

**ЎЗБЕКИСТОН АЛОҲА ВА АХБОРОТЛАШТИРИШ АГЕНТЛИГИ
ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ
УРҒАНЧ ФИЛИАЛИ**

ШАРИПОВ М.С., ЖУМАНИЁЗОВА Ғ

JAVASCRIPT

(WEB-ТЕХНОЛОГИЯЛАР ФАНИДАН МАЪРУЗАЛАР МАТНИ)

УРҒАНЧ-2006

МУНДАРИЖА

1. КИРИШ. ДАСТЛАБКИ КАДАМЛАР.	2
2. ХОДИСАЛАР.	4
3. ФУНКЦИЯЛАР.	5
4. HTML ХУЖЖАТЛАР ВА JAVASCRIPT ДАГИ ОБЪЕКТЛАР ИЕРАРХИЯСИ.	7
5. ФРЕЙМЛАР.	10
6. VRML-САХНАЛАРИНИ ДИНАМИК ЯРАТИШ.	20
7. КИСМ. ХОЛАТЛАР САТРИ.	22
8. САТР ХОЛАТИ ВА ТАЙМЕРЛАР.	22
9. JAVASCRIPT ДА ФОРМАЛАР.	26
10. ФОРМАГА КИРИТИЛГАН МАЪЛУМОТНИ.	30
11. IMAGE ОБЪЕКТИ.	32
12. КАТЛАМЛАР.	37
13. КАВАТЛАР 2.	40
14. JAVASCRIPT 1.2 ДА ХОДИСАЛАР МОДЕМИ.	45
15. ХОДИСАНИ КАМРАБ ОЛИШ.	47
16. DRAG ВА DROP.	49
17. MOUSEDOWN, MOUSEMOVE ВА MOUSEUP.	51
18. ХАРАКАТЛАНАЁТГАН ОБЪЕКТЛАРНИ КУРСАТИШ.	53
19. АДАБИЁТЛАР.	55

Кириш. Дастлабки кадамлар.

JavaScript – Netscape фирмаси томонидан ишлаб чиқилган скриптлардан тузилган янги тилдир. JavaScript ёрдамида сиз ажойиб Web саҳифалар яратишингиз мумкин. JavaScript ёрдамида нималар қилиш мумкинлигини қурасиз ва хаттоки қандай қилинишини билиб олишингиз мумкин.

JavaScript – бу Java эмас.

Қупчилик одамлар JavaScript – бу худди Javaнинг узи деб билишади, чунки уларнинг номи бир хилликка эга. Ҳақиқатда эса бундай эмас. Хозирча бу тиллар орасидаги фарқларнинг барчасини қурсатиб ўтиш ортиқча деб ўйлайман, шундай экан JavaScript бу Java эмаслигини эса тўтсангиз бас.

JavaScriptни ишлатиш(запуск).

JavaScriptда ёзилган скриптларни чиқариш учун нима қилиш зарур? Сизга JavaScript билан ишлаш қобилиятига эга бўлган браузер талаб қилинади. Масалан, Netscape Navigator (2.0 версиясидан бошлаб) ёки Microsoft Internet Explorer (MSIE – 3.0 версиясидан бошлаб). Бу браузерлар кенг қўлланилиб бошлангандан бошлаб, қупчилик одамлар JavaScript тилида скриптлар билан ишлаш имкониятига эга бўлди.

Албатта бу берилган дастурни ўқишдан олдин сиз HTML тилининг асослари билан танишган бўлишингиз шарт. HTML тили ҳақида Yahoo қидириш серверида “HTML” қалит сузи бўйича қидирувни ташкил қилиш керак.

HTML – Саҳифада JavaScriptни жойлаштириш.

JavaScript нинг скрипт кодлари HTML саҳифада тугридан тугри жойлаштирилади. Бунинг қандай қилинишини қуриш учун қуйидаги оддий мисол орқали қуриб чиқамиз:

```
<HTML>
```

```
<BODY>
```

```
<BR>
```

Бу оддий HTML ҳужжат.

```
</BR>
```

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

```
document.write (“Бу эса JavaScript”)
```

```
</script>
```

```
<BR>
```

Яна HTML. Ҳужжат.

```
</BR>
```

```
</BODY>
```

```
</HTML>
```

Бир қарашда бу мисол оддий HTML документни эслатади. Биргина бу ерда янги конструкция:

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

```
document write (“Бу эса JavaScript”)
```

</script>

Бу хакикатан ҳам JavaScript коди. Бу скриптнинг кандай ишлашини куриш учун жорий мисолни HTML файли сифатида ёзиб олинг ва браузерга юкланг.

Бу эса уша файл бажарилишининг натижаси (Агар JavaScriptни куллаб-куватловчи браузердан фойдалансангиз сизда 3 сатр булади):

Бу оддий HTML хужжат.

Бу эса JavaScript

Яна HTML. Хужжат.

Шуни айтиш керакки, берилган скрипт унчалик фойдали эмас, буни HTMLнинг «хакикий» тилида ёзганда ҳам булар эди.

Биз шунчаки <script> тегининг белгисини куриб чикдик. <script> ва </script> теглари орасидаги барча нарсалар JavaScript асосий буйруқлардан ташкил топади булардан бири document.write буйругини ишлатилишини куришингиз мумкин. document.write буйруги қачонки жорий документга(хужжатга) бирор нарса ёзиш керак булса ишлатилади. Демак, бизнинг кичкина дастуримиз HTML документда “ Бу эса JavaScript ” фразасини ёзади.

JavaScriptни куллаб-куватламайдиган браузерлар.

Агар браузер JavaScriptни қабул қилмаса, будай саҳифа кандай қуринишда булади. JavaScriptни куллаб-куватламайдиган браузерлар <script> тегини “танимайди”. Улар буни оддий матн сифатида чоп қилади. Бошқача қилиб айтганда бизнинг дастуримизда келтирилган JavaScript коди HTML документдан тугри марказида пайдо булади. Албатта бу бизнинг мақсадларимизга тугри келмайди. Бу ҳол учун махсус усул мавжуд ва бунда қириш скрипт коди браузернинг эски версияларидан беркитилади. Бунинг учун HTML нинг <!-- > коментария теги ишлатилади. Натижада бизнинг қириш кодимизнинг янги қуриниши қуйидагича булади:

<HTML>

<BODY>

Бу оддий HTML документ.

</BR>

<script language="JavaScript">

<!-- from old browsers>

document.write (“ Бу эса JavaScript”)

//...

</script>

Яна HTML документ.

</BR>

</BODY>

</HTML>

Браузер JavaScriptни куллаб-куватламаган ҳолда қуйидагича чоп қилар эди:

Бу оддий HTML документ.

Яна HTML документ.

Броузер JavaScriptни куллаб-куватламаган холда ва HTML нинг комментория теги йук булганда куйидагича чоп килар эди:

Бу оддий HTML документ.

document.write (“Бу эса JavaScript!”)

Яна HTML документ.

Илтимос, шунга эътиборингизни каратингки, сиз JavaScript нинг дастлабки кодини тула равишда беркита олмайсиз.

ХОДИСАЛАР.

Ходисалар ва уларни ишлаб чиқарувчилар JavaScript тилида дастурлаштиришнинг энг зарур қисми ҳисобланади. Ходисалар фойдаланувчининг у ёки бу ҳаракатларидан ташкил топади. Агар у қайсидир тугмани босса, мурожаатини кесиб утса, “Mouse Over” ходисаси юз беради. Ходисаларнинг бир неча турлари мавжуд. Биз JavaScript дастуримизда улардан бир нечасини бажаришимиз мумкин ва буни ходисаларни ишловчи махсус дастур ёрдамида амалга оширишимиз мумкин. Демак, тугмачани босиш орқали тушиб кетувчи ойна ходисасини яратиш мумкин. Бу шуни англатадики, Click – босиш ходисаси билан ойна яратиш реакцияси булиши керак.

Жорий вақтда биз ишлатишимиз керак булган дастур – ходисаларни ишлаб чиқарувчи OnClick деб аталади. Ва у компьютерга ходиса руй берганда нима қилиш кераклигини хабар беради.

Берилаган мисол бир нечта янги хусусиятларга эга, уларни тартиб буйича куриб чиқамиз. Биринчи янги хусусият бу OnClick=”alert(‘Y₀’)” булиб у <input> тегига тегишли. Айтиб утганимиздек, бу тугмани босганингизда нима содир булишини аниқлайди. Шу тарика, агар Click ходисаси юз берадиган булса, компьютер alert(Y₀) чакиригини бажариши керак. Бунинг узи JavaScript тилидаги кодга мисол булади (Шунга эътибор каратингки, бу ҳолатда биз хатто <script> тегида фойдаланмадик).

Alert функцияси тушувчи дарчаларни ҳосил қилиш имконини беради. Уни чакиришда сиз кавслар ичига бирор сатр беришингиз керак. Бизнинг ҳолатда бу ‘Y₀’. Ва бу худди шу матн тушаётган дарчада пайдо булади. Шу тарика, укувчи тугмачани шикирлатган пайтда, бизнинг скрипт ‘Y₀’ матнли дарча ҳосил қилади.

Берилган мисолда бир қанча чалқашликлар тугилиши мумкин: биз document.write() буйругида жуфт куштирнок, alert() конструкциясида эса бирлик куштирнок ишлатдик. Нега? Купчилик ҳолларда 2 та ҳолдан ҳам фойдаланиш мумкин. Бирок, сунги мисолда биз OnClick=”alert (‘Y₀’)” да бирлик ва 2 лик куштирнок ишлатдик. Агар биз OnClick=”alert (“Y₀”)” деб ишлатганда компьютер будай скриптга тушунмаган булар эди. Ва қайси конструкция OnClick функциясига тегишли, қайсиси тегишли эмаслигини аниқлай олмасди. Шунинг учун бундай ҳолларда 2 та куштирнок турини

ишлатишга мажбурсиз. Куштирнокни кандай тартибда ишлатишингиз муҳим рол уйнамайди, дастлаб иккилик куштирнок, сунгра бирилик куштирнок ва аксинча. Яъни сиз куйидагича ёзишингиз мумкин: `OnClick='alert ("Ү0")'`.

ФУНКЦИЯЛАР.

JavaScript тилида тузилган купчилик дастурларда функциялардан фойдаланамиз. Шунинг учун ҳозир бу тилнинг муҳим элементи ҳақида гапириш жуда муҳим. Функциялар купчилик ҳолларда узида бир неча буйруқларни боғлаш усулини номоён қилади. Келинг мисол учун бирор бир матнни уч марта бирданига чон қилувчи скрипт ёзамиз. Дастлаб оддий йулни курайлик:

```
<html >
< script language= "JavaScript">
< ! - - hide >
document.Write( "Менинг саҳифамга хуш келибсиз! < br >");
document.Write( "Бу JavaScript! < br >");
document.Write( "Бу JavaScript! < br >");
document.Write( "Бу JavaScript! < br >");
< /Script >
< /html >
```

Ва бу скрипт қушимча матнни босиб чиқаради:

Менинг саҳифамга хуш келибсиз!

Бу JavaScript!

Бу JavaScript!

Бу JavaScript!

Агар дастлабки кодига қарайдиган булсак, керакли натижани олиш учун коднинг аниқ бир қисми 3 марта такрорланган. Ахир бу самаралими?

Йўқ биз ушбу масалани осонроқ ечишимиз мумкин:

```
< html >
< script language= "JavaScript">
< ! - - hide
function myfunction () {
document. Write( `Менинг саҳифамга хуш келибсиз! < br >`);
document. Write( `Бу JavaScript! < br >`);
}
myfunction ();
myfunction ();
myfunction ();
< /Script >
< /html >
```

{ } кавс ичида жойлашган барча скриптлар `myfunction()` функциястга тегишли. Бу шуни англатадики `document.write()` дан иборат 2 та буйруқ

бирлаштирилди ва улар функция чакирилиши билан биргаликда бажарилади. Ҳақиқатдан ҳам бизнинг мисолимизда функция 3 марта чакирилган ва буни функциянинг узи аниқлагандан сунг myfunction() сатрини 3 марта ёзганимизни куришимиз мумкин. Ва уз навбатида бу шуни билдирадики, функциянинг таркибидаги нарсалар (буйруқлар, {} кавс ичидаги курсатилганлар) 3 марта бажарилган. Бу функциянинг ишлатилишига етарли даражадаги оддий мисол ва сизда нима учун функция JavaScript учун шу даража муҳим деган савол тугилади. Ушбу кулланмани уқиш мабойнида сиз албатта функциянинг фойдасини билиб оласиз. Функцияни чиқаришда узгарувчиларнинг узатиш имкониятининг узи бизнинг скриптлар учун жуда ихчамлик яратади ва буни биз кейинроқ кураимиз.

Функциялар яна ходисаларни ишловчи процедуралар билан биргаликда ишлатилиши мумкин. Навбатдаги мисолни кураимиз:

```
< html >
< script language= “JavaScript”>
< !-- hide
Function calculation( )
Var x = 12;
Var y = 5;
Var result = x+y;
Alert (result);
// - -
</script>
</head>
<body>
<form>
<input type = “button” value = “calculate” onclick = “calculation()”>
</form>
</body>
</html>
```

Бу ерда тугмачани босиш билан calculation() функциясини чиқариш жараёни руй беради. Сизган булсангиз, бу функция x,y ва result узгарувчиларидан фойдаланиб, кушиш амалини бажаради. Узгарувчиларни var калит сузи билан эълон қилиш мумкин. Узгарувчилар турли катталиқларни сон, сатр ва ҳақозаларни саклаш учун ишлатилади. Скриптнинг var result = x+y сатри браузерга result узгарувчисини ҳосил қилиш ва унга x+y арифметик амални жойлаштиришни (5+12) хабар беради. Бундан сунг result узгарувчиси 17 қийматни олади. Бу холда alert (result) ва alert(17) буйруги бир хил вазифани бажаради. Бошқача қилиб айтганда, бу 17 сон ёзилган дарчага эга буламиз.

HTML ХУЖЖАТЛАР ВА JAVASCRIPT ДАГИ ОБЪЕКТЛАР ИЕРАРХИЯСИ

Web саҳифанинг барча элементлари JavaScript тилидаги иерархик тузилиши буйича курилади. Хар бир элемент объект курилишида номаён булади. Ва бундан хар бир объект аниқ таркибга ва усулга эга булади. Уз навбатида JavaScript тили Web саҳифа объектларини осон бошқариш имкониятини беради. Ва бунда сиз объектлар иерархиясини тушунишингиз жуда муҳим. Буларнинг қандай руй беришини навбатдаги мисол орқали тез тушунишингиз мумкин.

Оддий HTML саҳифани курамитиз:

```
<html>
<head>
<title My homepage >
</head>
<body bgcolor = # ffffff>
<center>
<img scr = home.gif name="pic1" width = 200 height = 100>
</center>
<p>
<form name = "My form">
Name:
<input type = "text" name = "home" value = " " > <br>
e – mail:
<input type="button" value="pushme" name= "mybutton" onclick = "alert(Y)">
</form>
<p>
<center>
<img scr = "my.gif" name = "pic4" width = 300 height = 15>
<p>
<a href = http://maks/index.html"> </a>
</center>
</body>
</html>
```

Демак биз 2 та тасвир, 1 та муружат ва қандайдир матн чиқариш учун 2 майдондан иборат форма ва 1 та тугмачага эгамиз. JavaScript нуктаий назаридан браузер дарчаси – аллақандай Window объекти булиб ҳисобланади. Бу объект уз навбатида бир неча расмийлаштириш элементларига эга. Дарча ичкарасига биз html документини урнатишимиз урнатишимиз мумкин (ёки бошқа тилдаги файл, ҳозирча html файли билан чегараланиб турамитиз). Бундай саҳифа document объектидан сира ҳам фарқ қилмайди. Бу шуни англатадики document объекти JavaScript тилида ҳозирги пайтда юкланган html документни намоён этади. Document объекти JavaScript тилида жуда муҳим объектлардан бири ҳисобланади ва сиз ундан куп марта фойдаланасиз. Масалан Document объектининг таркибига , Web саҳифанинг

фонинг ранги киради. Шундай килиб биз учун шуниси муҳимки html объектларидан барчаси document объектининг таркибига киради. Html объектларига мисол, мурожат ёки тулдирилган форма була олади.

