

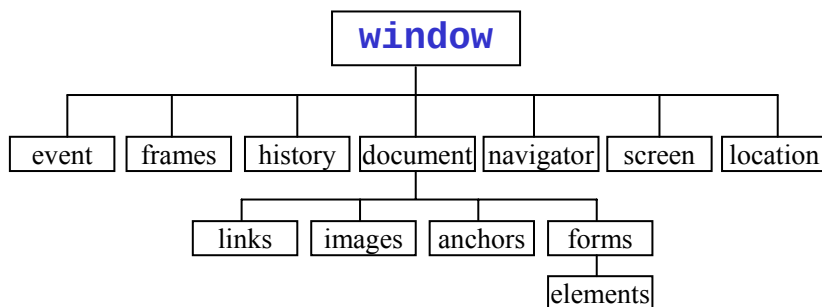
11 ЛАБОРАТОРИЯЛЫҚ ЖҰМЫС

Браузер объектілері мен оқиғалары

1. Құжаттың объектілік моделі

Браузер экранын өзгерту немесе жаңа терезелер жасау үшін браузердің ішкі мүмкіндіктерімен танысып, оның ішкі объектілерін пайдалану керек. Браузер тәгтерді экранда көрсетіп қана қоймай, ол құжаттың иерархиялық моделіне сәйкес объектілер тұрғызады. Браузер жұмыс кезінде модельді тұрақты түрде бақылап отырады. Егер объект қасиеті өзгерсе, экрандағы бейне де өзгереді. JavaScript тілі арқылы объект қасиеттерін өзгертуге болады, сондықтан ол экран динамикасын жүзеге асыра алады.

Кез келген құжаттың объектілік моделі әрқашанда нақты сайттың бейнесінен тәуелсіз түрдегі бір құрылымда болады (сурет).



Схемада көрсетілген барлық «объектілер» негізінде бір объектінің даналары болып табылады. Браузер әр нақты құжат үшін сол даналарды жеке-жеке түрде құрастыра алады. Сол даналарды бұдан кейін «объектілер» деп атаймыз.

2. Window объектісі

Window иерархияның жоғарғы жағында орналасады. Бұл қалған объектілердің одан төмен орналасып, соның қасиеттері түрінде қарастырылатынын білдіреді. Үш тәсілді біз бұрыннан білеміз, олар: alert, prompt және confirm.

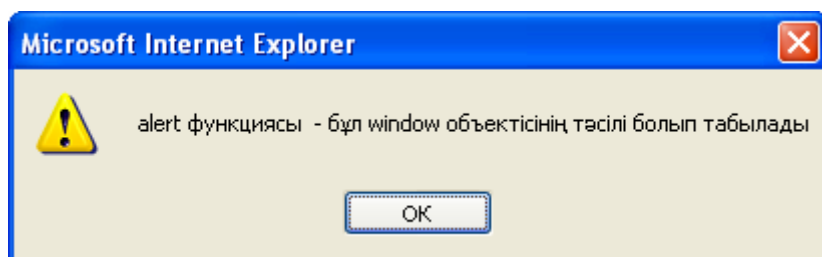
Егер келесі түрде жазылса:

```
Window.alert ("alert функциясы – бұл window объектісінің тәсілі болып табылады");
```

– онда браузер осы хабарламаны шығаратын қосалқы терезе ашады.

