясы true мәнін қайтарады, ал егер Отмена батырмасы шертілсе, False мәні қайтарылады.

Олардың орнына соларға парапар<Enter> және<Esc> пернелерін пайдалануға болады.

### Өрнектер:

y=x++ мына екі меншіктеулерге эквивалентті: y=x; y=y+1,

y=++x мына екі меншіктеулерге эквивалентті: x=x+1; y=x,

y=x-- мына екі меншіктеулерге эквивалентті: y=x; x=x-1,

y=--x мына екі меншіктеулерге эквивалентті: x=x-1; y=x.

Мысалы:

1) var i = 2;

  i++; // i = i + 1 қысқа жазба үшін

  alert(i); // 3

2) alert(5 % 2); // 1, (5-тің 2-ге бөліндісінің қалдығы)

## 2) Арнайы символдарды кірістіру

Біз Alert() ішінде «<br>» (және басқа HTML тегтерін) пайдалана алмаймыз, бірақ JS-те мәтінді жаңа жолға аударуға мүмкіндік беретін арнайы \n символы бар. Мысалы:

*alert('Жолды \n ауыстыратын \n хабарлама');*

Төмендегі кестеде JavaScript-те бар арнайы символдар бар.

*Кесте 2. Арнайы символдар*

Арнайы символ коды	Мағынасы
\n	Жолды ауыстыру
\\	Кері слэш
\t	Табуляция
\'	Бір тырнақша
\"	Қос тырнақша
\&	Амперсанд

Мысалы:

*alert('Амперсанд \& кері слэш \\және бір тырнақшаны \'*  
*қамтитын хабарлама');*



## 2) Тармақталу операторлары

### ➤ IF шартты операторы.

If операторын берілген шарт ақиқат болғанда ғана командаларды орындау үшін қолданылады:

```
if(шарт)
{
    тұжырым
}
```

Мысалы:

```
var a=7;
var b=7;
if (a==b) {
    document.write("Сандар тең");
}
```

### ➤ IF..ELSE операторы.

Бұл операторды берілген шарт ақиқат, ал екіншісі жалған болғанда командаларды орындау үшін қолданылады:

```
if(шарт)
{
    тұжырым1
}
else
{
    тұжырым2
}
```

Мысалы:

```
var a=9;
var b=7;
if (a==b) {
    document.write("Сандар тең");
}
else {
    document.write('Сандар тең емес');
}
```

### **Бақылау сұрақтары:**

1. JavaScript-те қандай арифметикалық әрекеттер операторлары бар?
2. Меншіктеу операторы қандай?
3. Шартты операторлардың қандай түрлері бар?

## 1.4 ЦИКЛДІК ОПЕРАТОРЛАР

**Мақсаты:** JavaScript тілінің циклдік операторларымен танысу

**Жоспары:**

1. Циклдік операторлар.

## 2. Циклдардың орындалуын үзу (тоқтату, кідірту) операторлары.

### 1) Циклдік операторлар

#### ➤ FOR циклі.

Белгілі бір әрекеттердің қайталанып орындалуын цикл операторларының көмегімен ұйымдастыруға болатындығы бізге белгілі. JavaScript бағдарламалау тілінде циклді ұйымдастырудың екі түрі бар:

1. For циклі.

2. While циклі.

For циклінің жазылуы:

```
for(атау; шарт; счетчиктердің өзгеру секциялары)
{
    тұжырым
}
```

#### ➤ WHILE циклі.

Циклдің орындалуы: алдымен шарт тексеріледі. Егер ол ақиқат болса, командалар орындалады. Келесі жолы да осы әрекеттер қайталанады, яғни шарт тексеріледі, егер ол ақиқат болса, цикл орындалады, т.с.с. Кезекті тексеру кезінде шарт жалған болған кезде, цикл жұмысы аяқталады. Циклде шарт алдын ала тексерілетін болғандықтан, ол бірде бір рет орындалмауы да мүмкін.

While циклінің жазылуы:

```
while(шарт)
{
    Өрнек
}
```

Мысалы,

```
Var i = 1;
var sum = 0; while(i<= 100)
{
    sum += i;
    i ++;
}
alert("Косынды1 + 2 + ... + 100=" + sum) ;
```

#### ➤ DO ... WHILE циклі.

Do...while – бұл while циклінің басқа түрі. Бұл цикл блок кодын *бір рет* орындайды, және берілген шарт ақиқат болғанша циклді қайталап отырады.

Жазылуы:

```
do
{
    орындалатын код
}
while (айнымалы<=соңғы мән);
```

## 2) Циклдардың орындалуын үзу (тоқтату, кідірту) операторлары

*Break* және *continue* командалары цикл командаларының орындалу ретін өзгерту үшін қолданылады. **Continue** командасы циклдің одан кейінгі тұрған барлық командаларын аттап өтіп, цикл параметрінің келесі мәніне көшіреді. Ал **break** командасы жалпы цикл орындалуын аяқтап, одан кейінгі келесі командаларға көшіреді.

For немесе while ағымды циклдарының операторларын тоқтату үшін **break** операторы қызмет етеді.

**Continue** операторы for немесе while-дің ішіндегі ағымды итерациясының орындалуын тоқтату үшін және келесі итерацияның басына өту үшін қолданылады.

**Бақылау сұрақтары:**

1. Циклдердің қандай түрлері бар?
2. Continue командасы
3. break командасы

### 1.5 ЖОЛДЫҚ АМАЛДАР. ФУНКЦИЯЛАР

**Мақсаты:** JavaScript тіліндегі жолдық амалдармен және функциялармен танысу

**Жоспары:**

1. Жолдық амалдар.
2. Функциялар.

#### 1) Жолдық амалдар

Жолдармен жұмыс істейтін мынандай амалдар бар:

➤ "+" - s1+s2 жолдарын қосу (конкатенация), s1 жолының символдарынан кейін s2 жолының символдары жазылады.

➤ **eval(s)** – JavaScript-тің кірістірілген функциясы. Ол JavaScript-тің бір немесе бірнеше операторларды қамтитын s жолымен берілген кодты орындайды. Бұл функцияны оператордың орындалуы үшін ғана емес, өрнекті есептеу үшін ғана емес, өрнекті есептеу үшін де қолдануға болады. Ол берілген кодтағы соңғы есептелген өрнектің мәнін қайтарады.

➤ **parseFloat(s)** – JavaScript-тің кірістірілген функциясы. S жолында қамтылатын, санға кірмейтін жолдың басынан бірінші символына дейінгі заттық санын табады (Float типті). Егер сан табылмаса, NaN ("Not a Number") мәнін береді.

➤ **parseInt(s)** – бүтін сандарға арналған (Integer). Негізі автоматты түрде табылады.

➤ **parseInt(s,n)** –n негізі бойынша бүтін сандар үшін арналған (2 -ден 36-дейін).

## 2) Функциялар

Программа құрғанда ондағы логикалық тәуелсіз бөліктерді жеке-жеке бөліп қарауға болады. Программаны мұндай бөліктерге бөліп қарастыру оның жұмысын түсінуді жеңілдетеді.

**Функция** - JavaScript тілінің негізгі элементі. Функция мынадай түрде сипатталады: `function F(v) {s}`

Мұндағы *F* – функцияны шақыруға болатын ат тағайындайтын функция идентификаторы; *V* - үтір арқылы бөлініп жазылатын функция параметрлерінің тізімі. *S* - нәтиже алу үшін орындалатын іс-әрекеттерден тұратын функция тұлғасы, яғни операторлар тізбегі.

Ол мыналардан құралады:

- function сөзінен;
- функция атауынан;
- жай жақшаға алынған, үтір арқылы бөлінген, аргументтер тізімінен;
- фигуралық жақшаға алынған функция денесінен.

Жазылуы:

```
function myFunction(arg1, arg2, ...)
{
  ...
  Операторлар тізбегі
  ...
}
```

Мұндағы, *myFunction* — функция атауы, *arg1, arg2* — параметрлер тізімі

Мысалы:

```
function Factorial(n)
{
  if((n<0)||((round(n)!=n)))
  {
    alert("Factorial функциясы келесі аргументте анықталмаған"+n);
    return NaN;
  }
  else
  {
    result=(n*Factorial(n-1));
    return result;
  }
}
```

**Мысалдар:**

1. `function showTime() {`  
// функция экранға уақытты шығарады}
2. `function do_Calc() {`  
// функция есептеулерді орындайды}
3. `function askUser() {`  
// функция пайдаланушыны сұратады}

```
4. function randColor() {  
  // функция кездесок түстерді шығарады}
```

#### **Бақылау сұрақтары:**

1. Жолдармен жұмыс істейтін қандай амалдар бар?
2. Функция деген не?
3. Ол қалай жазылады?

### **1.6 JAVASCRIPT ТІЛІНІҢ ОБЪЕКТІЛЕРІ**

**Мақсаты:** JavaScript тілінің құрамындағы объектілермен танысу.

#### **Жоспары:**

1. Тілдің құрамындағы объектілер. Олардың қасиеттері мен тәсілдерін пайдалану. *Date* объектісі.
2. *Array* объектісі.
3. *Math* объектісі.
4. *Window* объектісі.
5. *Document* класы.

#### **1)Тілдің құрамындағы объектілер. Олардың қасиеттері мен тәсілдерін пайдалану. *Date* объектісі**

Объект — бұл мәліметтер мен функциялар жиынынан тұратын бірыңғай конструкция немесе, JavaScript терминологиясында, *қасиеттер* мен *тәсілдер* жиыны.

JavaScript тілінің ішкі объектілерін қарастыруды өте пайдалы болып саналатын *Date* объектісінен бастайық. Бұл объект күн-ай мерзімімен (дата) және уақытпен жұмыс істеу үшін керек.

Объект экземплярын жасау үшін (*Date* объектісінің ғана емес, одан басқасының да) JavaScript тілінде *new* түйінді сөзі қолданылады:

```
var now = new Date();
```

Енді *now* айнымалысы *Date* объектісі экземпляры болып табылады да, ол үстіміздегі дата мен уақытты береді.

Жалпы экземпляр жасау былай орындалады:

```
var айнымалы = new Date(параметрлер);
```

Мысал:

```
var now = new Date();
```

```
var birthday =
```

```
  new Date(1954, 1, 8);
```

```
var bell = new Date(2003, 0, 14, 12, 20, 0);
```

*Date* объектісі экземпляры құрылғаннан кейін, оның ішкі мәліметтерін көруге болады, оны өзгерту мүмкіндігі де бар. Ол үшін көптеген тәсілдер бар, олардың тізімі тілге арналған кітап қосымшаларында келтіріледі.

Объект тәсілі атауы (тәсіл – JavaScript терминологиясында функция) экземпляр атауынан нүктемен бөлініп жазылып тұрады.

Былай жазуға болады:

```
var year = bell.getYear();
```

*year* айнымалысы мәні 2003 болады.

Date объектісінің бірнеше қарапайым скриптерін қарастырайық.

### **Ағымдағы күн және уақыт**

3) `var now = new Date();`

```
alert("Бүгін:" + now.getDate() + "/" + (now.getMonth() + 1) + "/" +  
now.getFullYear() + "\n
```

Қазір:

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

Осы кодтарды жазған кезде скрипті орындау мынадай хабарлама шығуына себепші болады.

*Date ішкі объектісі және оның тәсілдері*

3) Тәсіл түрі – `setYear()`. Жыл нөмірін тағайындайды.

Мысалы:

```
var d = new Date();
```

```
d.setYear(2004);
```

```
alert(d.getFullYear());
```

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

```
var d = new Date(2005,1,2);
```

```
var m = d.getMonth();
```

```
alert(m);
```

Объект — бұл мәліметтер мен функциялар жиынынан тұратын бірыңғай конструкция немесе, JavaScript терминологиясында, *қасиеттер* мен *тәсілдер* жиыны.

*Функция* = тәсіл (метод).

*Айнымалы* = қасиет (свойства).

Инкапсуляция термині «қара жәшік» ретінде қарастырылатын объектінің ішкі құрылымын жасыру деген сөз. Объектінің қасиеттері белгілі болып саналады, яғни олар – сырттан қол жеткізуге болатын айнымалылар. Бірақ бұл функциялар қалай құрылған, олар қандай алгоритммен жұмыс істейді, ол туралы программалаушыға айтылмайды. Программалаушы немесе объектіні тұтынушы адам объектінің қосымша ішкі функциялары мен айнымалылары бар ма, олар қол жеткізуге болатын қасиеттер мен тәсілдермен қалай байланысқан, ол жағын білмейді.

JavaScript жаңа объектілерді программалауға және олардың ішкі объектілерін өзгертуге мүмкіндік береді. Енді бірнеше ішкі объектілер жұмысын қарастырайық.

JavaScript тілінің ішкі объектілерін қарастыруды өте пайдалы болып саналатын *Date* объектісінен бастайық. Бұл объект күн-ай мерзімімен (датамен) және уақытпен жұмыс істеу үшін керек.

Объект экземплярын жасау үшін (*Date* объектісінің ғана емес, одан басқасының да) JavaScript тілінде *new* түйінді сөзі қолданылады:

```
var now = new Date();
```

Енді *now* айнымалысы *Date* объектісі экземпляры болып табылады

## 2) Array объектісі

Бұл объект мәліметтер жиымын (массивін) жасау үшін керек. *Массив* — элементтердің реттелген жиыны. Жеке элементті көрсету оның атауы мен индексін (нөмір) көрсету арқылы орындалады. JavaScript тілінде элементтерді нөмірлеу нөлден басталады.

Массивті үш түрлі тәсілмен құруға болады:

1-тәсіл.

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

2-тәсіл.

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

3-тәсіл.

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

Жиымның жеке элементтерін пайдалану үшін былай жазылады:

*массив\_атауы [индекс]*

Төмендегі скрипт:

```
var dayNames = new Array(«жексенбі», «дүйсенбі», «сейсенбі»,  
«сәрсенбі», «бейсенбі», «жұма», «сенбі»); alert(dayNames[0]); жұмысы  
нәтижесінде alert терезесіне «жексенбі» мәтіні шығады.
```

*Кесте 3. Array объектісінің әдістері:*

<b>Array әдісі</b>	<b>Сипаттамасы</b>
concat	Екі жиымды бір жиымға біріктіреді. Мысалы: a = new Array(0,1,2,3,4); //дайын мәндері бар b = new Array(5,6,7,8,9); //жиым құрады c = a.concat(b);
join	Жиымның барлық элементтерін бір жолға біріктіреді. Мысалы: a = new Array(0,1,2,3,4); b = a.join("-"); // "0-1-2-3-4" жолдарын қайтарады.
pop	Соңғы элементті жиымнан жояды және осы элементті қайтарады.
push	Жиым соңына бір не одан көп элементті қосады және соңғы қосылған элементті қайтарады.
reverse	Жиымды нөлдік элемент соңғы болатындай және т.б етіп төңкереді.
shift	Жиымның нөлдік элементін жояды және оны қайтарады.

slice	Жиымның бір бөлігін ерекшелейді. Синтаксисі: <i>slice(start, [end])</i> . Мысалы: <code>newArray = myArray.slice(0, -1);</code>
splice	Жиымға элементтер қосады және ондағыны жояды.
sort	Жиым элементтерін сұрыптайды.
toString	Жиымның жолдық берілу түрін қайтарады.
unshift	Жиымның бас жағына бір не бірнеше элементтерді қосады.

### 3) Math объектісі

Math математикалық классы — математик-программисттерге тригонометриялық және логарифмдік функциялар жиынын береді. Яғни ол тұрақтылар мен әдістерді қамтитын класс. Олардың шақырылу түрі:

*Math.константа*

*Math.функция(i..)*

Кесте 4. Math класының тұрақтылары:

Тұрақтылар	Түсініктемесі
E	E саны (натурал логарифмдерге негізделген)
LN10	Натурал логарифм (ln10)
LN2	Натурал логарифм (ln2)
LOG10E	10 негізді е логарифмі (log10e)
LOG2E	2 негізді е логарифмі (log2e)
PI	P тұрақтысы ("пи" саны)
SQRT1_2	0,5-тің квадрат түбірін есептейді (1/2 саны)
SQRT2	2-нің квадрат түбірін есептейді

### 4) Window объектісі

**Window** иерархияның төбесінде орналасады. Бұл қалған объектілердің одан төмен орналасып, соның қасиеттері түрінде қарастырылатынын білдіреді. Window объектісінде браузер құжатты көрсетіп тұрған терезенің қасиеттері мен тәсілдері де болады. Үш тәсіл бізге бұған дейін белгілі болған, олар: alert, prompt және confirm.

Window иерархияның төбесінде орналасады. Бұл қалған объектілердің одан төмен орналасып, соның қасиеттері түрінде қарастырылатынын білдіреді. Window объектісінде браузер құжатты көрсетіп тұрған терезенің қасиеттері мен тәсілдері де болады.

Үш тәсіл бізге бұған дейін белгілі болған, олар:

alert, prompt және confirm.

Егер былай жазылса:

*Window.alert ("alert функциясы – бұл window объектісінің әдісі");*

— онда браузер осы хабарламаны шығаратын қосалқы терезе ашады (суретті қ.)

Бірақ бұрын былай жазатынбыз:



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

Басты объект болып саналатын window объектісінің атауын жазбаса да болады.

Көбінесе тұтынушыға бір нәрсе көрсету үшін бұрынғы терезе ашық тұрғанда, жаңа терезе ашуға тура келеді. Жаңа терезе ашу үшін **window** объектісінің **open** тәсілін қолданамыз. Оны мына тәсілдердің бірімен аша аламыз:

*var айнымалы = open();*

*var айнымалы = open(файл);*

*var айнымалы = open(файл,терезе\_атауы);*

*var айнымалы = open(файл, терезе\_атауы, терезе\_параметрлері);*

Мұндағы:

– айнымалы — айнымалы. Ашылатын терезеге нұсқауыш;

– файл — сөз тіркесі (строка). Ашылған терезеде көрсетілетін файл аты. Егер бұл параметр берілмесе, бос терезе ашылады;

– терезе\_атауы — сөз тіркесі. Терезе аты сол терезеге мәлімет шығару үшін керек (window объектісінің name қасиеті мәні);

– терезе\_параметрлері — сөз тіркесі. Терезе қасиеттерін сипаттау. Егер параметр берілмесе, келісім бойынша қасиеттер қолданылады.

Параметрлер арасындағы үтірден соң, бос орын қоймаған дұрыс, браузерлер онсыз да жұмыс істей береді.

*Кесте 5. Терезе параметрлерін сипаттау*

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

*Мысал.* Ит суреті бар dog01.jpg терезе ашу.

<HTML>

<HEAD>

<TITLE>терезе</TITLE>

</HEAD>

<BODY bgcolor=white text=black>

```

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

```

Терезеде көрсетілетін файл ретінде мыналарды көрсете аламыз:

- HTML-файл, мысалы, file1.htm;
- сурет файлы, мысалы, ./pic/fish.gif;
- мәтіндік файл, мысалы, 010401.txt.

Close тәсілі ашық терезені жабады. Мұндағы close() немесе window.close() ағымдағы терезені жабады. Ал win.close() немесе window.win.close() — осы терезеден open тәсілі арқылы ашылған win нұсқауышы бар терезені жабады:

```
var win = window.open(...);
```

Терезенің өлшемі мен орнын анықтау.

➤ **moveBy(X,Y)** – терезені ағымдағы терезе орнына қарағанда санақ нүктесі сол жақ жоғарғы бұрыштан X шамаға (көлденеңінен, пикселмен) және Y шамаға (тігінен) жылжытады.

➤ **moveTo(X,Y)** – терезені санақ нүктесі сол жақ жоғарғы бұрыштан (X,Y) нүктесіне жылжытады.

➤ **resizeBy(X,Y)** – терезе өлшемін оның ағымдағы өлшеміне қарағанда X шамаға (көлденеңінен, пикселмен) және Y шамаға (тігінен) өзгертеді.

➤ **resizeTo(X,Y)** – терезенің ені X (пикселмен) және биіктігі Y болатындай абсолюттік өлшемін береді.

➤ **scroll(X, Y)** – терезенің жүйелік аумақтағы деректерін (X,Y) нүктесіне ауыстырады, яғни айналдыру жолақтарын X пен Y-ке ығыстырады.

➤ **scrollBy(X,Y)** - терезенің жүйелік аумақтағы деректерін ағымдағы орнына қарағанда көлденеңінен X-пикселге және тігінен Y-пикселге айналдырады.

➤ **scrollTo(X,Y)** - терезенің жүйелік аумақтағы деректерін (X,Y) нүктесіне айналдырады.

## 5) Document класы

**Document объектісі** (ол document объектісі қасиеттері болып табылады) <html>...</html> блоктарында орналасқан тегтерді

модельдейтін объектілер үшін қажет. Document объектісінің кең тараған тәсілі — write тәсілі. Бұл тәсіл құжат терезесіне мәлімет жазу үшін керек. Write тәсілін қолдану алдында жазба ашу керек (open тәсілі), ал шығарған соң, оны жабу керек (close тәсілі).

Open тәсілі форматы: *айнымалы.document.open()*;

Мұндағы: айнымалы — бұл мәлімет жазылатын терезеге нұсқауыш.

Write тәсілі форматы:

*document.write(жол)*; немесе *айнымалы.document.write(жол)*;

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

*document.close()*; немесе *айнымалы.document.close()*; айнымалы — жазба шығарылмай жабылатын терезеге нұсқауыш.

*document.open* тәсілін қолдану міндетті емес, өйткені *document.write* тәсілі жабылған құжатқа мәлімет жазу кезінде оның ішіндегі ескі мәліметті өшіріп, оны жаңа жазба үшін ашады.

Window объектісінің document өрісіндегі Document класының бір түрі.

Шақырылуы:

*терезе атауы.document* немесе жай *document* (ағымды терезе үшін)

*Кесте 6. Window объектісінің әдістері:*

<b>Әдістер</b>	<b>Мағынасы</b>
document.clear()	W атаулы терезені немесе фреймді тазалау.
document.close()	Буфердан қалған ақпаратты экранға шығарады және құжаттан шығу жолын жабады.
document.open ()	write(...) операторы көмегімен экранға мәтін жазу үшін, құжатқа кіру жолын ашады.
document.writeln(...)	Құжатқа қалған мәндерді жазу.
document.write(1 мән,2мән,...)	write(...) ұқсас, бірақ жаңа жолға өту символын кірістіру арқылы.

**Бақылау сұрақтары:**

1. JavaScript-тің қандай объектілері бар?
2. Date ішкі объектісі және оның тәсілдері?
3. Array объектісінің қызметі?
4. Math математикалық класының қызметі?
5. Document объектісі қызметі?
6. Жаңа терезе ашу үшін қандай объект қолданылады?
7. Терезені сипаттау параметрлері?

## 1.7 JAVASCRIPT ТІЛІНІҢ ОҚИҒАЛАРДЫ ӨНДЕУ МҮМКІНДІКТЕРІН ПАЙДАЛАНУ

**Мақсаты:** JavaScript тілінің оқиғаларды өңдеу мүмкіндіктерімен танысу  
**Жоспары:**

1. JavaScript тілінің оқиғаларды өңдеу мүмкіндіктері.
2. Тышқанды шерту, курсорды қозғалту, пернелерді басу әрекеттерін бағдарламалау.