Куйидаги расмда мисолимиз оркали яратилган html саҳифанинг объектлари иерархияси тасвирланган:

Албатта биз бу иерархия ёрдамида турли объектлар хақида маълумотга эга булишимиз ва уни бошқариш имконига эга булишимиз керак. Бунинг учун биз JavaScript тилида турли объектларга рухсат кандай ташкил этилганлигини билишимиз шарт. Курииб турибдики иерархик тузилишга кура хар бир объект уз номига эга. Мос равишда агар сиз html саҳифадаги 1-расмга мурожат қилмоқчи булсангиз объектлар иерархиясидаги уз йуналишингизни танлаб олишингиз керак ва энг юкори чуққидан бошлаш керак. Бу тузилишдаги 1-объект document деб аталади. Саҳифадаги 1-расм images [0] объекти каби тасвирланган. Бу шуни англатадики JavaScript да document.images[0] деб ёзиб бу объектга рухсат олган булаемиз. Масалан агарда сиз форманинг 1-элементига кандай матн киритилганини билмоқчи булсангиз датслаб кандай килиб объектга рухсат олишни аниқлаб олишингиз керак ва яна объектлар иерархиясининг энг юкорисидан бошлайемиз. Сунгра elements[0] объектга йул курсатаемиз ва кетма–кетлик билан барча объектларни номини ёзамиз ва натижада шу нарса аниқланадики 1-майдонга рухсатни шуларни ёзиш оркали олиш мумкин:

Document.forms[0].Elements[0]. Энди фойдаланувчи томонидан киритилган матнни кандай килиб билиш мумкин. Объектнинг қайси тури ёки хоссаларидан бири бу информацияга рухсат бериши мумкинлигини аниқлаш учун JavaScript хақидаги маълумотномаларга мурожат қилиш керак. У ерда сиз киритилган матн майдонига мос келувчи элемент value хоссаси борлиги ва у киритилган матнга мос келади. Демак энди биз киритилган қийматни уқиш учун барча керакли нарсаларга эгамиз. Бунинг учун JavaScript сатрида куйидагини ёзиш мумкин.

Name = document.forms[0].elements[0].value

Олинган сатр **name** узгарувчисига юкланади. Энди биз бу узгарувчидан хоҳлаган керакли пайтда фойдаланишимиз мумкин. Масалан, биз alert (“Hi”+name) буйруги ёзилган дарча яратишимиз мумкин. Натижада, агар фойдаланувчи бу майдонга ‘Stefan’ сузини киритган булса alert (“Hi”+name) буйругига кура янги дарчада ‘Hi Stefan’ саломи пайдо булади. Агар сиз катта саҳифалар билан иш курсангиз турли объектларга номлар буйича адреслаш процедурасида жуда чалқашлик булиши мумкин. Масалан document.forms[3].elements[17] document.forms[2].elements[18] объектларига мурожат қилиш керак булсин? Бундай муоммадан кутилиш учун турли объектларга узингиз ном беришингиз мумкин. Буларни кандай қилишни мисол оркали куришингиз мумкин:

```
<form name = “myForm”
```

```
  Name
```

```
  < input type = “text” name = “kirit “ value = “ “ > <br>
```

бу ёзув шуни билдирадики, forms[0] объекти энди 2-чи исм myForm га эга булди. Худди шундай elements[0] урнига kirit ни ёзишингиз мумкин (<input> тегида охирги атрибути name да курсатилган). Шу тарика

```
  name = document.forms[0].elements[0].value урнига
```

```
  name = document.myForm.kirit.value ни ёзишингиз мумкин. Бу
```

сезиларли даражада JavaScript да дастурлашни соддалаштиради, айникса катта Web-саҳифалар куп микдорда объектларга эга булса. (Номларни ёзишда регистрларни, ката-кичик харфлар холатини эътиборга олиш керак, яъни myForm урнига myform ёзиш мумкин эмас) JavaScript да объектларнинг купчилик хусусиятларида рухсат факат укиш учун эмас. Яна сиз уларга янги кийматлар ёзиш имкониятига хам эгасиз.

JavaScript воситалари оркали курсатилган майдонга янги сатр ёзишингиз мумкин.

```
<form name = “myForm”
```

```
<input type = “text” name = “input” value = “ха ха ха”
```

```
<input type = “button” value = “write”
```

```
  onclick = “ document.myForm.input.value = ‘уф! ;’”
```

Дастлаб куйидаги мисолни куриб чикамиз:

Скриптни даслабки коди:

```
< html >
```

```
< head >
```

```
< title>Objects < /title >
```

```
< script language = “JavaScript”>
```

```
< !- -hide
```

```
  function first() {
```

```
// форма киритилган
```

```
// матн майдонида дарча хосил килади
```

```
alert (“Текст элементининг киймати :”+ document.myForm.myText.value) ;
```

```
}
```

```
  function second() {
```

```
// Жорий функция кучирувчилар кийматини аниклайди.
```

```
  Var mystring = “ The checkbox is ” ;
```

```
// Кучирувчи учганми ёки йук?
```

```
  If (document.myForm.mycheckbox.checked.) mystring += “checked”
```

```
  else mystring += “notchecked”;
```

```
// Экранга сатрни чикариш
```

```
  alert ( mystring ) ;
```

```
}
```

```
//- -
```

```
< /script >
```

```
< /head >
```

```

<body bgcolor = lightblue >
  < form name = “myForm >
    < input type = “text” name = “myText” value = “xa xa xa” >
    < input type = “button” name=“button1” value = “button1” >
      onclick = “first ()”
    <br>
    < input type = “checkbox” name = “mycheckbox”>
    < input type = “button” name=“button 2” value = “button2” >
      onclick = “second()”
    </form >
    < p > < br > < br >
< script language = “ JavaScript ”>
  <!-- - hide
  document.write( “Фон ранги бу : ” ) ;
  document.write( document.bgcolor+ “<br>” ) ;
  document.write( “ 2 чи button даги матн :” ) ;
  document.write( “ document.myForm.button 2.value” ) ;
// --
</script >
</body >
</html >

```

Location объекти

JavaScript да window ва document объектларидан ташкари location деб аталувчи яна бир муҳим объект мавжуд. Бу объектда HTML документ юкланган адрес сакланади. Агар сиз <http://www.xyz.com/page/html> саҳифасини юклаган булсангиз, у холда location.href нинг киймати ҳам худди шу адресга мос тушади. Шундай булсада биз учун жуда муҳим нарса шуки, сиз location.href га янги киймат ёзишингиз мумкин. Масалан, куйидаги мисолда тугмача жорий дарчадан янги саҳифага утишни юклайди :

```

< form >
< input type = button value=“Yahoo”>
  onclick = “location.href = ` http://www.Yahoo.com ` ; ”
</form >

```

ФРЕЙМЛАР.

Фрейм яратиш.

Фрейм ва JavaScript кандай қилиб бирга ишлайди деган савол тез-тез бериладиган саволлардан биридир. Дастлаб биз фрейм нима ва у нима учун ишлатилади деган саволни тушунтирамиз. Сунгра JavaScript ва фреймлар кандай биргаликда ишлатилишини куриб чикамиз. Умуман олганда браузер

дарчаси бир неча алохида фреймларга ажратилиши мумкин. Бу шуни англатадики, фрейм бу браузер дарчасидаги тугри туртбурчак шаклидаги бирор бир белгиланган майдон. Хар бир фрейм экранга узининг документини чикаради. Шу тарика мисол учун сиз 2 та фрейм яратмоқчисиз. Биринчи фреймга сиз Netscape фирмасининг “Уй саҳифасини” юклашингиз, 2 чига эса Microsoft фирмасиникини юклашингиз мумкин. Фреймлар яратиш HTML тили масаласи булсади, биз бу ерда бу жараённинг асосий куринишини тавсифлаб утамиз. Фрейм яратиш учун бизга <frameset> ва <frame> теглари керак булади. 2 та фрейм яратувчи HTML саҳифа куйидаги куринишда булиши мумкин :

```
<html >
<frameset rows “ 50% , 50% ”
  <frame src = “ page 1.htm ” name = “frame1”
  <frame src = “ page 2.htm ” name = “frame2”
</frameset >
</html >
```

Натижада 2 та фрейм яратилади. <frameset> тегида rows хоссасининг фойдаланилганини сезган булсангиз керак. Бу фреймлар бир-бири устидан жойланганини билдиради. Юкори фреймга page1.htm саҳифаси юкланади, пастки фреймга эса page2.htm документи жойлаштирилади. Тула яратилган фрейм таркиби куйидаги кифада булади :

Агар сиз документлар усти-устига эмас, ёнма-ён туришини хоҳласангиз, у холда <frameset> тегида rows эмас cols сузини ёзиш керак. “ 50% , 50% ” фрагменти фреймларни канчалик катталигини билдиради. Агар сиз ҳисоб-китоб билан узингизни кийнашни истамасангиз ва ҳисобда барибир 100% ҳисоб-китоб булишини истасангиз “ 50% * % ” ёзишингиз мумкин. Фрейм улчамини нукталарда бермоқчи булсангиз сондан сунг % белгисини куймаслигингиз керак. Истаган фреймга ном бериш учун < frame > тегида name атрибутидан фойдалансангиз етарли. Бундай имконият бизга JavaScript тилида фреймларга руҳсат учун керак булади. Web - саҳифалар яратишда бир неча куйилган < frameset > тегларидан фойдаланишингиз мумкин.

```
<frameset cols = “50%,50%”>
<frameset rows=“50%,50%”>
<frame src=“cell.htm”>
<frame src=“cell.htm”>
</frameset>
<frameset rows=“33%,33%,33%”>
< frame src=“cell.htm”>
```

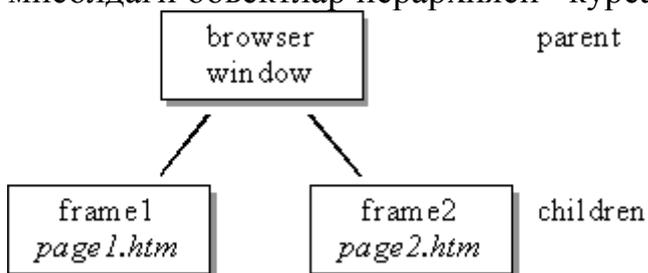
```
<frame src="cell.htm">
<frame src="cell.htm">
</frameset>
</frameset>
```

Натижада керак булган фреймлар таркиби куйидагича булади:

<frameset> тилида border параметрини ишлатган холда фреймлар орасидаги йугинликни беришингиз мумкин. Агар border=0 деб берилса бу сиз фреймлар орасида кандайдир масофа булишини хохламаётганингизни билдиради.

Фреймлар Ва Javascript

Энди келинг JavaScript да фреймларни браузер дарчасида кандай “куринишни” куриб чикамиз. Бу максатда ушбу кисмни берилишида келтирилган биринчи мисолдагидек иккита фрейм яратамиз. Куриб утганимиздек JavaScript web-сахифадаги курсатилганларнинг барча элементларини аллакандай иерархик таркибдаги куринишда ташкил килади. Худди шулар фреймларга хам таалуклидир. Навбатдаги расмда биринчи мисолдаги объектлар иерархияси курсатилган:

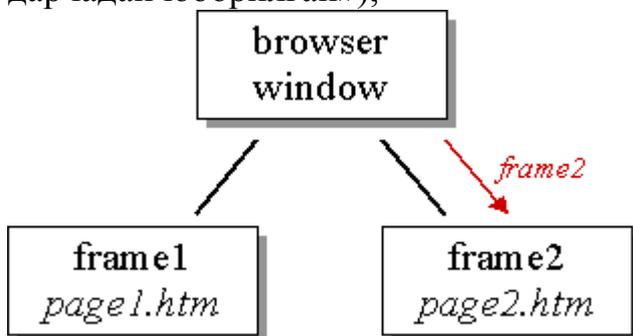


Иерархия чуққисида браузер дарчаси жойлашган (browser window). Жорий холатда у иккита фреймга ажратилган. Шу тарика дарча объект сифатида тур бошлиги яъни ушбу иерархиянинг онаси (parent), иккита фрейм эса мос равишда уларнинг авлодлари (children) хисобланади. Биз бу иккита фреймга frame1 ва frame2 деб ном бердик. Улар ёрдамида биз келтирилган иккита фреймлар билан маълумот алмашишимиз мумкин. Скрипт ёрдамида куйидаги масалани ечиш мумкин: Айтайлик, фойдаланувчи кандайдир мурожатни биринчи фреймда фаоллаштира, бироқ унга мос сахифа уша фреймда эмас, бошкасида юкланиши керак. Ушбу масалага мисол булиб мундарижа тузишни келтирсак ва бунда битта фрейм узгармас булиб қолади, лекин фойдаланувчига бир неча турли мурожатларни урганишга таклиф килади.

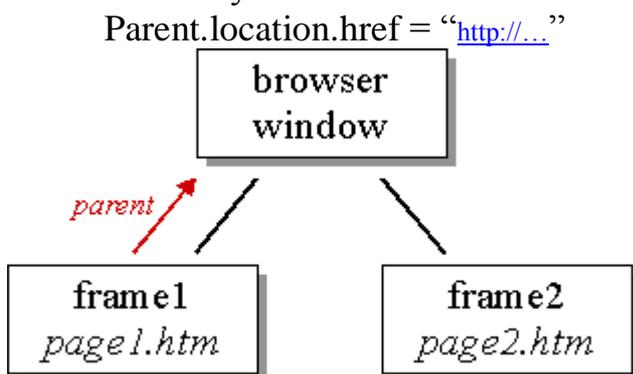
Ушбу масалани ечиш учун биз 3 та холни куриб чиқишимиз керак:

- асосий дарчалар фрейм авлод фреймга рухсат олади
- авлод-фрейм она дарчалар фреймга рухсат олади
- авлод фрейм бошка бир авлод фреймга рухсат олади

Объект нукталари назаридан «дарча» (window) иккита курсатилган фрейм frame1 ва frame2 деб номланади. Олдинги расмда куриш мумкинки, она дарча ва фреймлар орасида тугри олака мавжуд. Яъни агар сиз она дарча учун скрипт ёзсангиз яъни бу фреймларни яратувчи саҳифалар учун скрипт ёзсангиз у холда бу фреймларга уларни номлари оркали мурожаат қилишингиз мумкин. Маслан: `Frame2.document.write` («ушбу хабар она дарчадан юборилган»);

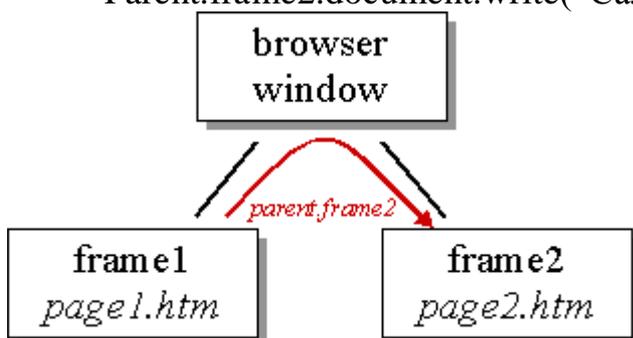


Айрим холларда сизга 1 фреймда туриб она дарчага рухсат олиш зарур бўлиб қолади. Масалан, бу агар сиз келаси утишда фреймлардан қутилишда зарур бўлади. Бундай холда фреймларни йукотиш факатгина мавжуд фреймлар урнига янги саҳифани юклаш оркали бажарилади. Бундай холда бу юкланиш она дарча саҳифасига бўлади. Буни она дарча - parent (ёки она фрейм) га унинг авлодлари бўлмиш фреймлардан рухсат олиш билан амалга ошириш мумкин. Янги документ юклаш учун `location.href` га янги URL адресларни киритиш керак. Фреймлардан қутилиш истагида эканмиз она дарчада `location` объектидан фойдаланиш керак. (шуни эслатамизки ҳар бир фреймга алоҳида саҳифа юклаш мумкин ва ҳар бир фрейм узининг `location` объектига эга). Демак биз она дарчага ушбу буйруқ оркали янги саҳифа юклашимиз мумкин:



Ва нихоят, сизга тез – тез учраб турадиган масалаларни бори бу 1 авлод фреймда туриб 2 авлод фреймга рухсат таъимнлаш масаласидир. Бизнинг масаламизда қандай қилиб иккита фрейм орасида ҳеч қандай тугри олака йуклигини куриш мумкин. Ва шунинг учун 2 чи фрейм борлиги ҳақида ҳеч нарсани билмайдиган frame 1 да туриб frame 2 ни чиқаришимиз мумкин эмас. Она дарча нуктаси назарига кура 2 чи фрейм мавжуд ва у frame 2 деб номланади ва 1 чи фреймдан она дарчага parent номи билан мурожаат қилса бўлади. Шу тариқа 2 чи фреймга жойлашган документга рухсатни қуйидагини ёзиш оркали олиш мумкин:

Parent.frame2.document.write(“Салом, бу чакирик / чи фреймдан! ”)



Навигацион панеллар (катновчи йулкалар)

Келинг, катнавчи йулкалар қандай яратилишини куриб чиқамиз. Битта фреймда Бир неча мурожатларни яратамиз. Лекин, агар фойдаланувчи улардан биттасини фаоллаштира мос саҳифа уша фреймда эмас, бошқасида жойлашсади.