Біз бұрын былай жазатынбыз:

```
alert("alert функциясы – бұл window объектісінің тәсілі болып табылады");
```



Негізгі объект болып саналатын window объектісінің атын жазбауға да болады. Келесі жолдардың нәтижелері бірдей болады:

```
window.alert("Ат айналып қазығын табар"); alert("Ат айналып қазығын табар");
```

2.1. Жаңа терезелер ашу

Көп жағдайда пайдаланушыға жұмыс нәтижесін көрсету үшін бұрынғы терезе ашық тұрғанда, жаңа терезе ашуға тура келеді. Жаңа терезе ашу үшін window объектісінің open тәсілін қолданамыз. Оны келесі тәсілдердің бірін қолдану арқылы жазуға болады:

```
var айнымалы = open();
```

```
var айнымалы = open(файл);
```

```
var айнымалы = open(файл, терезе_аты);
```

```
var айнымалы = open(файл, терезе_аты, терезе_параметрлері);
```

Мұндағы:

- айнымалы – ашылатын терезеге нұсқаушы;

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

- терезе_аты — 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, үнсіз келісім бойынша тағайындалған қасиеттер қолданылады. Параметрлер арасындағы үтірден соң, бос орын қоймаған дұрыс.

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. Пілдердің суреті slon.jpg орналасқан жеке терезе ашу

```
<HTML>
  <HEAD>
    <TITLE>Браузерді тексеру</TITLE>
  </HEAD>
  <BODY bgcolor=white text=black>
    <H2>Браузерді тексеру</H2> <HR>
    <SCRIPT language=JavaScript>
      <!--
        var win = open("slon.jpg","",
          "width = 320,height = 260"+
          "resizable =0,scrollbars =1"+
          "menubar =0,location = 1" +
          "status = 0, toolbar = no");
      //-->
    </SCRIPT>
  <P>
```

Негізгі мәтінге оралу үшін браузердің саймандар тақтасында орналасқан

Артқа қарай (Назад)батырмасын басыңыз.

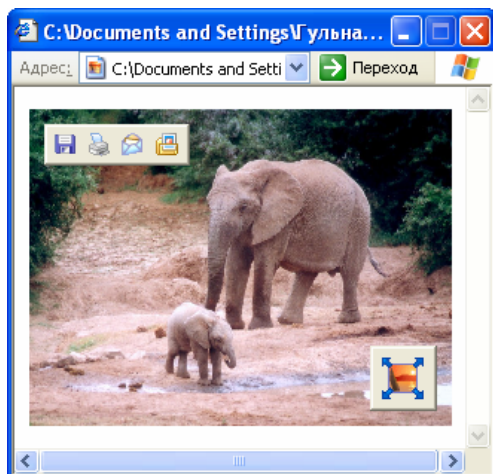
```
</BODY>
```

```
</HTML>
```

Осы программа нәтижесі:

Терезеде көрсетілетін файл ретінде мыналарды көрсете аламыз:

- HTML-файл, мысалы, file1.htm;



Мұнан кейін әр түрлі параметрлерді өзгертіп көруге болады.

- 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(...);
```

3. Document объектісі

Document объектісі (ол document объектісі қасиеттері болып табылады) <html> ...</html> блоктарында орналасқан тәгтерді 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(жазба);           немесе
```

```
айнымалы.document.write(жазба);
```

Мұндағы : айнымалы – бұл шығарылатын мәлімет көрсетілетін терезеге нұсқауыш. Егер айнымалы берілмесе, онда мәлімет ағымдағы терезеге жіберіледі; жазба – шығарылатын мәлімет көрсетілетін сөз тіркесі.

Close тәсілі форматы:

```
document.close();           немесе
```

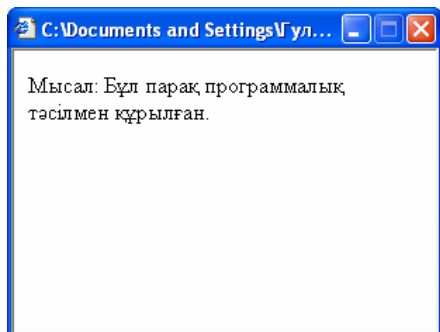
```
айнымалы.document.close();
```

Мұндағы 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.