### 1) JavaScript тілінің оқиғаларды өңдеу мүмкіндіктері

Ең алдымен Web-бағдарламалау тілінің қайсысы екендігіне қарамастан **оқиғалар** ұғымының не екендігін анықтайық. Ол компьютерде жазба жазып отыр. Бұл пайдаланушының компьютерде атқарған жұмысының бәрі, машинаның (компьютердің) көзімен қарағанда – оқиға болып табылады. Пайдаланушының пернетақтадағы пернені басуы, тышқанды қозғауы, жаңа терезе ашып, оны жабуы, сілтемемен басқа параққа ауысуы, форманың өрісін толтыруы, сілтемеге тышқанды апаруы – осының барлығы оқиға болып табылады.

JavaScript осы оқиғаларға жауап қайтаруды бағдарламалауға, сөйтіп парақшаны жандандыруға мүмкіндік жасайтын икемді құралдар ұсынады. Әр оқиғаға белгілі бір әрекетті меншіктеуге мүмкіндік беретін өңдеуіш болады.

Алдымен HTML-құжатқа JavaScript-терді және оның функциясы – alert-ті қалай қоюды қарастырайық, бұл бізге оқиға өңдеуіштің мысалы үшін ғана қажет болып тұр.

Мысалы:

```
<script language=javascript>  
    alert("Сіз басқа батырманы бастыңыз!!!");  
</script>
```

Формада орындалған бір әрекетке (оқиғаға) жауап ретінде JavaScript сценарийі орындалады. Оқиға – бұл тұтынушының формамен атқаратын белгілі бір әрекеті, мысалы, батырманы шерту, екі батырманың бірін таңдау немесе тышқан курсорын форма элементінен тысқа шығару. Браузер сервердегі web-парақты оқып бітірген кезді де оқиға болды деп санайды. Осындай оқиғалардың бірі болса, JavaScript сценарийі жұмысқа кірісе алады.

Форма элементі үшін оқиға ашылу тегінің атрибуты түрінде жазылады. Атрибутта оқиға жүзеге асқанда орындалуы тиіс функция атауы да көрсетілуі мүмкін.

Тұтынушы тышқанды шерткенде, оқиға болды деп саналады. Көбінесе тұтынушы HTML формаларындағы элементтерге тиіскенде, белгілі бір оқиға жүзеге асады, сондықтан оқиғаны өңдеу форма элементтері параметрлерінде көрсетіледі. Оқиғаны өңдеу параметрінің атауы **on** әріптерінен басталады да, одан кейін оқиға атауы жазылды.

Мысалы, Click оқиғасын өңдеу ісі onClick болып жазылады, onClick командасы әрекетін көрсету үшін, alert.тәсілін қолданайық. Келесі код берілген:

```
<A HREF="http://www.mail.ru" onClick="alert('Поштаны қарау!');">Шерту</A>
```

Сілтемені шерткенде, "Поштаны қарау!" деген сөздер жазылған ақпараттық терезе шығады.

## 2) Тышқанды шерту, курсорды қозғалту, пернелерді басу әрекеттерін бағдарламалау

Ал енді оқиғаны қарастырайық:

1. **onmouseover** – тышқанның меңзері объектіні көрсеткен сайын пайда болады. Мұнда объект ретінде сілтеме де, батырма да, форма өрісі де бола алады. Оқиға өңдеуші onmouseover JavaScript-тің бағдарламаларын орындайды. Мысалы:

```
<a href="#" onmouseover="alert('Сіз тышқанды сілтемеге қойдыңыз');">Сілтеме</a>
```

Оқиға өңдеушінің жазылуында тег атрибуттарымен бірге, оның JavaScript екенін көрсетпей-ақ жазуға болады екен. Онда тек JavaScript-тің тағы бір ерекшелігін: барлық функциялар мен операторлардың барлығы нүктелі үтірмен аяқталуы тиіс екендігін ұмытпау керек. Міне осымен бірге біз alert функциясының не істейтінін де білдік.

onMouseOver (тышқан курсорын алып бару) өңдеушіні қарастырайық.

```
<A HREF=http://www.mail.ru onMouseOver="window.status='Пошта қызметі';return true ">Сілтеме</A>
```

onMouseOver гипермәтін сілтемесін өңдейді. Ол гиперсілтеме ішінде пайдаланылады. Гипермәтіндік сілтеме форматы өзгермейді. onMouseOver оқиға өңдеуші бірден URL адресінен соң қойылады. Мұнда объект рөлін терезе (window) атқарады, ал терезе қасиеті (property) болып status есептеледі.

Браузер onMouseOver=" " сөзін кездестірген кезде, оқиға (event) орындалады.

Терезенің қасиеті болған соң, басқа объектілердің де қасиеті болуы тиіс. Енді тағы бір қасиетті – фон түсін өзгертейік. HTML тілінде парақтың фон түсін BGCOLOR командасы басқарады. Сондықтан фон түсін өзгерту үшін status сөзін bgColor сөзіне алмастырамыз. Мұнда мәтін жазу керек емес, сол себепті оны түске алмастырамыз, қызғылт (pink) түсті алайық. Нәтижесі мынадай болады ...

```
<A HREF="http://www.mail.ru" onMouseOver="document.bgColor='pink'; return true">Шерту</A>
```

Қос тырнақша ішіне мәтін тәрізді жеке элементтер орналастырылады. Егер екі команданы қатар бір команда тәрізді орындағымыз келсе, онда тырнақшаны бірінші команданың алдына және екінші команданың соңына қою керек. Осындай тәсілмен бұл екеуінің бір оқиға екені білдіреді. Бізге әлі жалғыз тырнақшалар да (апострофтар) керек болады.

Оқиға өңдеуіштерді қарастырайық.

Кесте 7. Оқиға өңдеуіштері

Оқиға	Сипаттамасы
onload	ашу (құжатты жүктеу). Әдетте <body> тегінде қолданылады. Мысалы: <body onload="alert('Парақ жүктелді');">
onunload	сіз парақтан шықтыңыз. Мысалы: <body onunload="alert('Кіргеніңізге рахмет!!!');">
onclick	пайдаланушы объектіні шертті
ondblclick	пайдаланушы объектіні екі рет шертті
onmouseover	тышқанның меңзері объектіні көрсеткен сайын пайда болады.
onmousemove	пайдаланушы тышқанды экран бетімен қозғайды
onmouseout	пайдаланушы объектіден тышқан тінтуірін алып кетті
onfocus	өріс фокусты пернетақтадан енгізу немесе тышқанмен шерту арқылы алған кезде пайда болады;
onblur	форманың select, text немесе textarea өрістері фокустарын жоғалтқан кезде пайда болады;
onsubmit	пайдаланушы форманы Web-серверге жөнелткен кезде пайда болады.
onselect	пайдаланушы кейбір мәтінді text немесе textarea өрісінің ішінен таңдаған кезде пайда болады;
onchange	форманың select, text немесе textarea өрістері фокустарын жоғалтқан кезде немесе мәндерін өзгерткен кезде пайда болады.

Сонымен біз жалпы мақсаттағы көбірек қолданылатын оқиғаларды қарастырдық. Көптеген оқиғалар нақты объектілерге бекітіледі. Атап айтқанда, жоғарыда айтылған оқиғалар форма объектілеріне жатады.

**Бақылау сұрақтары:**

1. JavaScript-тің alert функциясының қызметі?
2. Click оқиғасының қызметі?
3. Тышқанды шерту, курсорды қозғалту, пернелерді басудың қандай оқиғалары бар?

## 2 ПРАКТИКАЛЫҚ ЖҰМЫСТАР

### 2.1 Практикалық жұмыс №1 НЕГІЗГІ ОПЕРАТОРЛАР

**Жұмыстың мақсаты:** JavaScript тілінде арифметикалық операторларды қолданып бағдарлама құрып үйрену.

#### Қысқаша мәліметтер:

##### 1.1.Құжаттың құрылымы

JavaScript скриптінің коды HTML-бетте орналасады. Скрипт екі — `<SCRIPT>...</SCRIPT>` тегтерінің арасында орналасады. Скриптік бағдарламаларды қолдамайтын браузерлер, оларды тастап кетуі үшін, оларды комментарийлер блоктарының ішінде орналастыру керек. Оның жалпы жазылу түрі мынандай:

```
<SCRIPT LANGUAGE="JavaScript">
  <!--
  ...
  JavaScript бағдарламасы
  ...
  //-->
</SCRIPT>
```

Мұндағы: Language атрибуты берілген бағдарлама қай тілде жазылғанын көрсетеді. Комментарийлер `<!--...-->` жақшаларға алынады, Html-комментарийдің тегі жабылуы үшін `"/"` екі қиғаш сызықшалар қолданылады.

**Құжаттың жалпы құрылымы:** JavaScript тілінде жазылған бағдарлама құрылымының функциялары HTML-құжатының `<Head>` бөлімінде орналасады. Html-құжатының мысалы:

```
<HTML>
<HEAD>
  <TITLE> JavaScript тіліндегі бағдарлама мысалы</TITLE>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
    ...
    JavaScript бағдарламасы
    ...
    //-->
  </SCRIPT>
</HEAD>
<BODY>
  ...
  HTML-құжатының мәтіні және и JavaScript-те функцияны шақыру
  ...
</BODY>
</HTML>
```

## 1.2. Ақпаратты шығару.

JavaScript тілі ақпаратты шығарудың екі тәсілін ұсынады: alert әдісі және write әдісі.

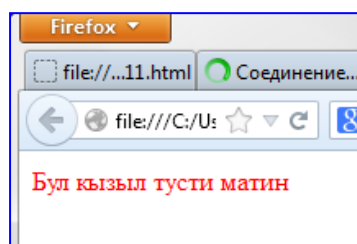
**1-тәсіл: alert әдісі.** Бұл әдіс көмегімен браузер экранында кішігірім хабарламалар шығаруға болады. Олар бөлек сұхбаттық терезеде пайда болады.

### **2-тәсіл: write әдісі**

Бұл әдіс *document* объектісімен ұсынылады. Анықталған параметрлері көрсетілген *document.write* әдісін шақырғанда, браузер терезесінде мәтін пайда болады.

#### **Мысал-1.**

```
<html>
<body>
<SCRIPT LANGUAGE="javascript">
document.write("<FONT COLOR='RED'> Бұл қызыл түсті
мәтін</FONT>")
</SCRIPT>
</body>
</html>
```



Сурет 8. Бағдарлама нәтижесі

**3-тәсіл: Prompt функциясы** ақпарат енгізу үшін қажет. Оның жазылуы:

```
prompt(">жазу", "үндеу бойынша енгізілетін мән");
```

мұнда экранға ішінде екі батырмасы бар терезе шығады. Біз жазба мәліметті енгізу жолына жазамыз да, ОК басамыз. Сонда терезе жоқ болады да, терезеге енгізілген мәліметтер шығады.

## 1.3. Ұғымдар: «унарлы», «бинарлы», «операнд»

Айнымалылармен жұмыс жасау үшін JavaScript барлық стандартты операторлары қолданады, бұлардың көбісі басқа бағдарламалау тілдерінде де бар.

Операторларда барлық бағдарламалау тілдерінде қолданылатын өзіндік ұғымдары бар.

– *Операнд* - бұл операторда қолданылатын айнымалы. Мысалы:  $5 * 2$  - оң және сол операндтарымен көбейтілген оператор. Басқаша атауы: «оператор аргументі».

– *Унарлы* - бұл бір мәнге қолданылатын оператор.

Мысалы, унарлы оператор азайту "-" қарама-қарсы мәнге ауыстырылады:

```
var x = 1;
```



```

alert( -x ); // -1, унарлы азайту
alert( -(x+2) ); // -3, унарлы азайту x+2 қосу нәтижесіне меншіктелген.
alert( -(-3) ); // 3

```

– Бинарлы деп — екі операндта қолданылатын операторды айтады. Дәл сол азайту бинарлы формада да қолданылады:

```

var x = 1, y = 3;
alert( y - x ); // 2, бинарлы азайту

```

JavaScript-тегі унарлының "+" және бинарлының "-" жұмыстары ерекшеленеді. Бұл шындығында әр түрлі операторлар. Бинарлы қосу - операндтарды қосады, ал унарлы - арифметикалық тұрғыда ештеңе жасамайды, бірақ операндты сандық типке әкеледі.

#### **1.4. Арифметикалық операторлар**

Кейбір JavaScript тілін сүйемелдейтін браузерлердің бұрынғы нұсқалары оларды орындамайды. Ондай браузерлер скриптерді түсінбейтән болғандықтан, JavaScript тілінің операторларын комментарий тегтерінің <!--...--> ішіне орналастыру қалыптасқан. Скриптердегі қолданылатын тілді анықтайтын параметрлердің бірі <Script> тегінің language атрибуты болып табылады.

Кесте 8. JavaScript-тегі арифметикалық операторлары

Оператор	Атауы
+	қосу
-	азайту
*	көбейту
/	бөлу
%	бүтінсанды бөліндіден қалған қалдық
<b>Икрементті және декрементті операторлар</b>	
++	Операнд мәнін бірге өсіру
--	Операнд мәнін бірге кеміту

Мысалдар:

1) alert(2 + 2); // 4

2) var i = 2;  
i = (2 + i) \* 3 / i;  
alert(i); // 6

3) alert(5 % 2); // 1, (5-тің 2-ге бөліндісінің қалдығы)  
alert(8 % 3); // 2, (8-дің 3-ке бөліндісінің қалдығы)  
alert(6 % 3); // 0, (6-ның 3-ке бөліндісінің қалдығы)

4) Инкрементті ++1-ге өсіру:  
var i = 2;  
i++; // i = i + 1 (қысқа жазба үшін)  
alert(i); // 3

5) Декрементті – 1-ге кеміту:  
var i = 2;  
i--; // i = i – 1 (қысқа жазба үшін)  
alert(i); // 1

6) Инкрементті/декрементті кез-келген өрнелерде қолдануға болады.

```
var i = 1;  
alert(2 * ++i); // 4 (мұнда ++i айнымалыны өсіреді, содан соң мәнін а-ға қайтарады)
```

```
var i = 1;  
alert(2 * i++); // 2, (мұнда а-ға i-дің ескі мәні қайтарылады)
```

### **1.5. Жолдарды қосу, бинарлы +.**

Егер бинарлы оператор + жолдарға қолданса, ол екеуін бір жолға қосады:

```
var a = "ол" + "студент";  
alert(a); // ол студент
```

*Егер де бір аргумент жол болса, онда екіншісі де жолға өзгереді.*

Операнд-жолы оң немесе сол жақта болуы міндетті емес, кез келген жағдайда жол емес аргумент жолға өзгереді. Мысалы:

```
alert('1' + 2); // "12"  
alert(2 + '1'); // "21"
```

Бұл бинарлы оператордың мүмкіндігі "+".

*Қалған арифметикалық операторлар сандармен ғана жұмыс істейді және үнемі аргументті саңдық мәнге әкеледі.* Мысалы:

```
alert('1' - 2); // -1  
alert(6 / '2'); // 3
```

### **1.6. Меншіктеу.**

Меншіктеу операторы «=» мәніне ұқсайды:

```
var i = 1 + 2;  
alert(i); // 3
```

Ол оң жақта тұрған айнымалы мәнін меншіктейді. Бұл мән өте күрделі және де құрамында басқа да айнымалылар болуы мүмкін

```
var a = 1;  
var b = 2;  
a = b + a + 3;  
alert(a); // 6
```

Бірден бірнеше өрнектерді шығаруға болады:

```
var a = 1;  
var b = 2;  
var c = 3 - (a = b + 1);  
alert(a); // 3  
alert(c); // 0
```

### **1.7. Меншіктеуі бар операторларды шақыру**

Көп жағдайда операторды айнымалыға меншіктеп, нәтижесін сол жерде сақтау қажет болады. Мысалы,

```
n = n + 5; d = d * 2;
```

Біріктірілген операторлардың көмегімен (+, -, \*, /, >>=, <<=, >>>=, &=, |=, ^=) бұл жазбаны қысқартуға болады:

```
var n = 2;
```

n += 5; // сонда n=7 (n = n + 5 )  
n \*= 2; // сонда n=14 (n = n \* 2 )  
alert(n); // 14

Кесте 9. Мәншіктеу операторлары

Оператор	Әрекет	Мысал	X қабылдайтын мән	Паскальда ұқсас
=	Айнымалы мәнін тағайындайды	X=500;	500	X:=500;
+=	Айнымалы мәнін көрсетілген мәнмен арттырады	X=500; X+=100;	600	X:=500; X:=X+100;
-=	Айнымалы мәнін көрсетілген мәнмен азайтады	X=500; X-=12;	488	X:=500; X:=X-12;
*=	Айнымалы мәнін көрсетілген мәнмен көбейтеді	X=500; X*=2;	1000	X:=500; X:=X*2;
/=	Айнымалы мәнін көрсетілген мәнмен бөледі	X=500; X/=2;	250	X:=500; X:=X/2;
%=	Айнымалы мәнін көрсетілген мәнмен бөледі және қалдығын қайтарады	X=500; X%=5;	0	X:=500; X:=X mod 5;

Кесте 10. Салыстыру амалдары

Операция (амал)	Әрекет	Мысал	Паскальда ұқсас
= =	Тең	X==10	X=10
!=	Тең емес	X!=5	X<>5
>	үлкен	X>0	X>0
<	кіші	X<4	X<4
>=	Үлкен немесе тең	X>=Y	X>=Y
<=	кіші немесе тең	X<=5	X<=5

Кесте 11. Логикалық амалдар

Операция (амал)	Әрекет	Мысал	Паскальда ұқсас
&&	And амалына ұқсас	X>=2 && y>=2	(X>=2)and(Y>=2)
	or амалына ұқсас	x>0    y>0	(x>0)or(y>0)
!	not амалына ұқсас	!(1 < x && x < 10)	Not((1 < x) and ( x < 10))

### **Өрнектер:**

$y=x++$  мына екі меншіктеулерге эквивалентті:  $y=x$ ;  $y=y+1$ ,

$y=++x$  мына екі меншіктеулерге эквивалентті:  $x=x+1$ ;  $y=x$ ,

$y=x--$  мына екі меншіктеулерге эквивалентті:  $y=x$ ;  $x=x-1$ ,

$y=--x$  мына екі меншіктеулерге эквивалентті:  $x=x-1$ ;  $y=x$ .

Мысалы:

1) `var i = 2;`

`i++; // i = i + 1 қысқа жазба үшін`

`alert(i); // 3`

2) `alert(5 % 2); // 1, (5-тің 2-ге бөліндісінің қалдығы)`

**Мысал-2.** "Person" атаулы объект құрып, объектіге 4 қасиет кірістіретін бағдарламаны құру.

```
<html>
```

```
<body>
```

```
<script>
```

```
var person=new Object();
```

```
person.firstname="Arman";
```

```
person.lastname="Cakenov";
```

```
person.age=23;
```

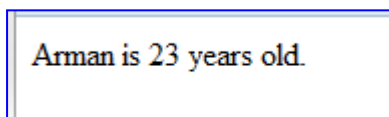
```
person.eyecolor="blue";
```

```
document.write(person.firstname + " is " + person.age + " years old.");
```

```
</script>
```

```
</body>
```

```
</html>
```



Сурет 9. Бағдарлама нәтижесі

**Мысал-3.** Келесі формуланы есептейтін бағдарлама құру керек.

```
<html>
```

```
<body>
```

```
<script>
```

```
var i = 2;
```

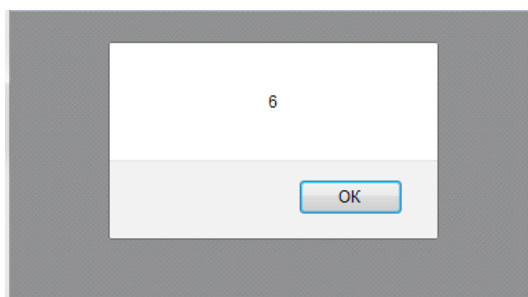
```
  i = (2 + i) * 3 / i;
```

```
  alert(i);
```

```
</script>
```

```
</body>
```

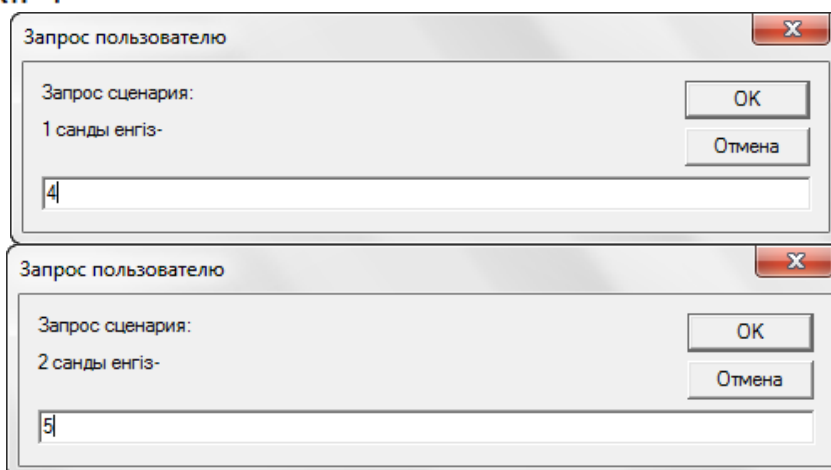
```
</html>
```



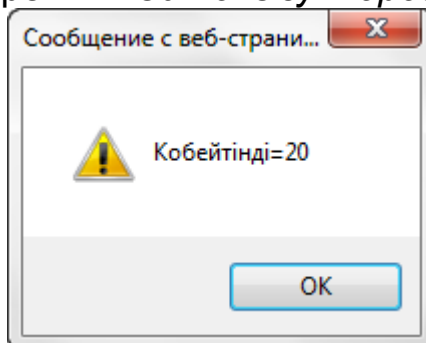
Сурет 10. Бағдарлама нәтижесі

**Мысал-4.** Берілген санның көбейтіндісін есептейтін бағдарлама құру керек.

```
<SCRIPT language=JavaScript>  
  
var str=prompt("1 санды енгіз-"," ");  
  x=parseInt(str)  
  str=prompt("2 санды енгіз-"," ");  
  y=parseInt(str)  
  P=x*y  
  alert("Кобейтінді="+P);  
</SCRIPT>
```



Сурет 11. Сан енгізу терезесі



Сурет 12. Бағдарлама нәтижесі

#### Өзіндік тапсырмалар:

1. Экранда бір қызыл және бір көк мәтін жолдарын шығаратын скриптті шығару керек:

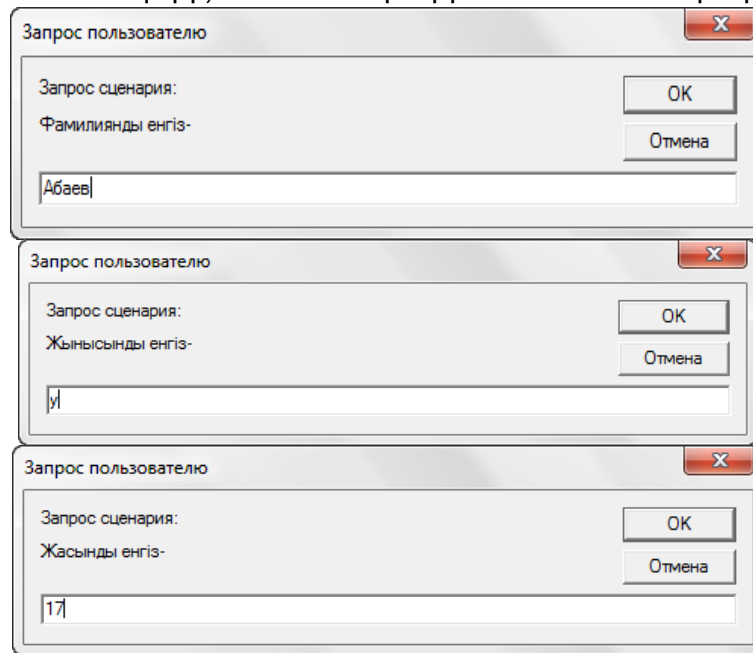
**Білім инемен құдық қазғандай.**  
**Өнер – білім бұлағы, білім - өмір шырағы.**

2. Берілген санның квадраттарының қосындысын есептейтін бағдарлама құрыңыз:  $Z=x^2+y^2$ .

3. Берілген сандардың кубтарының айырымын есептейтін бағдарлама құрыңыз:  $Z=x^3-y^3$ .

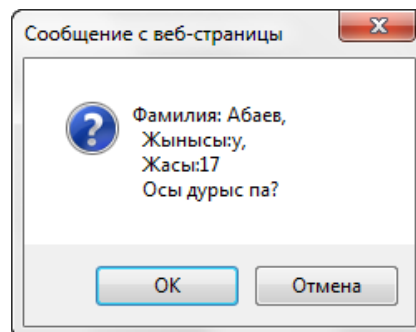
4. а,в,с қабырғалары бар үшбұрыштың ауданын Герон формуласы бойынша есептеу керек:  $S = \sqrt{p(p-a)(p-b)(p-c)}$ ;  $P = (a+b+c)/2$ .

5. Дискриминантты есептейтін бағдарламаны құрыңыз.
6. Өзіңіздің аты жөніңізді, жынысыңызды және жасыңызды енгізіңіз.



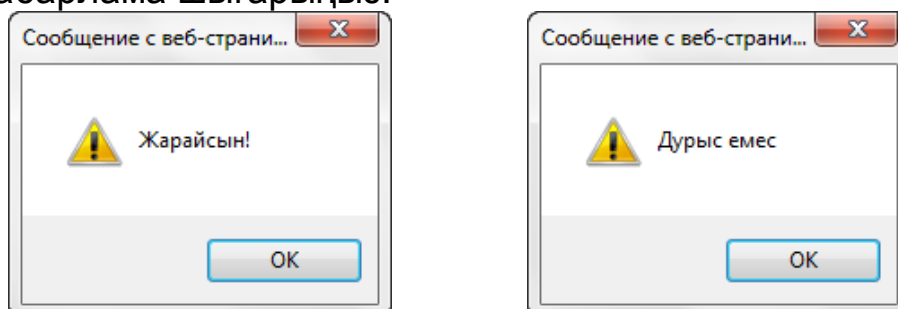
Сурет 13. Тапсырма түрі

Дұрыстығын сұраңыз:



Сурет 14. Тапсырма нәтижесі

Егер дұрыс болса, сәлемдесу шығарыңыз, керісінше жағдайда, қате туралы хабарлама шығарыңыз.



Сурет 15. Тапсырма нәтижесі

**Өзін-өзі бақылауға арналған сұрақтар:**

1. JavaScript тілі қандай тіл?
2. JavaScript тегтері қай тегтердің арасында орналасады?
3. Ақпаратты шығарудың қандай тәсілдері бар?
4. JavaScript-те қандай арифметикалық әрекеттер операторлары бар?
5. Меншіктеу операторы қандай?

## 2.2 Практикалық жұмыс №2

### JAVASCRIPT ТІЛІНІҢ ШАРТТЫ ОПЕРАТОРЛАРЫ

**Жұмыстың мақсаты:** JavaScript тілінің шартты операторларын қолданып есептерді шығарып үйрену.

**Қысқаша мәліметтер:**

#### 2.1.Шартты операторлардың түрлері.

JavaScript тілінің шартты операторлары көмегімен командалардың орындалуын өзгерте аласыз.

JavaScript те келесідей шартты операторлар бар:

- **if операторы** - берілген шарт ақиқат болғанда ғана командаларды орындау үшін қолданылады;
- **if..else операторы** - берілген шарт ақиқат, ал екіншісі жалған болғанда командаларды орындау үшін қолданылады;
- **if..else if..else операторы** – команданың анықталған шарты кезінде бірнеше блоктардың тек біреуінде орындалуы үшін қолданылады;
- **switch операторы** – бұл операторды да, команданың анықталған шарты кезінде бірнеше блоктардың тек біреуінде орындалуы үшін қолданылады.

*Кесте 12. JavaScript пен Pascal айырмашылығы*

JavaScript	Pascal
if (a==2) z=2; else z=3	if a=2 then z:=2 else z:=3;
if (x>=2 && x<=6) {y=0; z=1} else {y=1; z=0}	if (x>=2)and(x<=6) then begin y:=0; z:=1; end else begin y:=1; z:=0; end;

*Паскальдан айырмашылығы:*

1. Then жоқ.
2. Егер else бірдей жолда болса, else алдында нүкте үтір орнатылады.
3. Барлық шарт жақшаға алынады. Қарапайым шарттарды жақшаға алмауға болады.
4. Операторлық жақшаларының (begin end) орнына, фигуралық жақшалар қолданылады.

#### 2.2. IF шартты операторы.

If операторын берілген шарт ақиқат болғанда ғана командаларды орындау үшін қолданылады:

```
if(шарт)
{
    тұжырым
}
```

**Мысал-1.**

```
var a=7;
var b=7;
```

```

if (a==b) {
    document.write("Сандар тең");
}

```

### 2.3. IF..ELSE операторы.

Бұл операторды берілген шарт ақиқат, ал екіншісі жалған болғанда командаларды орындау үшін қолданылады:

```

if(шарт)
{
    тұжырым1
}
else
{
    тұжырым2
}

```

#### **Мысал-2.**

```

var a=9;
var b=7;
if (a==b) {
    document.write("Сандар тең");
}
else {
    document.write('Сандар тең емес');
}

```

#### **Мысал-3.**

$$y = \begin{cases} x^2 + x + 1, & x \leq 0 \\ x - \sqrt{x+1}, & x > 0 \end{cases}$$
 функциясын есептеу керек. Екі түрлі жолды

қолданып есептейік.

А)

```

<html>
<body>
<script>
var x = -22;
if (x <= 0) y = x*x+x+1;
else y = x-Math.sqrt(x+1);
{
    alert ("x=" +x+ "y=" +y);
}
</script>
</body>
</html>

```

Б)

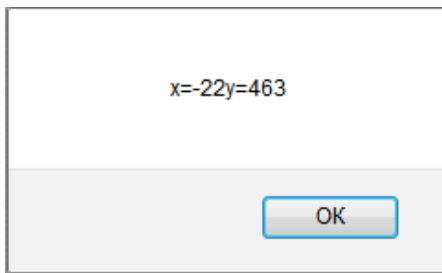
```

<html>
<body>
<script>
var x = prompt ("x-", "1");
y=(x >= 0) ? 2*x+1: x-
Math.sqrt(x+1);
    alert (+x+ "y=" +y);
</script>
</body>
</html>

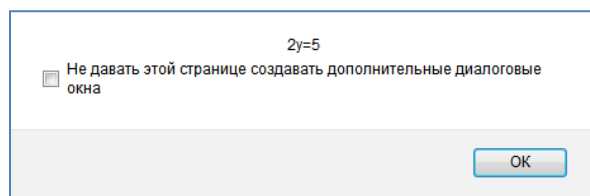
```

а)





Сурет-16. Нәтиже



ә)

Сурет-17. Нәтиже

### Өзіндік тапсырмалар:

1. Есептеңіз:

а)

$$y = \begin{cases} x, & x > 1 \\ 1-x, & x \leq 1 \end{cases}$$

ә)

$$y = \begin{cases} 2x+1, & x < 0 \\ x^2+1, & x \geq 2 \end{cases}$$

б)

$$z = \begin{cases} 5x+2y, & x+y \leq 0 \\ \frac{2}{x+y}, & x+y > 0 \end{cases}$$

2.  $x, y, z$  нақты сандары берілген.  $x, y, z$  қабырғалы ұзындықты үшбұрыш бар ма екенін анықтау керек.

3.  $x, y, z$  нақты сандары берілген. Табу керек:

а)  $\max(x, y, z)$ ;

б)  $\min(x, y, z)$ .

4.  $x, y, z$  нақты сандары берілген. Есептеңіз:

а.  $\max(x+y+z, xyz)$ ;

б.  $\min(x+y, z/2, xyz+1)$ .

### Өзін-өзі бақылауға арналған сұрақтар:

1. Шартты операторлардың қандай түрлері бар?

2. If операторы қай кезде қолданылады?

3. IF..ELSE операторы қай кезде қолданылады?

## 2.3 Практикалық жұмыс №3 JAVASCRIPT ТІЛІНДЕГІ ЦИКЛДІК ОПЕРАТОРЛАР

**Жұмыстың мақсаты:** JavaScript тіліндегі циклдік операторларды қолданып есептерді шығарып үйрену.

### Қысқаша мәліметтер:

#### 3.1. FOR циклі

Белгілі бір әрекеттердің қайталанып орындалуын цикл операторларының көмегімен ұйымдастыруға болатындығы бізге белгілі. JavaScript бағдарламалау тілінде циклді ұйымдастырудың екі түрі бар:

1. For циклі.

2. While циклі.

For циклі скрипт неше рет орындалатыны алдын ала белгілі болған жағдайда қолданылады.

For циклінің жазылуы:

```

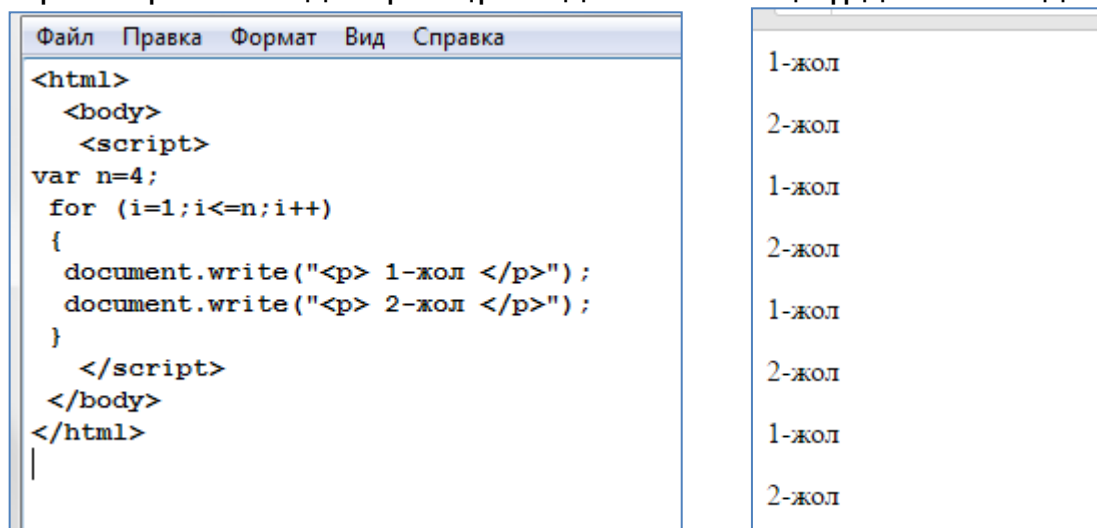
for(айнымалы =бастапқы_мән;
айнымалы <=соңғы_мән;
айнымалы = айнымалы +қадам)
{
    орындалатын код
}

```

### Паскальдан айырмашылығы:

1. Цикл кез-келген қадаммен болуы мүмкін.
2. do жоқ.
3. Параметрдің бастапқы және соңғы мәндері, сондай-ақ қадам, циклдің тақырыбындағы жақшаларда көрсетіледі.
4. Операторлық жақшаларының (begin end) орнына, фигуралық жақшалар қолданылады.

**Мысал-1.** Берілген сөздерді төрт рет қайталап шығаратын программа құру керек. Бұл мысалда бүкіл құжат динамикалық түрде жасалады.



Сурет 18. Бағдарлама коды мен нәтижесі

**Мысал-2.** Төмендегі мысал  $i=0$  басталатын циклді анықтайды.  $i \leq 5$  болғанша цикл орындала береді. Циклдің әрбір итерациясында  $i$  1-ге көбейіп отырады.

```

<html>
  <body>
    <script type="text/javascript">
var i=0;
for (i=0;i<=5;i++)
{
  document.write("i саны =" + i);
  document.write("<br />");
}
    </script>
  </body>
</html>

```

```

i саны =0
i саны =1
i саны =2
i саны =3
i саны =4
i саны =5

```

Сурет 19. Бағдарлама нәтижесі

### 3.2. WHILE циклі.

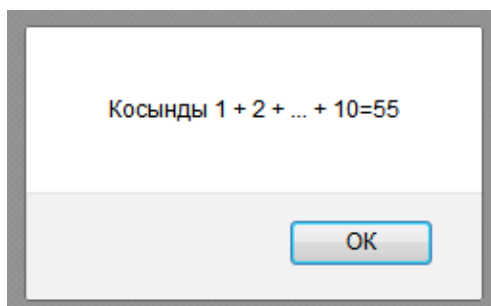
Циклдің орындалуы: алдымен шарт тексеріледі. Егер ол ақиқат болса, командалар орындалады. Келесі жолы да осы әрекеттер қайталанады, яғни шарт тексеріледі, егер ол ақиқат болса, цикл орындалады, т.с.с. Кезекті тексеру кезінде шарт жалған болған кезде, цикл жұмысы аяқталады. Циклде шарт алдын ала тексерілетін болғандықтан, ол бірде бір рет орындалмауы да мүмкін.

While циклінің жазылуы:

```
while(айнымалы <=соңғы_мән)
{
    орындалатын код
}
```

**Мысал-3.**  $S = 1 + 2 + \dots + 10$  қосындысын есептеу керек.

```
<html>
<body>
<script>
var i = 1;
var sum = 0;
while (i <= 10)
{
    sum += i;
    i++;
}
alert("Қосынды 1 + 2 + ... + 10=" +sum);
</script>
</body>
</html>
```



Сурет 20. Бағдарлама нәтижесі

### 3.3. DO ... WHILE циклі.

Do...while – бұл while циклінің басқа түрі. Бұл цикл блок кодын *бір рет* орындайды, және берілген шарт ақиқат болғанша циклді қайталап отырады.

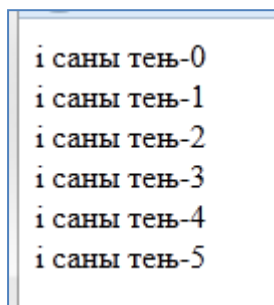
Жазылуы:

```
do
{
    орындалатын код
}
```

```
}  
while (айнымалы<=соңғы мән);
```

**Мысал-4.** Төмендегі мысал  $i=0$  басталатын циклді анықтайды.  $i \leq 5$  болғанша цикл орындала береді. Циклдің әрбір итерациясында  $i$  1-ге көбейіп отырады.

```
<html>  
<body>  
  <script type="text/javascript">  
    var i=0;  
    do  
    {  
      document.write("i саны тең-" + i);  
      document.write("<br />");  
      i++;  
    }  
    while (i<=5);  
  </script>  
</body>  
</html>
```



```
i саны тең-0  
i саны тең-1  
i саны тең-2  
i саны тең-3  
i саны тең-4  
i саны тең-5
```

Сурет 21. Бағдарлама коды мен нәтижесі

### Өзіндік тапсырмалар:

1. Қосындыны есептеу керек:  $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ .
2.  $P=1*2*...*10$  сандарының көбейтіндісін есептеңіз.
3. 1-ден 100-ге дейінгі сандардың арифметикалық ортасын анықтаңыз.
4.  $y = a^x$  функциясын  $a=1,2,3,4,5,6,7$  болғандағы мәндерін анықтау керек. Мұндағы цикл,  $a$ -ның өсу ретімен орындалады.

5. Есептеңіз: 
$$S = \sum_{i=1}^k \frac{x^{i+1}}{i}$$
.

6. Есептеңіз: 
$$p = \prod_{k=1}^n a^2 + k/2 - 1.$$

### Өзін-өзі бақылауға арналған сұрақтар:

1. Цикл деген не?
2. JavaScript бағдарламалау тілінде циклді ұйымдастырудың неше түрі бар?
3. For циклі қай кезде қолданылады?
4. While циклінің Do...while циклінен айырмашылығы неде?

## 2.4 Практикалық жұмыс №4

### JAVASCRIPT ОҚИҒАЛАРЫ

**Жұмыстың мақсаты:** JavaScript тілінің оқиғаларын қолданып есептерді шығарып үйрену.

**Қысқаша мәліметтер:**

#### JavaScript оқиғалары.

JavaScript-тің бірнеше оқиғаларын қарастырайық:

➤ **OnLoad оқиғасы.** Бұл оқиға бірденені тиегеннен кейін, мысалы, браузер терезесінде бетті ашқаннан кейін орындалады. Ол суреттерімен қоса барлық беттің тиелгенінен кейін ғана орындалды деп саналады. OnLoad оқиғасын функция бет ашылғаннан кейін бірден орындалу үшін сценарий ішінде қолдану өте ыңғайлы.

➤ **OnClick оқиғасы.** Бұл оқиға беттің белгілі бір жерінде тышқанмен шерткен кезде орындалады. Кейінірек сіз беттің көптеген элементтері (гипермәтін, сурет, батырма және т.б.) onClick оқиғасын қабылдай алатынын білесіз. Оны тұтынушымен бірлесіп әрекет ететін сценарий құрғыңыз келгенде қолдануға болады.

➤ **OnMouseover оқиғасы.** Бұл оқиға onClick оқиғасына ұқсас болғанымен, тышқан шертуінен кейін емес, курсорды беттің белгілі элементіне жақындаған кезде орындалады. OnMouseover оқиғасын кез-келген Web-бет объектімен (мәтінмен, суретпен, батырмамен, гипермәтінмен және т.б.) байланыстыруға болады. Сонымен қатар оны интерактивтілік деңгейін көтеру үшін қолдануға болады.

➤ **OnMouseout оқиғасы.** Бұл оқиға onMouseover оқиғасы сияқты, бірақ тышқан объектіден алынған жағдайда ғана орындалады.

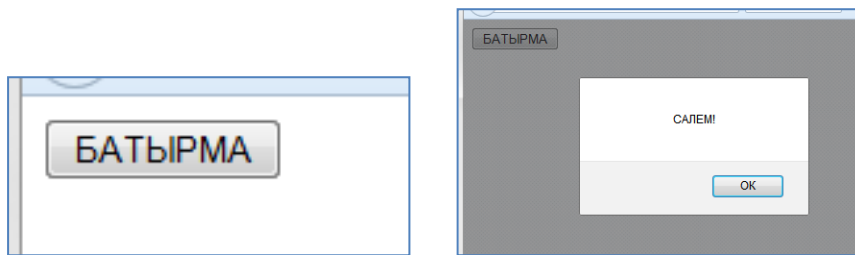
OnLoad оқиғасын қолдану. OnLoad оқиғасын JavaScript функциясын шақыру үшін қолданғанда, келесі *оқиғаны өңдеушіні* (event handler) Web-бетінің <body> тегіне орналастыру керек:

```
<body onLoad="Функция аты()">
```

**Ескерту.** Оқиғаны өңдеуші – бұл оқиға суреттелетін және пайда болу нәтижесінде кейбір әрекет орындалатын, мысалы, орындалуға JavaScript функциясы енгізілетін, код үзіндісі.

Мысал-1. Батырмаға OnClick оқиғасын қолдану.

```
<html>
<head>
</head>
<body>
  <script language="JavaScript">
  </script>
<FORM>
  <INPUT type=button value="БАТЫРМА" onclick="alert('САЛЕМ!') ; ">
</FORM>
</body>
</html>
```



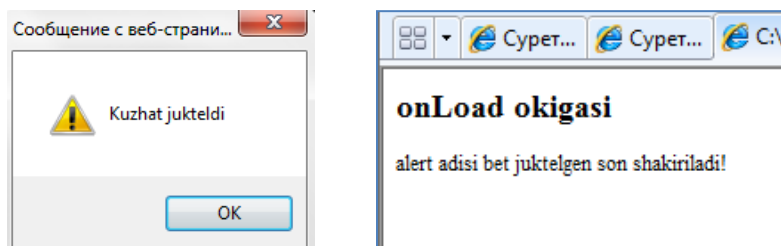
Сурет 22. Батырмаға OnClick оқиғасын қолдану

**Мысал-2.** Веб-бет жүктелгеннен кейін alert() әдісі шақырылатын бағдарламаны құрыңыз. **Load** оқиғасы body элементі үшін құжаттың жүктелуі аяқталғанда пайда болады.

```
<html>
<head> </head>

<body onLoad="alert('Kuzhat jukteldi') ">
  <h2> onLoad okigasi </h2>
<p>
  alert adisi bet juktelgen son shakiriladi!
</body>

</html>
```



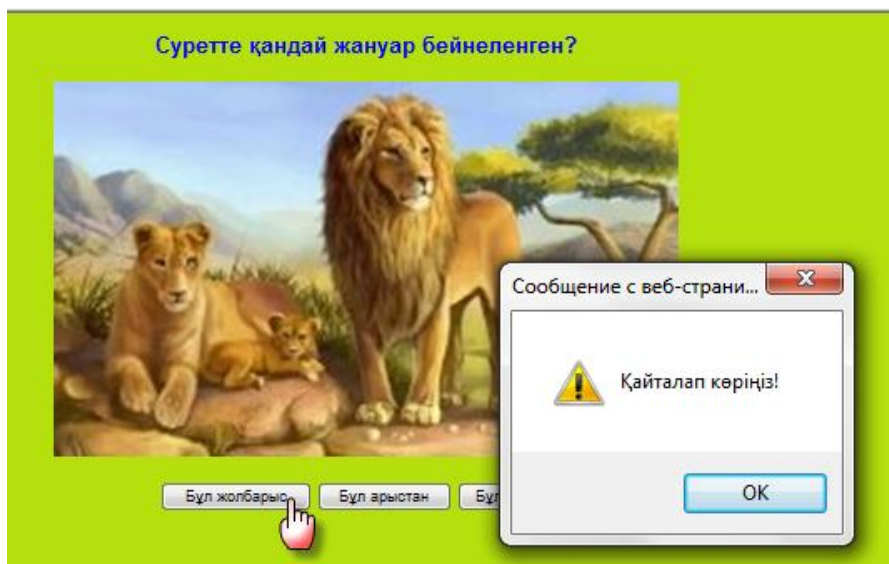
Сурет 23. Load оқиғасын қолдану

### Өзіндік тапсырмалар:

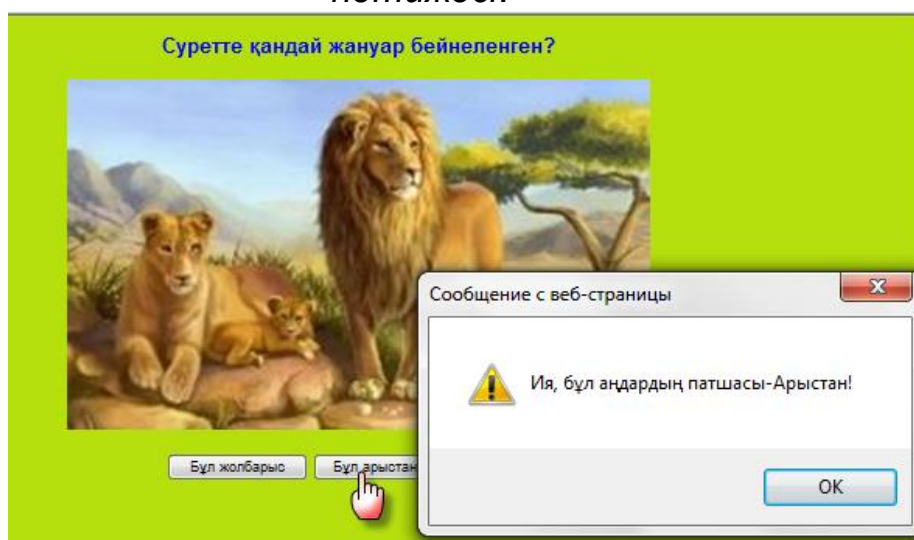
1. Alert() әдісі арқылы батырманы шерткенде аты жөніңізді шығаратын бағдарлама құрыңыз.
2. Қарастырылған элементтерді қолданып, суреттегідей құжатты құрыңыз.



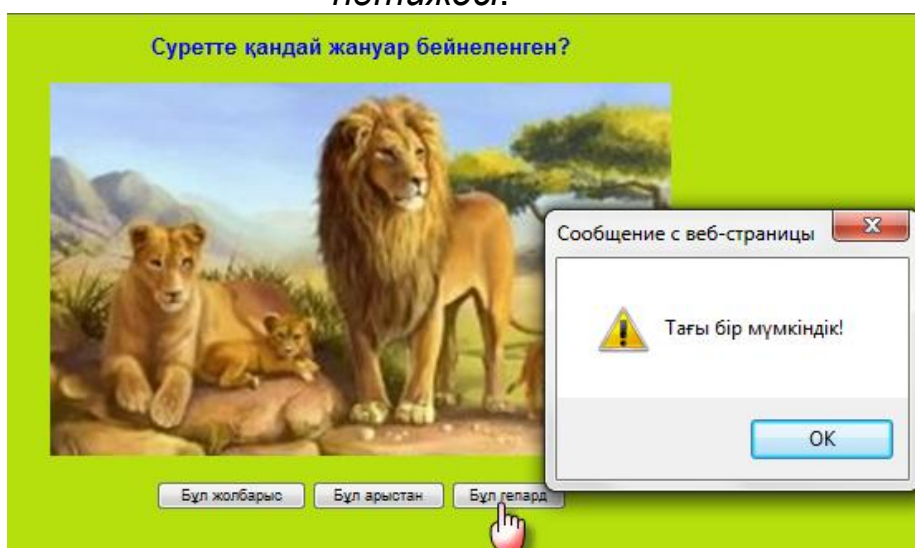
Сурет 24. Титулды бет



Сурет 25. *onClick* оқиғасын қолданып, бірінші батырманы шерту нәтижесі.



Сурет 26. *onClick* оқиғасын қолданып, екінші батырманы шерту нәтижесі.



Сурет 27. *onClick* оқиғасын қолданып, үшінші батырманы шерту нәтижесі.

### **Өзін-өзі бақылауға арналған сұрақтар:**

1. JavaScript тілінің қандай оқиғалары мен объектілерін білесіз?
2. onLoad оқиғасың қызметі қандай?
3. onclick оқиғасың қызметі қандай?

## **2.5 Практикалық жұмыс №5 JAVASCRIPT ФУНКЦИЯЛАРЫ**

**Жұмыстың мақсаты:** JavaScript тілінің функцияларын қолданып есептерді шығарып үйрену.

### **Қысқаша мәліметтер:**

#### **5.1. JavaScript функциялары**

Программа құрғанда ондағы логикалық тәуелсіз бөліктерді жеке-жеке бөліп қарауға болады. Программаны мұндай бөліктерге бөліп қарастыру оның жұмысын түсінуді жеңілдетеді.

**Функция** - JavaScript тілінің негізгі элементі. Функция мынадай түрде сипатталады: `function F(v) {s}`

Мұндағы F – функцияны шақыруға болатын ат тағайындайтын функция идентификаторы; V- үтір арқылы бөлініп жазылатын функция параметрлерінің тізімі. S- нәтиже алу үшін орындалатын іс-әрекеттерден тұратын функция тұлғасы, яғни операторлар тізбегі.

Ол мыналардан құралады:

- function сөзінен;
- функция атауынан;
- жай жақшаға алынған, үтір арқылы бөлінген, аргументтер тізімінен;
- фигуралық жақшаға алынған функция денесінен.

#### **5.2. Функцияның жазылуы:**

```
function myFunction(arg1, arg2, ...)
{
  ...
  Операторлар тізбегі
  ...
}
```

Мұндағы, *myFunction* — функция атауы, *arg1, arg2* — параметрлер тізімі

#### **5.3. Функцияны шақыру**

<i>Функция_атауы (үтір арқылы аргументтердің тізімі)</i>
--

Аргумент ретінде тұрақты, айнымалы, өрнек бола алады, көбінесе, басқа функцияны шақыру.

**Мысал-1.** Функция арқылы есімдерді шығару.

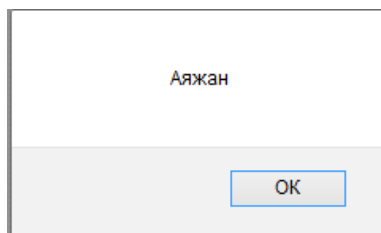
```
<html>
<head> <title> Салемдесу </title>
</head>
```



```

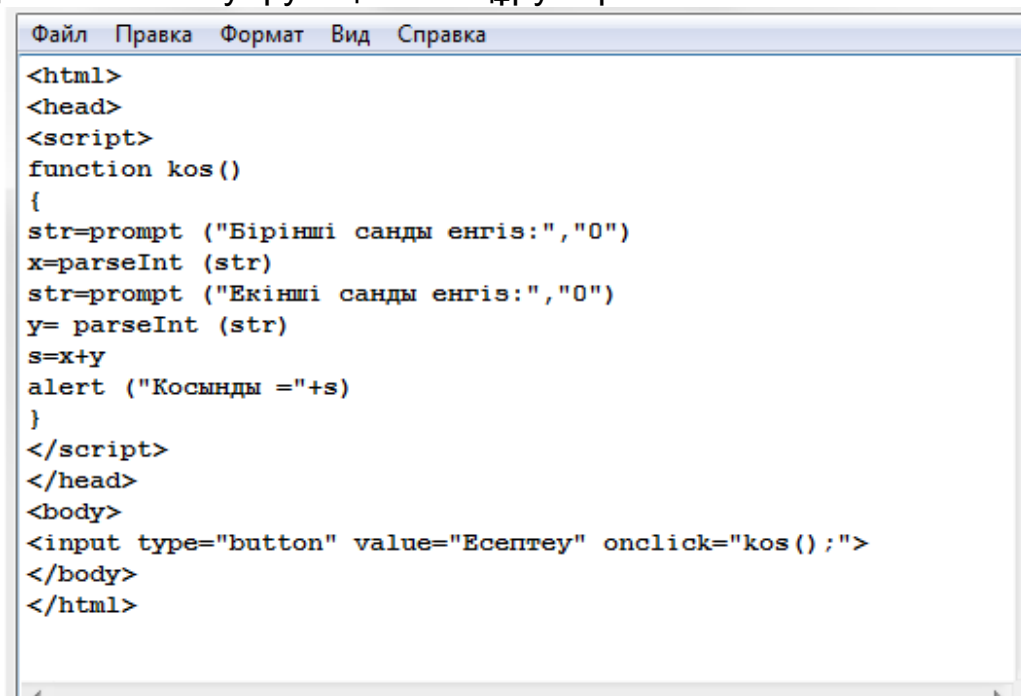
<body>
<script language="JavaScript">
{
function Hello (xname)
  alert (' '+xname)
Hello ('Аяжан ');
Hello (' Сания ');
Hello (' Оксана ');
}
</SCRIPT>
</body>
</html>

```

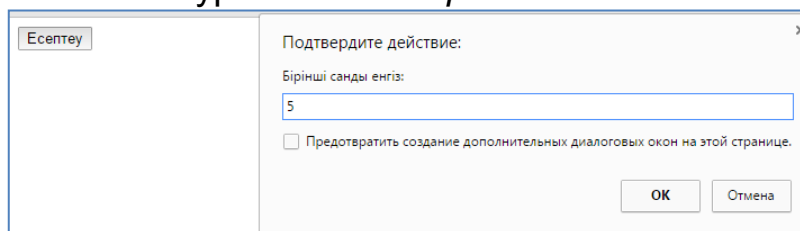


Сурет 28. Функция арқылы есімдерді шығару.

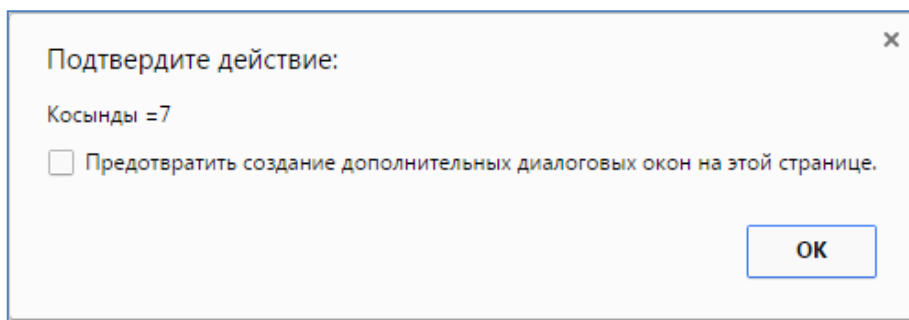
**Мысал-2.** Батырманы шерткенде шақырылатын, екі санды енгізіп және қосындысын есептеу функциясын құру керек.



Сурет 29. Бағдарлама коды

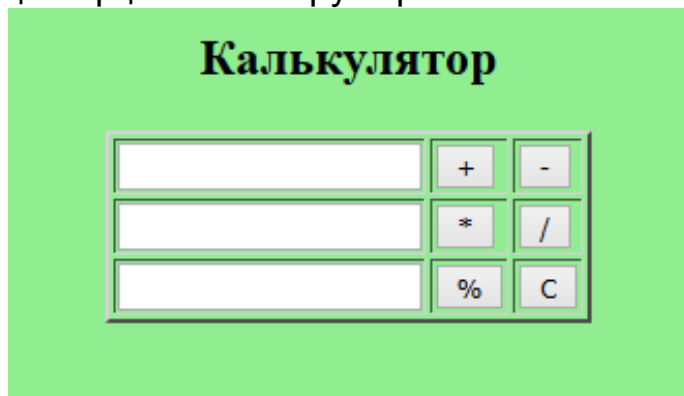


Сурет 30. Нәтиже



Сурет 31. Нәтиже

**Мысал-3.** Қарапайым калькулятор құру бағдарламасы. Арифметикалық амалдарды функция арқылы шығару керек.



Сурет 32. Қарапайым калькулятор түрі

1. Әр батырмаларды кесте арқылы қоршаймыз. Кестенің *border* параметрін қолданамыз. Айнымалыларды енгізу үшін екі мәтіндік өрісті құрамыз - *op1* және *op2*, нәтижесі *nat* өрісінде шығарылады.

```

<body bgcolor= lightgreen>
<center><h2>Калькулятор</h2>
<table border="2">
<tr>
<td><input id="op1"></td>
<td><input type="button" onclick="plus()"value="+"></td>
<td><input type="button" onclick="minus()"value="-"></td>
</tr>
<tr>
<td><input id="op2"></td>
<td><input type="button" onclick="kob()"value="*"></td>
<td><input type="button" onclick="bolu()"value="/"></td>
</tr>
<tr>
<td><input id="nat"></td>
<td><input type="button" onclick="proc()"value="%"></td>
<td><input type="reset" value="C" id="resetButton"/></td>
</tr>
</table></center>

```

2. Енді скриптің арасында функцияны қолданып арифметикалық амалдарды орындаймыз. Қосуды есептейтін функцияның жазылуы:

```
<script language="javascript">
function plus()
{
nat.value=op1.value-(-1)*op2.value;
}
```

Бұл жерде «+» таңбасының орнына –(-) деп береміз. Өйткені егер «+» таңбасын беретін болсақ, **конкатенация** (екі мәтіндік элементті біріктіру) амалы орындалып кетеді. Мысалы, 5+2=52.

Ал, қалған арифметикалық амалдар стандартты түрде жазылады:

«-» — азайту;

«\*» — көбейту;

«/» — бөлу.

3. Осы үш амалды өз бетіңізше орындаңыз.

4. Пайызды есептеу және өрістерді тазалау батырмаларын қосыңыз.

5. Өз қалауыңыз бойынша басқа да батырмалар қосуыңызға болады..



Сурет 33. Калькулятордың орындалуы

#### Өзіндік тапсырмалар:

1. Параметр түрінде үш санды алатын және олардың көбейтіндісін табатын функцияны құрыңыз. Сандар функциядан тыс енгізіледі.
2. Параметр түрінде 4 санды алатын және олардың айырмасын табатын функцияны құрыңыз.
3. Үшбұрыштың ауданын есептеу керек.

#### Өзін-өзі бақылауға арналған сұрақтар:

1. Функция деген не?
2. Функцияның жазылуы қандай?
3. Ол не үшін қолданылады?

## 2.6 Практикалық жұмыс №6 MATH ОБЪЕКТІСІ

**Жұмыстың мақсаты:** JavaScript тілінің math объектісін қолданып есептерді шығарып үйрену.

**Қысқаша мәліметтер:**

### 6.1.Math математикалық класы

Math математикалық класы — математик-бағдарламалаушыларға тригонометриялық және логарифмдік функциялар жиынын береді. Яғни ол тұрақтылар мен әдістерді қамтитын класс. Олардың шақырылу түрі:

*Math.константа*

*Math.функция(i..)*

*Кесте 13. Math класының тұрақтылары*

Тұрақтылар	Түсініктемесі	Жазылуы
E	E саны (натурал логарифмдерге негізделген) (2,718281828)	alert(Math.E) var x = Math.log(Math.E); // 1
LN10	10 санының натурал логарифмі (ln10)	alert(Math.LN10)
LN2	2 санының натурал логарифмі (ln2)	alert(Math.LN2)
PI	P тұрақтысы ("пи" саны)	alert(Math.PI) var z=Math.tan(Math.PI/4); // 0.9999999999999999
SQRT1_2	½ санының квадрат түбірін есептейді	alert(Math.SQRT1_2)
SQRT2	2-нің квадрат түбірін есептейді	alert(Math.SQRT2)
LOG2E	e санының екілік логарифмі (log2e) 1.4426950408889634	alert(Math.log2e)
LOG10E	e санының ондық логарифмі (log10e) 0.4342944819032518	alert(Math.log10e)

*Кесте 14. Math класының әдістері (стандартты функциялары)*

Әдісі	Түсініктемесі	Жазылуы
abs(сан)	Санның модулін береді.	Var x=-5; var y; y=Math.abs(x) ; alert(y);
sqrt(сан)	Санның квадрат түбірін береді.	y=Math. sqrt (x)
pow(негізі, дәрежесі)	Санның негізінің көрсетілген дәрежесін шығарады.	Math.pow(5.3) 5 <sup>3</sup> =125.
random()	0 мен 1 диапазонындағы кездейсоқ санды шығарады.	random() alert(Math. random());
ceil(сан)	Жоғарғы жағына қарай дөңгелектейді, яғни ең кіші бүтін санды қайтарады, ол аргументтен үлкен немесе тең.	y=Math. ceil (-5. 3) y=-5 var x = Math.ceil(9.2); // 10

floor(сан)	Төменгі жағына қарай дөңгелектейді, яғни ең үлкен бүтін санды қайтарады, ол аргументтен кіші немесе тең.	y=Math.floor(5.3) y= 5 var y = Math.floor(-5.9); // -6
round(сан)	Көрсетілген аргументті бүтін мәнге айналдырады.	var n = Math.round(-5.5); // -5 var m = Math.round(-5.6); // -6
cos(сан)	Санның косинусын есептейді.	y=Math.cos(x)
sin(сан)	Санның синусын есептейді.	var x = Math.sin(90); // 0.8939966636005579
tan(сан)	Санның тангенсін есептейді.	y=Math.tan(x)
asin(сан) acos(сан) atan(сан)	Кері тригонометриялық функциялар	var y = Math.acos(1); // 1
exp(сан)	Экспонентаны есептейді.	y=Math.exp(x)
log(сан)	Санның натурал логарифмін есептейді.	var z = Math.log(10); // 2.302585092994046
max(arg1,arg2) min(arg1,arg2)	Екі аргументтің үлкенін немесе кішісін табу	Var x=2; var y=Math.max(x,10) ; alert(y); var max = Math.max(1, 2, 3, -9, 46, -23); // 46

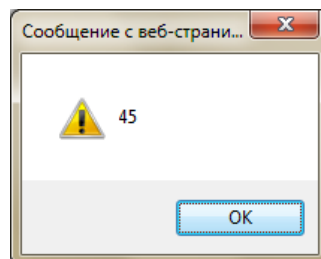
JavaScript операторында әдісті іске қосу үшін (әдетте, оны әдісті шақыру деп атайды), оған сілтеме жасау керек.

**Мысал-1.** 5-тің 10 дәрежесін табыңыз. Нәтижесін *st* айнымалысына жазу керек.

```
<script>
var st = Math.pow(2, 10);
alert(st);
</script>
```

**Мысал-2.** Осы скрипте Math объектісіне *floor* әдісі қолданды. Әдістің параметрі ретінде *d1* айнымалысының мәні қолданылған. Бұл үшін *Math.floor(d1)* конструкциясы пайдаланылды. Бұл әдіс аргументтің бөлшек бөлігін алып тастайды. Бұл мән *d2* айнымалысына беріледі және *alert(d2)* әдісінің көмегімен терезеге шығарылады.

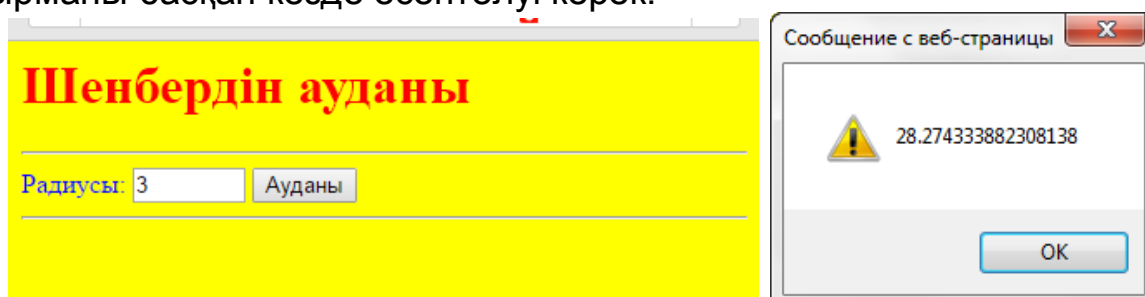
```
<script>
d1=45.6;
d2=Math.floor(d1);
alert(d2);
</script>
```



Сурет 34. Нәтиже

### Өзіндік тапсырмалар:

1. 215 санының квадрат түбірін табыңыз.
2. 587 санының квадрат түбірін табыңыз. Оны жоғары және төмен жағына қарай дөңгелектеңіз ('floor' және 'ceil' қолданыңыз).
3.  $\sin(x^2)+\cos(x^2)$  есептеңіз. Нәтижесін «round» арқылы бүтін мәнге айналдырыңыз.
4. a және b айнымалылары берілген. Олардың айырымдарының модулін табыңыз.
5. Мына сандар берілген: 4, -2, 5, 19, -130, 0, 10. Осылардың ішінен минималды и максималды санды табыңыз.
6. Шаршының ауданын есептейтін бағдарламаны жазыңыз.
7. Шеңбердің радиусын есептейтін бағдарламаны жазыңыз. Есептеу радиусын форманың мәтіндік өрісіне енгізу керек. Ауданы батырманы басқан кезде есептелуі керек.



Сурет 35. Шеңбердің радиусын есептеу

### Өзін-өзі бақылауға арналған сұрақтар:

1. Math математикалық классы не үшін арналған?
2. Math класының қандай тұрақтылары білесіз?
3. Math класының қандай әдістері білесіз?

## 2.7 Практикалық сабақ №7 ARRAY ОБЪЕКТИСІ

**Жұмыстың мақсаты:** JavaScript тілінде array объектісін қолданып есептерді шығарып үйрену.

### Қысқаша мәліметтер:

#### 7.1. Array объектісі

JavaScript тілінде массив (жиым) құрылымының ерекшелігі элементтер типтері әртүрлі және элементтер санын өзгертуге болады. Элемент индексі нөлден басталады. Массив объект ретінде екі жолмен, яғни object() және array() конструкторының көмегімен құрылады.

```
Mas=new object();  
Mas[0]=45; Mas[1]=2.5; ... Mas[10]=27;
```

```
A=new array();  
A[0]=6; A[8]=89;
```

Массив элементтерінің саны *length* қасиетімен анықталады. Мысалы: a.length=9;

*Жалпы жазылуы:*

```
var myArray = new Array();
...
// элементтер санын анықтау
alert(myArray.length)
```

*Кесте 15. Array объектісінің әдістері*

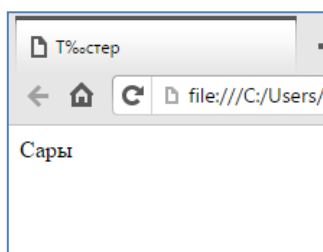
<b>Әдістер</b>	<b>Түсініктемесі</b>	<b>Жазылуы</b>
Length	Массив элементінің ұзындығы	массив_атауы.length
concat()	Массивтерді біріктіру	массив_атауы1.concat(массив2)
slice()	Массивтің бір бөлігін көшіреді	массив_атауы.slice()
Array.join()	Массивтің барлық элементтерін жолдық типке аударады.	массив_атауы.join(жол)
Array.join(бөлгіш)	Массив элементтері «бөлгіш» жолы арқылы бөлінген бір жолда жиым элементтерінің тізімін алу.	массив_атауы.join( )
Array.reverse()	Массив элементтерінің орналасу ретін анықтайды, яғни кері ретпен орналастырады.	массив_атауы.reverse()
Array.sort()	Массив элементтерін алфавит бойынша сұрыптайды. Егер массив сандық мәліметтерді қамтыса, онда сұрыптау кезінде олардың жолдық көрсетілімдері қолданылады.	массив_атауы.sort();
Array.sort(сұрыптау_функциясы)	Массив элементтерін пайдаланушының критерийлері бойынша сұрыптайды.	массив_атауы.sort();
Splice	Индекс бойынша элементті жою.	Splice (индекс, количество [, элем1 [, элем2 [, ...элемN]]]) массив_атауы.splice(0,2)
unshift()	Массивтің басына жаңа элемент қосады.	fruit.unshift("абрикос");
shift()	Массивтің басындағы бірінші элементті жояды.	массив_атауы.shift()
pop()	Массивтің соңғы элементін жояды.	массив_атауы.pop();
push()	Массивтің соңына элемент қосады.	fruit.push("шие", "абрикос");

*Ескерту:* массивтердегі индекстік нөмірлеу 1-ден емес, 0-ден басталады.

**Мысал-1.** Түстерді нөмірі бойынша шығаратын массивті құру керек.

```
<html>
<head>
<title> Түстер </title>
<script language="JavaScript">
<!-- hide
var colors = new Array ( );
colors[0] = 'Қызыл';
colors[1] = 'Сары';
colors[2] = 'Қоңыр';
colors[3] = 'Жасыл';
colors[4] = 'Көк';
colors[5] = 'Көгілдір';
colors[6] = 'Күлгін';
document.write(colors[1]);
// -->
</script>
</head>
<body>

</body>
</html>
```



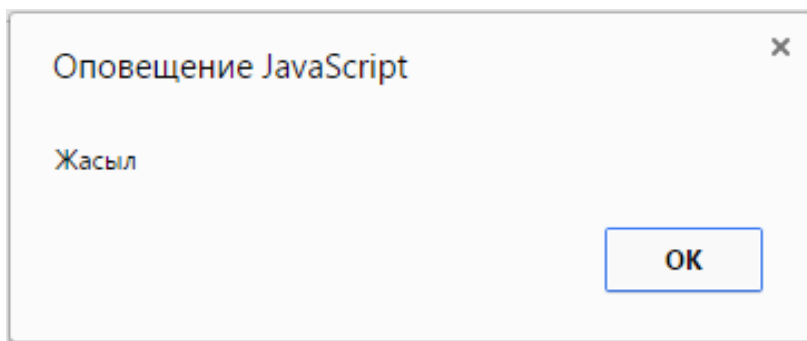
Сурет 36. Түс атауын шығару

**Мысал-2.** Түстерді нөмірі бойынша шығаратын массивті бөлек терезеде шығару керек.

```
<script language="JavaScript">
var colors = new Array ( );
colors[0] = 'Қызыл';
colors[1] = 'Сары';
colors[2] = 'Қоңыр';
colors[3] = 'Жасыл';
colors[4] = 'Көк';
colors[5] = 'Көгілдір';
colors[6] = 'Күлгін';
alert(colors[3]); </script>
```



Нәтиже:



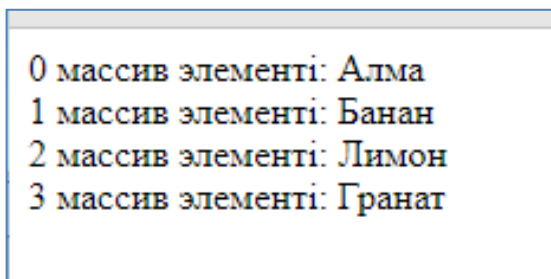
Сурет 37. Бағдарлама нәтижесі

**Мысал-3.** [1, 2, 3, 4, 5] элементтері бар массив берілген. *For* циклі көмегімен экранға осы элементтердің барлығын шығару керек.

```
var arr = [1, 2, 3, 4, 5];
for (var i = 0; i < arr.length; i++) {
    document.write(arr[i] + '<br>');
}
```

**Мысал-4.** *Concat()* әдісі көмегімен екі немесе бірнеше массивті бір массивке біріктіру керек.

```
<SCRIPT language=JavaScript>
// 1 массивті құрамыз
var mas1=["Алма", "Банан"];
// 2 массивті құрамыз
var mas2=["Лимон"];
// 3 массивті құрамыз
var mas3=["Гранат"];
//массивтерді біріктіріп, нәтижесін mas айнымалысына жазамыз
mas=mas1.concat(mas2, mas3);
// mas массивінің құрамын бетке шығарамыз.
var i;
for (i in mas)
{
    document.write(i + " массив элементі: " + mas[i] + "<br />");
}
</SCRIPT>
```



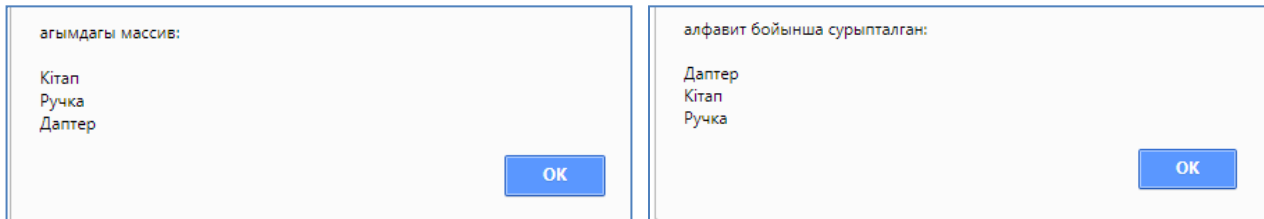
Сурет 38. Бағдарлама нәтижесі

**Мысал-5.** *myArray* массиві берілген. Осы массивті алфавит бойынша сұрыптау керек.

```

<SCRIPT language=JavaScript>
var myArray = new Array()
  myArray[0] = 'Кітап'
  myArray[1] = 'Ручка'
  myArray[2] = 'Дәптер'
alert('Ағымдағы массив:\n\n' + myArray.join('\n'))
// алфавит бойынша сұрыптау
  myArray.sort();
alert('Алфавит бойынша сұрыпталған:\n\n' + myArray.join('\n'))
</SCRIPT>

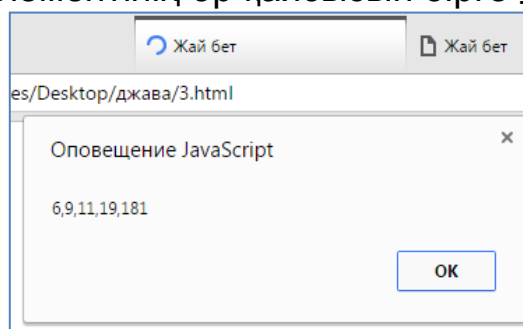
```



Сурет 39. Бағдарлама нәтижелері

### Өзіндік тапсырмалар:

1. «Алматы, Астана, Қостанай, Шымкент, Қарағанды, Семей, Павлодар» қала атауларынан тұратын массив берілген. Оларды алфавит бойынша, кері ретпен сұрыптаңыз, ұзындығын санаңыз.
2. push() әдісін қолданып, «Роза, Астра, Түймедақ» массивінің соңына «Гүлдер» элементін қосыңыз.
3. unshift() әдісін қолданып, «Роза, Астра, Түймедақ» массивінің алдына «Гүлдер» элементін қосыңыз.
4. concat() әдісін қолданып, «Алматы, Астана, Қостанай, Шымкент, Қарағанды, Семей, Павлодар» бірінші массивіне, «Қазақстан қалалары» екінші массивін біріктіріп үшінші массивті шығарыңыз.
5. Массивтің бес элементінің әр қайсысын бірге ұлғайту керек.



Сурет 40. Бағдарлама нәтижесі

6. Апта күндері массивін құрыңыз. For циклы көмегімен барлық апта күндерін шығарыңыз, ал демалыс күндерін қарайтылған (жирный) түрінде шығарыңыз.
7. Апта күндері массивін құрыңыз. For циклы көмегімен барлық апта күндерін шығарыңыз, ал ағымдағы күнді курсив түрінде шығарыңыз. Ағымдағы күн day айнымалысында сақталуы керек.

8. [15, 20, 30, 50, 274, 6000] массиві берілген. 1, 2 немесе 5 цифрларынан басталатын массивтегі сандарды ғана экранға шығару керек.
9. 1, 2, 5, 9, 4, 13, 4, 10 элементтері бар массив берілген. For циклі және if операторы көмегімен 4-тең болатын элементі бар екенін анықтау керек. Егер бар болса – экранға 'Бар!' деп шығару керек.
10. Кез-келген элементтері бар массив берілген. Массивтегі оң элементтердің қосындысын табу керек.
11. 2, 5, 9, 15, 0, 4 элементтері бар массив берілген. For циклі және if операторы көмегімен 3-тен үлкен, бірақ 10нан кіші элементтерді шығару керек.
12. [2, 3, 4, 5] элементтері бар массив берілген. Бұл массивте элементтердің көбейтіндісін табу үшін for циклын пайдаланыңыз.
13. 4, 2, 5, 19, 13, 0, 10 элементтері бар массив берілген. Оның элементтерінің кубтарының қосындысының квадрат түбірін табу керек. Мұны шешу үшін for циклын пайдаланыңыз.
14. arr массиві берілген. Оның элементтерінің арифметикалық ортасын табыңыз. Жиымдағы тапсырманы 8, 15, 20, 25, 49, 76 элементтерімен тексеріңіз.
15. Төменде код берілген.

```

<script type='text/javascript'>
var country=["Казахстан", "Германия", "Италия", "Россия",
"Австралия", "Оңтүстік Америка", "Евразия", "Африка",
"Антарктида"];
var animal=["Аю", "Жолбарыс", "Аққу", "Торғай", "Қарлығаш",
"Көгершін", "Арыстан", "Тиін"];
// мұнда код жазыңыз
</script>

```

Осы код бойынша келесі тапсырмаларды орындаңыз:

- *Country* массивінен мемлекеттерді жойып, нәтижесін *continent* массивіне сақтаңыз (**splice()** әдісі).
- *animal* массивінен құс болмайтын аңдарды жойып, нәтижесін *bird* массивіне сақтаңыз (**splice()** әдісі).
- Алынған массивтерді *contibird* массивіне біріктіріп, нәтижесін экранға шығарыңыз (**concat()** әдісі).
- Алынған массивтің бас жағына 'Солтүстік Америка' элементін қосыңыз, ал соңғы жағына 'Тотықұс' элементін қосыңыз және нәтижесін экранға шығарыңыз ( **push()** және **unshift()** әдістері).
- *Contibird* массивін сұрыптаңыз, нәтижесін экранға шығарыңыз (**sort()** әдісі).

**Өзін-өзі бақылауға арналған сұрақтар:**

1. JavaScript тілінің қандай оқиғалары мен объектілерін білесіз?
2. Array объектісі не үшін арналған?

3. Массивті сұрыптау әдісі қандай?
4. Массивті біріктіру әдісі қандай?
5. push() және unshift() әдістерінің атқаратын қызметі?

## 2.8 Практикалық сабақ №8 STRING ОБЪЕКТІСІ

**Жұмыстың мақсаты:** JavaScript тілінде жолдық функцияларды қолданып есептерді шығарып үйрену.

### Қысқаша мәліметтер:

JavaScript-да кез келген мәтіндік деректер жол болып табылады. Бірқатар басқа тілдерде болатын «символдың» жеке түрі жоқ.

Жолдық объектіні құру үшін келесі өрнек пайдаланылады:

*айнымалы\_атауы = new String ("жолдық\_мәні")*

Дегенмен, жолдық объектіні қарапайым меншіктеу операторы арқылы құруға болады:

*айнымалы\_атауы = "жолдық\_мәні"*

Жолдар қосарлы немесе жалғыз тырнақшалар арқылы жасалады:

```
var text = "жол";
```

```
var anotherText = 'тағы бір жол';
```

```
var str = "012345";
```

JavaScript-да қосарланған және жалғыз тырнақшалар арасында ешқандай айырмашылық жоқ.

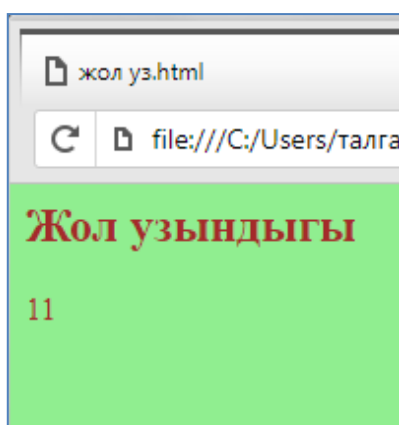
*Кесте 16. String объектісінің әдістері*

Әдіс атауы	Тағайындалуы	Жазылуы
Length	жолдың ұзындығы	x = mystr.length
toLowerCase	төменгі регистр кіші әріп	x.toLowerCase()
toUpperCase	жоғарғы регистр	x.toUpperCase()
Split	массивке жолды бөлу	split([separator][, limit])
Substr	Ішкі жолды қиып алу	жол.substr(индекс [,ұзындық])
Substring	Ішкі жолды қиып алу	str.substring(6,10)
Slice	Ішкі жолды қиып алу	document.write(str.slice(2, 5));
Concat	жолдарды қосу	жол1.concat(жол2)
indexOf	сәйкестіктерді басынан іздеу	str.indexOf("горы")
lastIndexOf	сәйкестіктерді соңынан іздеу	lastIndexOf(searchValue[, fromIndex])
replace	жолдағы мәтінді ауыстыру	x.replace('HTML','JavaScript')

charAt	Жол таңбасы (символ)	жол.charAt(индекс)
charCodeAt	коды бойынша символдық таңба	str.charCodeAt(i)
String.fromCharCode	таңба (символ) коды	fromCharCode(num1,...,numN)

**Мысал-1.** Берілген жолдың ұзындын санау керек.

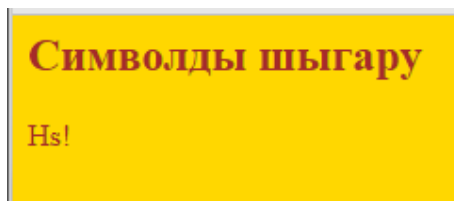
```
<html>
<head> </head>
<body bgcolor=lightgreen text = brown>
  <h2> Жол ұзындығы </h2>
  <script>
    mystr = «Мен-студент»
    x = mystr.length
    document.writeln(x)
  </script>
</body>
</html>
```



Сурет 41. Бағдарлама нәтижесі

**Мысал-2.** Берілген жолдың сөздерінің алғашқы әріптерін шығару керек.

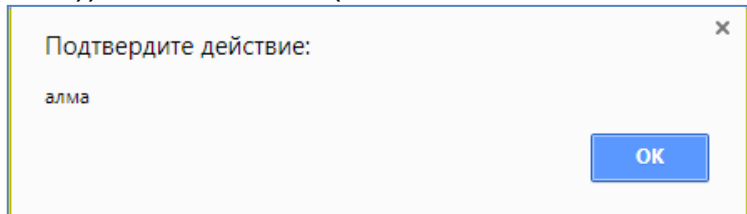
```
<script>
  x = «Hello students!»
  a = x.charAt(0)
  b = x.charAt(6)
  c = x.charAt(14)
  document.writeln(a,b,c)
</script>
```



Сурет 42. Сөздерінің алғашқы әріптерін шығару

**Мысал-3.** «Анель алма алды» жолы берілген. Осы жолдан «алма» сөзін үш түрлі тәсілмен (substr, substring, slice) қиып алу керек.

```
var str = 'Анель алма алды';  
alert(str.substr(6, 4)); //substr (қай символдан бастап, неше  
символ)  
alert(str.substring(6,10)); //substring (қай символдан бастап, неше  
символ)  
alert(str.slice(6,10)); //slice (қай символдан бастап, неше символ)
```



Сурет 43. Жолдан «алма» сөзін қиып алу  
**Өзіндік тапсырмалар:**

1. Кез келген жолдың ұзындын санап, нәтижесін бөлек терезеге шығару керек.
2. «Жоғары», «оқу», «орны» сөздерін біріктіріп, бір сөйлем шығару керек. concat әдісін қолданыңыз.
3. «Аю@Бота@Тай» жолы берілген. @ символын !' символымен алмастыру керек. Replace әдісін қолданыңыз.

**Өзін-өзі бақылауға арналған сұрақтар:**

1. Жол дегеніміз не?
2. Жолдың қандай әдістері бар?
3. Жолдағы мәтінді ауыстыру әдісі?
4. Ішкі жолды қиып алу әдісі?

## 2.9 Практикалық жұмыс №9 DATE ОБЪЕКТІСІ

**Жұмыстың мақсаты:** JavaScript тілінің Date объектісін қолданып есептерді шығарып үйрену.

### Қысқаша мәліметтер:

JavaScript тілінің ішкі объектілерін қарастыруды өте пайдалы болып саналатын Date объектісінен бастайық. Бұл объект күн-ай мерзімімен (датамен) және уақытпен жұмыс істеу үшін керек.

### 9.1. Date объектісінің экзеплярін жасау.

Объект экзеплярін жасау үшін (Date объектісінің ғана емес, одан басқасының да) JavaScript тілінде new түйінді сөзі қолданылады:

```
var now = new Date();
```

Енді now айнымалысы Date объектісі экзепляры болып табылады да, ол үстіміздегі дата мен уақытты береді.

Ағымдағы күн мен уақыты бар объектінің экзеплярін құру: new Date()

```
var a = new Date();
```

Ағымдағы күн мен уақыты «a» айнымалысында.

Конструктордың берілген аргументтерімен күн мен уақыты бар объектінің экзеплярін құру:

```
new Date(жыл, ай, күн[,сағат [,минут [,секунд [,мс]?]?]??)
```

Мысалы

```
var c = new Date(1958, 4, 21, 10, 15);
```

«с» айнымалысында күні – 21 мамыр 1958 және 10 сағат 15 мин. Айлар нөмірі 0–ден басталады.

Жалпы экземпляр жасау былай орындалады:

```
var айнымалы = new Date(параметрлер);
```

Мысал,

```
var now = new Date();
```

```
var birthday =
```

```
new Date(1954, 1, 8);
```

```
var bell = new Date(2003, 0, 14, 12, 20, 0);
```

## **9.2. Date объектісінің әдістері**

Әдіс бұл объект үшін немесе объектімен орындалатын әрекет. Әрбір объектінің әдістері бар (Кесте 17 мен кесте 18).

*Кесте 17. Күн мен уақытты алу әдістері*

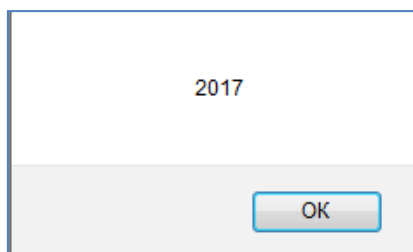
<b>Әдіс</b>	<b>Түсінігі</b>
getTime()	Объектінің қарапайым мәнін көрсетеді (қайтарады).
getFullYear()	Жергілікті уақыт бойынша жылдың санын көрсетеді
getMonth()	Айды жергілікті уақыт бойынша көрсетеді. (санау 0-ден басталады, яғни 0-қаңтар)
getDate()	Жергілікті уақыт бойынша айдың күнін көрсетеді.
getDay()	Жергілікті уақыт бойынша апта күнін көрсетеді. (0-жексенбі, ... , 6-сенбі)
getHours()	Жергілікті сағатты көрсетеді. (0-ден 23-ке дейін)
getMinutes()	Жергілікті уақыт бойынша минутты көрсетеді. (0-ден 23-ке дейін)
getSeconds()	Жергілікті уақыт бойынша секундты көрсетеді. (0-ден 23-ке дейін)

*Кесте 18. Күн мен уақытты орнату әдістері*

<b>Әдіс</b>	<b>Түсінігі</b>
setDate()	мерзім түрінде күнді орнату
setMonth()	айды орнату
setFullYear()	жылды орнату
setHours()	сағатты орнату
setMinutes()	минутты орнату
setSeconds()	секундты орнату
setMilliseconds()	миллисекундты орнату

**Мысал-1.** Бөлек сұхбаттық терезеге ағымдағы жылдың нөмірін шығару керек.

```
<body>
<script>
  var today= new Date();
  var god=today.getFullYear();
  alert(god);
</script>
</body>
```

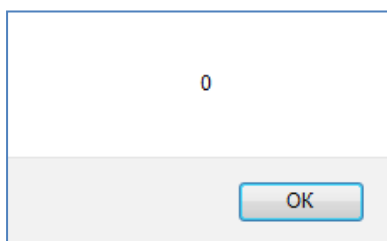


Сурет 44. Бағдарлама нәтижесі

Бұл сценарий `Date` объектісінің данасын жасайды. Ол `today` атауын және ағымдағы күнді қамтиды. `today` объектісіне `getFullYear()` әдісі пайдалану үшін `today.getFullYear()` конструкциясы пайдаланылады. Ол сол күннен бастап жыл мәнін көрсетеді. Бұл мән `god` айнымалысымен тағайындалған және **`alert(god)`** әдісін қолдану арқылы терезеде көрсетіледі.

**Мысал-2.** 21 мамыр 2017 жыл күні үшін апта күнінің нөмірі терезесін шығару керек.

```
<body>
<script>
  var dr = new Date(2017, 4, 21);
  dn=dr.getDay()
  alert(dn)
</script>
</body>
```



Сурет 45. Бағдарлама нәтижесі

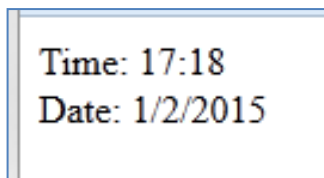
Бұл сценарийде `Date` объектісінің данасын жасайды. Оның `dr` атауы бар және 21 мамыр 2017 күнін қамтиды. `getDay()` әдісін `dr` пайдалану үшін `dr.getDay()` пайдаланылады. Ол бұл күн үшін апта күнін қайтарады. Бұл мән `dn` айнымалысымен тағайындалады және **`alert(dn)`** әдісін қолдану арқылы терезесінде көрсетіледі. Сәрсенбі 0 нөмірі болып көрсетіледі.

**Мысал-3.** `Date` объектісін қолданып, күн мен уақытты шығаратын скрипт жазу.

```
<script language="JavaScript">
<!-- hide
now= new Date();
```



```
document.write("Time: " + now.getHours() + ":" + now.getMinutes() +
"<br>");
document.write("Date: " + (now.getMonth() + 1) + "/" + now.getDate() + "/"
+
(1900 + now.getYear()));
// -->
</script>
```

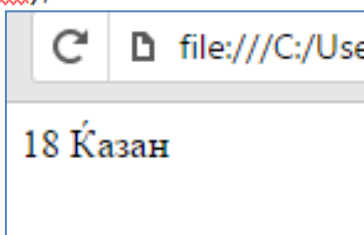


Сурет 46. Бағдарлама нәтижесі

Мұнда *now* атаулы *Date* объектісінде көрсетілген, күн мен уақытты экранға шығару үшін *getHours()* әдісі қолданылды. Және де жылдың жазбасына 1900 саны қосылған, өйткені *getYear()* әдісі 1900 жылдан кейін өткен жылдардың санын көрсетеді.

**Мысал-4.** Орнатылған күн нөмірі мен орнатылған айдың нөмірі бойынша ай атауын шығару.

```
var months = ["Қаңтар", "Ақпан", "Наурыз", "Сәуір", "Мамыр",
"Маусым",
"Шілде", "Сәуір", "Қыркүйек", "Қазан", "Қараша",
"Желтоқсан"];
var myDate = new Date();
myDate.setDate(18);
myDate.setMonth(9);
var fullDate = myDate.getDate() + " " + months[myDate.getMonth()];
document.write(fullDate);
```



Сурет 47. Ай атауын шығару

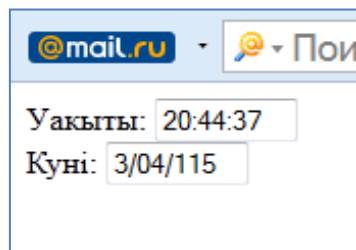
**Мысал-5.** *Date* объектісін қолданып, экранда жұмыс істеп тұрған сағаттың бейнесін құратын скриптті жазып шығару.

```
<html>
<head>
<script Language="JavaScript">
<!-- hide
var timeStr, dateStr;
function clock() {
now= new Date();
// уақыт
hours= now.getHours();
```

```

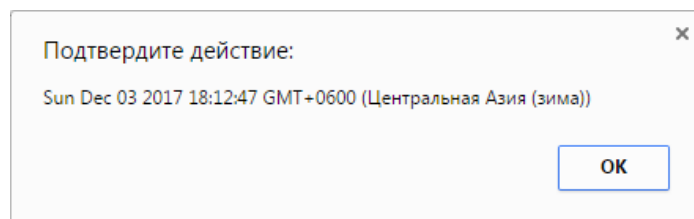
minutes= now.getMinutes();
seconds= now.getSeconds();
timeStr= "" + hours;
timeStr+= ((minutes < 10) ? ":0" : ":") + minutes;
timeStr+= ((seconds < 10) ? ":0" : ":") + seconds;
document.clock.time.value = timeStr;
// күн
date= now.getDate();
month= now.getMonth()+1;
year= now.getYear();
dateStr= "" + month;
dateStr+= ((date < 10) ? "/0" : "/") + date;
dateStr+= "/" + year;
document.clock.date.value = dateStr;
Timer= setTimeout("clock()",1000);
}
// -->
</script>
</head>
<body onLoad="clock()">
<form name="clock">
Уақыты:
<input type="text" name="time" size="8" value=""><br>
Күні:
<input type="text" name="date" size="8" value="">
</form>
</body>
</html>

```



Сурет 48. Бағдарлама нәтижесі  
**Өзіндік тапсырмалар:**

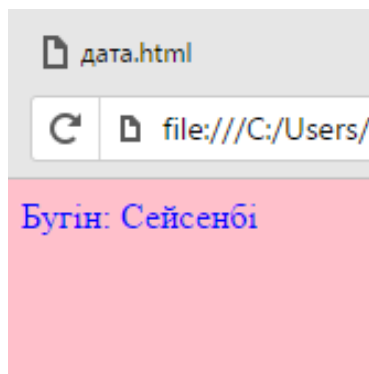
1. Бөлек сұхбаттық терезеге ағымдағы айдың, күннің нөмірін шығару керек.



Сурет 49. Бағдарлама нәтижесі

2.21 мамыр 2017 жыл күні үшін апта күнінің нөмірі терезесін шығару керек.

3. Орнатылған апта күнінің нөмірі бойынша апта күнінің атауын шығару.



Сурет 50. Бағдарлама нәтижесі

4. Күннің уақытына және сағатқа байланысты, уақыт мезгілі мен ағымдағы уақытын шығару керек.

Кайырлы кеш, ағымдағы уақыт: 19:48:35

Сурет 51. Бағдарлама нәтижесі

**Өзін-өзі бақылауға арналған сұрақтар:**

1. JavaScript тілінің қандай оқиғалары мен объектілерін білесіз?
2. OnLoad оқиғасың қызметі қандай?
3. onClick оқиғасың қызметі қандай?
4. Date объектісі не үшін арналған?

## 2.10 Практикалық жұмыс №10 WINDOW ОБЪЕКТІСІ

**Жұмыстың мақсаты:** JavaScript тілінің Window объектісін қолданып есептерді шығарып үйрену.

**Қысқаша мәліметтер:**

### 10.1. Window объектісі

Window иерархияның төбесінде орналасады. Бұл қалған объектілердің одан төмен орналасып, соның қасиеттері түрінде қарастырылатынын білдіреді. Window объектісінде браузер құжатты көрсетіп тұрған терезенің қасиеттері мен тәсілдері де болады. Үш тәсіл бізге бұған дейін белгілі болған, олар: alert, prompt және confirm.

Window иерархияның төбесінде орналасады. Бұл қалған объектілердің одан төмен орналасып, соның қасиеттері түрінде қарастырылатынын білдіреді. Window объектісінде браузер құжатты көрсетіп тұрған терезенің қасиеттері мен тәсілдері де болады.

Үш тәсіл бізге бұған дейін белгілі болған, олар:

alert, prompt және confirm.

Егер былай жазылса:

*Window.alert ("alert функциясы- бұл window объектісінің әдісі");*

— онда браузер осы хабарламаны шығаратын қосалқы терезе ашады.

Бірақ бұрын былай жазатынбыз:

*alert("alert функциясы- бұл window объектісінің әдісі ");*

Басты объект болып саналатын window объектісінің атын жазбаса да болады.

Window объектісі объект иерархиясының жоғарғы жағында болып табылады және барлық басқа браузер нысандарын қамтиды. Window объектісі ағымдағы шолғыш терезесін және оның мазмұнын сипаттайды.

*Кесте 19. Window объектісінің әдістері*

<b>Әдіс</b>	<b>Түсініктемесі</b>
open	Жаңа браузер терезесін ашады.
close	Браузердің терезесін жабады.
alert, prompt, confirm	Стандартты сұхбаттық панельдер.
setInterval	берілген миллисекунд саны арқылы функцияның қайта орындалуын көрсетеді.
setTimeout	Миллисекундпен берілген саны арқылы функцияны іске қосады.
clearInterval	setInterval әдісінің әрекетін алып тастайды.
clearTimeout	setTimeout әдісінің әрекетін алып тастайды.

### **Жаңа терезені ашу және жабу**

```
var айнымалы = open();  
var айнымалы = open(файл);  
var айнымалы = open(файл, терезе_атауы);  
var айнымалы = open(файл, терезе_атауы, терезе_параметрі);  
айнымалы.close();
```

– *айнымалы* – құрылатын терезенің көшірмесі.

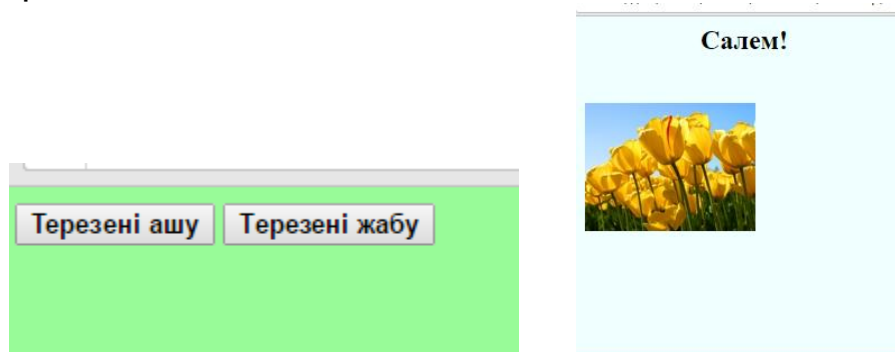
– *файл* - Сіз жинақталатын терезесінде көрсеткіңіз келетін файлдың жол атауы. Егер көрсетілмесе, бос терезе құрылады. Содан кейін терезесінде құжаттың қалыптастыру document.write пайдаланып, қарқынды іске асырылуы тиіс.

– *терезе\_атауы* - терезе жол атауы; Ол құрылған терезеге (объект терезе «ID» құны) хабарлау үшін пайдаланылады.

– *терезе\_параметрі* – жол, терезе қасиеттері; Терезе параметрлері *терезе\_параметрі* жолында белгіленеді:

```
"параметр1 = мән 1, параметр2 = мән2,..., параметрN = мән N"
```

**Мысал-1.** Батырманы басқанда жаңа терезе жасайтын, ол терезеге бар html-құжаты (11.html) шығарылады және басқа түймені басқанда құрылған терезе жабылатын қосымшаны жазыңыз.

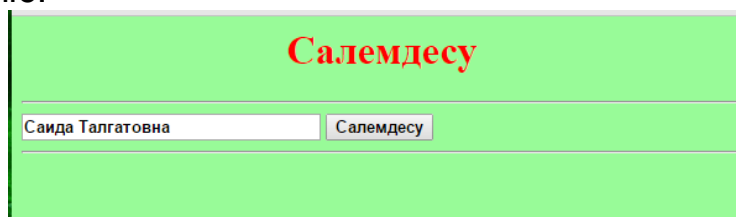


Сурет 52. Бағдарлама нәтижесі

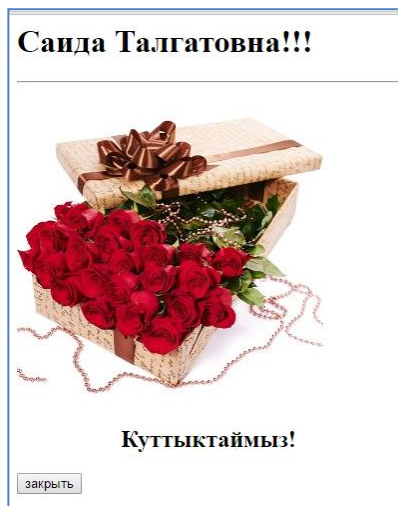
```
<html>
<head>
<style type="text/css">
body {background-color:PaleGreen;}
input {font-weight:bold;}
</style>
<script>
function op()
{
win=window.open("11.html", "www","width=300, height=300");
}
</script>
</head>
<body>
<form>
<input type=button value="Терезені ашу" OnClick="op()">
<input type=button value="Терезені жабу" OnClick="win.close()">
</form>
</body>
</html>
```

### Өзіндік тапсырмалар:

1. Форма өрісіне адамның есімін енгізетін және батырманы басқанда адамға сәлемдесу жасайтын жаңа терезені құратын қосымшаны жазыңыз. екінші терезедегі құжат document.write() әдісі көмегімен динамикалық қалыптастырылады. Жаңа құжатты жабу батырмасы сол жерде болуы тиіс.



Сурет 53.Веб-бет үлгісі



Сурет 54. Екінші веб-беттің үлгісі

2. Форманың мәтіндік өрістерінен тұратын мәліметтерді біріктіріп, жеке бетте осы деректер бойынша өмірбаянды қалыптастыру керек.

Сауалнама	
Аты	Бакенова
Тегі	Саида
Әкесінің аты	Талгатовна
Туған жылы	1996
Туған жері	Қостанай
Ұнататын ісім	кино
Ұнатпайтын ісім	ыдыс жуу
<input type="button" value="Өмірбаян терезесін ашу"/>	

Мен туралы
Мен Бакенова Саида Талгатовна 1996 жылы Қостанайда дүниеге келдім. Ұнататын ісім - комедия жанрындағы кинолады көру. Ал ұнатпайтын ісім - ыдыс жуу.
<input type="button" value="Шығу"/>

Сурет 55. Сауалнама деректерін енгізу үлгісі

**Өзін-өзі бақылауға арналған сұрақтар:**

1. Window объектісі не үшін арналған?
2. Window объектісінің қандай әдістерін білесіз?
3. Жаңа терезені ашу және жабу қалай жүзеге асады?

## 2.11 Практикалық жұмыс №11 ФОРМА ЭЛЕМЕНТТЕРІ – ЖАЛАУШАЛАР, РАДИОБАТЫРМАЛАР, ТІЗІМДЕР

**Жұмыстың мақсаты:** JavaScript тілінің форма элементтерін – жалаушаларды, радиобатырмаларды, тізімдерді қолданып есептерді шығарып үйрену.

**Қысқаша мәліметтер:**

### 11.1. Input объектісінің checked қасиеті

Жалаушалар мен радиобатырмалар құжатта `<input>` тегімен анықталады. Атрибуттары:

- `type="checkbox"` – жалауша
- `type="radio"` – радиобатырма

Жалаушалар бірі басқана байланысты емес. Оларды кез келген комбинацияда орнатуға және тастауға болады. Радиобатырмалар бірнеше баламадан бір нұсқаны таңдау үшін тағайындалған.

Жалаушалар мен радиобатырмалар input объектісінің түрлері. Input объектісінің checked қасиеті бар. Бұл қасиет құсбелгі белгіленгендігіне мәні true болады, (радиобатырма таңдалған) және false –кері жағдайда.

**Мысал-1.** Көбейту кестесін тексеру үшін қосымша жазыңыз. Дұрыс мысалдарды жалаушалармен белгілеу қажет. Егер қателер жоқ болса, форма жолында «дұрыс», олай болмаған жағдайда «дұрыс емес» сөзі шығарылсын.

Сурет 56. Веб-бет үлгісі

```

<head>
<script>
function checkAll()
{
    var ans='Дұрыс емес';
    // p1 айнымалысы c1 атаулы жалауша
    var p1=document.getElementById('c1');
    // p2 айнымалысы c2 атаулы жалауша
    var p2=document.getElementById('c2');
    // p3 айнымалысы c3 атаулы жалауша
    var p3=document.getElementById('c3');
    // p4 айнымалысы result атаулы мәтіндік өріс
    var p4=document.getElementById('result');
    // егер c1 жалаушасы көрсетілсе, онда c2 көрсетілмеген
    // және c3 жалаушасы көрсетілген
    if (p1.checked && !p2.checked && p3.checked) ans='Дұрыс';
    // нәтижесі
    p4.value=ans;
}
</script>
</head>
<body>
<p> Дұрыс тужырымдаманы белгіленіз:</p>
<form>
<input type="checkbox" id="c1" name="c1"> 2*2=4 <br/>

```



```

<input type="checkbox" id="c2" name="c2"> 2*5=30<br/>
<input type="checkbox" id="c3" name="c3"> 6*8=48<br/>
<input type="button" value="Тексеру" onClick="checkAll();">
<input type="text" id="result" name="result" value="">
</form>
</body>

```

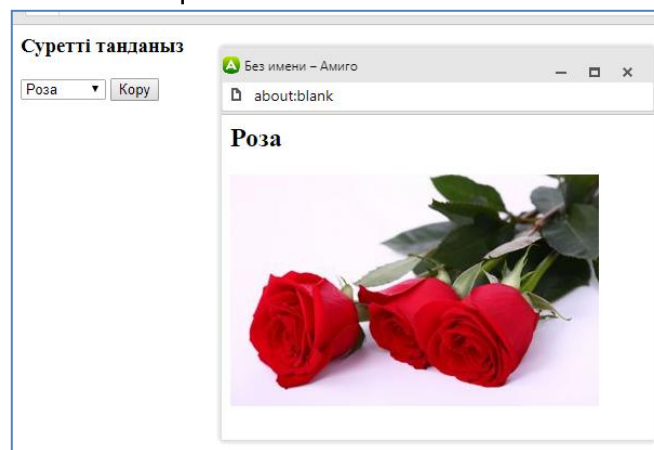
## 11.2. Option объектісінің қасиеттері

Ашылмалы тізімді құру үшін <select> элементі пайдаланылады, ол бірнеше <option> элементтерді қамтиды.

Бұл option нысандардың әрқайсысы мынадай қасиеттерге ие:

- text – тізім жолдың мәтінін тиісті.
- selected – егер жол таңдалса true тең және false – кері жағдайда.

**Мысал-2.** Тізімінен суретті таңдайтын және батырмасын басқан кезде таңдалған атауын тақырып түрінде және суреттің өзін жаңа терезеге шығаратын қосымша жазыңыз.



Сурет 57. Веб-бет үлгісі

```

<title> Тізім </title>
<script type="text/JavaScript">
function GO()
{
var p1=document.getElementById('k1');
var p2=document.getElementById('k2');
var p3=document.getElementById('k3');
//жаңа құжаттың тұрақты бөлігін қалыптастыру
st1="<body><h2>"
st3="</h2><body>"
//жаңа құжаттың өзгермелі бөлігін қалыптастыру
if (p1.selected)
{ st2=p1.text; st4=" 'orhideya.jpg' " }
if (p2.selected)
{ st2=p2.text; st4=" 'roza.jpg' " }
var win=window.open("","","width=400,height=300");

```

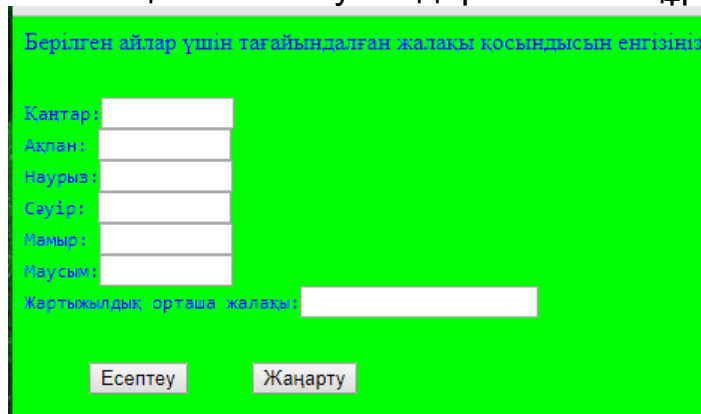


```

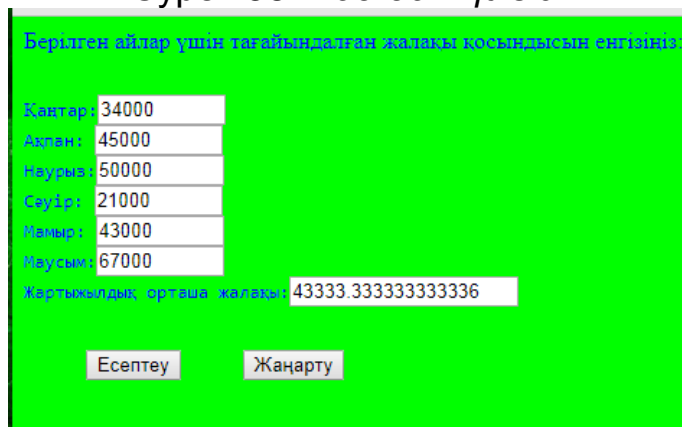
win.document.open();
// қалыптасқан кужат
str=st1+st2+st3+st4+st5
//терезеге шығару
win.document.write(str);
win.document.close();
}
</script>
</head>
<body>
<form>
<h3> Суретті танданыз</h3>
<select>
<option id='k1'> Орхидея </option>
<option id='k2'> Роза </option>
</select>
<input type="button" value="Копу" onclick="GO()">
</form>
</body>

```

**Мысал-3.** Орташа жалақыны есптеу бағдарламасын құру керек.



Сурет 58. Веб-бет үлгісі



Сурет 59. Нәтижесі

```

<head>
<title> Орташа табысты есептеу </title>

```

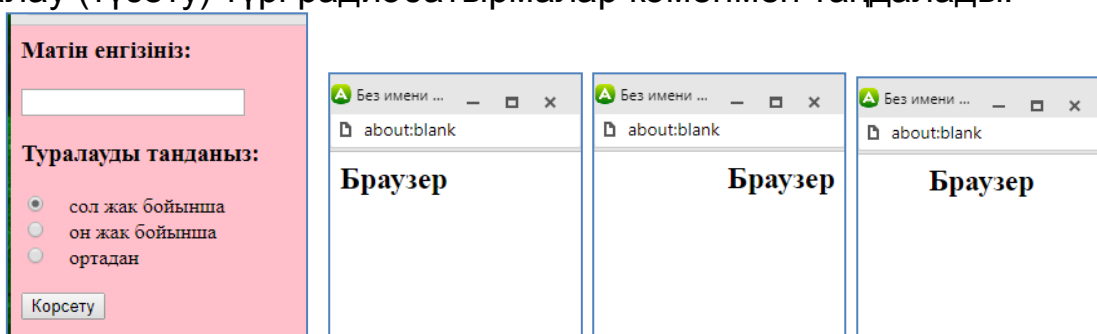
```

<script language= "JavaScript">
<!--//
function val (obj)
{var a1=1*obj.num1.value;
var a2=1*obj.num2.value;
var a3=1*obj.num3.value;
var a4=1*obj.num4.value;
var a5=1*obj.num5.value;
var a6=1*obj.num6.value;
var s =(a1+a2+a3+a4+a5+a6)/6
obj.res.value=s
}
//-->
</script>
</head>
<body bgcolor=lime text=blue>
Берілген айлар үшін тағайындалған жалақы қосындысын енгізіңіз:
<pre>
<form name="form1">
Қаңтар:<input type="text" size=8 name="num1">
Ақпан: <input type="text" size=8 name="num2">
Наурыз:<input type="text" size=8 name="num3">
Сәуір: <input type="text" size=8 name="num4">
Мамыр: <input type="text" size=8 name="num5">
Маусым:<input type="text" size=8 name="num6">
Жартыжылдық орташа жалақы:<input type="text" size=18
name="res"<br>
<br>
<input type="button" value=Есептеу onClick="val(form1)">
<input type="reset" value="Жаңарту">
</form></pre></body></html>

```

### Өзіндік тапсырмалар:

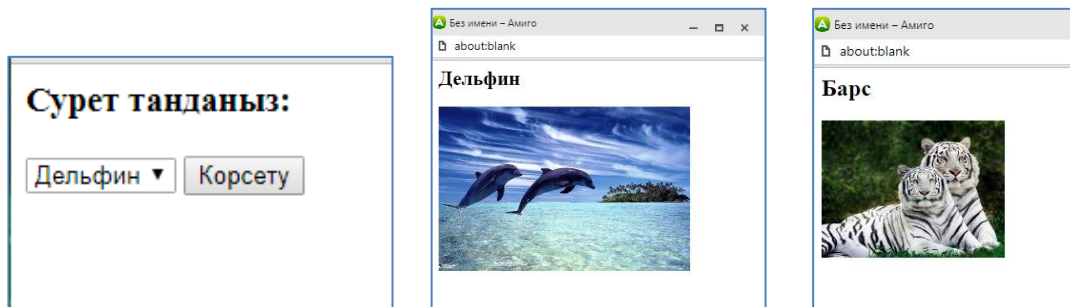
1. Жаңа терезе құратын, оған форманың мәтіндік өрісіне енгізілген мәтінді тақырып түрінде шығаратын қосымшаны жазыңыз. Тақырып үшін туралау (түзету) түрі радиобатырмалар көмегімен таңдалады.



Сурет 60. Веб-бет үлгісі

2. Көбейту кестесін тексеру үшін қосымша жазыңыз. Дұрыс мысалдарды жалаушалармен белгілеу қажет. Егер қателер жоқ болса, «Тексеру» батырмасын шерткенде басқа терезеде «дұрыс», олай болмаған жағдайда «дұрыс емес» сөзі шығарылсын.

3. Тізімнен таңдаған жануардың атауына байланысты, *Көрсету* батырмасын шерткенде сәйкесінше суреті шығатын бағдарлама құру керек.



Сурет 61. Веб-бет үлгісі

**Өзін-өзі бақылауға арналған сұрақтар:**

1. Input объектісінің checked қасиеті не үшін арналған?
2. Option объектісінің қандай қасиеттері бар?
3. Checkbox қасиеті не үшін арналған?

## 2.12 Практикалық жұмыс №12 IMAGE ОБЪЕКТІСІ

**Жұмыстың мақсаты:** JavaScript тілінің Image объектісін қолданып есептерді шығарып үйрену.

### Қысқаша мәліметтер:

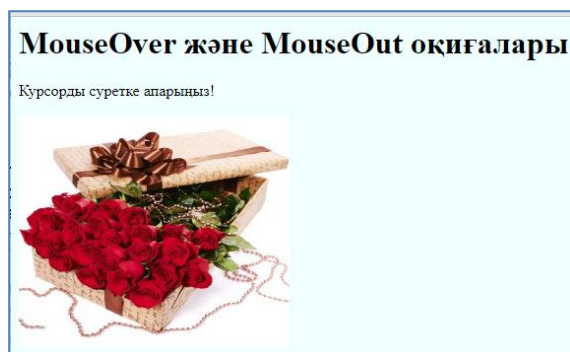
Құжаттағы барлық суреттер (img элементтері) Image объектісінің данасы болып табылады. Әрбір Image объектінің src қасиеті бар, оны өзгертуге болады. Мұны пайдалана отырып, web-бетте қатысатын графикалық бейнелерге өзгерістер енгізуге болады.

### 12.1. MouseOver және MouseOut тышқанның оқиғалары

Mouseover және mouseout оқиғалары, тышқан меңзері элементке жылжығанда немесе оның шектері тиісінше кеткенде болып жатады. Оқиғаларды өңдеушілердің onmouseover және onmouseout тиісінше аттары бар. 1-мысалда бұл қасиетті өзгертеміз, қандай mouseover немесе mouseout оқиғасы туындайтынына қарамастан.

**Мысал-1.** Мысалда меңзерді суретке әкелген кезде, оның орнында басқа сурет пайда болады.

Оқиғаны өңдеуге арналған логикалық параметрі бар doEvent функциясы құрылды. Орнату кезінде курсорды суретке әкелгенде (MouseOver оқиғасы туындайды) параметрі true doEvent функциясы шақырылады. Бұл жағдайда суреттің src қасиетінің мәні b.jpg өзгереді. Курсорды суреттен алып тастаған кезінде (onMouse оқиғасы туындайды) параметрі false doEvent функциясы шақырылады. Бұл жағдайда суреттің src қасиетінің мәні feя.jpg өзгереді.



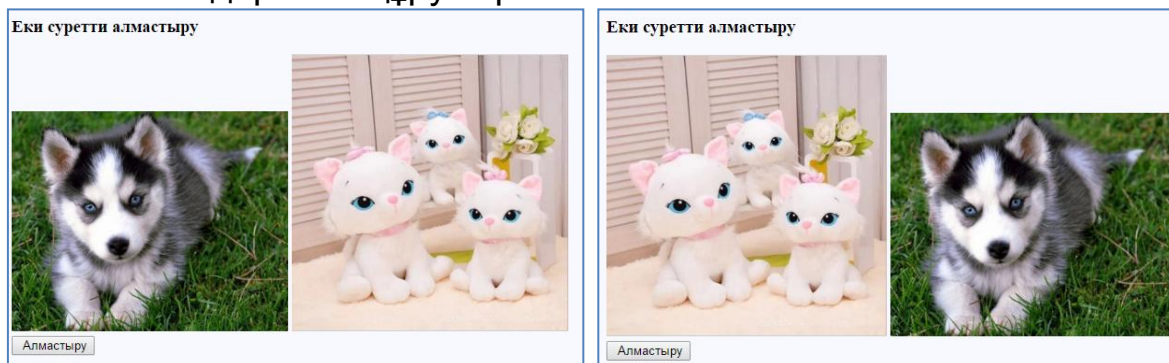
Сурет 62. Веб-бет үлгісі

```

<head>
<style>
body {background-color:azure;}
img {width:275; height:235} </style>
<script>
function doEvent(type)
{
    // cv айнымалысы pic атаулы img объектісі
    var cv=document.getElementById('pic');
    // егер функцияның параметрі true болса, онда
    // cv объектісінің src қасиетін б.jpg ауыстырамыз
    if(type) cv.src="б.jpg";
    // кері жағдайда фея.jpeg суретін көрсетеміз
    else cv.src="фея.jpeg";
}
</script>
</head>
<body>
    <h1> MouseOver және MouseOut оқиғалары</H1>
    <p>Курсорды суретке апарыңыз!</p>
    
</body>

```

**Мысал-2.** Алмастыру батырмасын шерткенде, екі сурет бір-бірімен алмасатын бағдарлама құру керек.



Сурет 63. Алмастыру бағдарламасы

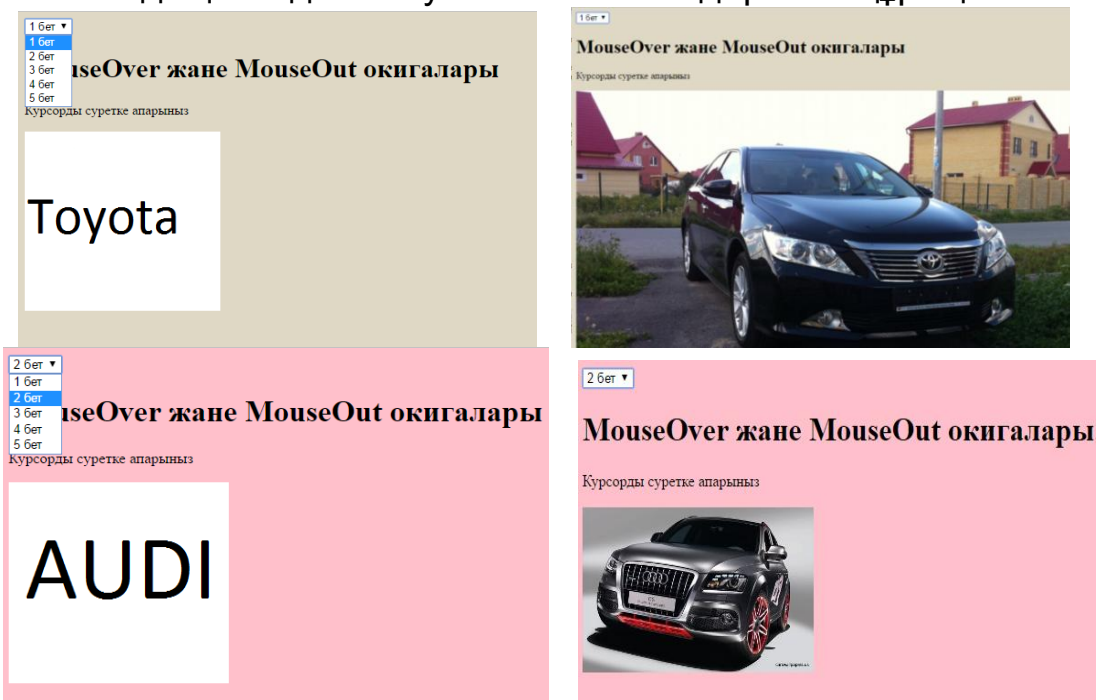
```

<script language="JavaScript">
<!--//
function chpict ()
{var d=document
var k=d.pm1.src
d.pm1.src=d.pm2.src
d.pm2.src=k
}
//-->
</script>
</head>
<body bgcolor=F8F8FF>
<h3> Екі суретті алмастыру </h3>
<IMG src="11.jpg" name=pm1 width=300>
<IMG src="12.jpg" name=pm2 width=300>
<FORM name="form1">
<input
type="button" value = " Алмастыру" onClick="chpict () ">
</form>
</body>

```

### Өзіндік тапсырмалар:

1. Қажетті бетті таңдағанда сол веб бет ашылатын, тінтуірді автокөлік атауына әкелген кезде, оның фотосы пайда болады. Курсорды алып тастаған кезінде қайтадан атауы шығатын бағдарлама құрыңыз.



Сурет 64. Веб-беттер үлгісі

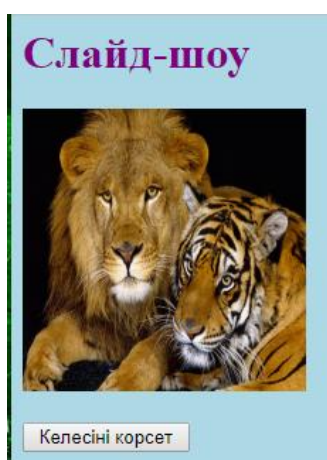
2. «Баста» батырsmасын шерткенде, терезедегі суреттер бірнеше секунд сайын ауысып отыратын бағдарлама құрыңыз.





Сурет 65. Веб-беттер үлгісі

3. Осы суреттер үшін слайд-шоу ұйымдастыру, батырманы басқан кезде әрбір келесі сурет автоматты түрде көрсетілетін бағдарлама құрыңыз.



Сурет 66. Веб-беттер үлгісі

**Өзін-өзі бақылауға арналған сұрақтар:**

1. Image объектісінің қандай қасиеттері бар?
2. mouseover және mouseout оқиғалары не үшін арналған?

## ҚОЛДАНЫЛҒАН ӘДЕБИЕТТЕР

1. Айзекс С. Dynamic HTML: пер. с англ. – СПб.: БХВ-Петербург, 2001. -496 с.
2. Айзекс С., «Dynamic HTML. Секреты создания интерактивных Web - страниц» М.: Издательский дом «Вильяме», 2008.
3. Айтбенова А.А. WEB-дизайн негіздері: жоғарғы оқу орындарының студенттеріне арналған оқу құралы. –Қостанай: ҚМПИ, 2015. - 237б.
4. Бөрібаев Б., Мадьярова Г.А. WEB-технологиялар: Оқулық. - Алматы: ЖШС РПБК «Дәуір», 2011.- 360 б.
5. Дмитриева М.В. Самоучитель JavaScript. - СПб.: БХВ-Петербург, 2003. - 512 с.: ил.
6. Дригалкин В.В. HTML в примерах. Как создать свой Web-сайт: Самоучитель. 2004.
7. Дронов В.А. JavaScript в web-дизайне. – СПб.: БХВ-Петербург, 2002. -880 с.
8. Полонская Е.Л. Язык HTML. Самоучитель.: - М.: Издательский дом «Вильяме», 2003. – 320с.
9. Смирнова И.Е. Начала Web–дизайна. – СПб.: БХВ-Петербург, 2003. -256 с.

## ҚҰЖАТТАҒЫ ТҮСТЕРДІ АНЫҚТАУ

Түсті анықтау үшін қажетті атрибутты пайдаланғаннан кейін не түс атауын (тырнақша ішінде), не оның он алты сандық мәнін қою керек. Түсті он алты сандық санмен белгілеу үшін ол түсті үш құраушыға бөліп белгілеуге болады: қызыл (R), жасыл (G), көк (B); оның әрқайсысы 0-ден бастап FF-ке дейінгі мәнге ие.

RGB форматындағы түсті жазу мысалдары кестеде берілген:

Түс		RGB	Түс		RGB
<b>black</b>	қара	000000	<b>purple</b>	күлгін	FF00FF
<b>white</b>	ақ	FFFFFF	<b>yellow</b>	сары	FFFF00
<b>red</b>	қызыл	FF0000	<b>brown</b>	қоңыр	996633
<b>green</b>	жасыл	00FF00	<b>orange</b>	қызыл сары	FF8000
<b>azure</b>	көгілдір (бирюзовый)	00FFFF	<b>violet</b>	күлгін (лиловый)	8000FF
<b>blue</b>	көк	0000FF	<b>gray</b>	сұр	A0A0A0
<b>silver</b>	күміс түсті	C0C0C0	<b>fuchsia</b>	ашық күлгін	FF00FF
<b>maroon</b>	қанық қызыл	800000	<b>lime</b>	ашық жасыл	00FF00
<b>olive</b>	зәйтүн түсті	808000	<b>teal</b>	көк-жасыл	008080
<b>navy</b>	қанық көк	000080	<b>aqua</b>	ашық көк	00FFFF

Аталған түстерден басқа қосымша түстер де бар. Олардың барлық саны 216 түс. Түсті белгілеу үшін аталған терминдердің орнына RGB-кодтарын пайдалануға болады, мысалы: #FF0000 – қызыл. RGB-кодтарды ауыстыра отырып, мәтін мен фонның қажетті түстерін таңдап алуға болады.



**ТЕСТІЛІК ТАПСЫРМАЛАР**

1. Web-түйін немесе web-сайт дегеніміз не:
  - A) бір авторға немесе бір басылымға жататын және ортақ гиперсілтемелер мен өзара байланысқан web-беттер тобын айтады;
  - B) Бір авторға немесе бірнеше басылымға жататын гиперсілтемелер тобын айтады;
  - C) құжат тақырыптары мен абзацтар белгілеу, гипермәтінде арнайы белгілер орнатуды айтады;
  - D) HTML форматында орындалатын бөлек құжатты айтады;
  - E) Өз айналым жолақтары бар құжаттың ауданын айтады.
2. HTML–құжаты немесе Web–құжаты немесе Web–бет дегеніміз не:
  - A) бір авторға немесе бір басылымға жататын және ортақ гиперсілтемелер мен өзара байланысқан web-беттер тобын айтады;
  - B) Бір авторға немесе бірнеше басылымға жататын гиперсілтемелер тобын айтады;
  - C) құжат тақырыптары мен абзацтар белгілеу, гипермәтінде арнайы белгілер орнатуды айтады;
  - D) HTML форматында орындалатын бөлек құжатты айтады;
  - E) Өз айналым жолақтары бар құжаттың ауданын айтады.
3. Құжат фрагменттерін белгілеу үшін және оның экранда бейнеленуін басқаруға арналған HTML тілінің арнайы элементтері, яғни компоненттері:
  - A) Тегтер.
  - B) Блоктар.
  - C) Тізімдер.
  - D) Сілтемелер.
  - E) Кестелер.
4. Мына тұжырымдардың қайсысы дұрыс:
  - A) HTML тілінің барлық тегтер «>» символынан басталады.
  - B) Блокнот арқылы гипермәтіндік құжатты жасауға болмайды.
  - C) Тегтер жазу кезінде бас әріп пен кіші әріп бірдей болып саналады.
  - D) HTML құжатына комментарийлер енгізуге болмайды.
  - E) HTML мәтіндік элементтері ішіне <, >, ”, @ символдар жазылады.
5. HTML-да контейнерлер жиынтығы болып табылатын құжат қалай басталады және қалай аяқталады (яғни, сипатталады):
  - A) функцияның жазылуымен;
  - B) <> (осы түрдегі жақшалар);
  - C) Тегалармен;
  - D) </ >;

- Е) Айнымалылармен.
6. JavaScript - бұл:
- A) Бағдарламалаудың әмбебап тілі;
  - B) Web-түйін беттерін жандандыру құралы;
  - C) Web-түйін беттерін көру сценарийлерін басқару тілі;
  - D) апплеттерді қосу тілі;
  - E) CGI сценарийі.
7. JavaScript-тегі объект... сипатталады:
- A) HTML-белгісінің ішінднгі орнымен;
  - B) Объект конструкторымен;
  - C) Қасиет, әдіс және оқиғамен;
  - D) Класстармен;
  - E) Тегтермен.
8. HTML–дегі JavaScript бағдарламасын кірістіру үшін қандай тег қолданылады:
- A) <scrit>;
  - B) <script>;
  - C) <body / script >;
  - D) <head/ script >;
  - E) < script / body >.
9. JavaScript. Берілген объектінің қасиеті status болып табылады:
- A) Window;
  - B) Frame;
  - C) Document;
  - D) History;
  - E) String.
10. JavaScript-тің Math класының қызметі:
- A) глобальды әдістер мен қасиеттерді бір объектіге біріктіру;
  - B) математик-программисттерге тригонометриялық және логарифмдік функциялар жиынын береді;
  - C) жаңа объект үшін күн және уақыт беріледі;
  - D) Web-браузер терезесін немесе кадрын көрсетеді;
  - E) URL-дің бір немесе бірнеше үзінділерін қамтитын, оқу және жазу мүмкіндігі бар жолды көрсетеді.
11. JavaScript-тің Date класының қызметі:
- A) глобальды әдістер мен қасиеттерді бір объектіге біріктіру;
  - B) математик-программисттерге тригонометриялық және логарифмдік функциялар жиынын береді;
  - C) жаңа объект үшін күн және уақыт беріледі;
  - D) Web-браузер терезесін немесе кадрын көрсетеді;
  - E) URL-дің бір немесе бірнеше үзінділерін қамтитын, оқу және жазу мүмкіндігі бар жолды көрсетеді.
12. JavaScript-тің Window класының қызметі:
- A) глобальды әдістер мен қасиеттерді бір объектіге біріктіру;

- B) математик-программисттерге тригонометриялық және логарифмдік функциялар жиынын береді;
  - C) жаңа объект үшін күн және уақыт беріледі;
  - D) Web-браузер терезесін немесе кадрын көрсетеді;
  - E) URL-дің бір немесе бірнеше үзінділерін қамтитын, оқу және жазу мүмкіндігі бар жолды көрсетеді.
13. JavaScript-тің Date класының `getDay()` әдісінің қызметі:
- апта күнін көрсетеді;
  - сағат өрісінің мәнін көрсетеді;
  - жыл өрісінің мәнін көрсетеді;
  - минут өрісінің мәнін көрсетеді;
  - ай өрісінің мәнін көрсетеді.
14. JavaScript-тің Date класының `getHours()` әдісінің қызметі:
- A) апта күнін көрсетеді;
  - B) сағат өрісінің мәнін көрсетеді;
  - C) жыл өрісінің мәнін көрсетеді;
  - D) минут өрісінің мәнін көрсетеді;
  - E) ай өрісінің мәнін көрсетеді.
15. JavaScript-тің Date класының `getMinutes()` әдісінің қызметі:
- A) апта күнін көрсетеді;
  - B) сағат өрісінің мәнін көрсетеді;
  - C) жыл өрісінің мәнін көрсетеді;
  - D) минут өрісінің мәнін көрсетеді;
  - E) ай өрісінің мәнін көрсетеді.
16. JavaScript-тің Date класының `getMonth()` әдісінің қызметі:
- A) апта күнін көрсетеді;
  - B) сағат өрісінің мәнін көрсетеді;
  - C) жыл өрісінің мәнін көрсетеді;
  - D) минут өрісінің мәнін көрсетеді;
  - E) ай өрісінің мәнін көрсетеді.
17. JavaScript-тің Date класының `getFullYear()` әдісінің қызметі:
- A) апта күнін көрсетеді;
  - B) сағат өрісінің мәнін көрсетеді;
  - C) жыл өрісінің мәнін көрсетеді;
  - D) минут өрісінің мәнін көрсетеді;
  - E) ай өрісінің мәнін көрсетеді.
18. JavaScript. `alert()` ...әдісін шақырғанда:
- A) ескерту терезесі құрылады;
  - B) `true` мәні қайтарылады;
  - C) мұндай әдіс жоқ;
  - D) енгізу өрісі бар терезе құрылады;
  - E) терезе жабылады.
19. JavaScript. `confirm()` әдісі нені қайтарады:
- A) `true/false` мәнін;

- B) әдіс аргументін;
  - C) void(x) ;
  - D) true мәнін;
  - E) false мәнін.
20. open() әдісінің параметрлері көрсетілмесе, онда:  
ағымды терезе қайта көрсетіледі;  
скриптегі қателік көрсетіледі;  
ешнәрсе өзгермейді;  
жаңа терезеге бос құжат жүктеледі;  
жаңа терезеге бос бет жүктеледі.
21. Батырмаға арналған қай оқиға:  
A) onChange  
B) onClick  
C) onBlur  
D) onload  
E) unload
22. Submit оқиғасы қай кезде орындалады:  
A) Тышқан гипермәтіндік сілтемеден өткенде;  
B) Сілтеменің тік төртбұрыш шекарасынан өткенде;  
C) Сурет батырманы таңдағанда;  
D) Кез-келген батырманы басқанда;  
E) Енгізу өрісінен фокусты алғанда.
23. Құжаттың 8-ші суретіне өту ():  
A) document.images[7];  
B) document.Family;  
C) document["family"];  
D) document.images7;  
E) document.images[8].
24. Image класының объектісін сипаттайтын оқиға:  
A) Load  
B) onmouseover  
C) onmouseout (IE)  
D) OnBlur()  
E) OnFocus()
25. JavaScript тіліндегі бағдарламаларды бөлек файл ретінде ...  
кеңейтпесімен орналастыруға болады:  
A) \*.js;  
B) \*.is;  
C) \*.ps;  
D) \*.vs;  
E) \*.sj.
26. Netscape фирмасымен өңделген, скриптерді құру үшін арналған  
тіл:  
A) JavaScript;

- B) Html;
  - C) VisualBasic;
  - D) Delphi;
  - E) Pascal.
27. Экранға хабарлама терезесін шығару:
- A) Prompt;
  - B) Alert();
  - C) Prompt("", "");
  - D) <input>;
  - E) Function.
28. alert("Web-бағдарлама") командасы нені білдіреді:
- A) Экранға хабарлама терезесін шығару
  - B) Пайдаланушының сұраныс терезесін шығару;
  - C) Экранға мәтін шығару;
  - D) Жалғыз ОК батырмасы бар ақпараттық тақта құру;
  - E) Жалғыз ОК батырмасы және мәтіні бар ақпараттық тақта құру.
29. confirm функциясы ... білдіреді:
- A) Экранға хабарлама терезесін шығару
  - B) Пайдаланушының сұраныс терезесін шығару;
  - C) Экранға мәтін шығару;
  - D) Екі ОК және ЖОЮ батырмалары бар ақпараттық тақта құру;
  - E) Екі ОК және ЖОЮ батырмалары және мәтіні бар ақпараттық тақта құру.
30. Жалауша қай команда арқылы беріледі:
- A) <INPUT type=checkbox>;
  - B) <INPUT type=button>;
  - C) <INPUT type=radio>;
  - D) <INPUT type=reset>;
  - E) <INPUT type=text>.
31. Батырма қай команда арқылы беріледі:
- A) <INPUT type=checkbox>;
  - B) <INPUT type=button>;
  - C) <INPUT type=radio>;
  - D) <INPUT type=reset>;
  - E) <INPUT type=text>.
32. Радиобатырма қай команда арқылы беріледі:
- A) <INPUT type=checkbox>;
  - B) <INPUT type=button>;
  - C) <INPUT type=radio>;
  - D) <INPUT type=reset>;
  - E) <INPUT type=text>.
33. Тастау батырмасы қай команда арқылы беріледі:
- A) <INPUT type=checkbox>;
  - B) <INPUT type=button>;

- C) <INPUT type=radio>;
  - D) <INPUT type=reset>;
  - E) <INPUT type=text>.
34. <img> объектісінің onclick оқиғасы ... білдіреді:
- A) Бейненің үстін тышқанмен шерту;
  - B) Бейненің үстіне тышқан көрсеткішін апару;
  - C) Бейнеден тышқан көрсеткішін алып тастау;
  - D) Бейнені екі рет шерту;
  - E) Бейненің үстінен тышқанның сол батырмасын шерту.
35. <img> объектісінің onmouseover оқиғасы ... білдіреді:
- F) Бейненің үстін тышқанмен шерту;
  - G) Бейненің үстіне тышқан көрсеткішін апару;
  - H) Бейнеден тышқан көрсеткішін алып тастау;
  - I) Бейнені екі рет шерту;
  - J) Бейненің үстінен тышқанның сол батырмасын шерту.
36. <img> объектісінің onmouseout оқиғасы ... білдіреді:
- K) Бейненің үстін тышқанмен шерту;
  - L) Бейненің үстіне тышқан көрсеткішін апару;
  - M) Бейнеден тышқан көрсеткішін алып тастау;
  - N) Бейнені екі рет шерту;
  - O) Бейненің үстінен тышқанның сол батырмасын шерту.
37. <img> объектісінің ondblclick оқиғасы ... білдіреді:
- P) Бейненің үстін тышқанмен шерту;
  - Q) Бейненің үстіне тышқан көрсеткішін апару;
  - R) Бейнеден тышқан көрсеткішін алып тастау;
  - S) Бейнені екі рет шерту;
  - T) Бейненің үстінен тышқанның сол батырмасын шерту.
38. JavaScript тіліндегі % таңбасы ... білдіреді:
- A) Қосу;
  - B) Алу;
  - C) Көбейту;
  - D) Пайыз;
  - E) Бөліндіден қалған қалдық.\*
39. JavaScript тіліндегі / \* ... \* / таңбасы ... білдіреді:
- A) түсінік;
  - B) жолдық түсінік;
  - C) Блоктық түсінік;
  - D) ұзын түсінік;
  - E) қысқа түсінік.
40. Web-тің негізгі қызметі қандай:
- A) Бағдарламалау.
  - B) қажетті ақпаратты іздеу, жинастыру және оны экранға шығаруды ұйымдастыру.
  - C) Есеп шығару.

- D) Графиктер салу.
  - E) Фотосуреттерді монтаждау.
41. Гиперсілтеме дегеніміз...
- A) Веб-беттерге сілтеме ретінде тақырыпқа байланысты басқа да веб-бетте жазылған сөз не сөз тіркестерін арнайы команда бойынша белгілеп қою.
  - B) сілтеме енгізілген мәтінді.
  - C) браузер мен серверді программалық қосу.
  - D) интернет серверінде файлдар түрінде сақталған құжаттарды қабылдап, экранға шақыру.
  - E) файлдарды барлық компьютер серверінде сақтап FTP арқылы қызмет көрсететін арнайы компьютерлер (файлдарды сақтау қоймалары).
42. Фрейм (Frame) дегеніміз...
- A) Өз айналым жолақтары бар құжаттың ауданы;
  - B) клиент компьютеріне бөлек файл түрінде берілетін және web-беті қараған кезде жүктелетін бағдарлама;
  - C) web-бет құрамына енген және оның мүмкіндіктерін кеңейту үшін арналған бағдарлама;
  - D) серверде жұмыс істеген кезде web-беттердің мүмкіндіктерін кеңейтуге мүмкіндік беретін бағдарламалардың жалпы атауы;
  - E) web-беттерді көру үшін арналған бағдарлама.
43. Скрипт немесе сценарий (Script) дегеніміз...
- A) Өз айналым жолақтары бар құжаттың ауданы;
  - B) клиент компьютеріне бөлек файл түрінде берілетін және web-беті қараған кезде жүктелетін бағдарлама;
  - C) web-бет құрамына енген және оның мүмкіндіктерін кеңейту үшін арналған бағдарлама;
  - D) серверде жұмыс істеген кезде web-беттердің мүмкіндіктерін кеңейтуге мүмкіндік беретін бағдарламалардың жалпы атауы;
  - E) web-беттерді көру үшін арналған бағдарлама.
44. Браузер (Browser) дегеніміз...
- A) Өз айналым жолақтары бар құжаттың ауданы;
  - B) клиент компьютеріне бөлек файл түрінде берілетін және web-беті қараған кезде жүктелетін бағдарлама;
  - C) web-бет құрамына енген және оның мүмкіндіктерін кеңейту үшін арналған бағдарлама;
  - D) серверде жұмыс істеген кезде web-беттердің мүмкіндіктерін кеңейтуге мүмкіндік беретін бағдарламалардың жалпы атауы;
  - E) web-беттерді көру үшін арналған бағдарлама.
45. Тег дегеніміз...
- A) Өз айналым жолақтары бар құжаттың ауданы;
  - B) элементтерінің параметрі немесе қасиеті;

- C) web-бет құрамына енген және оның мүмкіндіктерін кеңейту үшін арналған бағдарлама;
  - D) серверде жұмыс істеген кезде web-беттердің мүмкіндіктерін кеңейтуге мүмкіндік беретін бағдарламалардың жалпы атауы;
  - E) элементтің бастапқы немесе соңғы маркерлері.
46. Атрибут...
- A) Өз айналым жолақтары бар құжаттың ауданы;
  - B) элементтерінің параметрі немесе қасиеті;
  - C) web-бет құрамына енген және оның мүмкіндіктерін кеңейту үшін арналған бағдарлама;
  - D) серверде жұмыс істеген кезде web-беттердің мүмкіндіктерін кеңейтуге мүмкіндік беретін бағдарламалардың жалпы атауы;
  - E) элементтің бастапқы немесе соңғы маркерлері.
47. JavaScript. Логикалық ЖӘНЕ операциясын белгілейтін таңба?
- A) ||
  - B) &&
  - C) ??
  - D) //
  - E) !
48. JavaScript. Логикалық НЕМЕСЕ белгілейтін таңба?
- A) ||
  - B) //
  - C) ??
  - D) &&
  - E) !
49. JavaScript. Confirm() функциясы ...
- A) Бір ғана ОК батырмасы бар терезе шығарады
  - B) Ақпаратты еңгізу үшін қолданылатын функция
  - C) ОК және ОТМЕНА батырмалары бар терезе шығады.
  - D) Көлемді мәтінді экранға шығаратын функция
  - E) Айнымалыларды сипаттау үшін қолданылады
50. JavaScript тіліндегі "/" таңбасы ... білдіреді:
- A) түсінік;
  - B) жолдық түсінік;
  - C) Блоктық түсінік;
  - D) ұзын түсінік;
  - E) қысқа түсінік.
51. JavaScript тіліндегі ++ таңбасы ... білдіреді:
- A) Алу немесе қосу;
  - B) Бірге көбею;
  - C) Салыстыру операциясы;
  - D) Логикалық «және»;
  - E) Логикалық «немесе».
52. JavaScript тіліндегі -- таңбасы ... білдіреді:



- A) Алу немесе қосу;
  - B) Бірге азаю;
  - C) Салыстыру операциясы;
  - D) Логикалық «және»;
  - E) Логикалық «немесе».
53. JavaScript тіліндегі ! таңбасы ... білдіреді:
- A) Алу немесе қосу;
  - B) Логикалық терістеу;
  - C) Салыстыру операциясы;
  - D) Логикалық «және»;
  - E) Логикалық «немесе».
54. JavaScript тіліндегі = таңбасы ... білдіреді:
- A) меншіктеу;
  - B) Логикалық терістеу;
  - C) Салыстыру операциясы;
  - D) Логикалық «және»;
  - E) Логикалық «немесе».
55. Мәліметтерді сақтау үшін арналған жадының атаулы аудыны:
- A) Мәліметтер;
  - B) Командалар;
  - C) Айнымалылар;
  - D) Тұрақтылар;
  - E) Өрнектер.
56. Цикл командасының жалпы түрі:
- A) While (шарт) команда;
  - B) While(шарт, команда);
  - C) While(шарт, команда);
  - D) While(шарт), (команда);
  - E) While(шарт, «команда»).
57. for циклының жалпы түрі:
- A) For(басы; шарт; жетілдіру); команда;
  - B) For(басы, шарт, жетілдіру); команда;
  - C) For(басы, шарт), команда;
  - D) For(басы, шарт); жетілдіру; команда;
  - E) For(басы, шарт); (жетілдіру, команда).
58. break командасы... мүмкіндік береді:
- A) циклдің орындалуын т тоқтату;
  - B) циклді бірнеше секундқа тоқтату;
  - C) командалардың орындалуын өткізіп отыру;
  - D) циклдан кейінгі және цикл денесінің соңына дейінгі барлық командалардың орындалуын өткізу;
  - E) циклды орындау.
59. continue командасы... мүмкіндік береді:
- A) циклдің орындалуын лезде тоқтату;

- B) циклді бірнеше секундқа тоқтату;
  - C) командалардың орындалуын өткізіп отыру;
  - D) циклдан кейінгі және цикл денесінің соңына дейінгі барлық командалардың орындалуын өткізу;
  - E) циклды орындау.
60. If шартты операторының толық түрде жазылуы:
- A) if(шарт) { тұжырым };
  - B) if(шарт) { тұжырым 1} else{ тұжырым 2};
  - C) if(шарт) { тұжырым 1} then { тұжырым 2};
  - D) if(шарт) { тұжырым 1} then { тұжырым 2} else{ тұжырым 3};
  - E) дұрыс жауабы жоқ.
61. Функция дегеніміз...
- A) бұл оны шақырғанда орындалатын кодтың атаулы үзіндісі;
  - B) фигуралық жақшаға алынған дене;
  - C) меншіктелетін оператор;
  - D) дұрыс жауабы жоқ;
  - E) жолдық амалдар.
62. Жолдармен жұмыс істейтін eval(s) амалы ...
- A) n негізі бойынша бүтін сандар үшін арналған (2 -ден 36-дейін);
  - B) S жолында қамтылатын, санға кірмейтін жолдың басынан бірінші символына дейінгі заттық санын табады;
  - C) ол JavaScript –тің бір немесе бірнеше операторларды қамтитын s жолымен берілген кодты орындайды;
  - D) бүтін сандарға арналған (Integer). Негізі автоматты түрде табылады;
  - E) жолды жояды.
63. Жолдармен жұмыс істейтін parseFloat(s) амалы...
- A) n негізі бойынша бүтін сандар үшін арналған (2 -ден 36-дейін);
  - B) S жолында қамтылатын, санға кірмейтін жолдың басынан бірінші символына дейінгі заттық санын табады;
  - C) ол JavaScript –тің бір немесе бірнеше операторларды қамтитын s жолымен берілген кодты орындайды;
  - D) бүтін сандарға арналған (Integer). Негізі автоматты түрде табылады;
  - E) жолды жояды.
64. Жолдармен жұмыс істейтін parseInt(s) амалы...
- A) n негізі бойынша бүтін сандар үшін арналған (2 -ден 36-дейін);
  - B) S жолында қамтылатын, санға кірмейтін жолдың басынан бірінші символына дейінгі заттық санын табады;
  - C) ол JavaScript –тің бір немесе бірнеше операторларды қамтитын s жолымен берілген кодты орындайды;

- D) бүтін сандарға арналған (Integer). Негізі автоматты түрде табылады;
- E) жолды жояды.
65. Жолдармен жұмыс істейтін `parseInt(s,n)` амалы...
- A) `n` негізі бойынша бүтін сандар үшін арналған (2 -ден 36-дейін);
- B) `S` жолында қамтылатын, санға кірмейтін жолдың басынан бірінші символына дейінгі заттық санын табады;
- C) ол JavaScript –тің бір немесе бірнеше операторларды қамтитын `s` жолымен берілген кодты орындайды;
- D) бүтін сандарға арналған (Integer). Негізі автоматты түрде табылады;
- E) жолды жояды.
66. JavaScript тілінде процедуралар бар ма:
- A) Жоқ;
- B) Бар;
- C) Бірнешеуі бар;
- D) Жоқ, тек функциялар бар;
- E) Функциялар да жоқ.
67. Array объектісі ... үшін арналған:
- A) Мәліметтер жиымын (массивін) сақтау;
- B) Айнымалылардың мәндерін сақтау;
- C) Тұрақтылардың мәндерін сақтау;
- D) Тұрақтылардың жиымын (массивін) сақтау;
- E) Жолдық айнымалылардың жиымын (массивін) сақтау.
68. JavaScript-те глобальды әдістер мен қасиеттерді бір объектіге біріктіретін класс:
- A) Global класы;
- B) Window класы;
- C) Math класы;
- D) Location класы;
- E) Date класы.
69. JavaScript-те Web-браузер терезесін немесе кадрын көрсететін класс:
- A) Global класы;
- B) Window класы;
- C) Math класы;
- D) Location класы;
- E) Date класы.
70. JavaScript-те URL-дің бір немесе бірнеше үзінділерін қамтитын, оқу және жазу мүмкіндігі бар жолды көрсететін класс:
- F) Global класы;
- G) Window класы;
- H) Math класы;

- I) Location класы;
- J) Date класы.

71. JavaScript-те экрандағы гиперсілтемені немесе HTML-құжаттағы клиент жағындағы бейненің облысының үзіндісін көрсететін класс:

- A) Link класы;
- B) Window класы;
- C) Math класы;
- D) Location класы;
- E) History класы.

72. JavaScript-те пайдаланушы терезеде Forward және Back өзіндік батырмаларын құра алатын класс:

- A) Link класы;
- B) Window класы;
- C) Math класы;
- D) Location класы;
- E) History класы.

73. web-сценарийлерді құру үшін ... тіл қолданылады:

- A) JavaScript;
- B) Html;
- C) VisualBasic;
- D) Delphi;
- E) Pascal.

74. JavaScript. Prompt() функциясы...

- A) Бір ғана ОК батырмасы бар терезе шығарады
- B) ОК және ОТМЕНА батырмалары бар терезе шығады.
- C) Көлемді мәтінді экранға шығаратын функция
- D) Ақпаратты енгізу үшін қолданылатын функция
- E) Айнымалыларды сипаттау үшін қолданылады

75. JavaScript. <script>...</script> тегтері не үшін қажет?

- A) Комментарий енгізу үшін
- B) Скриптер енгізу үшін
- C) Формалар енгізу үшін
- D) Сырғымалы жолдары жасау үшін
- E) HTML-да мұндай скриптер қолданылмайды

76. JavaScript тілінде шығарылатын жазбаны жаңа жолға көшіру үшін мынадай таңбалар қолданылады:

- A) \'
- B) \n
- C) n.
- D) «+»
- E) «\*»

77. JavaScript тілінде айнымалыларды сипаттау үшін қандай түйінді сөз қолданылады?

- A) Alert

- B) Prompt
- C) Concat
- D) Slice
- E) Var

78. PI константасы үшін JavaScript тілінде қолданылатын қасиет?

- A) Math.PI
- B) Math.E
- C) Math.LN10
- D) Math.SQRT314\_10
- E) Math(3.14)

79.  $e=2.71828$  константасы үшін JavaScript тілінде қолданылатын қасиет?

- A) Math.SQRT314\_10
- B) Math.PI
- C) Math.E
- D) Math.LN10
- E) Math(2.71828)

80. 10 санның натуралдық логарифмі үшін JavaScript тілінде қолданылатын қасиет?

- A) Math.PI
- B) Math.E
- C) Math.sin(3,14)
- D) Math.LN10
- E) Math.SQRT314\_10

81. Аргументтің квадрат түбірі үшін JavaScript тілінде қолданылатын қасиет?

- A) Math.e(x)
- B) Math.sin()
- C) Math.E(x)
- D) Math.Ln()
- E) Math.sqrt()

82. X санының синусын табу үшін JavaScript тілінде қолданылатын қасиет?

- A) Math.SQRT(x)
- B) Math.asin()
- C) Math.sinus()
- D) Math.(x)
- E) Math.sin(x)

83.  $e^x$  табу үшін JavaScript тілінде қолданылатын қасиет?

- A) Math.e(x)
- B) Math.E(x)
- C) Math.ex()
- D) Math.exp
- E) Math.exp(x)

84. Аргументке ең жақын одан үлкен немесе соған тең бүтін санды табу үшін JavaScript тілінде қолданылатын әдіс?

- A) Math.>=(x)
- B) Math.floor(x)
- C) Math.ceil(x)
- D) Math.exp(x)
- E) Math.sin(x)

85. Аргументке ең жақын одан кіші немесе соған тең бүтін санды табу үшін JavaScript тілінде қолданылатын әдіс?

- A) Math.>=(x)
- B) Math.exp(x)
- C) Math.floor(x)
- D) Math.ceil(x)
- E) Math.sin(x)

86. Аргументті оған ең жақын бүтін санға дейін дөңгелектеу үшін JavaScript тілінде қолданылатын әдіс?

- A) Math.round(x)
- B) Math.floor(x)
- C) Math.ceil(x)
- D) Math.exp(x)
- E) Math.cel(x)

87.  $X^n$  мәнін табу үшін JavaScript тілінде қолданылатын әдіс?

- A) Math.round(x,n)
- B) Math.floor(x,n)
- C) Math.pow(x,n)
- D) Math.exp(x,n)
- E) Math.ceil(x,n)

88. JavaScript. setMonth() әдісі не береді?

- A) Ай нөмірін береді
- B) Ай нөмірін тағайындайды
- C) Апта күнін береді
- D) Ай атын береді
- E) Апта күнін тағайындайды

89. JavaScript тіліндегі length әдісінің атқаратын қызметі:

- A) Ол бастапқы жиымды өзгертпей, жиым элементтерінің «ажыратқыш» арқылы бөліп жібереді
- B) Жиым элементтерінің оның 1-элементті соңғысы болатындай етіп кері бағытта орын ауыстырады
- C) Жиым ұзындығын береді
- D) Жиым элементтерін өсу немесе лексографикалық тәртіпте орналастырады
- E) Бастапқы жиымнан бір позициядан бастап екінші позицияға дейінгі элементтерден тұратын жаңа жиым жасап жібереді

90. JavaScript. Бастапқы жиымды өзгертпей, жиым элементтерін «ажыратқыш» арқылы бөліп жіберетін әдіс:
- A) Slice
  - B) Join
  - C) Reverse
  - D) Concat
  - E) Sort
91. JavaScript тіліндегі while циклінің жалпы жазылу түрі:
- A) while(шарт)команда
  - B) while(цикл басы;шарт)команда;
  - C) while(цикл басы;шарт;қадамы)команда
  - D) while(шарт; қадамы)команда;
  - E) while(шарт)команда
92. JavaScript тіліндегі экранға шығарылатын мәндердің барлығын бір терезеде орналастыру үшін alert() функциясының орнына қандай команда жазу керек?
- A) Alert.write()
  - B) Document.write()
  - C) Promp()
  - D) Concat()
  - E) Slice()
93. JavaScript тілінде жүйелі жақшаға {} алынған командалар тізбегі қалай аталады?
- A) Тізбек
  - B) Функция
  - C) Блок
  - D) әдіс
  - E) Шарт
94. JavaScript тілінде экранға екі батырмасы (ОК және ОТМЕНА) бар терезесі шығаратын функция:
- A) alert()
  - B) prompt()
  - C) var()
  - D) confirm()
  - E) slice()
95. JavaScript тілінде  $x\%5$  деген жазу нені білдіреді?
- A) x-тің 5 пайызын табу
  - B) x-ті 5-ке бөлгендегі қалдықты табу
  - C) x-ті 5-ке бөлу
  - D) x-тің өсу қадамы 5-ке тең
  - E) x-тің кему қадамы 5-ке тең
96. Array объектісінің concat әдісі:
- A) Жиымға элементтер қосады және ондағыны жояды.
  - B) Екі жиымды бір жиымға біріктіреді

- C) Жиымның барлық элементтерін бір жолға біріктіреді.
  - D) Соңғы элементті жиымнан жояды
  - E) Жиым элементтерін сұрыптайды.
97. Array объектісінің sort әдісі:
- A) Екі жиымды бір жиымға біріктіреді
  - B) Жиымның барлық элементтерін бір жолға біріктіреді.
  - C) Жиымға элементтер қосады және ондағыны жояды.
  - D) Соңғы элементті жиымнан жояды
  - E) Жиым элементтерін сұрыптайды.
98. Array объектісінің splice әдісі:
- A) Екі жиымды бір жиымға біріктіреді
  - B) Жиымның барлық элементтерін бір жолға біріктіреді.
  - C) Соңғы элементті жиымнан жояды
  - D) Жиымға элементтер қосады және ондағыны жояды.
  - E) Жиым элементтерін сұрыптайды.
99. Array объектісінің pop әдісі:
- A) Екі жиымды бір жиымға біріктіреді
  - B) Жиымның барлық элементтерін бір жолға біріктіреді.
  - C) Соңғы элементті жиымнан жояды
  - D) Жиымға элементтер қосады және ондағыны жояды.
  - E) Жиым элементтерін сұрыптайды.
100. Array объектісінің join әдісі:
- A) Екі жиымды бір жиымға біріктіреді
  - B) Жиымның барлық элементтерін бір жолға біріктіреді.
  - C) Соңғы элементті жиымнан жояды
  - D) Жиымға элементтер қосады және ондағыны жояды.
  - E) Жиым элементтерін сұрыптайды.



## HTML ТІЛІНІҢ ТЕГТЕРІ ТУРАЛЫ АНЫҚТАМАЛЫҚ СӨЗДІК

Атауы	Тағайындалуы	Атрибуттары:
<head>...</head>	парақша тақырыбы	-
<title>...</title>	браузер терезесінің жоғарғы жолында бейнеленетін беттің тақырыбы	-
<body>...</body>	браузерде бейнелетін беттің мазмұны	<ul style="list-style-type: none"> <li>- <b>background</b> – құжат мәтіні бейнелетін фонды анықтайды.</li> <li>- <b>backcolor</b> =#ffffff (<b>yellow</b>) – фонның түсін анықтайды (сары).</li> <li>- <b>text</b> = #0000ff (<b>blue</b>) – мәтін түсін анықтайды (көк).</li> <li>- <b>Bgcolor</b> – құжат мәтінінің түсін анықтайды.</li> </ul>
<h1>..</h1> <h2>..</h2> ..... <h6>..</h6>	тақырып деңгейлері	<ul style="list-style-type: none"> <li><b>align = left</b> – сол жақ бойынша түзейді;</li> <li><b>align = right</b> – оң жақ бойынша;</li> <li><b>align = center</b> – орталық бойынша;</li> <li><b>align = justify</b> – ені бойынша</li> </ul>
<P>, </P>	абзац тегі	<b>align = left, center, right, justify</b> шегіністерін білдіреді.
 	жолды үзіп, жаңа жолға көшу тегі	-
<HR>	көлденең сызықты құру	<ul style="list-style-type: none"> <li>– <b>align</b> - түзету (мәндері - left, center, right);</li> <li>– <b>color</b> – жолақ түсі;</li> <li>– <b>size</b> – пиксель бойынша биіктігі;</li> <li>– <b>width</b> – экранның енінен пиксель немесе пайыз бойынша ені.</li> </ul>

<B>	қарайтылған мәтін	-
<I>	курсив	-
<U>	асты сызылған	-
<S>	сызылып тасталған	-
<BIG>	үлкен өлшемді әріптер	-
<SMALL>	кіші өлшемді әріптер	-
<SUB> ... </SUB>	төменгі индекс	-
<SUP> ... </SUP>	жоғарғы индекс	-
<Font>..</font>	мәтін қаріпінің түсін, өлшемі мен түрін өзгерту	<b>size</b> – қаріп өлшемін анықтайды. <b>color</b> – мәтін түсін анықтауға мүмкіндік береді. <b>face</b> – қолданыстағы қаріпті анықтайды.
<address> </address>	байланыс тізімі	-
<LH> </LH>	тізім тақырыбы	-
<LI>	тізім элементі	-
<UL>. . .</UL>	маркерленген тізім	<LI type = disk> - қарайтылған дөңгелек нүкте; <LI type = circle> - шеңбер; <LI type = square> - кішкентай қара шаршы.
<OL>. . .</OL>	нөмірленген тізім	Type=1 маркерді араб сандары түрінде орнатады; Type=A маркерді бас әріптер түрінде орнатады;

		Type=a маркерді кішкентай әріптер түрінде орнатады; Type=l маркерді үлкен рим сандары түрінде орнатады.
<DL>...</DL>	анықтау тізімі	-
<IMG>	Суретті орнату <IMG SRC="pic.gif">	<b>align</b> (top - жоғары, bottom - төмен, middle-ортада, left - сол жақта, right-оң жақта түзетілген); <b>border</b> - жақтау сызығы; <b>height</b> - суреттің биіктігі; <b>width</b> - көлденең өлшемі; <b>vspace</b> - суреттің жоғарғы және төменгі жақтары; <b>hspace</b> - суреттің сол және оң жақ шеттері.
<A href>...</A>	гиперсілтемелерді ұйымдастыру	-
<Marquee> </Marquee>		<b>align</b> (top - жоғары, bottom - төмен, middle-ортада); <b>bgcolor</b> - жүгіртпе жолдың фонының түсі; <b>direction</b> – жүгіртпе; <b>height</b> - жолдың бағыты; жүгіртпе жолдың биіктігі.
Embed	Аудио мен бейнені (видео) енгізу	-
<table>...</table>	Кесте құру	<b>align</b> - түзету (мәндері - left, center, right); <b>caption</b> кесте атауының элементі; <b>Border</b> - кестенің сыртқы жиектелуі; <b>Bgcolor</b> - ұяшықтарға фондық түс белгілеу;

		<b>Background</b> - ұяшықтарға фондық сурет белгілеу.
<TR>...</TR>	кесте жолдары	-
<TD>...</TD>	кесте ұяшықтары	-
<TH>...</TH>	кестенің бағандары мен жолдарына арналған атау	-
<Form> </Form>	толтырылатын форманы жасау	-
<Frameset> </Frameset>	экранды бірнеше көлденең және тік фреймдерге бөлу	-