Бунга мисол:

Дастлаб курсатилган фреймларни яратувчи скрипти ёзиб олишингиз керак. Бу документ худди олдинги қисмимизнинг берилишида куриб чиккан документдек қуринишда булади.

Fromes3.htm

```
<html>
<frameset rows="80%,20%">
<frame src="start.htm" name="main">
<frame src="menu.htm" name="menu">
</frameset>
</html>
```

бу ерда start.htm саҳифа асосий фрейм (main) да курсатиладиган саҳифа хисобланади. Бизда бу мавжуд саҳифа учун махсус талаблар йук.

Menu.htm

```
<html>
<head>
<script language="JavaScript">
<!-- hide

function load(url) {
  parent.main.location.href= url;
}

// -->
</script>
</head>
<body>

<a href="javascript:load('first.htm')">first</a>
<a href="second.htm" target="main">second</a>
<a href="third.htm" target="_top">third</a>
```

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

Куриб турганингиздек, янги сахифа юклаш урнига биз браузерга JavaScript да кандайдир буйрукни бажаришни таклиф киламиз. Яъни бунинг учун биз оддий href урнига JavaScript: параметридан фойдаланишимиз керак холос. Сунграк кавс ичида “first.htm” ни ишлатганимизни куришимиз мумкин. Бу сатр load функциясининг аргументи сифатида берилган. Load () функциясининг узи куйидаги тарзда аникланади.

```
Function load (url) {
Parent.main.location.href = url;
}
```

бу ерда кавс ичида url деб ёзилганини куришингиз мумкин. Бизнинг мисолимизда ‘first.htm’ сатри функция чакирилганда url узгарувчисига узлаштирилади. Ва бу янги узгарувчини load () функцияси *** фойдаланиш мумкин. Биз кейинрок узгарувчиларни бошка бир мисолда ишлатилиши билан танишамиз. Иккинчи мурожатда torget параметри катнашган. Аслини олганда Java S тегшили эмас. У html тилининг кулланмаларидан биридир. Куришиб турганидек, биз факат керакли фрейм номини курсатамиз. Бу холда курсатилган фрейм номидан олдин parent сузини ишлатиш шарт эмас, чунки torget параметри Java S эмас html тилининг функцияси хисобланади. Ва учунчи мурожатда torget ёрдамида кандай килиб фрейм кутилишни куришингиз мумкин. Агар сиз load () функцияси ёрдамида фреймларни йукотмокчи булсангиз факатгина унга

parent.location.href = url деб ёзсангиз етарли булади. Демак кайси усулни танлашингиз керак? Бу сизнинг нима килмокчилигингизга яъни script га боглик. Torget параметри жуда оддий ишлатилади. Сиз ундан янги сахифани бошка фреймга юкламокчи булган пайтингиздагина фойдаланишингиз мумкин. Java s тили асосида ечилган усул (бунга мисол биринчи мурожат булади) одатда агарда сиз мурожатни фаоллаштирганингизда 1 неча ходисалар руй бериш кузда тутилганда ишлатилади. Тез-тез учраб турадиган муоммалардан бири 2 сахифани 2 та фреймга юклаш шу турдаги масалаларга киради. Бу масалани torget параметри ёрдамида хам хал этиш мумкин, лекин Java S функциясидан фойдаланиш афзалрокдир. Фараз киламиз биз frame1 frame2 frame3 номли фреймларга эгамиз. Айтайлик frame1 да мурожатни фаоллаштирганда 2 та фреймнинг хар бирида хар хил Web сахифа юклансин. Ечим сифатида куйидаги функциядан фойдалансангиз булади.

```
Function load two ( ) {
Parent.frame 1.location.href = “first.htm”;
Parent.frame 2.location.href = “second.htm”;
}
```

Агарда шу функциянгиз янада ихчам булишини хохласангиз, у холда узгарувчини аргумент сифатида узатиш имкониятидан фойдалансангиз булади: Натижа куйидаги куринишда булади:

```
Function load two ( url1, url2 ) {
```

```

Parent.frame 1.location.href = url1;
Parent.frame 2.location.href = url2;
}

```

Бундан сунг load two("first.htm", "second.htm") ёки load two("first.htm", "forth.htm") функция чакиригини тахлил килса булади. Шубхасиз бу сизнинг функциягизни янада ихчам ва уни тез-тез, турли контекстларда ишлатишингиз мумкин булади.

ДАРЧА ВА ДИНАМИК ЯРАТИЛУВЧИ ДОКУМЕНТЛАР.

Дарча яратиш.

Броузерда янги дарча яратиш JavaS нинг энг зур кулайликларидан биридир. Биз бу янги дарчада янги документларни (масалан уша html документларини) ёки янги материалларни (динамик равишда) яратиб куйиш мумкин. Дастлаб биз янги дарча очиш кейин унга html – саҳифани кандай юклашни ва нихоят уни кандай ёпишни курамиз. Куйида келтирилган скрипт браузерда янги дарча очади ва унга кандайдир Web саҳифани юклайди.

```

<html>
<head>
<script language="JavaScript">
<!-- hide

function openWin() {
  myWin=open ("menu.html");
}

// -->
</script>
</head>
<body>

<form>
<input type="button" value="Open new window" onClick="openWin()">
</form>

</body>
</html>

```

Келтирилган мисолимизда янги дарча open усули ёрдамида menu.htm саҳифаси ёзилади. Биз дарча яратиш жараёнини бошқариш имкониятига эга эканлигингизни айтиб утишимиз керак. Масалан, янги дарча макола сатрига, ускуналар сатрига ёки мундарижага эга булиш мумкинлигини курсатишингиз мумкин. Масалан, навбатдаги скриптда 400x300 нуктали улчамдан иборат янги дарча очилади, уларга макола сатрига, на ускуналар панелига ва яна мундарижага эга.

```

<html>
<head>
<script language="JavaScript">
<!-- hide

```

```

function openWin2 () {
    myWin= open("ьутт.htm", "displayWindow",
        "width=400,height=300,status=no,toolbar=no,menubar=no");
}

// -->
</script>
</head>
<body>

<form>
<input type="button" value="Open new window" onClick="openWin2()">
</form>

</body>
</html>

```

Куриб турганингиздек дарча тартибини “width = 400, width = 300, status = no, toolbar=no,menubar = no” сатрида расмийлаштирамиз. Шунга эътиборингизни каратингки сизга бу сатрда символлар орасида буш жой куйиш талаб куйидмайди. Дарча бошкариладиган хоссалар руйхати

directories	yes no
height	<i>количество пикселей</i>
location	yes no
menubar	yes no
resizable	yes no
scrollbars	yes no
status	yes no
toolbar	yes no
width	<i>количество пикселей</i>

JavaS тилининг 1.2 версиясига бир неча янги хоссалар кушилган (яъни netscape navigator 4.0) сиз бу янги хосасаларни netscape 2x, 3x ёки Microsoft Internet Explorer 3x учун материал тайёрлашда ишлата олмайсиз чунки браузер Java s ни 1.2 версиясига тушунмайди.

Дарчанинг янги хоссалари

alwaysLowered	yes no
alwaysRaised	yes no
dependent	yes no
hotkeys	yes no
innerWidth	<i>количество пикселей (заменяет width)</i>
innerHeight	<i>количество пикселей (заменяет height)</i>
outerWidth	<i>количество пикселей</i>
outerHeight	<i>количество пикселей</i>
screenX	<i>количество пикселей</i>
screenY	<i>количество пикселей</i>
titlebar	yes no

z-lock	yes no
--------	--------

Биз бу хоссаларнинг изоҳини Java S тилининг 1.2 версияси тарифидан топишингиз мумкин. Сунгрок улардан бир нечасига тушинтириш ва мисоллар бериб утамиз. Масалан, энди сиз бу хоссалар ёрдамида янги очилаётган экраннинг қайси жойида жойлашишини аниқлашингиз мумкин. Java S нинг эски версиясида сузда бундай имконият йук.

Дарча номи.

Куриб турганингиздек дарча очаётганда бу учта аргументдан фойдаланишимиз керак:

```
myWin= open("bla.htm", "displayWindow",  
"width=400,height=300,status=no,toolbar=no,menubar=no");
```

2 – аргумент кимга керак? Бу дарча номи юкорида бу у target параметрида кандай ишлатилишини курган эдик. Демак, агар сиз дарча номини билсангиз у ерга янги саҳифа юклашингиз мумкин:

```
<a href="bla.html" target="displayWindow">
```

Бунда сиздан мос дарча номини курсатиш талаб килинади (агар шундай дарча мавжуд булмаса, шу номли янги дарча яратилади). Шунга эътиборингизни каратингки, бу ердаги mywin дарча номи эмас. Факатгина шу узгарувчи билан дарчага рухсатни олмайсиз. Модомики у оддий узгарувчи экан, унинг харакат майдони факатгина у аниқлаганунча скрипт билан белгиланади. Бундан ташқари дарча номи (жорий ҳолатда display window) ажойиб узгарувчи булиб уни ихтиёрий броузер дарчасидан туриб фойдаланса булади.

Дарчаларни ёпиш.

Сиз шунингдек Java S тили ёрдамида дарчаларни ёпиш имкониятига эгасиз. Буни қилишда сизга close () усули ёрдам беради. Келинг олдин қилганимиздек, янги дарча очамиз ва унга навбатдаги саҳифани юклаймиз.

```
<html>  
<script language="JavaScript">  
<!-- hide  
  
function closeIt () {  
    close();  
}  
  
// -->  
</script>  
  
<center>  
<form>  
<input type=button value="Close it" onClick="closeIt()">  
</form>  
</center>  
  
</html>
```

Энди эса, агар сиз янги дарча тугмачани боссангиз у ёпилади. Open () ва close () бу window объектининг усуллари ҳисобланади. Шуни эсда тутиш

керакки, биз факат open() ва close() деб эмас, балки window.open() ва window.close() деб ёзишимиз керак. Лекин бизникидай холатда window объектнинг тушириб колдириш мумкин, чунки агар сиз бу объектнинг факатгина 1 усулини чакирмоқчи булсангиз window ни колдирса булади. (Бу факат шу объект учун мумкин булган хол).

Документларни динамик яратиш.

Энди биз Java S нинг энг зур имконияти хисобланиш документларни динамик яратишларни куриб чикамиз. Яъни энди сиз скриптингизга Java S тилида узи янги html саҳифалар яратишга рухсат беришингиз керак. Бундан ташқари сиз шу тарика бошка Web документлар яъни vrlm – сахналар ва бошкалар каби. Кулайлик учун сиз бу документларни алоҳида дарча ёки фреймларда жойлаштиришингиз мумкин. Даствлаб биз оддий html документ яратамиз ва буни янги дарчада курсатамиз. Унинг скрипти куйидагича:

```
<html>
<head>
<script language="JavaScript">
<!-- hide

function openWin3() {
    myWin= open("", "displayWindow",
        "width=500,height=400,status=yes,toolbar=yes,menubar=yes");

    // открыть объект document для последующей печати
    myWin.document.open();

    // генерировать новый документ
    myWin.document.write("<html><head><title>On-the-fly");
    myWin.document.write("</title></head><body>");
    myWin.document.write("<center><font size="+3>");
    myWin.document.write("This HTML-document has been created ");
    myWin.document.write("with the help of JavaScript!");
    myWin.document.write("</font></center>");
    myWin.document.write("</body></html>");

    // закрыть документ - (но не окно!)
    myWin.document.close();
}

// -->
</script>
</head>
<body>

<form>
<input type=button value="On-the-fly" onClick="openWin3()">
</form>

</body>
</html>
```

Келинг mywin3() функциясини куриб чикамиз. Афтидан бу даствлаб браузерга янги дарча очамиз. Open() функциянинг 1 чи аргументи () буш сатр булгани учун яъни у шуни билдирадиги, бу жорий холатда url нинг аник

адресини курсатишини мумкин. Браузер фақат мавжуд документларни ишлаш эмас, балки у янги документларни кушимча равишда яратиши ҳам керак. Скриптда бу `mywin` узгарувчисини аниклаймиз. Ва унинг ёрдамида биз янги дарчага рухсат олиб биламиз. Шунга эътиборингизни каратингки, жорий ҳолатда ушбу мақсад билан биз дарча номига (`DisplayWindow`) мурожат қила олмаймиз. Дарчани очганимиздан сунг `document` нинг объектини ёзиш учун очишга жавоб келади. Бу ушбу буйрук ёрдамида амалга оширилади:

```
// документ объектини навбатдаги босмада очиш
```

```
myWin.document.open ();
```

Бу ерда биз `document` объектнинг усули `open ()` га мурожат қиламиз. Бирок бу усул `window` объектнинг `open ()` усули билан умуман бир хил эмас. Бу буйрук янги дарча очмайди, у фақат `document` ни навбатдаги босмага тайёрлайди. Бундан ташқари, бу `my win` олд кушимчасини `document open ()` олдидан кушишингиз керак, бу бизга янги дарча ёзиш имконини беради.

Куйидаги скрипт сатрларидаги `document.write ()` чакириги ёрдамида янги документ ёрдамида янги документ матни шаклланади:

```
// янги документ ташкил этиш
myWin.document.write("<html><head><title>On-the-fly");
myWin.document.write("</title></head><body>");
myWin.document.write("<center><font size=+3>");
myWin.document.write("This HTML-document has been created ");
myWin.document.write("with the help of JavaScript!");
myWin.document.write("</font></center>");
myWin.document.write("</body></html>");
```

қуриб турганингиздек, биз бу ерда `html` тилининг оддий тегларини ёзамиз. Яъни бу хақиқий `html` лойихасини ташкил этамиз. Бунда сиз `html` нинг ихтиёрий тегларидан фойдаланишингиз мумкин. Буни тугатиш учун биз яна документни ёпишимиз керак. Бу навбатдаги буйрук орқали бажарилади:

```
// документни ёпиш - ( дарчани эмас )
```

```
mywin.document.close ( );
```

Айтиб утганимиздек, сиз фақат документларни динамик яратиш эмас, балки уз ихтиёрингизга кура уларни у ёки бу фреймларда жойлаштиришимиз мумкин. Масалан, сиз 2 та фреймга эга булсангиз ва улар `frame1` ва `frame2` номли булса ва `frame2` да янги документ ташкил қилмоқчи булсангиз у ҳолда сиз `frame1` да туриб ушбуларни ёзсангиз етарли булади:

```
parent.frame2.document.open ();
```

```
parent.frame2.document.write("Here goes your HTML-code");
```

```
parent.frame2.document.close ();
```

§ VRML-САХНАЛАРИНИ ДИНАМИК ЯРАТИШ.