Мысал. Келесі суреттегі бейнені құру программасы:

```
var win = open("", "", "width=300,height=200"); win.document.write("Мысал: Бұл парақ ");  
win.document.write("программалық тәсілмен құрылған."); win.document.close();
```

Терезеге мәтін шығару кезінде write функциясы бірнеше рет қолданылғанымен, оның нәтижесі бір тұтас абзац ретінде шығарылады.



Write тәсілі арқылы терезеге жай мәтіннен басқа тәгтермен белгіленген мәтінді де шығаруға болады (сурет).

```
var win = open("", "", "width=300,height=200"); win.document.write("<H2>Мысал  
</H2>" +  
"<HR>" +  
"<P>" +  
"Бұл парақ" +  
" программалық тәсілмен құрылған.");  
win.document.close();
```

4. Оқиғалар

Браузер тұтынушы әрекеттерін қадағалап, соларға сәйкес әрекет ете алады. Егер терезеде тышқанды шерту сілтемеге сәйкес келсе, браузер жаңа құжат жүктейді. “Alt + ←” пернелерін басу терезені бұрынғы қалпына қайтарады.

Егер тұтынушы тышқан курсорын суретке алып барса, онда браузер шағын терезеге сол тәгіне, яғни суретке сәйкес alt-мәтін шығарады. Тұтынушы бағыттауыш пернелерді басса, браузер сәйкесінше терезедегі мәтінді жылжытады.

Бұлардың бәрі тышқан мен пернеліктің браузерге тұтынушының не істеп жатқаны жайлы « хабар беріп» отыратынын білдіреді. Оған қоса, браузер құжаттың жүктелуі жайлы да толық хабардар болып

отырады. Егер сурет жүктелмесе (дискіде жоқ, сервермен байланыс үзілді), браузер оның орнына тіктөртбұрыш қояды. Осындай информациялық хабарламалар *оқиғалар* (события) деп аталады. Кейбір оқиғаларға браузер бірден жауап береді, ал басқаларына жайбарақат қарап тұра беруі де мүмкін. Мұндайда браузерге не істеуі керек екендігі айтылмаған деп ұққан жөн.

HTML-код арқылы браузер болады(суретті):

```
<FORM>
  <INPUT type=button
    value="Кәдімгі батырма">
</FORM>
```

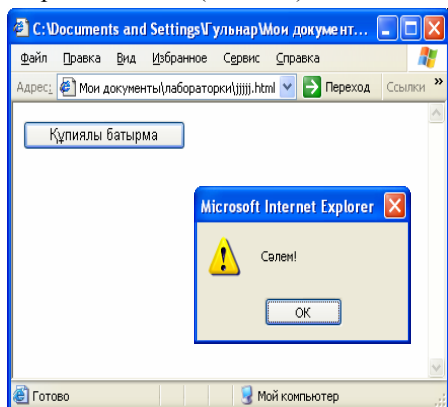
Батырма `<form>...</form>` арқылы берілген. `value` атрибуты береді, ал `type` атрибуты енгізу. Браузер батырманы шерткенде, оған экранда батырманы басқанды істемейді.

Ал енді басқа батырмаларды `<FORM>`

```
  <INPUT type=button value="Құпиялы батырма"
    onclick="alert('Сәлем!') ">
```

`</FORM>`

Экранға «Сәлем!» сөзі бар alert терезесі шығады (сурет). `<input>` тәгінің `onclick` атрибуты бар. Оның мәні ретінде: `"alert('Сәлем!')"` сөзі жазылған.