JavaS тилининг ихчамлигини намоиш қилиш учун, келинг энди VRML тилида динамик тарзда сахна ташкил этамиз. Эслатиб утамизки ,VRML вокеликни ветуал моделлаштириш каби кенгайтирилади. Яъни бу тил 3 улчовни сахналар яратиш тилидир. Оддий мисол келтирамиз кук куб. Бундан ташқари буни куриш учун сизнинг браузерингизга VRML нинг дастурий приставкаси керак булади. Киритилган скрипт VRML нинг браузерга кира олиш имкониятини аниқлайди.

Скриптнинг дастлабки коди:

```
<html>
<head>
<script language = "JavaScript">
<!-- hide
function vrml scene ( ) {
vrml = open ( " , ", display window", "width = 500,
           heidth = 400, status = yes, tool bar = yes, menu bar = yes " );
// документни куйидаги маълумот учун олиш
vrml.document.open ( "x - world /x - vrml " )
vr:=vrml.document;
//vrml сахнасини яратиш
vr.writeln ( "# vrml v1.0 ascii");
// ёритиш
vr.writeln ( "separator {d:rectional{");
vr.writeln ( "direction 3 -1 -2.5} ");
// камера
vr.writeln ( "Perspective Camera { postion -8.6 2.1 5.6");
vr.writeln ( "orientation -0.1352 -0.1233 1.1417");
vr.writeln ( "focal Distance 10.84");
// куб
vr.write ( "Separator { material { diffuse Color 001}");
vr.write ( "transform { transform -2.4 -2 1 rotation 0 0.5 1.9}");
vr.write ( "Cube {} }");
// документни ёпиш
vrml.document.close ( );
}
// - -
</script>
</head>
<body>
<form>
<input type = "button" value = "vrml on_the_fly" on click= "vrml scene ( )">
</from>
</body>
</html>
```

Курииб турибдики, скрипт матни олдинги мисолимиздагидай. Дастлаб янги дарча очилади. Сунгра унга маълумот чиқариш учун document очамиз.

Унга мос буйрукни батафсил куриб чикамиз:

```
// куйидаги маълумотни чикариш учун document очиш  
vrml.document.open ( “x – world /x – vrml ”);
```

Олдинги мисолларда бу кавс ичида хеч нарса курсатмаган эдик. У холда “x – world /x – vrml ” ёзув нимани англатади? Аслида олганда биз ушбу кулланма ёрдамида яратаётган document га MIME типини берамиз. Яъни биз бу билан браузерга бериладиган маълумотлар кандай типдалигини хабар киламиз. Агар биз уша жойда кавс ичида MIME нинг аник типини аниклаб олмаганимизда янги документга “ text/html” типи танланган булар эди. (Бу эса html файллар учун MIME типининг узидир).

Уч улчамли сахна яратиш учун биз vrml.document.write () кулланмасини ташкил килишимиз керак. Лекин бу жуда узун булганлиги сабабли, биз оддий vr дан vrml.document узгарувчисини аниклаб оламиз. Ва сунгра vrml.document.write () деб ёзиш урнига биз факат vr.write () деб ёзамиз. Буни тамомлагандан сунг биз vrml тилида оддий кулланмаларни ёзишимиз мумкин. Биз бу ерда vrml сахнасининг элементарини тавсифини келтириб утирмаймиз. Улар билан танишишни истовчилар учун интернетда бир неча яхши маълумот манбалари бор. Vrml тилида оддий матн куйидагича булади:

```
#vrml v1.0 ascii  
separator {  
  DirectionalLight { direction 3 -1 -2.5 }  
  Perspective Camera {  
    Position -8.6 2.1 5.6  
    Orientation -0.1352 -0.1233 1.1417  
    Focal Distance 10.84  
  }  
  separator {  
    Material {  
      Diffuse Color 001  
    }  
  }  
  Transform {  
    Translation -2.4 -21  
    Rotation 0 0.5 1.9  
  }  
  cube {}  
}
```

Бу document.write буйруги ёрдамида экранга чикармоқчи булган коднинг худди узи. Лекин оддий vrml файли билан юклаш мумкин булган имкониятни динамик тарзда сахна яратиш бемаънилиқдир.

КИСМ. ХОЛАТЛАР САТРИ
САТР ХОЛАТИ ВА ТАЙМЕРЛАР.
§ Сатр холати (холатлар сатри)

Java S да сиз томондан тузилган дастурлар ҳолатлар сатрига ёзув ёзиш мумкин яъни у сизни браузерингиз дарчасини пастки қисмида туртбурчак шаклида бўлади. Буни қилиш учун сизга атиги window.status да керакли сатрни ёзишингиз етарлидир. Навбатдаги мисолда шундай иккита тугмача ташкил қилинадик улар орқали сиз ҳолат сатрига бирор бир матнни ёзишингиз ва мос ҳолда сунгра уларни учришингиз мумкин:

```
<html>
<head>
<script language="JavaScript">
<!-- hide

function statbar(txt) {
    window.status = txt;
}

// -->
</script>
</head>
<body>

<form>
    <input type="button" name="look" value="Писать!"
        onClick="statbar('Привет! Это окно состо\яни\я!');">
    <input type="button" name="erase" value="Стереть!"
        onClick="statbar('');">
</form>

</body>
</html>
```

Демак, биз иккита тугмадан иборат форма ташкил этамиз. Бу иккала тугмача ҳам statbar() функциясини чиқаради. Ёзиш тугмачасидан чақурув қуйидагича бўлади: statbar ‘салом! Бу ҳолатлар сатри!’ Кавс ичига биз ‘салом! Бу ҳолатлар сатри!’ деб ёздик. Ва бу statbar функциясига юборилалиган матннинг узидир. Уз навбатида statbar() функциясини қандай аниқланганини қуришингиз мумкин:

```
function statbar(txt) {
    window.status = txt;
}
```

Функция саҳифасида txt сузи жойлаштирилган бу биз функцияга узатайтган сатр txt узгарувчисига узлаштиришини билдиради. Функцияга узгарувчилар қиритиш тез – тез қулланиладиган ва функцияга ихчамлик келтирадиган ҳолатдир. Сиз функцияга бир неча аргументларни қиритишингиз мумкин, фақат уларни вергул билан ажратиш керак.

Тхт сатри window.status = txt буйруги орқали ҳолатлар сатрига олиб борилади. Шунга мос ҳолда матнни ҳолатлар сатридан йукотиш window.status ёзувида буш сатрни узлаштириш орқали бажарилади.

Ҳолатлар сатрига матнни чиқариш механизми мурожатлар билан ишлашда жуда қулай. Жорий мурожатнинг url ини экранга чиқариш урнига, сиз оддийгина кейинги саҳифада нима дейилишини суз билан

тушинтиришингиз мумкин. Ушбу мисолдаги link – факатгина сичконча курсаткичини мурожат устига жойлаштиришни намоиш килади.

```
<a href="dontclick.htm"
  onMouseOver="window.status='Don\'t click me!'; return true;"
  onMouseOut="window.status='';">link</a>
```

onmouseover га нима учун true кийматни беришимиз керак деб сурашингиз мумкин. Аслини олганда бу шуни англатадики, браузер бундан кейин mouseover ходисасини ишлашнинг шахсий кодини бажармсалиги керак. Коидага кура браузер холатлар сатрида мос мурожатнинг url ини курсатади. Агар биз true кийматини бермасак, у холда бизнинг код бажарилгандан дарров браузер уз усули буйича холатлар сатрига ёзади, яъни бизнинг матн шу ернинг узида учиради ва укувчилар кура олмайди.

Бу скриптда сиз яна бир нарсани куришингиз мумкин. Айрим холларда сизга куштирок символларни чоп этиш керак булиб қолади. Масалан биз dont click me матнини чоп этишимиз керак ва биз уни onmouseover ходисаларни ишлаш процедурасига берсак унда бирлик куштироклар ишлатилади. Натижада биз dont ... деб ёзсак, браузер апостров символарида адашади. Ушбу муоммани ечиш учун бизга факат куштирок белгиси олдидан \ белгини куйиш етарли, бу жорий белги факат чоп этиш учун куйилганини билдиради (бу холат иккилик куштироклар учун хам мумкин).

§ Таймерлар.

Time out (ёки таймерлар) функцияси ёрдамида сиз компьютерни бир неча буйрукларини кандайдир вақтгача бажарилишини дастурлаштиришингиз мумкин. Навбатдаги скриптда шундай тугмача намоиш килинганки, у тушаётган дарчани дарров эмас, балки 3 секунддан сунг бажаради:

```
<script language="JavaScript">
<!-- hide

function timer() {
  setTimeout("alert('Врем\я истекло!')", 3000);
}

// -->
</script>

...

<form>
  <input type="button" value="Timer" onClick="timer()">
</form>
```

Бу ерда setTimeout() window объектининг усулларидан биридир. У вақт интервалини урнатади, мен уйлашимча бу кандай содир булишини англадингиз? Биринчи чакирув аргументи бу Java S коди булиб у берилган вақтда бажарилувчи хиобланади. Бизнинг холатда у “alert (‘вақт булди’)” га тенг. Илтимос шунга эътиборингизни каратингки Java S коди куштирокка олиниши керак. 2 чи аргументда компьютерга ушбу код қачон бажарилиши

хабар килинади. Бунда биз вақтни миллисекундларда курсатишимиз керак. (3000 миллисекунд = 3 секунд).

§ Прокрутка (Айлантириш).

Энди сиз кандай килиб холатлар сатрига ёзув ёзишни ва таймерлар билан кандай ишлашни билган вақтингизда айлантиришни бошқаришни урганишга утсак ҳам булади. Сиз аллақачон матнли холатлар сатридан жойлаштиришни куриб олдингиз. Интернетда бу усулдан ҳамма жойда кулланилади. Энди биз асосий чизгичда кандай килиб айланишни дастурлаштиришни куриб чикамиз. Хамда бу чизгичнинг барча мумкин булган ижобий узгартиришларини куриб чикамиз. Югурувчи сатр яратиш деярли оддий. Келинг дастлаб кандай килиб холатлар сатрида жойлашувчи матн – югурувчи сатрни яратиш мумкинлиги хақида уйланиб оламиз. Эхтимол, биз олдин холатлар сатрига бирор бир матн ёзиб олишимиз керакдир. Кейин вақт утиши билан биз у ерга факат уни сал чапрокка суриб уша матнни узини ёзишимиз керак. Агар биз буни бир неча марта килсак, фойдаланувчида у югурувчи сатр билан иш олиб бораётгандай таасурот колдиради. Лекин, биз шуни эсда тутишимиз керакки хар сафар холатлар сатрида матннинг кайси кисми куринишини хисоблаб боришимиз керак.(Коида буйича матннинг катталиги холатлар сатрини улчамини ошириб юборади). Бу тугмача дарча очади ва айлантиришга мисол келтиради.

Бу скриптнинг бошлангич коди ва мен унга бир неча изохлаб киритдим.

```
<html>
<head>
<script language="JavaScript">
<!-- hide

// define the text of the scroller
var scrtxt = "Это JavaScript! " +
    "Это JavaScript! " +
    "Это JavaScript!";
var len = scrtxt.length;
var width = 100;
var pos = -(width + 2);

function scroll() {

    // напечатать заданный текст справа и установить таймер

    // перейти на исходную позицию для следующего шага
    pos++;

    // вычленить видимую часть текста
    var scroller = "";
    if (pos == len) {
        pos = -(width + 2);
    }

    // если текст еще не дошел до левой границы, то мы должны
    // добавить перед ним несколько пробелов. В противном случае мы должны
    // вырезать начало текста (ту часть, что уже ушла за левую границу
```

```

if (pos < 0) {
  for (var i = 1; i <= Math.abs(pos); i++) {
    scroller = scroller + " ";
    scroller = scroller + scrtxt.substring(0, width - i + 1);
  }
} else {
  scroller = scroller + scrtxt.substring(pos, width + pos);
}

// разместить текст в строке состо\яни\я
window.status = scroller;

// вызвать эту функцию вновь через 100 миллисекунд
setTimeout("scroll()", 100);
}

// -->
</script>
</head>

<body onLoad="scroll()">
Это пример прокрутки в строке состояния средствами JavaScript.
</body>
</html>

```

Scroll() Функциясининг куп кисми маттни фойдаланувчи курадиган кисмини ажратиб олишдан иборат. Биз бу кодни кенг равишда тушунтириб утирмаймиз, факатгина сиз умуман бу айлангириш кандай жойлаштирилганини тушунсангиз булди. Ушбу жараённи ишга тушуриш учун биз `<body>` тегидан ёзилган `onload` ходисаларини ишлаш процедурасидан фойдаланамиз. Яъни `scroll()` функцияси HTML саҳифа юклангандан дарров чакирилади. `Onload` процедураси воситасида `scroll()` функциси чакирилади. `Scroll()` функциясида биз биринчи булиб таймерни урнатамиз. Бу билан биз `scroll()` функцияси хар 300 секундда чикарилишини кафолатлаймиз. Бунда матн бир кадамга утказилади ва бошка таймер юборилади ва шу тарика у бетухтов давом этади.

JAVASCRIPT DA FORMALAR.

Формалар интернетда кенг ишлатилади. Формага киритилгпн маълумотлар тез – тез кайта серверга юборилади ёки бирор бир адресга электрон почта оркали жунатилади. Муаммо шундай, фойдаланувчи томонидан киритилган маълумот тугрилигига ишониш. Уни интернетга юборишдан олдин Java S тили ёрдамида осон текшириш мумкин. Олдин бу формани текшириш кандай булишини курсатмокчи эдик, кейин эса интернетга маълумот жунатишининг кандай имкониятлари борлигини курсатамиз. Дастлаб биз оддий скрипт яратайлик. Айтайлик html-саҳифа 2 та матн киритувчи элементга эга булсин. Улардан бирига фойдаланувчи уз исмини, 2 – сига электрон почта манзилини киритсин. Сиз у ерга бирор бир маълумотни киритиб тугмани босиб курунг.

Агар сиз у ерга хеч нарса киритмаган булсангиз хатолик юз бериш хакида маълумот оласиз. 1 – элементдаги исталган маълумот тугрилиқ жихатдан текширилади. Албатта агар фойдаланувчи бошка номни киритган булса у буларни кафолатламайди. Браузер хатоларни сонларга хам каршилиқ курсатмайди. Масалан, сиз '17' деб киритсангиз у холда сиз 'Салом 17' деган саломни оласиз. Демак, бу текшириш идеал була олмайди.

Форманинг 2 чи элементи бир мунча мураккабирокдир. Унга оддий сатр киргизиб, масалан исмингизни. Сиз унга @ ни ёзмагунча бу элемент иш бажармайди. @ белгини мавжудлиги билан фойдаланувчи электрон почта манзилини тугри киритган була олади. Гарчи маъносиз булсада биргина @ белгиниг узи хам бу шартга тугри келади.интернетдаги хар бир электрон почта манзилида @ белги катнашади. Шундай экан бу белгига текшириш бу ерда жуда уринли. Скрипт бу 2 та элемент билан кандай ишланди ва унда текшириш кай курунишда булади. Бу куйидагича олиб борилади:

```
<html>
<head>
<script language="JavaScript">
<!-- Скрыть

function test1(form) {
    if (form.text1.value == "")
        alert("Пожалуйста, введите строку!")
    else {
        alert("Hi "+form.text1.value+"! Форма заполнена
корректно!");
    }
}

function test2(form) {
    if (form.text2.value == "" ||
        form.text2.value.indexOf('@', 0) == -1)
        alert("Неверно введен адрес e-mail!");
    else alert("OK!");
}
// -->
</script>
</head>

<body>
<form name="first">
Введите Ваше имя:<br>
<input type="text" name="text1">
<input type="button" name="button1" value="Проверка"
onClick="test1(this.form)">
<p>
```

```

Введите Ваш адрес e-mail:<br>
<input type="text" name="text2">
<input type="button" name="button2" value="Проверка"
onClick="test2(this.form)">
</body>
</html>

```

Дастлаб </body> булимидаги кодни куриб чикамиз. Биз бу ерда факат 2 та тугмачани матн киритиш учун яратамиз. Бу тугмачалар test1(...) ва test2 (...) функцияларини кай бири босилганини караб чиқаришда.

Функцияга аргумент сифатида this.form комбинациясини берамиз, чунки у кийинчилик бизга уша функцияга ва худди уша бизга керак элементга мурожат қилишга рухсат беради. Test1 (form) функция берилган сатр бушлигини текширади. If (form.text1.value= = "")... орқали бажарилади. Бу ерда узгарувчи 'form' га функция чакирилганда 'this form' дан олинган киймат берилади. Биз элементга киритилаётган сатрни form text ига value сузини кушиб аниқлаб олишимиз мумкин. Сатр буш эмаслигига ишониш учун уни " " билан таккослаб курамиз. Агар киритилган сатр " " билан мос келса у холда сатрга ҳеч нима киритилмаган булади ва булардан фойдаланувчи хато хақида хабар қабул қилади. Агар бирор бир нарса киритилган бўлса фойдаланувчи "ok" тасдигини олади. Кейинги муаммо шундан иборатки фойдаланувчи форма майдонига пробел киритиш мумкин ва бу тугри маълумот сифатида қабул қилинади! Агар сида бу пробелларни олиб ташлаш истаги бўлса у холда албатта унга бу имкониятни текширишни кушишингиз мумкин. Энди test2(form) функциясини курамиз. Бу ерда яна киритилган сатр бушлик билан текширилади. (уқувчи томонидан киритилган маълумот хақиқатдан бор йуклигига ишониш учун). Лекин биз if буйругига яна нимадир қушдик. // белгилар йигиндиси "ok" буйруги деб аталади. If буйрукги биринчи ва иккинчи таккослаш нима билан тугалланишини текширади. Агар булардан биттаси бажарилса if буйруги true кийматига эришади ва кейинги скрипт бажарилади. Суз билан тушунтирганда келтирилган сатр ёки буш ёки унда @ белгиси йуклиги хақида хатолик деган хабар оласиз. (if буйругидаги 2 чи оператор киритилган сатр @ белгисига эга эканлигини текширади).

Маълум бир белгилар борлигини текшириш.

Айрим холларда формага киритилаётганлар факат айрим белгилар ёки сонлардан иборат булиши яъни маълумотни чегаралаш керак булиб қолади. Келтирилган маълумот факат сонлардан иборат булган телефон номерлари хақида эслашимиз қифоя. Бунда сизга келтирилган маълумотлар сонми йукми, шуни текшириш керак булади. Бу холатда мураккаблик шундан иборатки купчилик одамлар телефон номерларига яна хар хил белгилар, масалан, 01234-5678, 01234/56789 ёки 1234_56789 кушишлари мумкин. Фойдаланувчидан телефон номерларида бундай белгиларни фойдаланишини ман қилиб бўлмайди. Шу боис биз скриптимизни сонлар ва бир неча

Белгиларни текширувчи процедурани кушишимиз керак. Скриптнинг дастлабки коди куйидагича:

```
<html>
<head>
<script language="JavaScript">
<!-- hide
function check(input) {
  var ok = true;

  for (var i = 0; i < input.length; i++) {
    var chr = input.charAt(i);
    var found = false;
    for (var j = 1; j < check.length; j++) {
      if (chr == check[j]) found = true;
    }
    if (!found) ok = false;
  }

  return ok;
}

function test(input) {

  if (!check(input, "1", "2", "3", "4",
    "5", "6", "7", "8", "9", "0", "/", "-", " ")) {

    alert("Input not ok.");
  }
  else {
    alert("Input ok!");
  }
}

// -->
</script>
</head>

<body>
<form>
Telephone:
<input type="text" name="telephone" value="">
<input type="button" value="Check"
  onClick="test(this.form.telephone.value)">
</form>
</body>
</html>
```

Формага киритилган маълумотни

такдим этиш.

Формага киритилган маълумотларни беришнинг кандай имкониятлари мавжуд? Форма маълумотларини беришнинг энг оддий усули бу электрон почта оркали узатишдир. (бу усулни кенгрок урганамиз). Агар сиз формага киритилаётган маълумотларга сервер куз-кулок булиб туришини хохласангиз у холда сиз CGI интерфейсидан фойдаланишингиз керак. У сизга маълумотларни автоматик равишда кайта номлашга имкон беради. Масалан, сервер мижозлардан айримлари кира олиш имконияти булган маълумотлар базасини яратиш мумкин. Бошка мисол- уаһоо сингари кидириш сахифалари. Одатда унда шахсий маълумотлар базасидан яратилган суровни кидириш учун форма ташкил килинган. Натижада фойдаланувчи мос тугмачани босишдан олдинрок жавобни олади. Унга токи жорий серверни куллаб-кувватловчи одамлар уларга курсатилган маълумотларни укиш ва кидиришни кутиб утиришга хожат колмайди. Буларнинг барчасини сервернинг узи бажаради. Java S бундай имкониятларни бажаришга йул куймайди.

Java S маълумотларни сервердаги ьирор бир файлга ёзиб олиш имконияти булмагани сабабли у ёрдамида укув такризлари китобини яратаолмайсиз. Буни факат CGI интерфейси оркали бажариш мумкин.

Албатта, сиз фойдаланувчилар электрон почта оркали юборган маълумотлар учун такриз китоби яратишингиз мумкин. Лекин бу холда сиз электрон почта оркали олинган такризларни кулда битта файлга киритишингиз керак. Агар сиз хар куни 1000 такриз олишни назарда тутмасангиз шундай килишингиз мумкин. Шунга мос скрипт оддий HTML матндан иборат булади ва хеч кандай Java S дастурлаштириш керак эмас. Албатта битта холни хисобга олмаганда, яъни агар сизга маълумотларни юборишдан олдин текшириш керак булса бу ерда хакикатан хам Java S керак булади. Факат биз хамма ерда хам ишламайдиган mailto буйругини киритамиз, у факат Microsoft Explorer 3.0 да куллаб-кувватланмайди.

```
<form method=post action="mailto:your.address@goes.here"
enctype="text/plain">
Нравится ли Вам эта страница?
  <input name="choice" type="radio" value="1">Вовсе нет.<br>
  <input name="choice" type="radio" value="2" CHECKED>Напрасная
трата времени.<br>
  <input name="choice" type="radio" value="3">Самый плохой сайт
в Сети.<br>
  <input name="submit" type="submit" value="Send">
</form>
```

Enctype="text/plane" параметри матнни худди узини хеч кодланган кисмларсиз юбориш учун ишлатилади. Бу сезиларли даражада бундай почтани укишни соддалаштиради.

Агар сиз формани сетга узатилмасдан текширишни истасангиз, у холда onSubmit ходисалар ишловчи дастурдан фойдаланишингиз мумкин. Сиз бу дастур учун < form > тегида чакирик жойлаштиришингиз мумкин. Масалан:

```
function validate() {
  // check if input ok
  // ...

  if (inputOK) return true
  else return false;
}

...

<form ... onSubmit="return validate()">

...
```

Агар шу тарика тузилган формада нотугри маълумотлар киритилган булса бу форма интернетга юборилмайди.

§ Форманинг аник элементини белгилаш.

Focus() усули ёрдамида формани янада самимий килиш мумкин. Демак, сиз кайси элемент биринчи булиб белгиланишини танлашингиз мумкин. Демак, сиз кайси элемент биринчи булиб белгиланишини танлашингиз мумкин. Ёки браузерга нотугри маълумотлар киритилган форматни белгилашни буюришингиз мумкин. Яъни браузернинг узи курсорни сиз курсатган формат элементига урнатади ва фойдаланувчига бирор нарса курсатишда форматни чиқаришга тугри келмайди. Буни куйидаги скрипт формати оркали килиш мумкин:

```
Function setfocus( ) {
  Docuent.first.text1.focus( );    }
```

бу ёзув мен курсатиб утган критилувчи матннинг биринчи элементини бегилайди. Сиз барча формаланинг жисмларни курсатишингиз керак, жорий холда у first деб аталади ва форманинг элементи номи text1. Агар сиз саҳифа юкланганда жорий элемент белгиланишини истасангиз бунинг учун сиз <body> тэгини onload атрибути билан тулдиришингиз мумкин. Бу шундай куринишда булади:

```
<body onload = "setfocus( )">
```

Яна бир мисол билан тулдиришингизга тугри келади:

```
Function setfocus( ) {
  Document.first.text1.focus( );
  Document.first.text1.select( ); }
```

Яна бир кдни куринг:

Бунда факат элемент белгиланмайди, балки ундаги матн ҳам белгиланади.

IMAGE ОБЪЕКТИ.

Web саҳифада тасвир.

Энди Java S нинг 1.1 версиясидан бошлаб ишлаш мумкин булган Image объектини куриб чикамиз. (яъни netscape navigator 3.0 билан). Image объекти ёрдамида сиз Web саҳифангизга график образлар билан узгартиришлар киритишингиз мумкин. Бу бизга қисман мултипликатция яратиш имконини беради. Шуни эслатамизки, фойдаланувчилар браузерларнинг эски версиялари орқали ушбу қисмда келтирилган скриптларда фойдалана олмайди. (Netscape Navigator ёки Internet Explorer 3.0 каби). Ёки яхши ҳолатда ҳам улардан тула эффект олиб булмайди.

Келинг дастлаб Java S да Web саҳифага тасвирни қандай қилиб жойлаштиришни куриб чикамиз. Биз қураётган тилда тасвир массив қуринишда берилади. Массив бу image деб номланади ва document объектининг асосларидан бири ҳисобланади. Web саҳифадаги ҳар бир тасвир тартиб рақамига эга: 1-тасвир 0 рақамини, 2-эса 1 ва х к каби рақамларни олади. Шу тариқа 1-тасвирга биз document.image[0] деб мурожат қилишимиз мумкин. Html документдаги ҳар бир тасвир Image объекти узининг аниқ бир хусусиятига эгаки, қайсики Java S тилидан унга мурожат қилиш мумкин. Масалан, сиз унинг хусусиятлари дан width ва height лар орқали тасвирнинг улчамини аниқлашингиз мумкин. Яъни document.image[0].width ёзув орқали Web саҳифадаги тасвирнинг кенглигини аниқлашингиз мумкин(нукталарда). Бахтга қарши, барча тасвирнинг индексларини узгартириб туриш қийин кечади, айниқса агар бир саҳифангизда улар етарли қуп булса. Ушбу муаммо тасвирларга узининг шахсий номини белгилаш орқали ҳам булади. Демак, агар сиз тасвирни ушбу тег

```
<img src = "img.gif" name = "myimage" width = "100" height="100">
```

орқали киритсангиз у ҳолда сиз унга document.myimage ёки document.images["myimage"] деб ёзиш билан мурожат қилишингиз мумкин.

§ Янги тасвирларни юклаш.

Албатта бу биз хоҳлаган нарса булмаса ҳам Web саҳифада тасвир улчамини қандай олишни билиб қуйган мумкин. Бу Web саҳифада тасвирлар алмашилиши жорий қилишни истаймиз ва бунинг учун бизга src атрибути керак булади. тегидаги ҳолатдаги каби src атрибути келтирилган тасвирнинг манзилини саклайди. Энди сиз Java S нинг 1.1 версиясида аллақачон Web саҳифага юкланган тасвирга янги адрес бериш имкониятига эгасиз. Ва натижада тасвир Web саҳифага эски адресни янги адрес билан алмаштирган ҳолда юкланади. Мисол учун ушбу ёзувни қураамиз:

```
<img src = "img1.gif" name = "Myimg" width = 100 height = 100>
```

Бу ерда img1.gif тасвир юкланади ва Myimg номини олади. Навбатдаги сатрда олдинги img1.gif тасвир аллақачон янги img2.gif билан алмашади:

```
Documunt.MyImages.src = "img2.gif";
```

Бунда янги тасвир хар доим эски тасвир улчамини олади. Ва сиз энди унинг улчамини узлаштираолмайсиз.

§ Тасвирларни огохлантирувчи юкланиши.

Бундай харакатларимизнинг камчиликларидан бири булиб, src га янги адрес берилгандан сунг унга мос тасвир юкланиш жараёни бошланиши хисобланади. Ва бу олдиндан килинмаганлиги сабабли янги тасвир интернет оркали узатилгунча ва уз жойига боргунча бир неча вақт утади. Айрим холларда бу мумкин, лекин бундай тез – тез тухталишларга куниш мумукин эмас. Бу холат билан биз нима килишимиз керак? Албатта, ба масаланинг ечими булиб, тасвирларни огохлантирувчи юкланишдан фойдаланиш хисобланади.

Бунинг учун бу янги Image объектини яратишимиз керак. Навбатдаги сатрларни куриб чикамиз:

```
HiddenImg = new Images();  
HiddenImg.sec = "img3.gif";
```

1 – сатрда янги Image объекти яратилади. 2 – сатрда кейинчалик hiddenimg объекти ёрдамида номаён буладиган тасвир адреси курсатилади. Куриб утганимиздек, src атрибутидаги янги адрес браузерини ушбу берилган адресли тасвирни юклашга мажбур килади. Шунинг учун, бизнинг мисолимиздаги 2-сатр бажарилганда img2.gif тасвир юклана бошлайди. Лекин HiddenImg нинг уз номидан келиб чиккан холда (“яширин тасвир”) браузер юклашни тамом килгандан сунг экранда тасвир пайдо булмайди. У факат компьютернинг хотирасида келгуси фойдаланиш учун сакланган булади. Экранга тасвирни чикариш учун биз

```
document.myImage.src = hiddenImg.src;
```

сатридан фойдаланамиз:

лекин энди тасвир хотирадан дарров олинади ва экранда намоён булади. Шу тарика биз тасвирни огохлантирувчи юкланишни бошкардик.

Сиз аник бир ходисага реакция сифатида тасвирларнинг алмашилишидан фойдаланган холда ажойиб натижа олишингиз мумкин. Масалан, сичконча курсаткичи сахифанинг аник бир жойига борганда тасвирни алмаштиришингиз мумкин. Навбатдаги мисолда оддий килиб сичконча курсаткичи тасвирга жойлаштирилганида текшириб куриш (умуман бунда сиз агар браузерингиз факатгина Java S 1.0 ни к-кув.диган булса, хатолар хакида хабар оласиз, бундан кутилишни сунгрок куриб чикамиз).

Хато! Рухсат килинмаган гиперматн объекти.

Ушбу мисолнинг дастлабки коди куйидаги куринишда булади:

```
<a href="#"  
  onMouseOver="document.myImage2.src='img2.gif'"  
  onMouseOut="document.myImage2.src='img1.gif'">  
</a>
```

Бунда навбатдаги муоммалар тугилиши мумкин:

- Укувчи Java S 1.1 к-кув.майдиган браузердан фойдаланяпти

- 2 чи тасвир юклатилмаган
- Бунинг учун биз Web саҳифадаги ҳар бир тасвир учун янги командалар ёзишимиз керак
- Биз шундай скриптга эга булишни хоҳлар эдикки, қайсики уни Web саҳифаларнинг купчилигида яна ва яна ишлатиш имкони бўлса.

Энди биз ушбу муоммаларни ечувчи скриптни тула вариантини куриб чиқамиз. Бу скрипт узун булганига қарамадан сиз уни бир ёзиб олишингиз билан ушбу муоммалар ҳақида қайгурмайсиз. Ушбу скрипт узини ихчамлигини саклаб қолиши учун 2 та шартни куриб чиқиш керак:

```

<html>
<head>

<script language="JavaScript">
<!-- hide

// *****
// Script from Stefan Koch - Voodoo's Intro to JavaScript
//      http://rummelplatz.uni-mannheim.de/~skoch/js/
//      JS-book: http://www.dpunkt.de/javascript
//      You can use this code if you leave this message
// *****

    // ok, у нас браузер с поддержкой JavaScript
    var browserOK = false;
    var pics;

// -->
</script>

<script language="JavaScript1.1">
<!-- hide

    // браузер с поддержкой JavaScript 1.1!
    browserOK = true;
    pics = new Array();

// -->
</script>

<script language="JavaScript">
<!-- hide

var objCount = 0; // количество изображений на web-странице

function preload(name, first, second) {

    // предварительная загрузка изображений и размещение их в массиве

```

```

    if (browserOK) {
        pics[objCount] = new Array(3);
        pics[objCount][0] = new Image();
        pics[objCount][0].src = first;
        pics[objCount][1] = new Image();
        pics[objCount][1].src = second;
        pics[objCount][2] = name;
        objCount++;
    }
}

function on(name){
    if (browserOK) {
        for (i = 0; i < objCount; i++) {
            if (document.images[pics[i][2]] != null)
                if (name != pics[i][2]) {
                    // вернуть в исходное состо\яние все другие
изображени\я
                    document.images[pics[i][2]].src = pics[i][0].src;
                } else {
                    // показывать вторую картинку, поскольку курсор
пересекает данное изображение
                    document.images[pics[i][2]].src = pics[i][1].src;
                }
            }
        }
    }
}

function off(){
    if (browserOK) {
        for (i = 0; i < objCount; i++) {
            // вернуть в исходное состо\яние все изображени\я
            if (document.images[pics[i][2]] != null)
                document.images[pics[i][2]].src = pics[i][0].src;
        }
    }
}

// заранее загружаемые изображени\я - Вы должны здесь указать
// изображени\я, которые нужно загрузить заранее, а также объект
Image,
// к которому они относ\ятс\я (первый аргумент). Именно эту
часть
// нужно корректировать, если Вы хотите использовать скрипт
// применительно к другим изображени\ям (конечно это не
освобождает
// Вас от об\язанности подредактировать в документе также и
раздел body)

preload("link1", "img1f.gif", "img1t.gif");
preload("link2", "img2f.gif", "img2t.gif");
preload("link3", "img3f.gif", "img3t.gif");

```

```

// -->
</script>
<head>

<body>
<a href="link1.htm" onMouseOver="on('link1') "
  onMouseOut="off()" ">
</a>

<a href="link2.htm" onMouseOver="on('link2') "
  onMouseOut="off()" ">
</a>

<a href="link3.htm" onMouseOver="on('link3') "
  onMouseOut="off()" ">
</a>
</body>
</html>

```

Берилган скрипт барча тасвирларни pics массивига жойлаштиради. Ушбу массив энг дастлаб чакирилган preload () функцияси ёрдамида яратилади. Preload () функциясини чакируви оддийгина куйидаги курунишда булади.

Preload (" Link 1", " img1f.gif", " img1f.gif");

Бу шуни англатадики, бу скрипт сервердан 2 та img1.gif ва img2.gif тасвирларини юклаши керак. Улардан биринчисини сичконча курсаткичи тасвир майдонига тушмаганда хосил буладиган тасвирдир. Фойдаланувчи сичконча курсаткичини тасвирга жойлаштира, у холда 2 чи тасвир пайдо булади. Preload () функциясини чакириш мабойнида " link 1 " сузини 1 чи аргумент сифатида курсатамиз ва Web саҳифадаги Image объектига олдиндан юклатилган 2 та тасвирни бериб куямиз. Агар сиз бизнинг мисолимизда < body > булимини куриб чиксангиз уша link 1 ном билан аталган тасвирни топасиз. Биз тасвирни Web-саҳифада хосил килиш учун скриптни ёзиб юрмасдан унинг тартиб ракамидан эмас балки унинг номидан фойдаланамиз. On () ва off () функцияларни иккаласи хам onMouseOver ва onMouseOut ходисаларни кайта ишлаш программаси воситасида чакирилади. Image элементининг узи MouseOver ва MouseOut ходисаларини бажариб билмагани учун биз бу тасвирларни мурожаат килишимиз керак. On () функцияси курсатилган тасвирдан бошқасини дастлабки холатига келтиришни куришимиз мумкин. Акс холда дарров бир неча тасвир белгиланган булиб қолади, шунинг учун буни килиш жуда зарур. (Гап шундаки агарда фойдаланувчи дархол дарча чегарасидан курсаткични тасвирга жойлаштира MouseOut ходисаси регистрация килинмайди).

9 КИСМ.КАТЛАМЛАР.

Катлам – бу Netscape Navigator 4.0 браузерининг ажойиб янги имкрниятларидан биридир. У Web – сахифаларни тасвир каби объектларини аник жойлаштиришга имкон яратади. Бундан ташкари сиз HTML сахифангизга турли объектларни куйишингиз мумкин. Хаттоки объектларни куринмас килишингиз мумкин. Java S тилида _____ ни осон бошкаришингиз мумкин. Жорий вақтда _____ дан сиз факат Netscape Navigator 4.0 фойдаланишингиз мумкин. Аслини олганда катлам деганда нимани тушунамиз? Буни тушунтириш осон ва буни оддий мисолда, бир неча когоз варагини олиб тушунтирамиз. Бир варақка матн ёзинг. Башкасига тасвир чизинг ва хакоза. Энди бу варақларни стол устига куйинг. Фараз киламиз хар бир варақ бирор бир катлам . у хар хил объектларни бизнинг мисолимизда матн ва тасвирни ичига олиши мумкин. Энди тасвирли варақни олиб стол буйлаб харакатлантирамиз. Диккат билан тасвирни варақ изидан харакатланишини кузатинг. Агар биз варақни уннга бурсак тасвир хам уннга бурилади! Катламлар узида турли объектларни саклаши мумкин, масалан тасвирлар, формалар, матнлар HTML сахифангизга куйилиши мумкин. Агар сиз бирор бир катламни харакатлантирсангиз у холда бу каватдаги колган барча объектлар хам бу харакатларни такрорлайди. Каватлар бир-бири билан стол устидаги оддий кагаз варақлари сингари жойлашиши мумкин. Хар бир каватда ялтирок майдон булиши мумкин. Варақда тешик яъни 1 чи варақдаги “ ялтирок майдон ” дан пастда жойлашган варақлар куринади.

Каватлар хосил килиш .

Кавт хосил килиш учун биз < layer > ва < ilayer > тегида фойдаланишимиз керак. Куйидаги параметрлардан фойдаланишингиз мумкин:

Name = “ layerName” Left = x position Top = y position z-index = layer index width = layer width clip = “ x1-offset, y1-offset,x2- offset,y2-offset ” above = “ layerName” below = “ layerName ” visibility show/hide/inherit bgcolor = “ rgbcolor ” background = “ imageurl ”	Варақ номи Чапдан юкори бурчакни абциссаси Чапдан юкори бурчакни ординатаси Кават учун индекс номери Кават кенглиги (нуктада) Каватнинг куринувчи кисмини белгилайди Бизнинг кават тагида кайси кават борлигини аниклайди Кайси кават бизнинг кават устидалигини аниклайди Бу каватнинг куриниши Фон рангги стандарт ранг номи ёки rgb ёзув Фон тасвир.
---	---

<layer> теги аник жоллаштиришингиз мумкин булган каватлар учун ишлатилади. Агар сиз кават холатини курсатмасангиз (left top параметрлари

ёрдамида) у холда жимлик холатига кура у дарчанинг юкори чап бурчакни эгаллайди.

<i>layer</i> тег ёрдамида

Келинг энди оддий мисолдан бошдаймиз. Биз 2 та кават хосил килмокчимиз. Улардан биринчисига тасвир, иккинчисига эса жойлаймиз. Бу килмокчи булган нарса ушбу матнни жорий тасвир устида курсатиш.

Дастлабки коди:

```
<html>
<layer name=pic z-index=0 left=200 top=100>
  <img src = "img.gif" width=160 height=120>
</layer>
<layer name=txt z-index=1 left=200 top=100>
<font size =+4> <i>layers-demo</i> </font>
</layer>
</html>
```

Куриб турибмизки <layer> теги ёрдамида биз 2 та кават шаклини аниклаймиз. 2 та кават хам 200/100 (left ва top параметрлари оркали) сингари жойлаштирилади. <layer> ва </layer> ёки <i>layer</i> ва </i> теглари орасидаги барча нарсалар тавсифланаётган каватга тегишлидир. Бундан ташкари биз курсатилган каватларнинг пайдо булиш тартибини z-index параметрлари оркали белгиладик яъни бизнинг холатда сиз браузерга матн тасвир юкорисида ёзилишни хабар берамиз. Умуман олганда, z-index нинг энг юкоридаги кавати барча каватлар устида курсатилади. Сиз z-index ни танлашда факатгина 0 ва 1 кийматлар билан чегараланган эмассиз-умуман истаган мусбат сонларни танлашингиз мукин.

JavaScript ва каватлар.

Энди биз каватларга Java S оркали кандай рухсат олишни куриб чикамиз. Фойдаланувчи тугмачани босиш оркали бирор бир каватни курсатиши ёки бекитиш имкониятига эга булувчи мисолни куришдан бошлаймиз. Дастлаб бу каватлар Java S да кай тарзда хосил булишини куришимиз керак. Одатдагидек бунинг бир неча усуллари бор. Энг яхшиси хар бир каватга ном бериб чикиш. Демак, агар биз

```
<layer ... name=mylayer>
```

```
...
```

```
</layer>
```

</layer> деб кават хосил килсак у холда биз унга кейинчалик document.layers["mylayer"] конструкцияси оркали рухсат олишимиз мумкин. Бу каватларга рухсатни биз бутун сонли индекс оркали олишимиз мумкин. Демак, энг пастки каватга рухсат document.layers[0] деб ёзиш билан олинади. Шунга эътибор каратингки индекс бу z-index параметр билан бир хил нарса эмас. Агар мисол учун сиз layer1 ва layer2 деб номланувчи z-index номерлари 17 ва 100 булган каторларга эга буламиз. У холда сиз уларга document.layers[17] ва document.layers[100] деб эмас балки document.layers[0] ва document.layers[1] оркали рухсат олишингиз мумкин. Каватлар Java S скриптлари ёрдамида узгартириб булувчи бир неча хусусиятларга эга.

Навбатдаги мисолда 1 – каватни бекитувчи ёки аксинчахосил килувчи тугмача ташкил килинган. (Netscape Novigator +4 ёки ундан юкори версиялари талаб килинади).

Дастлабки коди куйидаги курунишда булади:

```
<html>
<head>
<script Language = “JavaScript”
<!-- hide
    function showHide ( ) {
        if (document.layers[“mylayer”].visibility == “show”)
            document.layers[“mylayer”].visibility = “hide”
        else (document.layers[“mylayer”].visibility = “show”);
    }
    //--
</script>
</head>
<body>
<ilayer name = mymylayer visibility show
<font size =+1 color= “#0000ff”> <i> this text inside a layer
</i> </font>
</ilayer>
<form>
<input type = “button” value = “show/hide layer” onclick = “ShoeHide”>
</form>
</body>
</html>
```

Берилган тугмача ShowHide () функциясини чиқаради. Бу функция layer (mylayer) объектининг курунувчанлик хусусиятига рухсат олиш масаласи ечилган. Document.layers[“mylayer”].visibility параметрига “show” ёки “hide” каватини бериб сиз ушбу каватни курсатиш ёки беркитиш имкониятига эга буласиз. Show ва hide бу қирқиб олинган калит сузлар эмас балки сатр хисобланиб, сиз Document.layers[“mylayer”].visibility = show деб ёза олмайсиз.

§ Каватларни жойлаштириш.

Left ва top хусусиятлар берилган каватнинг ҳолатини белгилайди. Сиз уларга янги каватлар бериб уни узгартиришингиз мумкин. Масалан навбатдаги сатрда каватнинг горизантал ҳолати 200 нуктада бериляпти:

```
Document.layers[“mylayer2”].left = 200;
```

Энди биз каватларни аралаштириш дастурига утамиз, у браузер дарчасида қандайдир айланишлар яратади. Скрипт куйидаги курунишда:

```
<html>
<head>
<script Language = “JavaScript”
<!-- hide
```

```

function move ( ) {
if (pos<0) direction = true;
if (pos >200) direction = false;
if direction pos++ else pos --;
document.layers[“mylayer2”].left = pos;
}
//-- > </script>
</head>
<body onload = “setinterval (‘move ( )’,20)”
<ilayer name = mylayers2 left = 0>
<font size = +1 color = “#0000ff”> <i> this text is inside a layer </i>
</font>
</ilayer>
</body>
</html>

```

Биз mylayer2 номли кават яратдик. <body> тегида биз onload процедурасини ишлатганимизни куриш мумкин. Бизга саҳифа юклатилгандан дарров кават айланишини бошлаш зарур. Onload ходисаларини кайта ишлаш процедурасида биз setInterval () функциясидан фойдаланамиз. Бу Java S версиясининг янги усулларидан биридир. Ундан бирор бир функцияни яна ва кайта яна аник бир интервалада чакиришда фойдаланилади. Олдин биз бунинг учун setTimeout вақт функциясидан фойдаланганмиз. setInterval функцияси ҳам худди шундай ишлайди, бироқ сиз уни бир марта чакирсангиз етарли булади.

setInterval функциясида биз move () функциясини ҳар бир 20 миллисекунддан чакириш имконига эгамиз. move () функцияси эса уз навбатида ҳар 20 миллисекунддан каватни янги соҳага алмаштиради. Бу функцияни яна ва яна чакиришимиз натижасида матнимизни тез ҳаракатланишини кураемиз. move () функциясида киладиган нарсамиз шуки кават учун янги координата ҳисоблаш ва уни ёзиш: document.layers[“mylayer2”].left = pos.

10 ҚИСМ. КАВАТЛАР 2.

Биз аллақачон каватлар янги технологиясининг асосий тушунчаларига эга булдик. Бу қисмда куйидаги мавзуларни куриб чикамиз.

- Каватдан қирқиб олиш.
- Қуйилган каватлар.
- Ялтирок каватлар билан турли хил самаралар олиш.

Каватлардан қирқиб олиш.

Каватнинг қандайдир қисми (туртбурчак) қуринадиган қилиш мумкин. Бундай усул қирқиб олиш дейилади. Масалан, HTML саҳифасида куйидаги қирқиб олиш функциясини беришимиз мумкин:

```

<layer left = 0 top = 0 clip = "20,50,110,120">
<img src = "daviner.jpg" width = 209 height = 264>
</layer>

```

Берилган тасвир фрагменти 90x70 улчамга эга. Clip атрибутидаги 1 чи 2 та киймат киркилаётган кисмининг чапдан юкори бурчакни курсатада. Навбатдаги 2 та киймат пастки унг бурчакни курсатади. Java S тили ёрдамида киркилган кисмини бошкариб бундан хам кизикарли натижаларга эришиш мумкин. Аниқрок айтганда Layer объектини хусусиятлари булган clip.left, clip.top, clip.right ва clip.button ларни узгартиришингиз мумкин. Агарда бу хусусиятларга янги киймат берсангиз худди шу вақтнинг узида бу узгариш мос равишда кабул килинади. Куйидаги мисолда тасвирни киркиб олинган кисмини параметрлари динамик равишда узгариб туради ва фойдаланувчида тасвир аста-секин “усаётган” дай таасурот колади:

```

<html>
<head>
<script Language = "JavaScript"
<! - - hide
    var middle x, middle y, pos;
function start ( ) {
    // тасвир улчамини олади
var width = document.layers[ " imglayer "].document.davincn.width ;
var height = document.layers[ " imglayer "].document.davincn. height ;
    // тасвир маркази нуктасини аниқлайди
middle x = math.round ( width/2 ) ;
middle y = math.round ( height/2 ) ;
    // дастабки позиция
    // юбориш
    show ( )
}
function show ( ) {
    // киркилаётган кисмини улчамни катталаштириш
    pos+ = 2 ; //
    document.layers[ " imglayer "].clip.left = middle x-pos ;
    document.layers[ " imglayer "].clip.top = middle x-pos ;
    document.layers[ " imglayer "].clip.right = middle x+pos ;
    document.layers[ " imglayer "].clip.button = middle x+pos ;
    //
    if ( !( ( pos middlex) && ( pos middley )))
        setTimeout( " show ( ) ", 20 ) ;
}
    // - -
</script>
</form>
<body>
< layer name = " imglayer " clip = " 0,0,0,0 ">

```

```

< img name = devinci scr = “ devinci.jpg ” width = 209 height = 264 >
</ilayer >
< form >
< input type = button value = “ start ” onclick = “ start( ) ”
</form>
</body>
</html>

```

<body> тегида тавсифланган тугмача start () функциясини чакиради. Даглаб биз ишни тасвирни кайсидир нуктасидан бошлашни аниқлаймиз. Бу нуктанинг x ва y координаталарини midde x ва midde y узгарувчиларга юклаймиз. Бундан сунг биз тасвирни киркилган кисмига middeleX, middleY ва pos узгарувчиларига боглик равишда улчам берувчи show () функциясини чикарамиз. Хар сафар show () функцияси чакирилганда pos функцияси киймати автоматик равишда ошиб боради. Show () функцияси охирида setTimeout () чакириги ёрдамида таймер урнатамиз ва бунда бу функция ва яна чакирилади. Бу жараён қачонки бу киркилган кисм бутун булиб курунгунча давом этади. Шу ерда Start () функцияси бошида тасвирнинг улчамини олганимизни эътиборга олиб утиш керак.

```

Var width = document.layers[“imglayer”].document.doviuci.width;
Var height = document.layers[“imglayer”].document.doviuci.height;

```

Document.layers[“imglayer”] конструкцияси ёрдамида бу imglayer номли каватга мурожат қилишимиз мумкин. Бирок нима учун document.layers[“imglayer”] дан сунг document деб ёзилади?

Гап шундаки, хар бир кават узининг HTML саҳифасига яъни хар бир кават document объектига эга. Imglayer каватдаги тасвирга рухсат олиш учун биз Варақнинг қолган барча майдони тоза ушбу document объектига рухсат олишимиз керак. Юқорида келтирилган мисолимизда бундай тасвир dovinci номига эга эди. булиши керак.

§

Куриб утганимиздек, кават узида бир неча объектларни мужассамлаштириши мумкин. Улар хаттоки бошқа каватларни уз ичига олиши мумкин. Хақиқатан эса қуйилган каватлардан фойдаланишнинг бир неча сабаблари бор. Бир неча қуйилган каватларни номоён қилувчи мисолларни куриб чикамиз. Биринчи мисолда иккита бошқа (layer1 ва layer2) кават қуйилган кават (parentlayer деб номланади) ишлатилади.

Бу биринчи кават

Бу иккинчи кават

Бу асосий (она) кават

Move/stopparent	Move/stoplayer	Move/stoplayer
-----------------	----------------	----------------

Саҳифни очганимиздан сунг биз 3 та тугмача курамиз. Бу тугмачалар каватлар харакатини бошлаши ёки тухташи мумкин.

Энди даглабки код скриптини курамиз:

```

Html

```

```

Head

```

```

<script Language = “JavaScript”

```

```

<! - - hide

```

```

        // дастлабки позиция
var pos0 = 0 ;
var pos1 = -10 ;
var pos2 = -10 ;
        // харакат
var move0 = true ;
var move1 = false ;
var move2 = false ;
        // йуналиш
var dir0 = false ;
var dir1 = false ;
var dir2 = true ;
        function startstop ( which ) {
                if ( which == 0 ) move 0 = ! move 0
                if ( which == 1 ) move 1 = ! move 1
                if ( which == 2 ) move 2 = ! move 2
        }
function move ( ) {
        if ( move 0 ) {
                // parentlayer ни жойлаштириш
if ( dir 0 ) pos 0 - -
                else pos + + ;        if ( pos 0 < -100) dir 0 = false ;
if ( pos 0 100 ) dir 0 = true ;
                document.layers [ “ parentlayer ”].left = 100 + pos 0 ;
        }
        if ( move 1 ) {
                // parentlayer ни жойлаштириш
if ( dir 1 ) pos 1 - - else pos 1 + + ;
if ( pos 1 < -20 ) dir 1 = false ;
if ( pos 1 20 ) dir 1 = true ;
                document.layers [ “ parentlayer ”].layers[ “ layer ”].top = 10 + pos 1 ;
        }
        if ( move 2 ) {
                // parentlayer ни жойлаштириш
if ( dir 2 ) pos 2 - - else pos 2 + + ;
if ( pos 2 < -20 ) dir 2 = false ;
if ( pos 2 20 ) dir 2 = true ;
                document.layers [ “ parentlayer ”].layers[ “ layer 2 ”].top = 10 + pos 2
;
        }
}
// - -
</script >
</head >
< body onload = “ setInterval ( ‘ move ’ , 20 ) ”

```

```

    < ilayer name = parentlayer left = 100 top = 0 >
      < layer name = layer 2 z-index = 10 left = 0 top = -10 >
        // бу 1 чи кават
      < /layer >
      < layer name = layer 2 z-index = 20 left = 200 top = -10 >
        // бу 2 чи кават
      < /layer >
      < br > < br >
      // бу асосий ( она ) кават
    < /ilayer >
    < form >
      <input type = "button" value="Move/Stopparentlayer" onclick="startstop(0)
;”>
      <input type = "button" value="Move/Stopparentlayer1" onclick="startstop(1)
;”>
      <input type = "button" value="Move/Stopparentlayer2" onclick="startstop(2)
;”>
    </form>
  </body>
</html>

```

parentlayer ичида биз 2 та каватни аникладик. Булар худди кушилган каватларга эга булади. Java S тилида бу каватларга рухсатни кандай олиш мумкин. Бу кандай килинади, уни move () функциясида куриш мумкин:

```

document.layers [ " parentlayer "].left = 100 + pos 0 ;
- - - - -
document.layers [ " parentlayer "].layers[ " layer "].top = 10 + pos 1 ;
- - - - -
document.layers [ " parentlayer "].layers[ " layer 2 "].top = 10 + pos 2 ;

```

Куйилган каватларга рухсат олиш учун layer 1 ва layer 2 каватлар parentlayer ичида жойлашганлиги сабабли сизга document.layers ["layer 1"] ёки document.layers[" layer 2 "] деб ёзиш етарли булмайд.

Энди кандай килиб каватдан куринадиган майдонни куриб чикамиз. Навбвтадаги мисолда тасвирни киркиш ва жойлаштириш механизmidан фойдаланилган.

Мисолни чакириш учун ушбу тугмани босинг:

Test it

Скриптни дастлабки коди :

```

<html>
<head>
<script language = " JavaScript " >
< ! - - hide

```

```

var pos = 0 ; //дастлабки холат
var direction = false ;
function move n clip ( ) {
    if ( pos < -180 ) direction = true ;
    if ( pos 40 ) direction = false ;
    if ( direction ) pos + = 2 ;
    else pos - = 2 ;
    document.layers [ “ clippinglayer ”].layers[ “ imglayer ”].top = 100 + pos ;
}
// - -
</script>
</head>
< body onload = “ setInterval ( ‘ moveNclip ( ) ’ , 20 ) ; ”
< ilayer name = “clippinglayer” z-index = 0 clip = “ 20,100,200,160 ” top = 0 left =
0 >
< ilayer name = “ imglayer ” top = 0 left = 0 >
< img name = devici scr = “ davinci.jpg ” width = 209 height = 204 >
</ilayer>
</ilayer>
</body>
</html>

```

Ва яна куйилган каватга мурожатларни куриш мумкин:

```
document.layers [ “ clippinglayer ”].layers[ “ imglayer ”].top = 100 + pos ;
```

скриптнинг колган элементлари билан аллакачон таниш булишингиз керак.

11 Кисм javascript 1.2 да ходисалар модеми.

Янги ходисалар

JavaS 1.2 куйидаги ходисаларни кайта ишлашни куллаб куватлайди:

Abort	Dblclick	Keydown	Mousedown	Mouseup	Select
Blur	DragDrob	KeyPress	Mousemove	Move	Submit
Click	Error	Keyup	Mouseout	Reset	Unload
Change	Fox	Load	Mouseover	Resize	

Жадвални урганиш давомида куришимиз мумкинки бу янги ходисаларнинг бир нечтаси руёбга чикарилган. Биз бу кисмда уларнинг бир нечтасини куриб чикамиз. Датлаб resize ходисасини курайлик. Ушбу ходиса ёрдамида дарча улчами укувчи томонидан узгартирилганини аниклашингиз мумкин. Куйидаги скрипт буни кандай килишни курсатади:

```

<html>
<head>
<script language = “ JavaScript ” >
    window.onresize = message;
    function message ( ) {
        alert (“дарча улчами узгартирилди”);
    }
</script>

```

```
</head>
```

```
< body>
```

Илтимос бу дарча учамини узгартиринг

```
</body>
```

```
</html>
```

window.onresize = message Сатрида биз уша ходиса процедурасини беремиз. Аниқрок килиб айтганда фойдаланувчи дарча улчамини узгартирганидан дарров ушбу message () функциси чакирилади. Эхтимол сиз ходисаларни кайта ишловчи программаларни бундай тарзда бериш билан топип билмасангиз керак. Бирок бу ерда хеч кандай янгилик йук. Масалан, агар сизда button объекти бор булса, бу ходисани кайта ишлаш процедурасини куйидаги тарзда аниклаш мумкин.

```
<form name = “my form” >
```

```
<input type = “button” name = “my button”onclick = “alert (‘ дарча улчами узгартирилди’ )”>
```

```
</form>
```

Бирок сиз буни яна бир бошка курунишда беришингиз мумкин:

```
<form name = “my form” >
```

```
<input type = “button” name = “my button”>
```

```
</form>
```

```
<script language = “JavaScript”>
```

```
document.myform.mybutton.onclick = message;
```

```
function pressage ( ) {
```

```
alert+ (‘ дарча улчами узгартирилди!’);
```

```
 }
```

```
</script>
```

Уйлаш мумкинки 2 чи вариант бироз мураккаброк. Лекин нима учун биз айнан ундан 1 чи скрипта фойдаландик? Сабаби шундаки биз window объектини бирор бир аниқ тег ёрдамида аниқлай олмаймиз ва шунинг учун биз 2 чи вариантни куллашимизга тугри келади. 2 та мухим эслатма: 1 – дан window.onResize деб ёзмаслигингиз керак балки барчасини катта харфлар билан ёзиш керак. Агар сиз window.onResize = message () деб ёзсангиз браузер message () ни функция чикарувчи деб кабул килади. Лекин бизнинг холатда биз функцияни тугридан – тугри чикаришни хохлаймиз, факат ходисаларни кайта ишловчини аниқлаб куймокчимиз.

Event объекти.

Java S1.2 тилига янги event объекти кушилган. У узида бирор ходисани ёзиб борувчи хусусиятга эга. Хар сафар бирор ходиса юз берганда event объекти мос кайта ишловчи дастурга юборилади.

Навбатдаги мисолда экранга бирор бир тасвир чикарилади. Сиз унинг каергадир сичконча курсаткичи билан туртишингиз мумкин. Натижада экранда сичконча жорий вақтда турган нуктасининг координаталари курсатилган хабар чикади.

```
<layer>
```

```
<a href = “ # ” onclick = “alert(‘x:’ +event.x+’y:’+event y); return false;”>
<img src = davinci.jpg “width = 209 height = 264 border = 0”>
</a>
</layer>
```

куриб турганингиздай JavaS нинг олдинги версияларида бажарганимиздагидек <a> тэгда onclick ходисани кайта ишловчи программани жойлаштирдик. Бу ерда янги булиб event.x ва event.y лар билан яратилган хабар дарчаси хисобланади. Худди шунинг узи бизга сичконча координаталарни аниклаб берувчи бизга керак булган event объектида. Бундан ташкари мен барча буйрукларни <layer> тэгига жойлаштирдим. Шу туфайли биз жорий каватга нисбатан яъни бизнинг холатда тасвирнинг узига нисбатан координаталар хакидаги хабарни кобул киламиз. Акс холда биз браузер дарчасига нисбатан координаталар олган булар эдик. (return false инструкцияси бу ерда браузер жорий мурожатни узокрок кайта ишлаш учун фойдаланилган).

Event объекти куйидаги хусусиятларга эга:

Data – DragDrop ходисаси юз берганда тухтатилган объектларнинг URL адреслар массиви.

Layer x Каватга нисбатан курсаткичнинг горизонтал холати (нукталарда) resize ходисаси билан биргаликда бу хусусият браузер дарчаси кенглигини билдиради.

Layer y Каватга нисбатан курсаткичнинг вертикал холати (нукталарда) resize ходисаси билан биргаликда бу хусусият браузер дарчаси баландлигини билдиради.

Modifiers Модификатор калитини берувчи сатр – alt_mask, control_mask, meta_mask ёки saift_mask .

Page x Браузер дарчасига нисбатан курсаткичнинг горизонтал холати (нукталарда)

Page y Браузер дарчасига нисбатан курсаткичнинг вертикал холати (нукталарда)

Screen x Экранга нисбатан курсаткичнинг горизонтал холати. (нукталарда)

Screen y Экранга нисбатан курсаткичнинг вертикал холати. (нукталарда)

Target Дастлаб ходиса юборилган объектни намоён килувчи сатр.

Type Ходиса тилини курсатувчи сатр.

Which Сичконча ёки тугмачанинг ASCII киймати.

X layer x синоними

Y layer y синоними

§ Ходисани камраб олиш.

Java S тилининг асосий хусусиятларидан бири ходисани камраб олишдир.

Масалан, агар кимдир тугмачани туртса у холда шу тугмачага мос onclick ходисасини ишловчи дастур чакирилади. Сиз ходисаларни ишлаш ёрдамида дарча, документ ёки каватга мос курсатилган тугмача объектнинг ходисасини ишловчи чакирилишига қадар ходисаларни ишлаш ва камраб олишга эришасиз. Худди шундай сизнинг дарчангиз, документингиз ёки каватингиз

объекти узининг оддий адресини топгунга кадар ходиса хакида сигнални ишлаш мумкин. Бу нимага кераклигини куйидаги мисол ёрдамида курайик:

```
Html
Head
Script Language = "JavaScript"
Window.Capture Events (Event.click);
Window.onclick = haudle;
Function.haudle (L) {
    Alert ("window объекти бк ходисани камраб олади!");
    Return true;
}
</script>
</head>
<body>
<a href = "test.htm"> "Кликните" по этой ссылке; </a>
</body>
</html>
```

Куриб турганингдай бу <a> тегда ходисани ишловчи дастурни курсатмаймиз. Бунинг урнига

```
Window.Capture Events (Event.click);
```

Сатрни window объектини click ходисаси камраб олиши учун ёзилади. Одатда window объекти click ходисаси билан ишламайди. Бирок караб олиб у кайтадан уни window объектига йуллайди.

Шуни эътиборга олиш керакки Event.click даги click фрагменти бош харфлар билан ёзилиши керак. Агар сиз бир неча ходисаларни камрашини истасангиз у холда сизга уларни бир биридан / белгилари оркали ажратишингиз керак булади. Масалан:

```
Window.Capture Events (Event.click/Event.move);
```

Haudle () функциясида ходисани ишлаш дастури сифатида return true дан фойдаланамиз. Бу хакикатдан шуни англатадики, haudle () функцияси тамом булгандан сунг браузер мурожатнинг узини ишлаб чиқиш керак. Агар сиз унинг урнинга return false ёзсангиз шу ернинг узида барчаси тамом булади. Агар энди сиз <a> теги onclick ходисасини ишловчи дастур берсангиз, у холда ушбу ходисани руй беришида жорий дастур чикарилмаслигини тушуниб етасиз. Хайрон коларли жойи йукки, window объекти link объектига эга булгунга кадар, у ходиса хакида сигнални кулга киритади. Албатта агар сиз haudle () функцисини куйидагича аниклаймиз:

```
Function haudle (l) {
    Alert ("window объекти ушбу ходисани кулга киритади");
    Window.route Event (l);
    Return true;
}
```

R узгарувчиси Event объекти хисобланади.

Бундан ташкари сиз ихтиёрий объектга бевосита сигнал юборишингиз мумкин. Бунинг учун сиз `handle Event ()` усулидан фойдаланишимиз мумкин. Бу куйидаги курунишда булади. Бу куйидагича курунишда булади:

```
<html>
<Script Language = JavaScript>
window.Capture Events (Event.click);
window.onclick = handle;
function handle (l) {
document links (l);
}
</script >
<a href = "test.htm"> "Кликните" по этой ссылке </a> <br>
<a href = "test.htm">
onclick = "a href ('2 чи мурожат учун ходиса ишловчи!');"
2 чи мурожат </a>
</html>
```

Агар сиз хеч бир мурожатни туртмаган булсангиз ҳам `click` ходисасидаги силжаллар 2 чи мурожатни ишлаш учун юборилади.

Навбатдаги скрипт тугмача босилишидаги силжалларни сизнинг скриптингиз кандай силжишини намоён килади:

```
<html>
<Script Language = JavaScript>
window.Capture Events (Event.keypress);
window.keypress = pressed;
function pressed (l) {
alert ("key pressed! ASCII – value:" +l.which)
}
</script>
</html>
```

12 Қисм. Drag ва Drop.

§ Drag ва Drop нима.

Java S 1.2 ни ходисаларнинг янги модели ва каватлар механизми ёрдамида Web саҳифамизда `drag` ва `drop` схемасини ишлатиш имкониятига эга буламиз. Drag ва drop нима? Масалан, бир канча ОС лар (windows 95/NT ёки Mac OS) бизга файлларни ахлат кутисига олиб борган холда тозалашга рухсат беради. Бошқача килиб айтганда сиз файл тасвирига сичконча тугмасини туртасиз ва уни ахлат кутисига олиб борасиз – `drag` ва сунг уни у ерга ташлайсиз – `drop`. Биз бу ерда амалга оширмоқчи булган `drag` ва `drop` механизми Web саҳифада чегараланади. Шунинг учун сиз HTML саҳифадаги объектларни компьютерингизнинг каттик диски ёки бошқа хусусий харакатларга келтириш учун бу ерда келтирилган хаддан фойдалана олмайсиз. JavaScript +тили тугридан тугри Drag ва Drop механизмини куллаб кувватламайди. Бу бизда `image` объекти учун `drageable` ёки бирор бир хусусиятни белгилаш имкони йук деган маънони беради. Шунинг учун биз бунинг учун зарур кодни узимиз ёзишимиз керак. Лекин бу унчалик кийин эмас. Демак бизга

нима керак? Бизга икки нарсa керак. Биринчидан биз сичконча билан боглик аник ходисани аниклашимиз керак, яъни кандай тушунишимиз керак. Сунгра биз экран буйлаб объектларни кандай килиб утказишни уйлашимиз керак. Албатта биз объектларни яратиш ва уларни экран буйлаб жойлаштиришни кават сингари янги имкониятдан фойдаланган холда бажарамиз. Хар бир объект узининг кавати билан намоён булади.

JavaScript 1.2 да сичконча билан ишлашдаги ходисалар.

Биз сичконча билан ишлашда юз берадиган ходисалардан кайси бирини ишлатишимиз керак? Бизда MouseDrag каби ходиса йук лекин бу уни Mouse Dam ,Mouse Move ва Mouse up ходисаларни кайта ишлаб эришишимиз мумкин. Javas ни 1.2 версиясида ходисаларнинг янги модулидан фойдаланилади . Биз буларсиз ушбу масалани ечиб билмайбиз . Келинг бу янги моделнинг бир канча мухим кисмига яна бир куриб чикайлик. Фойдаланувчи браузер дарчасининг бирор бир жойида сичконча тугмасини босади. Бизнинг скрипт бу ходиса кайси объект (яъни кават) билан богликлигини ва бу ходисани бажариш керак.Бизга ушбу ходиса координата нукталарини билишимиз керак.Java S 1.2 да ушбу нукталарни координаталарини руй беришининг саклайдиган янги Event объекти ташкил килинган. Ходисани камраб олиниши бошка бир мухим моментни узх ичига олади. Масалан агар фойдаланувчи сичконча тугмачасини туртса, у холда унга мос ходиса тугридан-тугри Button объектига юборилади. Лекин бизнинг мисолимизда бу ходиса Window объектида ишлатилиши керак. Шунинг учун биз дарча объектига сичконча билан боглик ходиса хакидаги сигнални камраб олишга рухсат берамиз. Бу навбатдаги мисолда номоён булади. (Click ходисасида) сиз браузер дарчасининг ихтиёрий жойига сичконча билан туртишингиз мумкин. Бунда дарчада ушбу ходиса руй берган жойнинг координатаси хакидаги хабар дарчаси номоён булади.

Ушбу мисолнинг коди :

```
< html >
< script language = " Javascript " >
< ! - - hide
Window.Capture Events ( Event. Check ) ;
Wihdow . Onclick = displayCoords :
function display Coords (l)
  alert ("x : " + l.pageX + " y : " + l.pageY ) ;
}
//--
</script>
Ушбу дарчанинг кайсидир жойига туртинг!
</html>
```

Дастлаб биз window объекти click ходисаси хакидаги сигнални камраб олишини хабар киламиз. Бунинг учун биз Capture.Event () усулидан фойдаланамиз.

Window.onClick = displayCoords; сатри шуни англатадики, click ходисаси содир булганида нима содир булади. Аникрок килиб айтганда браузернинг click ходисасига реакцияси сифатида displayCoords () функцияси чакирилади (displayCoords дан сунг кавс куйиш керак эмаслигини эътиборга олинг). Уз навбатида displayCoords функцияси куйидаги куринишда аникланади:

```
Function displayCoords (l) {
```

```
Alert (“x:”+l.page.x+ “y:”+l.page.y);
```

Куриб турганингиздай бу функция аргументга эга (уни l деб атадик). Хакикитдан эса бу Event объекти булиб бу displayCoords () функциясига ишлатилишига берилади. Event объекти реге x ва page y каби хусусиятларга эга, улардан ходиса содир булган жой нуктасини координаталарини олиш мумкин. Хабар дарчаси эса факатгина шу кийматни курсатади.

MouseDown, MouseMove ва MouseUp.

Айтиб утганимиздек Java S да MouseDrag ъодисаси йук. Шунинг учун биз Drag ва Drop механизмини хосил килишда MouseDown, MouseMove ва MouseUp ходисаларидан фойдаланишимиз керак. Навбатдаги мисолимизда MouseMove ёрдамида сичкончанинг жорий координаталари холатлар сатрида намоён булади. Куришимиз мумкинки бу скрипт коди олдинги мисол билан бир хил:

```
< html >
```

```
  < script language = “ JavaScript ” >
```

```
  < ! - - hide
```

```
  Window.Capture Events ( Event. MouseMove ) ;
```

```
  Window.On MouseMove = displayCoords;
```

```
  function displayCoords (l) {
```

```
    status = “x : “ + l.pagex + “ y : “ + l.page y;
```

```
  }
```

```
  //--
```

```
</script>
```

Сичконча координаталари холатлар сатрида намоён булади

```
</html>
```

Event.MouseMove ни ёзишда сиз MouseMove сузини бош харфлар билан ёзишингиз кераклигига эътиборни каратинг. MouseMove ходисаси содир булганда кайси функция чакирилишини курсатишга келганда эса сиз уни кичкина харфлар билан ёзишингиз керак булади:

```
Window.on mousemove = ...
```

Энди биз охирги 2 та мисолни бирлаштиришимиз мумкин. Биз фойдаланувчи сичкончани тугмачасини босганда унинг координатаси пайдо булишини хохлаймиз. Мисолнинг коди куйидаги куринишда булади:

```
< html >
```

```
  < script language = “ JavaScript ” >
```

```
  < ! - - hide
```

```
  Window.CaptureEvents ( Event.MouseDown/Event.MouseUp ) ;
```

```
  Window.On MouseDown = StartDrag;
```

```
  Window.On MouseUp = EndDrag;
```

```

    Window.On MouseMove = MoveIt;
function StartDrag (l) {
    Window.CaptureEvents ( Event.MouseMove);
}
function MoveIt (l) {
    // Координаталарни курсатиш
status = “x: “ + l.pageX + “ y : “ + l.pageY;
}
function EndDrag (l) {
    window.releaseEvents (Event.MouseMove);
}
//--
</script>

```

Сичконча тугмаисини босинг ва уни юбормаган холда сичкончани узини харакатлантиринг. Сичконча координаталари холатлар сатрида номаён булади.

```
</html>
```

1 чи дан биз window объектини MouseDown ва MouseUp ходисаси хакида сичкончани кабул килишга мажбур киламиз.

```
Window.CaptureEvents ( Event.MouseDown/Event.MouseUp );
```

Куриб турганингиздек биз / (ёки) белгисидан window объекти курсатилган ходисалардан бир нечтасини кабул килиш керак деган мақсадда фойдаланамиз. Курсатилган ходисалар жойга эга булса унда нима содир булишини куйидаги 2 та сатр оркали тавсифланади.

```
Window.On MouseDown = StartDrag;
```

```
Window.On MouseUp = EndDrag;
```

Куйидаги сатрда MouseMove ходисаси руй бериб window объекти сигнал кабул килганда нима содир булиши аниқ булган.

```
Window.On MouseMove = MoveIt;
```

Лекин тухтанг ахир биз Event.MouseMove ни window.captureEvents () да аниқламадику! Бу шуни англатадики жорий ходиса window объекти томонидан камраб олинмайди. Унда нимага агар window объекти бу ходиса хакида хеч нимага эга булмаса, биз window объектига MoveIt () функциясини курсатамиз! Бу саволга жавобни MouseDown ходисаси содир булгандан сунг чакириладиган StartDrag () функциясида топишингиз мумкин:

```
function StartDrag (l) {
    Window.CaptureEvents ( Event.MouseMove);
}

```

Бу шуни билдирадики сичконча тугмачаси босилгандан дарров window объекти MouseMove ходисасини камираб олади. Ва агар MouseUp ходисаси содир булса биз MouseMove ходисасини тухтатишимиз керак. Бу EndDrag () функциясида ReleaseEvents () усули ёрдамида бажарилади.

```
function EndDrag (l) {
    window.releaseEvents (Event.MouseMove);
}

```

```
}
```

MoveIt () функцияси сичкончани координаталарини холатлар сатрига ёзади. Энди бизда Drag ва Drop механизмини амалга оширувчи скриптнинг барча элементлари бор. Ва биз энди объектларимизни экранда чизишни бошласак ҳам булади.

§ Харакатланаётган объектларни курсатиш.

Оддинги дарсларимизда биз кандай қилиб каватлар ёрдамида объектларни жойлаштиришни яратишни курган эдик. Бу энди фойдаланувчи қайси каватдаги сичконча тугмаси билан туртганини аниқлашимиз керак. Мисолни коди:

```
< html >
```

```
< head >
```

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

```
  < ! - -
```

```
var drag Obj = new array ( );
```

```
var   dx,dy;
```

```
Window.CaptureEvents ( Event.MouseDown/Event.MouseUp ) ;
```

```
Window.On MouseDown = StartDrag;
```

```
  Window.On MouseUp = EndDrag;
```

```
  Window.On MouseMove = MoveIt;
```

```
function StartDrag (l) {
```

```
  current Obj = which Obj (l);
```

```
  Window.CaptureEvents ( Event.MouseMove);
```

```
  }
```

```
  function MoveIt (l) {
```

```
    drag move Obj[current Obj].left = l.pageX - dx;
```

```
    drag move Obj[current Obj].top = l.pageY - dy;
```

```
  }
```

```
}
```

```
function EndDrag (l)  {
```

```
  currentObj null;
```

```
  window.releaseEvents (Event.MouseMove);
```

```
}
```

```
function init ( ) {
```

```
  drag Obj [0]=document.layers[layers0];
```

```
  drag Obj [1]=document.layers[layers1];
```

```
  drag Obj [2]=document.layers[layers2];
```

```
}
```

```
function which Obj (l)  {
```

```
//
```

```
var.hit=null;
```

```
for (var i=0, i<drag Obj.length: i++ ) {
```

```
if ((drag Obj[i] left<l.pageX) &&
```

```
  (drag Obj[i] left<drag Obj[i].clip.width l.pageX) &&
```

```
  (drag Obj[i] top<l.pageY) &&
```

```

        (drag Obj[i] top< drag Obj[i].clip.height l.pagex))
    }
        hit=i;
        dx=l.pagex – drag Obj[i].left;
        dy=l.pagey – drag Obj[i].top;
        break;
    }
}
return hit;
}
/-- >
</script>
</head>
body onload = “init ( )”
<layer name = “layer0” left=100 top=200 clip = “100,100” bgcolor = “0000ff”>
<font size = +1> object0 </font>
</layer>
<layer name = “layer1” left=300 top=200 clip = “100,100” bgcolor = “00ff00”>
<font size = +1> object1 </font>
</layer>
<layer name = “layer0” left=100 top=200 clip = “100,100” bgcolor = “ff0000”>
<font size = +1> object2 </font>
</layer>
</body>
</html>

```

HTML саҳифамизнинг <body> тегида 3 та каватлар аниқланганини куришимиз мумкин. Саҳифа юклатилгандан сунг OnLoad ходисани ишлаш процедураси ёрдамида <body> тегида init () функцияси чакирилади.

```

function init ( ) {
    // “Жойлаштириладиган” каватлар
    drag Obj [0]=document.layers[layers0];
    drag Obj [1]=document.layers[layers1];
    drag Obj [2]=document.layers[layers2];
}

```

Drag Obj массиви фойдаланувчи жойлаштирмокчи булган каватларни узининг ичига олади. Хар бир кават Drag Obj да бирор бир номерга эга. Уни биз кейинрок кураимиз.

Сичконча билан боғланган ходисани камраб олиш кодини (биз олдин ҳам ишлатган) яна ишлатганимизни куриш мумкин.

```

Window.CaptureEvents ( Event.MouseDown/Event.MouseUp ) ;

```

```

    Window.On MouseDown = StartDrag;

```

```

    Window.On MouseUp = EndDrag;

```

```

    Window.On MouseMove = MoveIt;

```

Биз Stand Drag функциясига куйидагини кушдик.

```

    Current Obj = which Obj (l);

```

Which Obj (l) функцияси кайси объектга туртиш бажарилганини аниқлайди. У мос кават номерини кайтаради. Агар ҳеч кайси кават белгиланмаган булса у null каватни кайтаради. Олинган киймат current Obj узгарувчисида сақланади. Бу шуни аниқлатадики, жорий вақтда биз жойлаштирмакчи булган каватнинг номерини current Obj дан олишимиз мумкин (агар ҳеч кайсини утказмакчи булмасак бу null каватда булади).

Which Obj функциясида сиз ҳар бир каватнинг left, top, width ва height хусусиятини текширишимиз мумкин. Ушбу кийматлар ёрдамида бир фойдаланувчининг кайси объектни тугмача билан туртганини текширишимиз мумкин.

Адабиётлар

1. Шафран Э. Создание Web-страниц: самоучитель-СПб:Питер,2001.-320с.
2. Игорь Шапошников HTML4 «БХВ-Петербург» 2002.
3. www.intuit.ru
4. www.citforum.ru