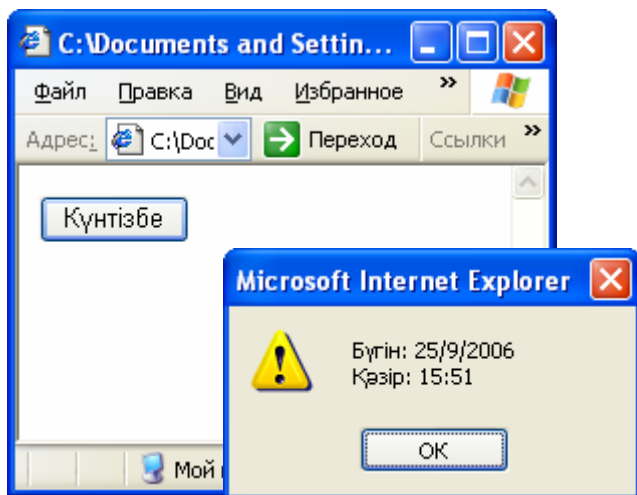
`onclick` атрибуты браузерге батырма басылғанда не істеу керек екендігін білдіріп тұр. Атрибут мәні ретінде ұзын сөз жазу қажет емес, сондықтан көбінесе былай программаланады.

```
<SCRIPT language=JavaScript>
<!--
function Fun()
{ var now = new Date() ;
  alert("Бүгін: "+now.getDate()+"/"+
    (now.getMonth()+1)+"/"+ now.getYear() +
    "\nКәзіп: "+now.getHours()+":"+now.getMinutes());
}
//-->
</SCRIPT>
<FORM>
<INPUT type=button
value=Күнтізбе onclick="Fun();" ">
</FORM>
```

`onclick` оқиғасы туғанда – батырманы шерткенде жұмыс істейтін `Fun()` функциясы сол оқиғаның өңдеушісі болып саналады.

салған батырма мынадай

тәгінің ішкі `<input>` тәгі батырма бетіндегі жазуды өрісінің типін сипаттайды. жауап бере ме? Әрине, ол көрсетеді, бірақ басқа ешнәрсе қарастырайық.



Осы программаны браузерге жүктеп бірнеше тәжірибе жүргізіндер.

1. Батырмадағы сөзді өзгертіп «Күнтізбе» орнына басқа мәтін жазыңдар.
2. Fun функциясы кодын басқаға, мысалы, мынаған өзгертіндер:

```
var sum = 0;
for (var i = 100; i; i--) sum += i;
alert("Қосынды 1 + 2 + ... + 100 = " + sum);
```

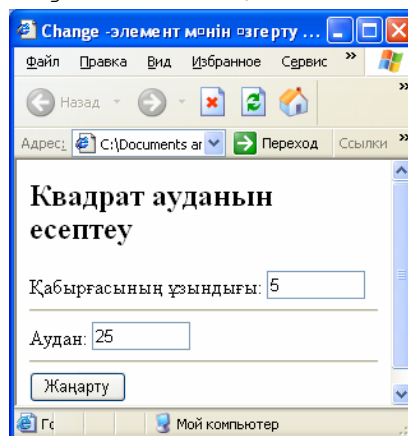
3. Өз функцияларыңды жазып шығыңдар. Мысалы, бір сан енгізіп оның квадратын немесе кубын табыңдар.

Сонымен, JavaScript HTML-кодтарында оқиғаларды өңдейтін программалар жазуға мүмкіндік береді де, оны интерактивті бет етуге, яғни тұтынушы әрекетіне жауап беретіндей күйге келтіре алады.

Тәгтегі onclick түйінді сөзі - «оқиғалық» атрибуттың аты, алдына «on» сөзі қойылып жазылады. Мұндай атрибут мәні скриптік код болып табылады. Егер оқиға орындалса, онда атрибут аты арқылы кодталған скрипт іске қосылады. Onclick атрибуты «экрандағы белгілі нүктені тышқанмен шерткенде» орындалатын оқиғаны өңдеуді жүзеге асырады.

Мысалы: Қабырғасының ұзындығы берілген жағдайда квадраттың ауданын анықтау программасын жазайық. Ауданның мәні қабырғаның мәні өзгерген сәтте есептелуі тиіс. Екі мәтіндік өріс енгізілген форма құрайық: оның бірі квадраттың қабырғасының мәнін енгізу, ал екіншісі аудан мәнін есептелген мәнін шығару үшін қажет. **Жаңарту** батырмасы форма өрістерін тазалайды. Квадрат ауданы change оқиғасы туындыған мезетте, формадағы num1 деген элемент мәні өзгергенде және элемент фокусын жоғалтқанда есептеледі.

```
<HTML>
<HEAD>
<title> Change -элемент мәнін өзгерту оқиғасы</title>
<script>
function srec(obj)
{ obj.res.value=obj.num1.value* obj.num1.value}
</script>
</HEAD>
<BODY>
<h2>Квадрат ауданын
<FORM name="form1">
Қабырғасының ұзындығы: <input
name="num1"
onChange="srec(form1)">
<hr>
Аудан: <input type="text"
name="res"><hr>
<input type="reset"
</FORM>
</BODY>
</HTML>
```



есептеу</h2>

type="text" size=7

size=7

value=Жаңарту>

5. Интерактивті құжат мысалы

Экранда бір енгізу өрісі және төмендегідей сипаттамалары бар төрт батырма болсын:

- ағымдағы уақытты береді;
- жаңа терезе ашып, оған сәлемдесу мәтінін шығарады да, терезені жабады;

- есептелетін өрнекті енгізу өрісі;
 - енгізу өрісіндегі арифметикалық өрнектің мәнін көрсету өрісі;
 - терезе жабу батырмасы.
- Енді осы әрекеттерді программалайық.

```
<HTML>
<HEAD> <TITLE>Интерактивті парақ</TITLE>
<SCRIPT language=JavaScript>
function CurTime()
{   var now = new Date() ;
alert("ЕҮрiн: "+now.getDate()+"/"+
(now.getMonth()+1)+"/"+ now.getYear() +
"\nКәзiп: "+now.getHours()+":"+
now.getMinutes());   }

function Hallo()
{ var win=window.open("", "",
"width=400,height=300");
win.document.open();
var str = "";
str += '<H1>Сәлем!</H1>';
str += '<HR>';
str += '<P>';
str += 'Веб-программалаушылар!Қатарларың толыға берсін!';
str += '<P>';
str += '<FORM>';
str += '<INPUT type=button value=Жабу';
str += ' onclick="window.close()">';
str += '</FORM>';
win.document.write(str);
win.document.close();   }
</SCRIPT>
</HEAD>
<BODY bgcolor=white text=black>
<H2>Интерактивті парақ</H2> <HR>
<FORM>
<INPUT type=button value=Уақыт
onclick="CurTime()">
<INPUT type=button value=Сәлем
onClick="Hallo()">
<INPUT type=text value="2*2" size=25
name=calc>
<INPUT type=button value=жауап
onclick="alert(eval(this.form.calc.value))">
<P>
<INPUT type=button value=жабу
onclick="window.close()">
</FORM>
</BODY>
</HTML>
```

Мұндағы: **CurTime** функциясы **Уақыт** батырмасын шертуді өңдейді. **Hallo** функциясы **Сәлем** батырмасын шертуді өңдейді. **alert** функциясы **Жауап** батырмасын шертуді өңдейді. alert функциясы аргументі болып eval функциясының мәні есептеледі. eval (document,forma,calc.value) Eval функциясы аргументі болып берілген енгізу өрісінің мәні болып табылады. Мұндағы: forma – бұл объект, ол мынадай блок бейнесі

```
<form name=forma ... >...</FORM>;
```

calc - бұл объект, ол мынадай тәг бейнесі

```
<INPUT name=calc ... >;
```

value – бұл calc объектісі қасиеті. Window.close (...) функциясы **Жабу** батырмасын өңдейді. CurTime және Hallo функциялары сипатталған <script>...</script> блогы HTML-коды тақырыбында келтірілген. Себебі браузер функция сипатталуын оларды пайдаланғанға дейін оқуы тиіс, құжат тақырыбы ең алдымен орындалады ғой.

Программа жұмысы нәтижесі:

