

## JavaScript Hammaga

JavaScript Hammaga.

Avtordan:

Salom azizlar bugundan boshlab JavaScriptni hamma funktsiyalarini va uni o'zini iloji boricha chuqur o'rganishni boshlaymiz. Bu kitob nafaqat boshlovchilar uchun balki javascript yozuvchilari uchun ham foydali bo'lishi mumkindir. Kitob nafaqat o'quvchilar uchun balki o'qituvchilar uchun ham yozilgandir, bu kitob yozishda men dars berayotgan malumotlarimda foydalandim va hamma chiqqan savollarni shu yerda ko'rsatib yechimini berib ketdim. Niyat qilaman bu kitob sizga yordam beradi.

Kitobimni men OpenOffice.org Writer'da yozdim va Mening asosiy Operatsion Sistemam Linuxdir, keling bularni qo'llab quvatlaymiz va open source programmalarini ko'payishiga o'z hisamizni qo'shamiz. Kitob sotish maqsadida yozilmagan va uni sotish qatidan man etiladi. Kimki bosmadan chiqarish yoki o'zgartirish kiritish niyatida bo'lsa unda afor yani men bilan maslahatlashishi va ruhsat olishi shartdir.

## JavaScript Hammaga

### JavaScript Nima?

Birinchi bor JavaScript 1995 chi yil o'rtaga chiqdi. JavaScript 1995 yili ishlatilishi boshladi, ammo 1998 yilga kelib keng qullanishga kirdi. Uning chiqishini asos sabablaridan biri bu Client tomonidan bo'ladigan kiritishlarni tekshirish uchun, yani server tomonida qilinadigan ishlarni bir qismini olib tashlash uchun.

Lekin shuncha vaqt o'tgandan so'ng javascript web development'ning eng asoslaridan biri bo'lib qoldi. Microsoft kompanyasi ham o'zining scripting language yani Vbscriptni chiqardi ammo lekin uni keng qo'llab bo'lmas edi, sababi Internet Explorer'dan boshqa hech qaysi Web Browser uni ishlata olmasdi. Keyin esa Microsoft ham Internet Explorer ichiga JavaScript'ni qo'shdi. Hozirgi kunga kelib hamma Web Browserlar JavaScriptni ishlata oladi.

JavaScript bu “Client side programming” yani kimki sahifangizni ochsa shu odamning tomonida ishlaydi. PHP, Perl, CGI, JSP lar kabi serverda ishlamaydi.

JavaScriptda siz DHTML yani Dynamic HTML sahifalari yoza olasiz. Bu degani sizning qandaydir harakatingizga qarab ish qilishi, va boshqalardir.

JavaScript'ning asosi uch qismidir bular:

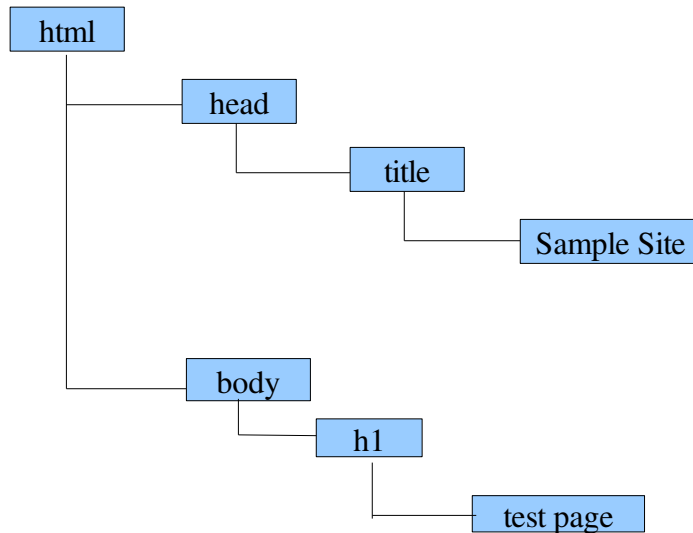
1. ECMAScript
2. The Document Object Model (DOM)
3. The Browser Object Model (BOM)

ECMAScript: Bu JavaScriptning asosi, yani programmalshidir, bu hech qanday browser'ga bog'liq bo'lmagan, hech qanday server ga bog'liq bo'lmagan qismidir. Buning ichida buyruqlar, script yozish usullari, o'zgaruvchan/o'zgarmlar, operatorlar, kalit so'zlar dir (syntax, types, keywords, operators, statements)

DOM( The Document Object Model): API (Application programming interface) for HTML and XML, yani programmalar yaratish ko'rinishidir. Agar boshqa tillarda misol uchun Java yoki Delphi, Visual Basic'da programmalar yozgan bo'lsangiz unda ular OOP Object Oriented Programming edi, ularda asos va uning ichida elementlari bor edi, bu yerda ham huddi shunday tarifdir. Misolga qarang:

```
<html>
  <head>
    <title>Sample Site</title>
  </head>
  <body>
    <h1> Test Page </h1>
  </body>
</html>
```

## JavaScript Hammaga



Mana yuqoridagi misoldan ko'rganingizdek bu yerda document'imizni obyektlarini ich ichida bo'lganligini chizib ko'rsatdik bu esa DOM'dir. <html> ningi ichida <head> uni ichida <title> va hkz.

BOM (Browser Object Model) Bu webbrowser bilan aloqali ishlardir. Yani bularga statusbar'ning tekstini o'zgartirish, toolbarni o'zgartirish va hkz, faqatgina browser bilan aloqali bo'lib web sahifaning o'ziga aloqali emasdir.

Bular:

1. Yangi browser ochish
2. Browserni yurgizish, katta kichik qilish, yopish
3. Cookie qulanishi
4. Ekran kattaligi
5. Toolbar, Navigator lar

Bular hammasi BOM o'z ichiga oladi.

## JavaScript Hammaga

### ECMAScript

Kim oldin Java, Perl kabi programmalash tillarida yozgan bo'lsa unda javascript ham unga oson va komandalarni tanishligini ko'radi.

Yozish:

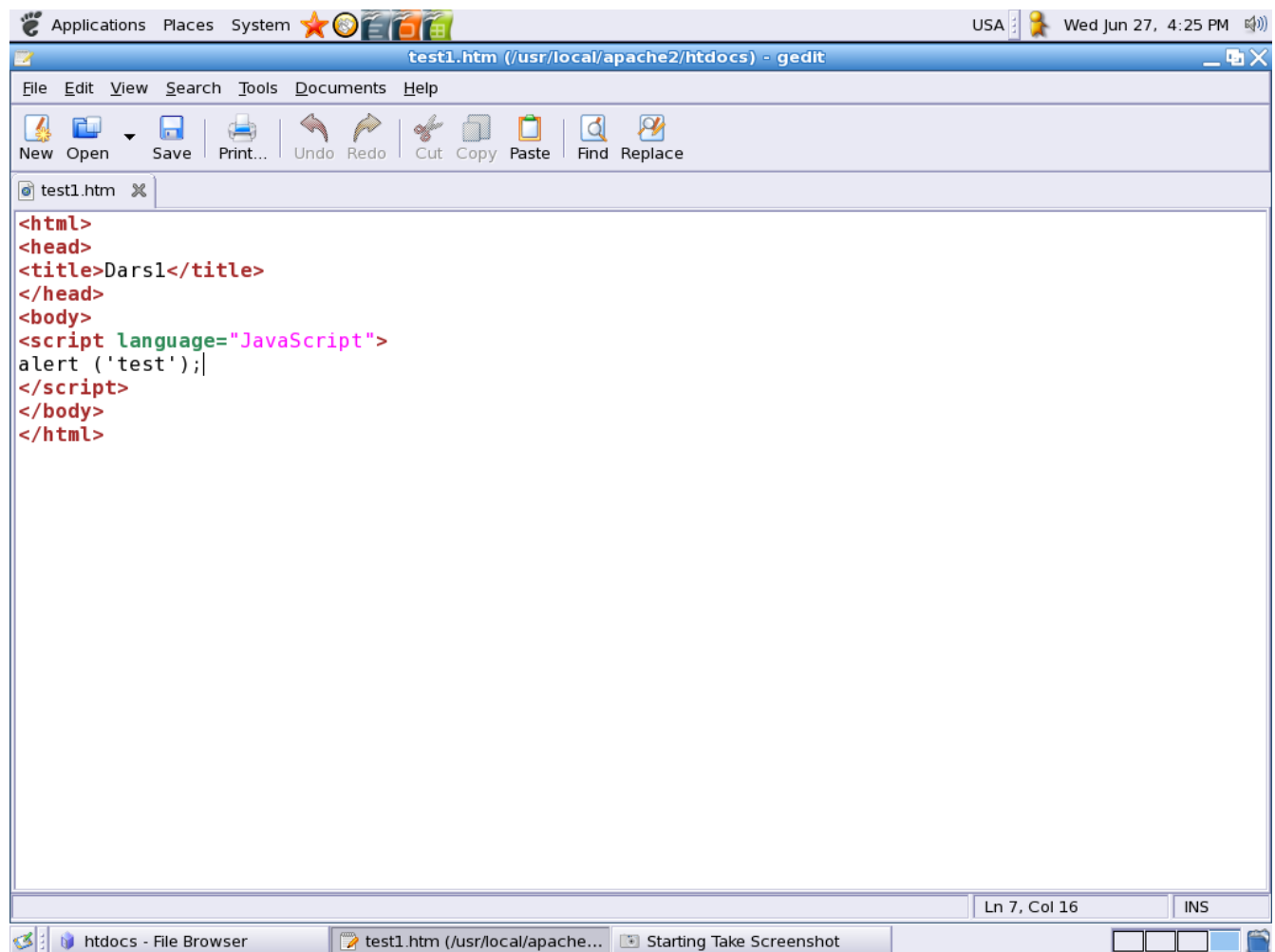
**Hamma yozuvlar case-sensitive (harflarni katta kichikligiga bog'liq):** Harflarni katta yoki kichikligi ahamiyati kattadir. Masalan “test” “Test” bilan bir hil emas, ular har hil deb qabul qilinadi.

HTML ko'dining ichida script yozish uchun siz

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

kabi qo'shimcha yozishingiz kerak bu ikta tagning ichiga siz javascript programmangizni yozasiz.

Quyida misollarimiz uchun kerakli bo'lganligi uchun hozir avvalam bor alert() funksiyasini ko'rib chiqamiz. Alert ('Test') bu oynamizga “Test” so'zini chiqaradi. Rasmga qarang:



## JavaScript Hammaga

**Variable lar(o'zgaruvchan va o'zgarmaslar):** O'zgarmas yoki o'zgaruvchanlarni tarifini berayotganingizda ularni har birini alohida yoki birga tariflashingiz kerak va ularga hohlasangiz boshidayoq qiymat berib qo'yishingiz mumkindir. Va var so'zi bilan boshlanishi shartdir. Istalgan ko'd ichida yozishingiz mumkindir. Misol:

```
var color = "white";  
var num = 123;  
var test = "salom", age = 32, site = "uzinfo.info";
```

```
var test1 = "hayot";  
//nimaladir bu yerda  
alert (test1);  
var test2 = 55;  
//qandaydir ko'd  
alert (test2);
```

Yana shuni esdan chiqarmaslik kerakki, bosh harfi; yoki harf, yoki ( ) yoki (\$) belgisi bilan boshlanishi shart.

```
var $test;  
var test;  
var _test;
```

**Nuqta vergul qoyish ixtiyoriy:** har bir qator tugashida nuqta vergul (;) qo'yish mumkin yoki qoymasa ham bo'ladi.

```
Var test1 = 1  
var test2 = 34;  
bularni ikkalasi ham to'g'ri yozilgandir.
```

**Komentariyalar:** Ayna C, Delphi, Java kabidir.

```
// bitta qatorni komentariyaga olish  
/* bir nechta  
qatorni komentariyaga olish  
*/
```

**Katta qavus:** Ko'd bloklari katta qavus orasida yoziladi "{ }" misol uchun if uchun quyida kichik blok.

```
if (color=="red") {  
color_back="blue";
```

## JavaScript Hammaga

```
alert(color_back);  
color="white";  
}
```

### O'zgarmas va O'zgaruvchanlar turlari:

**Undefined:** agarda o'zgarma/o'zgaruvchantanitilmagan bo'lsa yani:

```
var color;
```

bu yerda biz color degan o'zgaruvchan bor deb tanitdik ammo uni nima ekanligini ko'rsatmadik.

**Boolean:** ha/yoq (true/false) qiymatini ola oladigan o'zgaruvchan.

```
var tekshir = true;
```

```
var bor = false;
```

**Number:** bu son bo'lgan o'zgaruvchandir. Bir qancha turda ko'rsatish mumkin yani 8lik sanoq sistemasida yoki 10 lik sanoq sistemasida yoki 16 sanoq sistemasida. 8 lik sanoq sistemasida ko'rsatish uchun sonning oldingi qismiga 0 raqamini qo'yais agarda 16 sanoq sistemasi bo'lsa unda 0x qoyasiz oldiga.

```
var sakiz = 070; // (bu 10 lik sanoq sistemasida 56 degani)
```

```
var onolti = 0xAB; // (bu 10likda 171 degani)
```

```
var onlik = 4324; // (onlik sanoq sistemasi)
```

**String:** tekstlar uchundir, buning ichida istalgan narsa bo'lishi mumkin.

```
var tekst = "Toshkent eng chiroyli, gozalshahar. Katta katta bindar va hkzzz....";
```

string ichida quyidagi mahsus belgilarni ishlatishingiz mumkin.

```
\n    yangi qator
```

```
\t    Tab
```

```
\b    Backspace
```

```
\r    Enter
```

```
\f    Formfeed
```

```
\\    Backslash
```

```
\'    Single quote
```

```
\\"    Double quote
```

```
\Onnn sakizlik sanoq sistemasidagi son
```

```
\xnn  on oltilik sanoq sistemasi
```

## JavaScript Hammaga

misol:

```
var tekst = "Salom Halq, \n Tashkent \n  Uzbekistan";
```

```
alert (tekst);
```

shunda oynangizda quyidagicha oyna paydo bo'ladi.

Rasmdan ko'rishingiz mumkinki Tashkent bir qator pasda Uzbekistan esa ikki qator. Yani uzbekistan'gacha biz ikki marotaba \n belgisini ishlatdik.

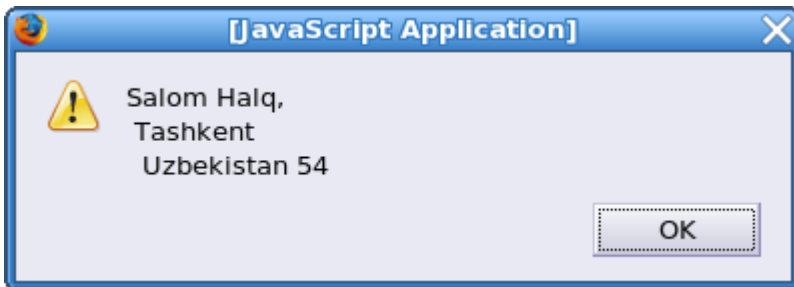


```
<script language="JavaScript">
```

```
var tekst = "Salom Halq, \n Tashkent \n  Uzbekistan \0654";
```

```
alert (tekst);
```

```
</script>
```



Mana yuqoridagi rasmda ko'rganingizdek sakizlik sanoq sistemasidagi 654 onlikda 54 deb ko'rsatib berdi.

**typeof** operatori: Bu operatorimiz bizaga o'zgaruvchani qaysi tip olganligini ko'rsatib beradi.

Misol:

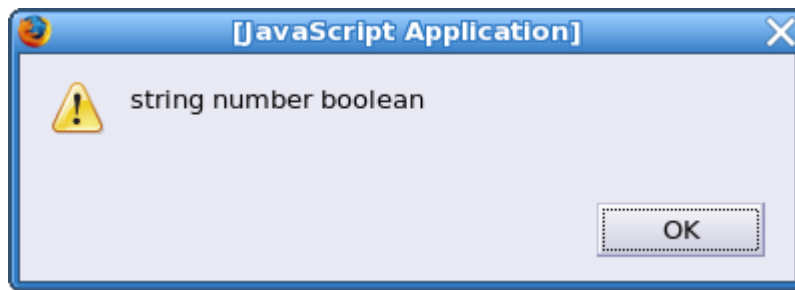
```
var Test = "Tekstimiz";
```

```
var test2 = true;
```

```
alert (typeof(Test) + ' ' + typeof(65) + ' ' + typeof(test2));
```

Shunda oynamizga quyidagi chiqadi:

## JavaScript Hammaga



Ko'rganingizdek birinchisi bizda string, ikkinchi son, uchinchi esa boolean.

### O'zgaruvchan tiplarini o'zgartirish

#### Dars 2

**length operatori:** Bu operator string bo'lgan o'zgaruvchani uzunligini beradi.

Misol:

```
var Color = "red";  
alert (Color.length);
```

Oynamizga 3 soni chiqadi.

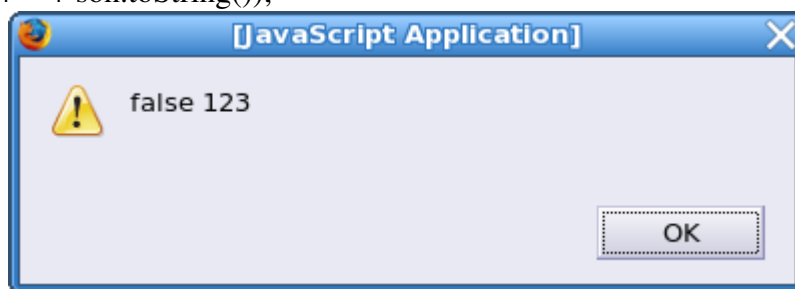
```
var tekst = "Bu yerda bizning tekst";  
alert (tekst.length);
```

oynamizda 12 soni chiqadi. Bu tekst'ning uzunligidir.

**String'ga qolgan tiplarni o'zgartirish:**

**toString() operatori:** Bu operator qolgan tiplardan yani number, boolean larni string'ga o'giradi.

```
var bor = false;  
var son = 123;  
alert (bor.toString() + ' ' + son.toString());
```



mana shu kabi oyna chiqadi.



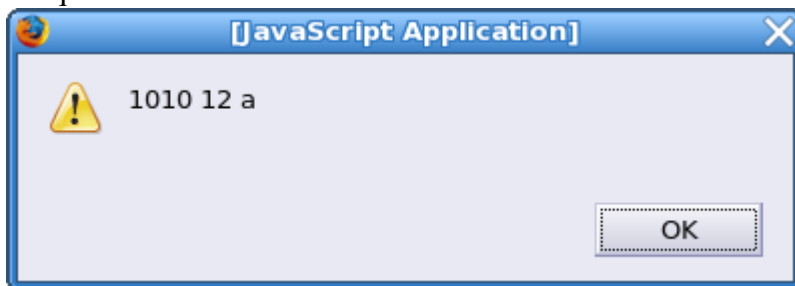
## JavaScript Hammaga

Yana bir hususiyati shundaki sonlarni tekstga o'girayotganda kirilgan sonni qaysi sanoq sistemasiga o'girishini ham ko'rsatishingiz mumkindir.

Misol:

```
var son = 10;  
alert (son.toString(2) + ' ' + son.toString(8) + ' ' + son.toString(16));
```

oynangizda quyidagi chiqadi



Numberga o'girish:

Stringdan Numberga o'tkazish `parseInt()` va `parseFloat()` operatorlari bilan amalga oshiriladi. Faqatgina string bo'lgan fayllargina Numberga yo'giriladi, qolganlari esa NaN bo'lib o'giriladi.

Misol;

```
var iNum = parseInt("1234");  
var fNum = parseFloat("12,43");
```

Bu yerda biz `parseInt` orqali Integer'ga o'girdik.

Integer: Bu butun sonlar

`parseFloat` orqali esa bi Float'ga o'girdik

Float: Qoldiq sonlar, yani nuqtali sonlar.

Agarda sonimizning ichida harf bo'lsa unda harfgachabo'lgan sonlarnigina Numberga o'giradi.

Misol:

```
var iNum = parseInt("254bgde");
```

Shunda `iNum=254` bo'ladi.

Qo'shimcha qilib siz ko'rsatilgan sonni qaysi sanoq sistemasida ekanligini ham ko'rsatishingiz mumkindir. Buning uchun:

```
var iNum = parseInt("10", 2) //ikkilik sanoq sistemasida "10" soni o'nlikda 2 degani iNum=2 bo'ladi  
var iNum = parseInt ("10", 8) //7 sanoq sistemasida "10" soni 10likda 8 degani  
var iNum = parseInt("10", 16) //16 sanoq sistemasida "10" soni 10likda 16 degani
```

Agarda siz faqatgina harflardantashkil topgan teksni `parseInt` yoki `parseFloat` bilan o'girsangiz unda

## JavaScript Hammaga

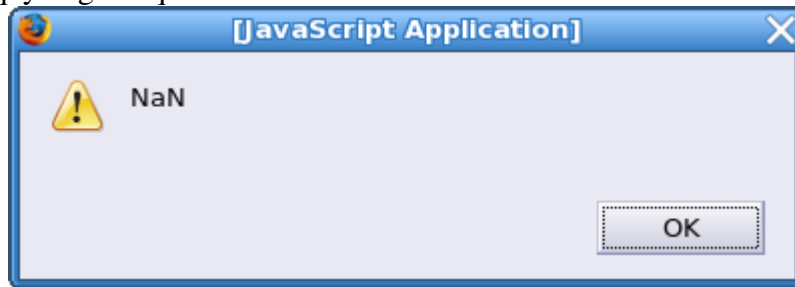
sizga NaN qaytaradi son bo'lmaganligi uchun va birinchi belgisi ham agarda harf bo'lsa undayam NaN qaytaradi.

Misol:

```
var iNum = parseInt("blue");
```

```
alert(iNum);
```

shunda oynangizga quyidagi chiqadi:



agarda

```
var iNum = parseInt("b123");
```

```
alert(iNum);
```

deb bersak unda ham NaN ko'rsatadi

Yana shuni etib o'tish kerakki siz yaratayotganingizda o'zgaruvchanni unda boshidayoq turini o'zgartirib yaratishingiz mumkin.

Buning uchun yaratilayotgan o'zgaruvchiturini Boolean, String, Number deb qavs ichida qiymatini berishingiz mumkindir.

Misol:

```
var bTest = Boolean("1") // true bo'ladi
```

```
var sTest = String("1") // "1" teksti
```

```
var nTest = Number("120"); //120 soni
```

Boolean uchun agarda ichida hech narsa bo'masa yoki 0 soni yoki null bo'lsa unda uni qiymati false bo'ladi, qolgan holatlarda true bo'ladi.

Number uchun agarda false yoki true qo'yilsa unda 0, 1 sonini oladi. Qolgan vaqtlarda yozilgan sonni oladi.

## Object class

Bu hamma obyektlarning eng boshi, hamma variables'lar shunda hususiyatlarini oladi va shundan yaratiladi. Shuning uchun bu class'ni biz hozir yuzaki bo'lsa ham qarab chiqamiz.

## JavaScript Hammaga

### Properties/Hususiylari:

**constructor** – Qaysi obyekt yaratganligini ko'rsatadi.

**prototype** – shu obyektning prototipi kim ekanligini ko'rsatadi, asosan bu Object dir

### Methods:

**hasOwnProperty(property):** Qandaydir o'zining qo'shimcha hususiyati bor yoki yoqligini aniqlaydi.

**isPrototypeOf(object):** Qandaydir objektning prototype'ligini bilish uchun ishlatiladi.

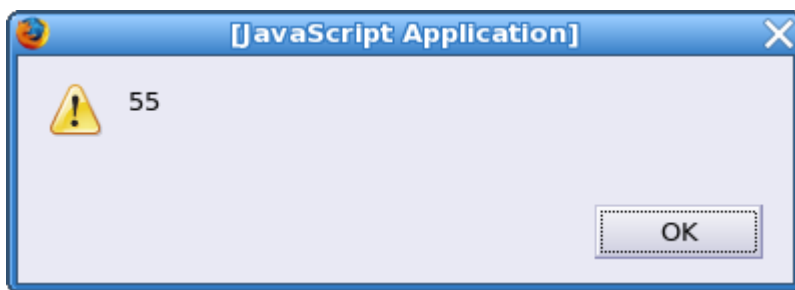
**propertyIsEnumerable(property):** bu hususiyatning for ... in kabi funktsiyalar bilan ishlatish mumkin yoki mumkin emasligini ko'rsatadi.

**toString():** Buni biz yuqorida ko'rib chiqqan edik.

**valueOf():** bu asosan aynan toString kabi ishlaydi. Yani qiymati nima ekanligini qaytaradi.

Misol:

```
var iNum = new Number(55);  
alert (iNum.valueOf());
```



### Number class:

Bu object classidan yaratilib qo'shimcha hususiyat va methodlari bilan keladi. Bu yerda ba'zi bir qo'shimcha hususiat va method'larini ko'rib chiqamiz.

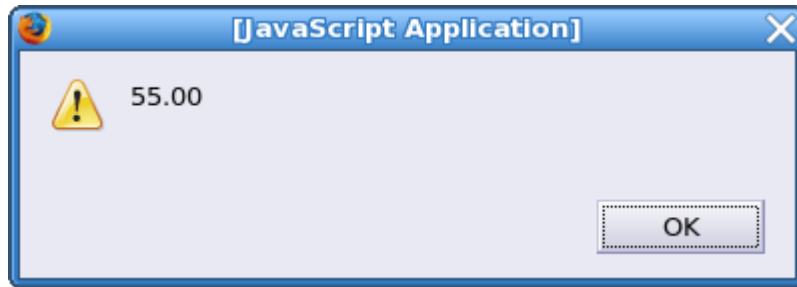
**ToFixed():** bu method sonni string'da qaytaradi va uning ichida ko'rsatilgan soncha qoldiq qismini ham qo'shib beradi.

Misol:

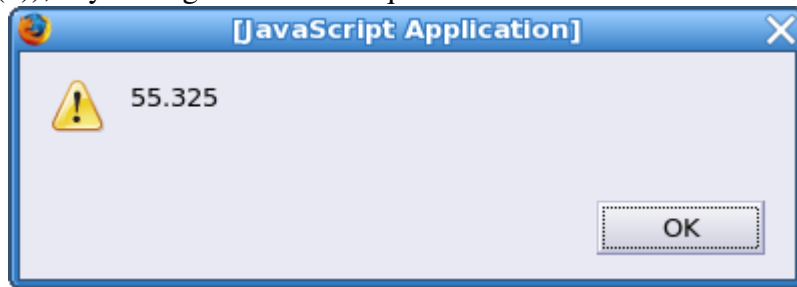
```
var iNum = 55;  
alert (iNum.toFixed(2)); // "55.00" qaytaradi.
```

Mana oynachaga qarang

## JavaScript Hammaga



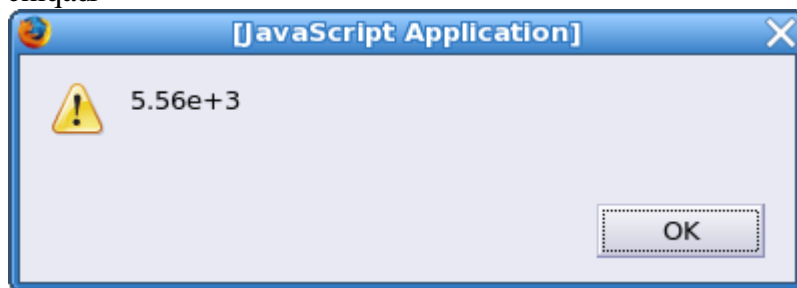
```
var iNum=55.3251;  
alert (iNum.toFixed(3)); //oynamizga "55.325" chiqadi
```



**toExponential():** Bu bizga sonimizni e formatda ko'rsatish uchun kerak.

Misol:

```
var iNum = new Number (5562);  
alert (iNum.toExponential(2));  
oynamizga quyidagi chiqadi
```



Yani ko'rganingizdek bu yerdabiz nuqtadan keyin 2 son ko'rsatish deb berdik va qolgan qismini olib tashlab e+3 qo'shimchasini qo'shib berdi.

Bu degani  $5.56 \cdot 1000$  ga ko'paytirilgani deganidir.

**toPrecision():** ham toFixed va toExponential ning ishini qiladi, qaysi birini ishlatish bu siz bergan ichki songa bog'liqdir.

Misol:

```
var iNum = 99;  
alert (iNum.toPrecision(1)); //yani bitta son bilan 99 ni tariflash kerak.
```

## JavaScript Hammaga

Oynamizga  $1e+2$  bilan ko'rsatadi yani 100 deganidir buning uchun u 99 ni 100 ga aylantirdi va toExponentialni ishlatdi.

Agarda biz

```
var iNum= 99;
```

```
alert (iNum.toPrecision(3)) //deganimizda oynamizga “99.0” chiqargan bo'lardi.
```

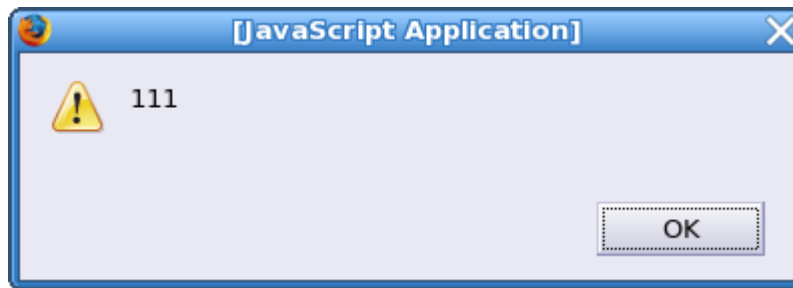
Boshqa tillardagi kabi javascriptda ham siz+ - / \* ishlarini qila olasiz.

```
var iNum = 55;
```

```
var iNum2 = 56;
```

```
var iRes = iNum+iNum2;
```

```
alert (iRes);
```



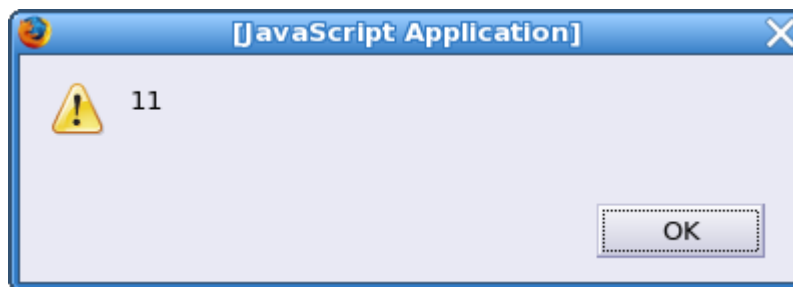
yoki

```
var iNum = 55;
```

```
var iNum2 = 5;
```

```
var iRes = iNum / iNum2;
```

```
alert (iRes);
```



### Math Object:

Bu number'lar bilan ishlashda bizga qo'shimcha object bo'lib uning ichida bir qancha funktsiyalar va o'zgarmaslar joylashgandir.

Bular:

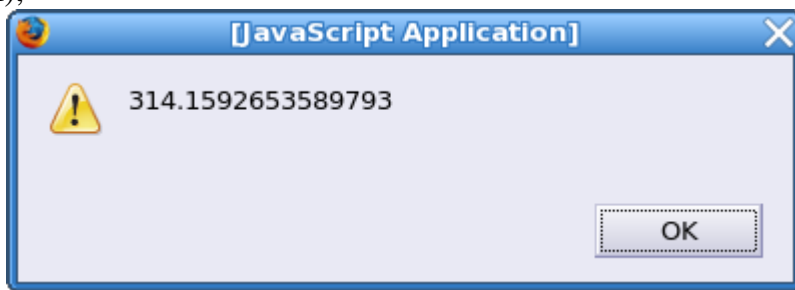
Math.E - Euler's constant (2.718281828459045)

## JavaScript Hammaga

Math.LN2 – Logorifm 2 (0.6931471805599453)  
Math.LN10- Logorifm 10 (2.302585092994046)  
Math.LOG2E – Logorofm 2 Euler'sconstant (1.4426950408889634)  
Math.LOG10E – Logorifm 10 of Euler's constant (0.4342944819032518)  
Math.PI - (3.141592653589793)  
Math.SQRT1\_2 – Ildiz ostida 1/2 (0.7071067811865476)  
Math.SQRT2 – Ildiz ostida 2 (1.4142135623730951)

misol:

```
var iNum = 100;  
alert (iNum*Math.PI);
```



Math object method'lari:

**Math.abs():** absolute value, yani sonning + - belgisiga qaramasdan qanday son ekanligini ko'rsatadi.

3 soni -3 va 3 uchun dir. Misol:

```
var iNum1 = -3;  
var iNum2 = 3;  
alert (Math.abs(iNum)); //oynamizga 3 sonini chiqaradi  
alert (Math.abs(iNum2)); //oynamizga 3 sonini chiqaradi.
```

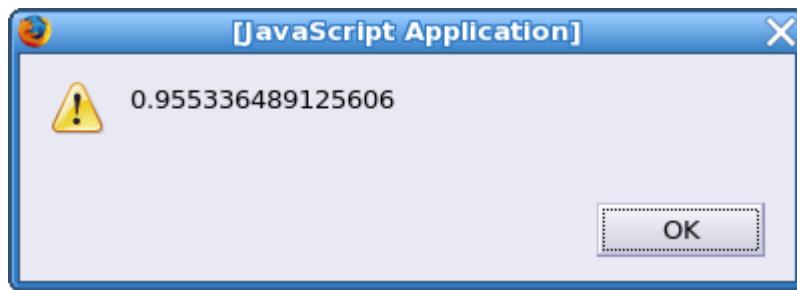
**Math.acos(), Math.asin(), Math.atan(), Math.atan2(x, y):** Bular arc cosinuz arc sinuslar va hkzlardir yani kiritilgan radian'dagi soning radian da arc cosinuz arc sinusdagi qiymatlari.  
atan2(x, y) x/y ning arc tangenisi.

**Math.cos(), Math.sin(), Math.tan():** cosinus, sinus, tangens larni radianda qaytaradi. Yani Math.cos(n) bu yerda n -1 va 1 sonlari orasidagi qiymat shunin cosinusda qiymatini radianda qaytaradi.

Misol:

```
var iNum = 0.3;  
alert (Math.cos(iNum));
```

## JavaScript Hammaga



Bu son .3 ning cosinus qiymatidir.

**Math.ceil(number):** number sonidan keyingi katta son yoki unga teng bo'lgan sonni qaytaradi.

Misol:

```
var iNum = 12.6;  
alert (Math.ceil(iNum)); //oynamizga 13 sonini chiqaradi
```

**Math.floor(number):** number sonidan kichik bo'lgan yoki teng bo'lgan keyingi integer soni.

```
var iNum = 12.6;  
alert (Math.floor(iNum)); //oynamizga 12 sonini chiqaradi
```

**Math.exp(number):** Bu yerda biz Euler's constant'ni number darajasini olamiz.

**Math.log(number):** numberni logarifmini olamiz.

**Math.sqrt(number):** Numberni ildizini olamiz

**Math.random():** Ihtiyoriy sonni bizga beradi. Va bu sonlar 0-1 orasidagi sonidir.

Misol:

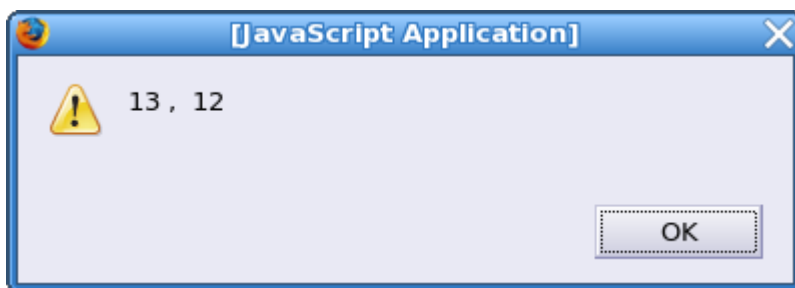
```
var iNum = Math.random();  
alert (iNum); //har doim har hil son beradi 0.2356415... 0.62541251... va hkz.
```

**Math.round(number):** Bu method esa number sonini yahlitlaydi, agar nuqtadan keyingi son .5 ga teng yoki katta bo'lsa sonimizga bir qo'shib beradi bo'lmasam soni o'zini qoldiqsiz qismini beradi.

Misol:

```
var iNum1 = 12.6;  
var iNum2 = 12.4;  
alert (Math.round(iNum1) + ' , ' + Math.round(iNum2));
```

## JavaScript Hammaga



Mana shu oynamizga chiqadi ko'rganingizdek birida 13 soni yani yuqoriga qarab yahlitladi birida esa pastga qarab yahlitladi.

**Math.pow(num1, num2):** num1 ni num2 darajasini beradi.

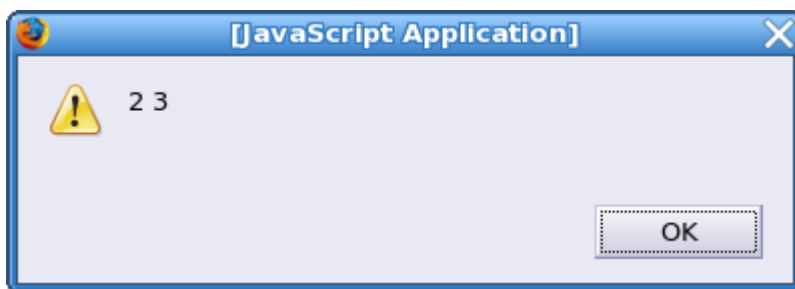
**Math.min(num1, num2), Math.max(num1, num2):** num1 va num2 ning kichigi va kattasini bizga beradi.

Misol:

```
var iNum1 = 2;
```

```
var iNum2 = 3;
```

```
alert (Math.min(iNum1, iNum2)+ ' ' + Math.max(iNum1, iNum2));
```



birinchisida eng kichigini ikkinchisida esa eng kattasini ko'rsatib berdi.

### If statement:

Shu yerda tushintirib o'tish lozim deb bildim sababi quyida biz bu If then 'ni ishlatamiz. If bu qandaydir qismni tog'ri yoki tog'ri emasligini, yoki bir biriga teng yoki teng emasligini tekshiruvchi va shunga qarab qandaydir ko'dni ishlab ishlamasligini hal qiluvchi operatoridir. Keyingi darslarimizda yanaham chuqurroq ko'rib chiqamiz.

Yozilishi:

```
if (condition) {
```

```
//qandaydir ko'd bu ko'd agarda condition bajarilsagina ishlaydi.
```

```
}
```

Misol:

```
var iNum=5;
```

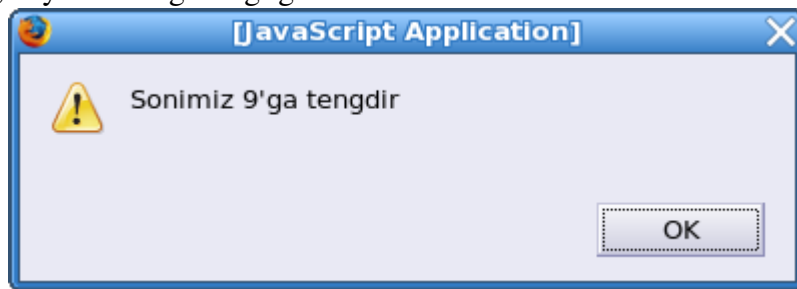
```
var iNum2 = 4;
```



## JavaScript Hammaga

```
var iResult = iNum + iNum2;  
if (iResult == 10) {  
  alert ('Sonimiz 10\'ga tengdir');  
}  
if (iResult == 9) {  
  alert ('Sonimiz 9\'ga tengdir');  
}
```

Shunda oynamizga quyidagi “Sonimiz 9'ga tengdir” degan yozuvi chiqadi. Bu yerda biz iResultni 10 ga tengligini tekshirdik, keyin esa 10ga tengligini tekshirdik.



Mana shu bilan ikkinchi darsimiz ham tugadi, keling savollarni ko'rib chiqaylik.

1. Qabdaydir son bor, agar qoldiqli bo'lsa “float” yozuvi agarda butun bo'lsa “integer” yozuvi chiqsin.
2. Faqatgina soning qoldiq qismini ko'rsatuvchi script yozing.
3. 1-100 orasida ihtiyor son chiqaruvchi script yozing
4. 5 ta son berilgan ulardan eng kichigini va eng kattasini ko'rsatuvchi script yozing.

### Yechimlar:

1-misol yechimi

```
<script language="JavaScript">  
var n1=421.1;  
if (n1==Math.ceil(n1)) {  
  alert("Integer");  
}  
if (n1<Math.ceil(n1)) {  
  alert("Float");  
}  
</script>
```

## JavaScript Hammaga

2-misol yechimi

```
<script language="JavaScript">
var n1=421.1;
alert (n1-Math.floor(n1));
</script>
```

3-misol yechimi

```
<script language="JavaScript">
var n1=Math.random();
alert(Math.round(n1*100));
</script>
```

4-misolning yechimi

```
<script language="JavaScript">
var n1=421, n2=151, n3=100, n4=176, n5=246
var nMin, nMax
nMin=Math.min(n1, n2);
nMax=Math.max(n1, n2);
nMin=Math.min(nMin, n3);
nMax=Math.max(nMax, n3);
nMin=Math.min(nMin, n4);
nMax=Math.max(nMax, n4);
nMin=Math.min(nMin, n5);
nMax=Math.max(nMax, n5);
alert(nMin + ' ' + nMax);

</script>
```

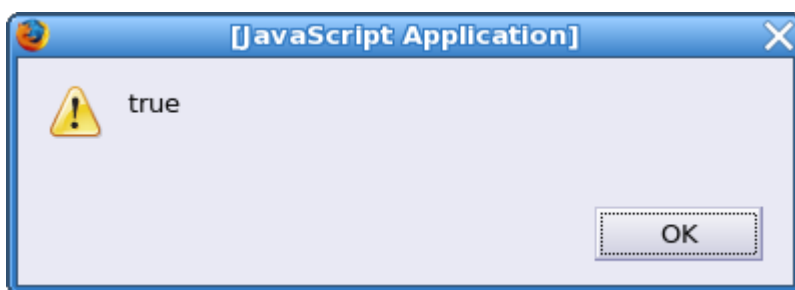
### Dars 3

#### String class:

**valueOf():** Bu operatorimiz toString()kabidir.

```
var sText = new String("Hamma hursand");
alert (sText.valueOf()==sText.toString());
```

## JavaScript Hammaga



Shu yerda berib o'tish kerak deb bildim. ASCII ko'dlar, tarjima qilinmagan va [http://www.klgtu.ru/ru/students/literature/inf\\_asu/80.html](http://www.klgtu.ru/ru/students/literature/inf_asu/80.html) sahifasidan olingandir.

ASCII-коды

Наиболее распространенным является код ASCII (American Standard Code for Information Interchange), который используется для внутреннего представления символьной информации в операционной системе MS DOS, в Блокноте операционной системы Windows'xx, а также для кодирования текстовых файлов в Интернет. Структура кода представлена в таблице (обозначения столбцов и строк выделены полужирно).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	...	...		0	@	P	'	p	A	P	a	...	...	...	p	Ё
1	...	...	!	1	A	Q	a	q	Б	С	б	...	...	...	с	ё
2	...	...	"	2	B	R	b	r	В	Т	в	...	...	...	т	€
3	...	...	#	3	C	S	c	s	Г	У	г	...	...	...	у	е
4	...	...	\$	4	D	T	d	t	Д	Ф	д	...	...	...	ф	ї
5	...	...	%	5	E	U	e	u	Е	X	е	...	...	...	х	і

## JavaScript Hammaga

6	...	...	&	6	F	V	f	v	Ж	Ц	ж	...	...	...	ц	Ў
7	...	...	'	7	G	W	g	w	З	Ч	з	...	...	...	ч	ў
8	...	...	(	8	H	X	h	x	И	Ш	и	...	...	...	ш	°
9	...	...	)	9	I	Y	i	y	Й	Щ	й	...	...	...	щ	·
A	...	...	*	:	J	Z	j	z	К	Ъ	к	...	...	...	ъ	·
B	...	...	+	;	K	[	k	{	Л	Ы	л	...	...	...	ы	√
C	...	...	,	<	L	\	l		М	Ь	м	...	...	...	ь	№
D	...	...	-	=	M	]	m	}	Н	Э	н	...	...	...	э	¤
E	...	...	.	>	N	^	n	~	О	Ю	о	...	...	...	ю	■
F	...	...	/	?	O	_	o	¸	П	Я	п	...	...	...	я	

Таблица кодов содержит 16 столбцов и 16 строк; каждая строка и столбец пронумерованы в шестнадцатеричной системе счисления цифрами от 0 до F. Шестнадцатеричное представление ASCII-кода складывается из номера столбца и номера строки, в которых располагается символ. Так, например, ASCII-код символа 1 есть число 3116, что по правилам перевода означает 1100012. В двоичной системе код представляется восемью разрядами, т.е. двоичный ASCII-код символа 1 есть 001100012.

Данная таблица делится на две части: столбцы с номерами от 0 до 7 составляют стандарт кода – неизменяемую часть; столбцы с номерами от 8 до F являются расширением кода и используются, в частности, для кодирования символов национальных алфавитов. В столбцах с номерами 0 и 1 находятся управляющие символы, которые используются, в частности, для управления принтером. Столбцы с номерами от 2 до 7 содержат знаки препинания, арифметических действий, некоторые служебные символы, а также заглавные и строчные буквы латинского алфавита. Расширение кода включает символы псевдографики, буквы национальных алфавитов и

## JavaScript Hammaga

другие символы.

В приведенной таблице в качестве национального выбран русский алфавит. Пустые ячейки означают, что они не используются, а ячейки с многоточием содержат символы, которые умышленно не показаны.

Пример 1. С помощью таблицы ASCII-кодов закодировать сообщение "группа", используя шестнадцатеричное представление кода.

Результат: A3 E0 E3 AF AF A0 (для простоты коды символов разделены пробелами)

### Qanday qilib 16 sanoq sistemasini 10lik sanoq sistemasiga o'giramiz?

Bu juda oson.masalan “e” belgisi 16 s/s da 65 degani ho'sh 10likdachi?

Har son 16ning darajasiga ko'paytiriladi uning o'rniga qarab darajasi nechki ekanligi beriladi.

$$65_{16}=101_{10} \quad 6*16^1+5*16^0=96+5=101$$

yoki

$$\text{“E” F0 buning onlikdagi qiymati } F=15 \quad 15*16^1+0*16^0=240$$

$$\text{va shu kabidir agarda bizda F17 bo'lganda unda biz } 15*16^2+1*16^1+7*16^0=3963 \text{ bo'lgan bo'lardi}$$

Keling endi String classining qo'shimcha method'larini ko'rib chiqamiz.

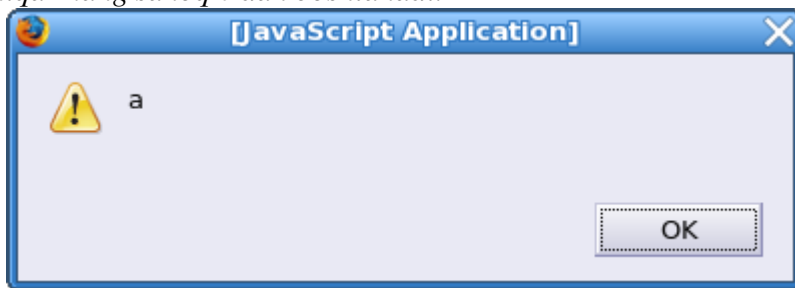
**charAt(n):** bu string'imizning ichidagi n' joyida qaysi belgi turganligini ko'rsatib beradi.

```
var sText = new String("Hammsi yahshi");
```

```
var sT = sText.charAt(1);
```

```
alert (sT);
```

Shu yerda esizdan chiqarmang sanoq O'dan boshlanadi.



Mana bizning oynamizga ikkinchi o'rinda turgan harfni ko'rsatdi.

**charCodeAt(n):** Bu bizning n o'rinda turgan harfning ASCII ko'dini beradi.

Misol:

## JavaScript Hammaga

```
var sText = new String("Hammasi yahshi");
var sT = sText.charCodeAt(1);
alert (sT); //oynamizga 97 sonini chiqaradi.
```

Misol yozing agarda birinchi harf kichkina harfda bo'lsa unda uni katta harfda ko'rsatsin, va agarda kattada bo'lsa hech narsa qilmasin, agar boshqa qanaydir belgi bo'lsa unda shu belgini ko'rsatsin.

61-7A 97-127 kichkina harflar

41-5A 65-95 katta harflar

Mana yechimi:

```
var sText = new String(".toshkent");
var nT
nT=sText.charCodeAt(0);
if (nT>96) {
if ( nT<128){
alert(sText.charAt(0));
}}
if (nT<65) {
alert(sText.charAt(0));
}
if (nT>127){
alert(sText.charAt(0));
}
```

**concat(text):** Bu ikta stringni birlashtirish uchun ishlatiladi.

Misol:

```
var sText = "Toshkent";
var sT=sText.concat (" chiroyli");
alert (sT); //oynamizga "Toshent chiroyli" degan yozuv chiqadi.
```

concat() operatorini ishlatish o'rniga siz "+" belgisini ham ishlatishingiz mumkin.

```
var sText = "Toshkent";
var sT=sText + " chiroyli";
alert (sT); //oynamizga "Toshent chiroyli" degan yozuv chiqadi.
```

**indexOf(harf), lastIndexOf(harf):** bular qandaydir harfni yoki harflar to'plamini berilgan string'ning ichidan qidiradi va agarda topsa unda uning joylashgan o'rnini ko'rsatadi, agar topa olmasa unda -1

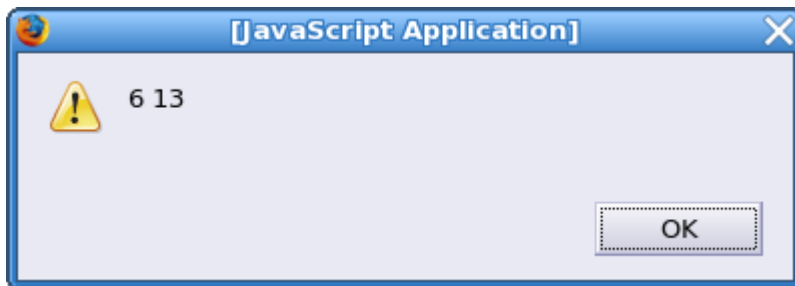
## JavaScript Hammaga

qaytaradi. `indexOf()` va `lastIndexOf()` ning bir biridan farqishuki `indexOf` qidirishni boshidan boshlaydi `lastIndexOf` esa ohiridan boshlaydi.

Misol:

```
var sText = "Toshkentda yangi uylar qurildi";  
alert(sText.indexOf("n") + ' ' + sText.lastIndexOf("n"));
```

oynamizga



6 va 13 sonlari chiqdi, bu degani boshidan 6nchi o'rinda n harfi bor va ohiridan qaraganda boshidan 13 o'rinda ham "n" harfi bor.

**localeCompare(string):** Bu operatorimiz, ikta stringni bir biri bilan solishtirish uchun ishlatiladi. Yani agarda ikta string teng bolsa 0 qiymatini beradi, agarda birinchi string ikkinchisidan keyin kelsa 1 agarda oldin kelsa unda -1 qiymatini qaytaradi.

Misol:

```
var sText = "Homhayol";  
alert (sText.localeCompare("Somsa")); // -1 qaytaradi  
alert (sText.localeCompare("Homhayol")); // 0 qaytaradi  
alert (sText.localeCompare("Asqar")); // 1 qaytaradi
```

Lekin shu narsani esdanchiqarmangki, bazi bir web browserlar boshqacha sonlar ko'rsatishi mumkin yani, -1 o'rniga -5 yoki shu kabi ammo lekin – belgisi bilan shuning uchun har doim -1 ga teng emas balki 0'dan kichikmi deb tekshirish maqsadga muvofiq bo'ladi.

O'zingiz yozing

```
var oStringObject1 = new String("yellow");  
var oStringObject2 = new String("brick");  
var iResult = sTestString.localeCompare("brick");  
if(iResult < 0) {  
    alert(oStringObject1 + " comes before " + oStringObject2);  
} else if (iResult > 0) {
```

## JavaScript Hammaga

```
    alert(oStringObject1 + “ comes after“ + oStringObject2);  
  } else {  
    alert(“The two strings are equal”);  
  }
```

tekshiruvni.

### **substring() slice() operatorlari:**

Ikkalasi ham deyarli bir hil ish qiladi. Ularni ishi shundan iboratki berilgan joydan yoki berilgan joydan berilgan joygacha bo'lgan string'ning qismini kesib olish uchundir.

Misol:

```
var sText = “Ham yonda pul”;  
alert (sText.slice(3)); //oynaga “yonda pul” chiqadi  
alert (sText.substring(3)); //oynaga “yonda pul” chiqadi  
alert(sText.slice(3,9)); //oynaga “yonda” chiqadi  
alert(sText.substring(3, 9)); //oynaga “yonda” chiqadi
```

Ho'sh bu yerda birinchi raqam qayerdan boshlashligi va ikkinchi son esa qayergacha kesish kerakligini ko'rsatmoqdamiz.

Nima farqi bor bu ikki operatorni?

Ularni farqi shundaki slice() “-” li sonni orqadan sanash kerak deb tushunadi, substring() esa uni 0 raqami deb qabul qiladi.

Misol:

```
var sText = “Ham yonda pul”;  
alert (sText.substring(3, -3)); //oynaga “Ham” chiqadi  
alert(sText.slice(3,-3)); //oynaga “yonda ” chiqadi
```

substring() har doim kichik raqamni boshlanish nuqtasi deb biladi katta son esa tugash shuning uchun sText.substring(3, -3) bizga “Ham” tekstini berdi.

### **ToLowerCase(), toUpperCase(), toLocaleLowerCase(), toLocaleUpperCase() operatorlari:**

Bu operatorlarimiz bizga berilgan string'ni tamoman kichik yoki katta harfga o'girish uchun ishlatiladi. toLocalLowerCase() va toLowerCase() larni farqi shundaki ba'zan locale'ga ham qarash uni inobatga olish kerak, bu misol uchun Turk tili uchun kerak bo'ladi.

Misol:

```
var sText=“Homhayol”;  
alert(sText.toLowerCase());//oynaga “homhayol” chiqadi
```



## JavaScript Hammaga

```
alert(sText.toUpperCase()); //oynaga "HOMHAYOL" chiqadi
```

Shundoq misol yozing, Unda tekst berilgan, ichidagi 1chi, o'rtadagi, va ohirgi harflarni katta harfda chiqarsin.

Mana uning ko'di:

```
var sText = "hamyonda pul";
var uzun = sText.length;
var sB;
sB=sText.slice(0, 1).toUpperCase()+sText.slice(1,uzun/2-1)+sText.slice(uzun/2-1,
uzun/2).toUpperCase()+sText.slice(uzun/2,uzun-1) +sText.slice(uzun-1, uzun).toUpperCase();
alert(sB);
```

O'zingiz yozing:

Ikta string tekstimiz bor ularni bir biri bilan tekshiring tengligini, shuni hisobga olinki Test=test deb hisoblaysiz.

```
var sText1 = "hamyonda pul";
var sText2 = "Hamyonda pul";
if (sText1.toLowerCase()==sText2.toLowerCase()) {
alert("Ular bir biriga tengdir");
}
```

Mana shu yerda biz String Number obyektlari haqida qisqacha tushunchani tugatamiz.  
Chuqurlashtirilgan keyingi darslarimizda o'tiladi.

## Dars 4

### Unary Operators (Unarniy operatorlar)

Bu operatorlar faqatgina bitta qiymat ola oladi va bu object yoki uni hususiyatlaridir.

#### **delete**

bu operatorimiz faqatginaobject'ning method'i yoki properties'ini qabul qila oladi. Va ularni o'chirib tashlaydi. Agarda bu property yoki method yozuvchi(programist) tarafidan berilgan bo'lmasa unda uni o'chira olmaydi.

Misol:

```
var o = new Object;
o.name="Nike";
alert(o.name); //oynaga "Nike" yozuvi chiqadi
delete o.name;
alert(o.name); //o'chirganimiz sababli "unedfined" chiqadi
```

## JavaScript Hammaga

agarda biz  
delete o.toString()  
berganimizda hato chiqaradiva ko'dimiz ishlamasdi.

### void

bu operatorimiz qandaydurfunktsiyani yoki method'ni chaqirganimizda orqaga qiymat qaytarmasligi uchun ishlatiladi. Yani deylik qandaydir funktsiya bajarilishi kerak lekin u qiymat qaytarmasligi kerak, ammo hamma funktsiyalar qiymat qaytarmasa ham “undefined” qaytaradi. Shuni olib tashlash uchun void ishlatiladi. Quyida misol keltiramiz, bu yerda biz hali ko'rmagan ko'mandalar bor ularga ahamiyat bermang, keyinchalik ko'rib chiqamiz. Hozir esa faqatgina void ishlashini ko'ring.

```
<html>
<head>
<title>Dars4</title>
</head>
<body>
<a href="javascript:window.open('about:blank')">Click Me void'siz</a>
<a href="javascript:void(window.open('about:blank'))">Click Me void'li</a>
</body>
</html>
```

bu yerda biz yangi oyna ochishimiz kerak edi, birinchi “Click Me void'siz” ni ishlatganimizda bizga yangi oyna ochdi, va bor oynani esa funktsiya qaytargan qiymat bilan to'ldirdi. Ikkinchisida esa “Click Me void'li” ni bosganda yangi oyna ochildi ammo eski oynamiz ham saqlanib qoldi.

### increment/decrement (1 ga oshirish yoki kamaytirish)

Bu operatorlarimiz qandaydir sonni 1 soniga oshirish uchun yoki kamaytirish uchun kerak bo'ladi. Buning uchun siz ++ yoki - - (ikkita minus belgisi) ishlatishingiz mumkindir.

Misol:

```
var iNum =5;
alert(++iNum);
iNum=5;
alert (--iNum);
```

Bu yerda biz iNum'ning qiymatini 6 ga oshirdik, keyin esa yana 5 qiymatini berdik va keyin 4 qiymatiga tushirdik.

## JavaScript Hammaga

Shunga o'hshash ikkinchi operatorimiz bor.

Agar siz qandaydir sondan o'zgaruvchandankeyin shu belgilarni -- ++ ishlatsangiz unda uning ishi shundaki, birdaniga qiymatni o'zgartimay ikkinchi qatordan o'zgartiradi.

Misol:

```
var iNum=5;
```

```
var iNum2=iNum--;
```

```
alert (iNum + ' ' + iNum2); //oynamizga 4 va 5 soni chiqadi, sababi shundaki iNum2 ga 5 sonini beradi undan so'nggina iNum ni qiymatini birga kamaytiradi.
```

### NOT operatori

Keling avvalam borikkilik sanoq sistemasini azgina qarab chiqamiz. Chunki not operatorini tushunish uchun ikkilik sanoq sistemasini bilishingiz kerakdir.

Ikkilik sanoq sistemasi 0 va 1 sonlaridan iborat bo'lgan sanoq sistemasidir.

Misol uchun:

4 soni ikkilik sanoq sistemasida 0100 deganidir ammo bu to'g'ri lekin sistema buni quyidagicha tushunadi 0000 0000 0000 0000 0000 0000 0000 0000 yani 32 ta 0 va bir raqamli yani 32 bit bu 4Byte deganidir lekin sistemada har 4 raqamdan keyin bo'sh joy bo'lmaydi siz qulay tushunishingiz uchun shuni qildim.

Yani 4 sonimiz bizda

0000 0000 0000 0000 0000 0000 0000 0100 kabi ko'rinadi.

Qanday qilib bir biriga o'g'iramyiz.

Bu juda oson eng ohiridan bo'shlab 1 2 4 8 16 ... 256 kabi 2 ning darajalarini qo'yib chiqasiz.

Endi not operatori nima qilishini ko'raylik. Uning ishi 0 sonini 1 ga 1 sonini 0 ga o'girish yani teskarisiga o'girishdir.

Not operatori ~ belgi bilan belgilanad.

```
var iNum=4;
```

```
var iNum2 = ~iNum;
```

```
alert (iNum2); //shunda oynaga -5 chiqadi nega?
```

Sababi:

sonlar sistemaga quyidagicha qayd etiladi deylik 4 soni

0000 0000 0000 0000 0000 0000 0000 0100

a -4 soni esa yuqoridagini teskarisi va unga 1 qoshilganidir

yani:

1111 1111 1111 1111 1111 1111 1111 1011

+0000 0000 0000 0000 0000 0000 0000 0001

=

1111 1111 1111 1111 1111 1111 1111 1100

bu bizning -4 sonimizdir.

## JavaScript Hammaga

Qayta o'girish ham shunday hammasini teskarisini olasiz va 1 sonini qo'shasiz.

Endi -5 soni chiqqanini sababi shundaki

hammasini teskarisini qildi ammo 1 qo'shmadi shunda bizda quyidagi hosil bo'ladi

1111 1111 1111 1111 1111 1111 1111 1011

bu bizda – son degani alert bilan oynaga yuborganimizda buni javascript minus 10 likda ko'rsatishi kerak.

Nima qiladi?

Hammasini teskarisini oladi va 1 sonini qo'shadi

0000 0000 0000 0000 0000 0000 0000 0101 //mana ko'rib turganingizdek bu 5 soni va oldiga – belgisini qo'yib ekranga chiqardi.

Agarda shu yerda nimanidir yahshi tushuntira olmagan bo'lsam menga yozing [uzinfo@inbox.ru](mailto:uzinfo@inbox.ru) iloji boricha tushuntirib kitobimga o'zgartirish kiritaman.

### AND operatori

Bu operator & belgisi bilan ishlaydi. U nima ish qiladi? Uning ishi quyidagi tablitsadan ko'rsa bo'ladi

Bit birinchi son	bit ikkinchi son	natija
1	1	1
1	0	0
0	1	0
0	0	0

ikta sonni and orqali birlashtirganimizda yuqoridagi chiqadi. Yani 1 & 1 bu 1 ga teng 1 & 0 0 ga teng. Yani agarda sonlarimizning birida 0 soni bo'lsa uni 0 ga o'giradi agarda ikkalasi ham 1 bo'lsa unda 1 beradi.

### OR operatori

| belgisi bilan ishlatiladi.

Bit birinchi son	bit ikkinchi son	natija
1	1	1
1	0	1
0	1	1
0	0	0

Bu and operatorining teskarisi agardakamida bitta 1 bo'lsa natijasi 1 chiqadi

### XOR operatori

## JavaScript Hammaga

^ belgisi bilan beriladi.

Bit birinchi son	bit ikkinchi son	natija
1	1	0
1	0	1
0	1	1
0	0	0

Bu agarda ikkala soni bir biriga teng bo'lsagina 0 natija beradi. Bo'lmasam 1 natija beradi.

### Left SHIFT

bitlarni chap tarafga surish. Bu operator << belgisi bilan belgilanadi.

Misol:

```
var iNum=2;           //0010
var iNum2=iNum<<2     //1000
```

Orqa qismi 0'lar bilan to'ldiriladi.

Nimaga kerak? Kichik misol kerakligiga bu qandaydir sonni 2 ga yoki uni darajasiga ko'paytirmoqchi ko'paytirmoqchi bo'lsangiz unda nima qilasiz yoki \* ishlatasiz yoki << shift ishlatishingiz mumkindir.

Misol:

```
var iNum=5;
alert (iNum<<1);           //”10” chiqadi
alert (iNum<<10);          //5*1024=5120
```

Ha esdan chiqarmang bunda uning minus plus belgisi saqlanib qoladi.

Yani -2 ni 5 ta sursak unda 64 emas balki -64 bo'lib qolaveradi.

Shu yerda etib o'tish joiz deb bildim, sonlar bitlarini eng birinchisi bu + - belgisining bitidir, agarda 1 bo'lsa unda minus son degani.

### Right SHIFT

Bu >> ham hudi shu ishni qiladi ammo farqi o'ng tarafga suradi, lekin birinchi bitga tegmaydi. Agar uni ham surish niyatingiz bo'lsa unda >>> ishlatasiz.

### Logical NOT

Bular Unary Operatorlaridan farqi shundaki bular faqat true yoki false qaytaradi. Yuqoridagi operatorlar esa son qaytaradi (ikkilik).

NOT ! Belgisi bilan beriladi.

Misol:

## JavaScript Hammaga

```
var bTek=false;
if (!bTek) {
  alert("Ura ishladi");
}
```

!bTek bizda true degani shuning uchun if statement ishlaydi.

### AND

Yuqoridagi end kabidir faqat farqi bu ham true yoki false qaytaradi va && belgisi bilan ishlatiladi.

Misol:

```
var bT = true;
var bF = false;
if (bT && bF) {
  alert ("Hopa chota ishladi");
}
```

Oynamizga hech qanday ma'lumot chiqmaydi sababi bittasi true bittasi esa false ohiri false bo'ladi.

### OR

Agarda bir qanchasidan bittasi true bo'lsani tekshiradi va || belgisi bilan ishlatiladi.

Misol:

```
var bT = true;
var bF=false;
if (bT || bF) {
  alert ("Ishladi");
}
```

bu yerda bittasi true shuning uchun oynamizga "Ishladi" chiqdi.

### Module operatori

Module % belgisi orqali hisoblanadi

Misol:

```
var iNum=45;
alert(iNum%2);           //oynamizga 1 sonini chiqaradi
```

### === operatori

Bu operator tekshirish operatoridir, aynan == kabidir ammo farqi shundaki objectlani bir biriga o'zgartirmaydi.

## JavaScript Hammaga

Misol:

```
var iNum=55
var sText="55";
alert (iNum==sText);      //oynada true chiqadi chunki bu ikkalasini tekshirishdan oldin turi
o'zgartiriladi
alert (iNum===sText);     //oynaga false chiqaradi.
```

### Conditional operator

Bu yerda qandaydir tekshiruvga ko'ra qiymat o'zgaruvchanga beriladi.

Misol:

```
var iNum=5, iNum2=6
var iMax=(iNum>iNum2) ? iNum: iNum2
alert (iMax);           //oynaga 6 soni chiqadi.
```

Yani bu yerda nima qildik? Bu yerda biz iNum va iNum2 ni bir biri bilan solishtirdik va agarda iNum katta iNum2 bo'lsa unda iNum ni iMax'ga berishni bo'lmasa iNum2 ni iMax'ga berishni aytdik.

### Qo'shimcha tenglik operatorlari

+= operatori misolga qarang

```
var iNum=10;
iNum=iNum+10;          //iNum=20 bo'ladi
```

shuni osonroq bajarish uchun

```
var iNum=10;
iNum +=10              //Bu yerda ham iNum=20 bo'ladi aynan yuqoridagidek.
```

Huddi shunday \*=, -=, /=, %=, <<=, >>=, >>>= kabi operatorlar ham bordir. Bular ham aynan yuqoridagidek.

Misol:

```
var iNum=5;
iNum *=5               //iNum=25
```

Masala: 167 ni 128 gako'paytiring, faqat \* ni ishlatmasdan.

### IF

```
if (condition) { statement } else {statement}
```

Yani yuqorida IF'ni ko'ribchiqgan edik man bu yerda biz uni chuqurroq ko'rib chiqamiz.

## JavaScript Hammaga

Misol:

```
var bTek = true;
if (bTek) {
  alert ("bTek = true ekan");
}
else {
  alert ("bTek = false eskan");
}
```

if (condition1) { statement } esle if (condition 2) { statement 2 } else { statement 3 }

Bu yeda esa biz ikta tekshirishni ishlatdik yani ikka marta tekshiradi agar tog'ri kemasda unda else ning ichidagini ishlatadi.

Misol:

```
var iNum=45;
if (iNum ==45) {
  alert ("son 45"); }
else if (iNum>45) {
  alert ("son 45 dan katta");
}
else {
  alert ("son 45 dan kichik");
}
```

Agarda istasangiz bir qancha esle if ishlatishingiz mumkindir.

### **do-while**

Bu operatorimiz tsikl/qaytariluvchi ishlar uchu kerak bo'ladi.

```
do {
  statement }
while (expression);
```

Misol:

```
var iNum=0;
do {
  iNum++;
}
while (iNum<10);
alert (iNum);
```



## JavaScript Hammaga

By yerda biz har gal iNum ni bir soniga oshirmoqdamiz va ohirida agarda iNum  $\geq 10$  bo'lsa unda chiqishi kerakligini va tsikl'ni tugatishligini while bilan berdik. Keyin esa oynaga iNum'ning qiymatini chiqardik. Esdan chiqarmang bu yerda qandaydir (statement) bajariladi undan keyin esatekshiruv ishi (while) bajariladi.

### 2. Masala:

teskt berilgan uni hamma harflarini katta harfgao'giring va oynaga chiqaring toLowerCase toUpperCase ishlatmang.

#### **while**

Aynan do-while kabidir farqi shundaki avval tekshiradi undan so'ngina agar qabul qilinda ishni bajaradi.

Misol:

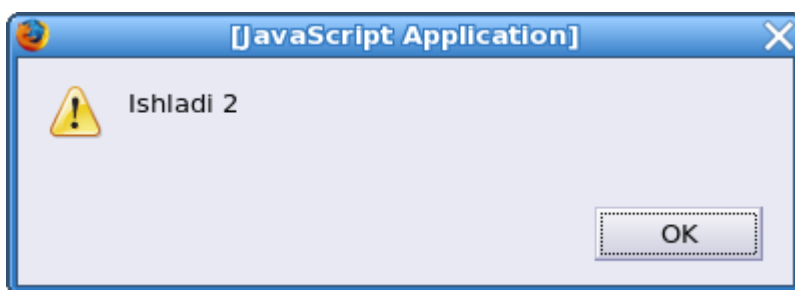
```
var iNum=0;
while (iNum=<10) {
iNum++;
}
alert(iNum);
```

Yoki quyidagi misolga qarang, bu yerda do-while va while'ning farqi yaqqol ko'rinadi.

```
var bT=false;
while (bT) {
alert("Ishladi");
}
```

```
do {
alert("Ishladi 2");
} while (bT);
```

## JavaScript Hammaga



Mana ko'rib turganingizdek oynangizda Ishladi 2 ning o'zigina ko'rindi.

### Kichik Imtihon

1. Komentariya uchun qanday belgilarni ishlataman? Qancha bo'lsa hammasini yozing.<sup>5</sup>
2. Oynaga "Toshkent Salom" so'zini chiqaring, va Salom so'zi ikkinchi qatorda chiqsin.<sup>5</sup>
3. Qandaydir o'zgaruvchan bor, uning turini oynaga chiqaring.<sup>5</sup>
4. Tekstimiz bor, uning uzunligini oynaga chaqing<sup>5</sup>
5. Ikta sonimiz bor, + - belgisidan qat'iy nazar ikkalasini qo'shing<sup>10</sup>
6. Ihtiyoriy qoldiqli son bor, agarda nuqtadan keyingi son 5 va unda katta son bo'lsa yuqoriga bo'lmasam pastga qarab butunlashtiring<sup>5</sup>
7. Sonimiz bor uni ildizini oynaga chiqaring<sup>10</sup>
8. 1-100 orasida ihtiyoriy butun son chiqaruvchi script yozing<sup>15</sup>
9. Qanday qilib 16 sanoq sistemadidan 10 likga og'iramiz, misol keltiring<sup>5</sup>
10. Ihtiyoriy teskt berilgan uni o'rtadagi harfini Code'ini oynagachiqaring<sup>5</sup>
11. Ihtiyoriy teskt berilgan undagi eng ohirgi turgan "n" harfini turgan joyini chiqarsin<sup>5</sup>
12. Ihtiyoriy tekst berilgan uni ohirgi uchta harfini kesib ko'rsatsin<sup>5</sup>
13. Tekst berilgan uni boshidagi, o'rtasidagi va ohiridagi harfni bosh harfga og'irib chiqaring<sup>20</sup>

### for

Bu ham qaytarilishdir, yani while'lar kabi, yozilishi

*for (initialization; expression; post-loop-expression) statement*

Misol:

```
for (var i=0; i<10; i++) {  
  alert (i);  
}
```

Bu yerda biz i ni aniqlavoldik, keyin esa 10dan kichik bo'lsa bajarni berdik, va keyingisida esa uni ortib borishini belgilab berdik. Ichida esa oynaga har bir sonni choqarishni berdik.

### for in

Bu asosan obyektning xususiyatlari bilan aloqali ishlar uchun kerak bo'ladi.

## JavaScript Hammaga

For (property in expression) statement

Misol uchun window obyektini xususiyatlarini ko'rish uchun.

```
for (sProp in window)
{ alert (sProp); }
```

Bu yerda bizga bir qancha window obyektining xususiyatlarini ko'rsatib beradi.

O'zingiz yozing

1. Tekst bor uni ichidagi harflarni katta harfga o'giring, butun tekstga toUpperCase ishlatmasdan.

Echimi:

```
<script language="javascript">
var sText='Sandjar./qani';
var sRes="";
for (var i=0; i<sText.length; i++) {

if ((sText.charCodeAt(i)>96)&& (sText.charCodeAt(i)<128)) {
sRes=sRes+sText.charAt(i).toUpperCase();
}
else {
sRes=sRes+sText.charAt(i);
}
}
alert (sRes);
</script>
```

### Labels

Bu qandaydir joyga belgi qoyib ketishdir, yani keyinchalik shu yerga qaytish uchun.

*label: statement*

misol:

```
bir: alert('123');
```

### break continue

Bu operatorlar, qandaydir qaytarilish operatorini sizga kerak bo'lgan joyda qilinayotgan ishini tugatishi uchun ishlatiladi.

Misol:

## JavaScript Hammaga

```
var iNum = 0;
for (var i=1; i<10; i++) {
  if (i%5==0){break;}
}
iNum++;
}
alert (iNum);
```

Bu yerda oynamizga 4 soni chiqadi sababi 5 soni 5 ga qoldiqsiz bo'linadi va uning modali 0 ga teng boladi va break ishlab for'ning ishini tugatadi.

continue esa kelgan joyidan yana boshiga borish uchun yani.

Misol

```
var iNum = 0;
for (var i=1; i<10; i++) {
  if (i%5==0){continue;}
  iNum++;
}
alert (iNum);
```

Bu yerda oynamizga 8 soni chiqadi, sababi iNum++ bir marotaba ishlamaydi, yani i=5 bo'lganda continue ishlab uni for'ning boshiga olib chiqadi va kelgan joyidan davom etib bir marotaba iNum++ ishlamay qoladi.

O'zingiz ishlang.

Ihtiyoriy tekst berilgan undagi ihtiyoriy bir harfni joyigacha bo'gan tekstni chiqaruvchi script yozing, indexOf ishlatmang.

Echimi:

```
<script language="javascript">
var sText = "Hamma hursand";
var sRes="";
for (var i=0; i<sText.length; i++) {
  if (sText.charAt(i)=="r"){break;}
  sRes=sRes+sText.charAt(i);
}
alert (sRes);
```

## JavaScript Hammaga

</script>

Ho'sh keling keyingi misolga qaraymiz.

```
var iNum=0;
tepa:
for (var i=0; i<10; i++) {
  for (var j=0; j<10; j++) {
    if (i==5 && j==5) {
      break tepa;
    }
    iNum++;
  }
}
alert (iNum);
```

Bu yerda esa biz i=5 va j=5 bo'lganida break qilib ikkala for'ning ichidan chiqib ketmoqdamiz, sababi tepa bu for'larning boshidir. Va oynamizga 55 soni chiqadi.

Agarda continue qilganimizda unda

```
var iNum=0;
tepa:
for (var i=0; i<10; i++) {
  for (var j=0; j<10; j++) {
    if (i==5 && j==5) {
      continue tepa;
    }
    iNum++;
  }
}
alert (iNum);
```

Bu yerda iNum=95 bo'ladi sababi ikkalasi ham 5 ga teng bo'lganda continue ishni yuqori tepa'dagi for'ga chiqarib yuboradi va u kelgan 6 sonidan boshlab yana davom etadi, va jo'zining 5ta qadamini yo'qotadi.

### Dars 5

**with**

## JavaScript Hammaga

Bu operator qandaydir obyekt bilan ishlash uchundir. Misolga qarang

```
var sMes = "Tekst";  
with (sMes) {  
  alert(toUpperCase());  
}
```

Bu yerda with ni ichida alert ochib toUpperCase berdik bu toUpperCase sMes'ga tegishli, sababi biz oldindan ko'rsatib qoydik with bilan qaysi obyekt biulan ishlashni boshlaganimizni.

### switch

Bu operatorimiz aynan if -esle if -else kabidir amma osonroq ko'rinishda, yani

```
switch (expression) {  
  case value: statement  
  break;  
  case value: statement  
  break;  
  ...  
  default: statement  
}
```

Misolda

```
switch(i) {  
  case 25: alert("25");  
  break;  
  case 35: alert("35");  
  break;  
  default: alert("son");  
}
```

bu misolimiz teng

```
if (i==25) alert("25");  
else if (i==35) alert("35");  
esle alert("son");
```

Bu ikkalasi bir biriga tengdir, faqat yuqoridagisi osonroq ko'rinishga ega va qulay yoziladi.

### Functions

Bu Programming'ning eng asosidir. JavaScript ko'dini istalgan joyida elon qilinadi, va uning ishi

## JavaScript Hammaga

qandaydir programmako'dini bir necha bora ishlatsangiz va sizga qandaydir informatsiyani ishlab chiqib uni javobi kerak bo'lsa.

```
function function_name(arg1, arg2, ... argn){  
statement  
}
```

Misol

```
<script language="javascript">  
function alert_me(sName, sMes){  
alert (sName + " Hush kelibsiz " + " "+sMes);  
}  
alert_me('Sinch', 'somsa olib keldingizmi?');  
</script>
```

Mana yuqorida biz funktsiyani elon qildik va nima ish qilishi kerakligini berib chiqdik, undan so'ng esa uni chaqirdik “alert\_me(...,...)” orqali.

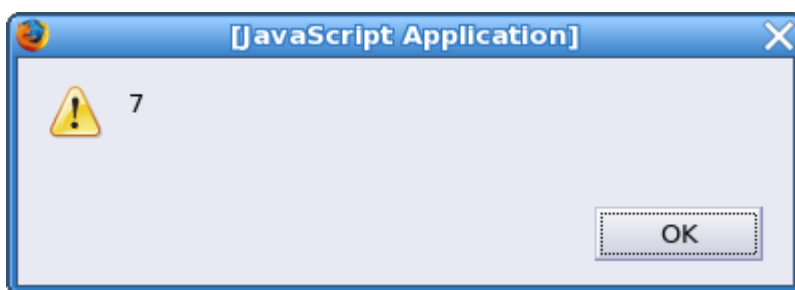
Agarda biz funktsiyamiz nimadir qaytarishini hohlasakchi? Ikta sonni bir biriga qo'shib uni qiymatini qaytaruvchi misol yozaylik.

```
var i=0;  
function qosh(num1, num2) {  
return num1+num2;  
}  
i=qosh(3, 4);  
alert (i);
```

mana bu bizning ko'dimiz, siz qiladigan ishingiz faqatgina return orqali kerakli qiymatni orqaga qaytarib yuborishdir.

Mana oynamizga:

## JavaScript Hammaga



1. O'zingiz yozing ikta son bor, shu ikta sonni bir birdan ayiruvchi funksiya yozing, lekin shunday ayirinki sonlarni orasidan kattasidan kichigini ayiring.

Echimi

```
var i=0;
function ayir(num1, num2) {
return Math.max(num1,num2)-Math.min(num1, num2);
}
i=ayir(6, 4);
alert (i);
```

ho'sh agarda siz funksiya nomli o'zgaruvchi (argument) bermagan bo'lsangizchi?

Unda siz bu argument'larga quyidagicha yetishishingiz mumkindir

Misol:

```
function oynaga() {
alert (arguments[0]);
alert(arguments[1]);
}
oynaga("birinchi", "ikkinchi");
```

va oynangizga "birinchi" va "ikkinchi" chiqadi. Ho'sh nechta argument borligini bilish uchun length funksiya ishlata'miz.

Yani arguments.length

2. O'zingiz yozing, qancha argument berishingiz nomalum lekin hammasi son bo'lishi kerak, shu sonlar qo'shilgani yani yegindisini oynaga chiqaring

Echim:

```
function yegindi() {
var yegindi=0;
for (var i=0; i<arguments.length; i++){
yegindi=yegindi+arguments[i];
}
```



## JavaScript Hammaga

```
return yegindi;  
}  
alert(yegindi(12, 13, 25, 50, 100));
```

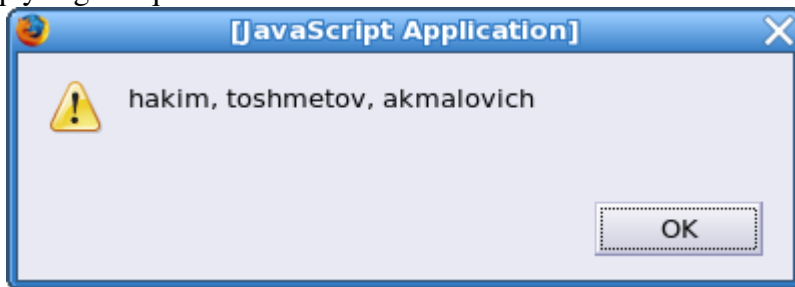
### Array

Bu bizning o'zgaruvchanlarimizning bir turidir. Yani bitta o'zgaruvchan nomini ich ichida bir qancha o'zgaruvchanlardir.

#### Misol

```
var oStudent= new Array ();  
oStudent['name']="hakim";  
oStudent['surname']="toshmetov";  
oStudent['lastname']="akmalovich";  
alert(oStudent['name']+" , " +oStudent['surname']+" , " +oStudent['lastname']);
```

Shunda oynamizga quyidagi chiqadi



Bu bizning bir o'zgaruvchan ichida ich ich bir qancha o'zgaruvchanlar ishlatishimizga yordam beradi. Yoki so'zlarning o'rniga son ham ishlatishingiz mumkindir

```
var oStudent= new Array (3);  
oStudent[0]="hakim";  
oStudent[1]="toshmetov";  
oStudent[2]="akmalovich";  
alert(oStudent[0] +", " +oStudent[1] +", " +oStudent[2]);
```

### Ich ichi Function

Siz funktsiya ichida yana bir funktsiya ochishingiz mumkindir. Misolga qarang

```
function myFuncA() {  
  var valueA="A";  
  myFuncB();  
  function myFuncB(){
```

## JavaScript Hammaga

```
var valueB="B";  
alert (valueA);  
alert(valueB);  
}}
```

Mana ko'rganingizdek biz bir funktsiyani ichida ikkinchi funktsiyani ochdik va uni chaqirdik.

### Anonim Function

Bu yerda biz qanday qilib anonimlashtirilgan funktsiya yaratish va ishlatishni ko'ramiz.

Buning uchun:

```
var Nimadir=function () {statement}
```

kabi ishlatishingiz mumkindir.

Buni chaqirish uchun Nimadir ni chaqirsangiz yetarlidir.

Yana shu yerda ko'rsatib ketish o'rinli deb bildim

quyida misol keltirilgan undan keyin uni ko'rib chiqamiz

Misol:

```
function showAll() {  
    alert("Employee " + this.name + " is " + this.age + " years old.");  
}
```

```
var employeeDB = [{name:"Alice", age:23, show:showAll},  
                  {name:"Fred", age:32, show:showAll},  
                  {name:"Jean", age:28, show:showAll},  
                  {name:"Steve", age:24, show:showAll}];
```

```
employeeDB[1].show();
```

### this

Deylik sizda qandaydir object'iz bor va uni ichida funktsiyasi bor deylik, va siz shu funktsiya ichida object'ning qandaydir elementini ishlatmoqchisiz shunda siz this.nimadir orqali chaqrishingiz mumkindir. Yani this aktiv bo'lgan objectga ko'rsatkichdir.

Bu yerda biz employeeDB o'zgaruvchisiga show degan funktsiyani berdik, ammo uni farqi shundaki uni o'zi funktsiya emas balki showAll() funktsiyasiga ko'rsatkichdir.

### Array bilan aloqali operatorlar

#### join()

Bu operatorimiz array'ni ichidagi hamma o'zgaruvchanlarni qo'shib bitta qiladi.

Misol:

```
var sText=new Array('Hakim', 'Nozim', 'Kozim');
```

## JavaScript Hammaga

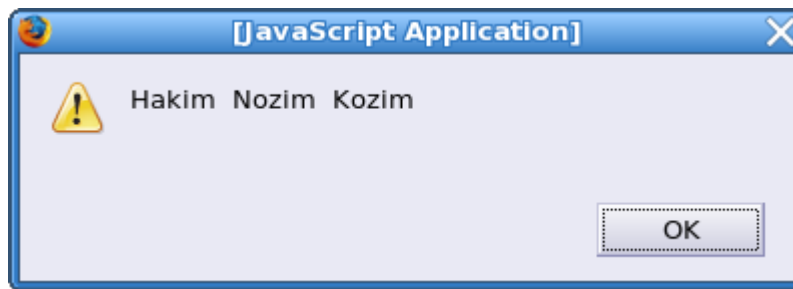
```
alert (sText.join(','));      //Oynamizga Hakim,Nozim,Kozim chiqadi
alert (sText.join('/'));     //Oynamizga Hakim/Nozim/Kozim
alert (sText.join(' '));     //Oynamizga Hakim Nozim Kozim chiqadi
```

### split()

Yuqoridagini teskarisi yani biz tekstni bo'lib tashlamoqchimiz bo'laklarga

Misol:

```
var sText='Hakim, Nozim, Kozim';
var sArray=sText.split(',');
alert(sArray[0]+ ' ' +sArray[1]+ ' ' +sArray[2]);
```



Mana ko'rganingizdek etilgan belgilardan tekstni bo'lib tashladi. Agarda ',' bermaganimizda nima bo'lardi yani quruq split("") ni o'zini ishlatganimizda? Unda hamma harf/belgilarni birma bir bo'lib tashlagan bo'lardi.

Bular qatorida yana huddi string kabi concat, slice operatorlari ham ishlaydi, ularni o'zingiz misollarda ko'rib chiqing.

### reverse()

Bu array'ning ichidagi o'zgaruvchanlarni teskarisiga aylantiradi, yani:

```
var aNum=['1', '2', '3'];
alert(aNum.reverse().toString());      //Oynamizga 3 2 1 chiqadi.
```

### sort()

Arrayni ichidagilarni tartiblaydi.

```
var aNum=['2', '1', '3'];
alert(aNum.sort().toString());
```

Shunda oynada 1 2 3 chiqadi

### splice()

## JavaScript Hammaga

Bu operatorimiz Array'ichidagi elementlarni o'chiradi yoki qo'shadi misol

```
var aText=['green', 'red', 'yellow'];  
aText.splice(2, 1);  
alert (aText.toString());
```

Bu yerda oynamizga green, red, yellow, blue qoyadi, birinchi son: bu qayerdan boshlashni, va ikkinchi son qancha o'chirish kerakligini bildiradi. 2Nchi elementdan keyin 1 ta elementni o'chirishni berdik.

```
var aText=['green', 'red'];  
aText.splice(2, 0, 'yellow', 'blue');  
alert (aText.toString());
```

Bu yerda oynamizga green, red, yellow, blue qoyadi, birinchi son: bu qayerdan boshlashni, va ikkinchi son qancha o'chirishni yani 0 ta element o'chir dedik.

Agarda almashtirmoqchi bo'lsak unda

```
var aText=['green', 'red', 'yellow'];  
aText.splice(2, 1, 'blue');  
alert (aText.toString());
```

Oynamizga green, red, blue so'zlari chiqadi.

### 3. O'zingiz ishlang:

Shunday script yozingki ruscha alifboda yozilgan tekstni lotincha alifboga o'girsin.

Echimi:

```
<script language="javascript">  
var sText='Салом';  
var sDone="";
```

```
function transr(txt) {  
var rus = new Array ('к', 'г', 'х', 'л', 'е', 'о', 'з', 'ы', 'э',  
'ъ', 'и', 'я', 'ж', 'щ', 'ч', 'б', 'р', 'т', 'п', 'м', 'д', 'ц',  
'ь', 'с', 'ю', 'у', 'н', 'й', 'ф', 'ш', 'в', 'а');  
rus['к']='K';  
rus['г']='G';  
rus['х']='H';  
rus['л']='L';
```

## JavaScript Hammaga

```
rus['e']='E';
rus['o']='O';
rus['з']='Z';
rus['ы']='';
rus['э']='E';
rus['ъ']='';
rus['и']='I';
rus['я']=' Ya';
rus['ж']=' J';
rus['щ']='Sh';
rus['ч']='Ch';
rus['б']='B';
rus['р']='R';
rus['т']='T';
rus['п']='P';
rus['м']='M';
rus['д']='D';
rus['ц']='Ts';
rus['ь']='';
rus['с']='S';
rus['ю']='Yu';
rus['у']='Y';
rus['н']='N';
rus['й']='Y';
rus['ф']='F';
rus['ш']=' Sh';
rus['в']='V';
rus['а']='A';
```

```
var latr=rus[txt.toLowerCase()];
if (!latr) {latr=txt;}
return latr;
}
```

```
for (var i=0; i<parseInt(sText.length, 10); i++)
```

## JavaScript Hammaga

```
{  
  
sDone+=transr(sText.substring(i, i+1));  
}  
alert (sDone);  
</script>
```

### Dars 6

#### Date

Bu Object Java objecti kabidir, kunlarni 12:00 (ertalabki/tungi) 1 Yanvar 1970 yildan boshlab sanab kelinadi.

Date objecti quyidagicha yaratiladi:

```
var myDate = new Date();           //Hozirgi vaqt  
var myDate = new Date(yyyy, mm, dd, hh, mm, ss);           //siz istagan vaqt boyicha  
var myDate = new Date(yyyy, mm, dd);  
var myDate = new Date("monthName dd, yyyy hh:mm:ss");  
var myDate = new Date("monthName dd, yyyy");  
var myDate = new Date(epochMilliseconds);
```

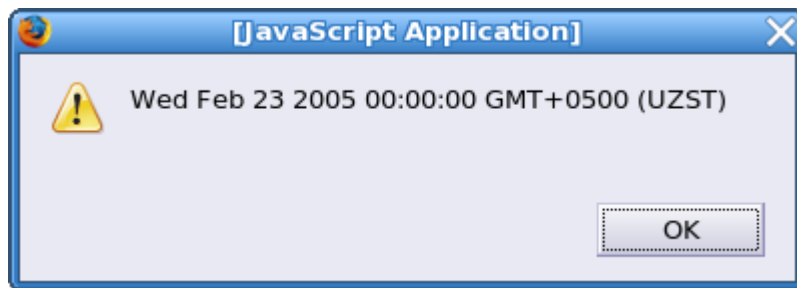
Misol:

```
var myDate=new Date();  
alert(myDate);
```

yoki

```
var myDate=new Date(2005, 01, 23);  
alert(myDate);
```

## JavaScript Hammaga



Shular bilan birga siz `parse()` `UTC()` method'larini ham ishlatishingiz mumkindir.

Misol

```
var myDate=new Date(Date.parse('Feb 23, 2005'));    //Bu yerda javascript sistemaning kun  
                                                    //yozilishiga qaraydi
```

```
alert(myDate);
```

yoki

```
var myDate=new Date(Date.UTC(2005, 01, 23));
```

```
alert(myDate);
```

Siz UTC da yil, oy, kun, soat, minut, second larni ham ko'rsatishingiz mumkin.

### **toDateString()**

Bu methodimiz bizga date'dagi ma'lumotni yil, oy, va kunini o'zini ko'rsatish uchun ishlatiladi.

Misol:

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

```
alert(myDate.toDateString());
```

### **toTimeString()**

Faqatgina vaqtni ko'rsatadi.

Misol

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

```
alert(myDate.toTimeString());
```

### **toLocaleString()**

Sizning sistemadagi Locale'ga qarab vaqt va kuni ko'rsatadi.

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

```
alert(myDate.toLocaleString());
```

## JavaScript Hammaga



### **toLocaleDateString(), toLocaleTimeString()**

Yuqorida etib o'tilgan toDateString lardan farqi shundaki bu yerda sistemani locale'i hisobga olinadi.

```
var myDate = new Date();  
alert(myDate.toLocaleDateString());  
alert(myDate.toLocaleTimeString());
```

Oynangizga sistemada qanday kun va vaqt ko'rsatishini belgilab bergan bo'lsangiz shunday ko'rsatib berdi.

### **toUTCString()**

UTC formatida kun va vaqtni ko'rsatadi.

```
var myDate = new Date();  
alert(myDate.toUTCString());
```

UTC'da sizga vaqtni ko'rsatib berdi.

Method	Tarifi
getTime()	Millisekundlarda kuni ko'rsatadi.
setTime(milliseconds)	Sets the milliseconds representation of the date.
getFullYear()	Yilni ikta sondamas to'la 4ta son bilan ko'rsatadi (2005).
getUTCFullYear()	Yilni UTC'da 4ta son bilan ko'rsatadi.
setFullYear(year)	Yilni belgilab berish, 4ta son bilan yoziladi.
setUTCFullYear(year)	Yilni UTC'da belgilab berish, 4ta son bilan yoziladi.
getMonth()	Oyni qaytaradi sonlarda, 0-Yanvar ... 11-Dekabr
getUTCMonth()	Oyni UTCda qaytaradi, 0-Yanvar ... 11-Dekabr.
setMonth(month)	Oylarni berish agarda 11 dan katta son bo'lsa unda yil qo'sha boshlaydi.
setUTCMonth(month)	UTC'da oy qo'shish 11 dan katta son yil qo'shadi.
getDate()	Oyning kunini ko'rsatadi.
getUTCDate()	UTCda oy kunini ko'rsatadi.
setDate(date)	Oy kunini belgilab beradi.
setUTCDate(date)	Sets the day of the month of the UTC date.
getDay()	Hafta kunini qaytaradi
getUTCDay()	Hafta kunini UTC bo'yicha qaytaradi.
setDay(day)	Hafta kunini belgilab beradi.
setUTCDay(day)	UTC boyicha hafta kunini belgilaydi.
getHours()	Soatni qaytaradi.
getUTCHours()	UTC bo'yicha soatni qaytaradi.
setHours(hours)	Soatni belgilab beradi.
setUTCHours(hours)	UTC bo'yicha soatni belgilab beradi.



## JavaScript Hammaga

getMinutes()	Minutlarni qaytaradi.
getUTCMinutes()	UTC bo'yicha minutlarni qaytaradi.
setMinutes(minutes)	Minutlarni belgilab beradi.
setUTCMinutes(minutes)	UTC bo'yicha minutlarni belgilaydi.
getSeconds()	Sekundlarni qaytaradi.
getUTCSeconds()	UTC bo'yicha sekundlarni qaytaradi.
setSeconds(seconds)	Sekundlarni belgilab beradi.
setUTCSeconds(seconds)	UTC bo'yicha sekundlarni belgilab beradi.
getMilliseconds()	Millisekundlarni qaytaradi.
getUTCMilliseconds()	UTC bo'yicha millisekundlarni qaytaradi.
setMilliseconds(milliseconds)	Millisekundlarni belgilab beradi.
setUTCMilliseconds(milliseconds)	UTC bo'yicha millisekundlarni belgilab beradi.

1. Ozingiz ishleng. Shunday misol yozing 5 sekundan keyin habar chiqarsin. Echim:

```
var bTrue=true;
var d = new Date();
function tDif(timer1){
var tDate= new Date();
var tDiferen =tDate.getSeconds()-timer1;
return tDiferen;
}
do {
if (tDif(d.getSeconds())>5) {break;}

}
while (bTrue);
alert("5 Sekund otdi");
```

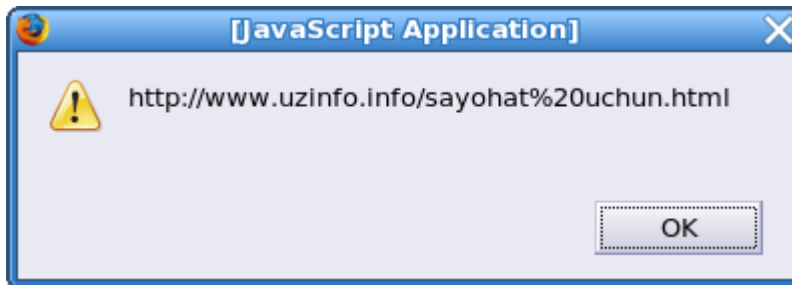
Mana ko'rganingizdek juda ham osondir.

### **encodeURIComponent() encodeURIComponent()**

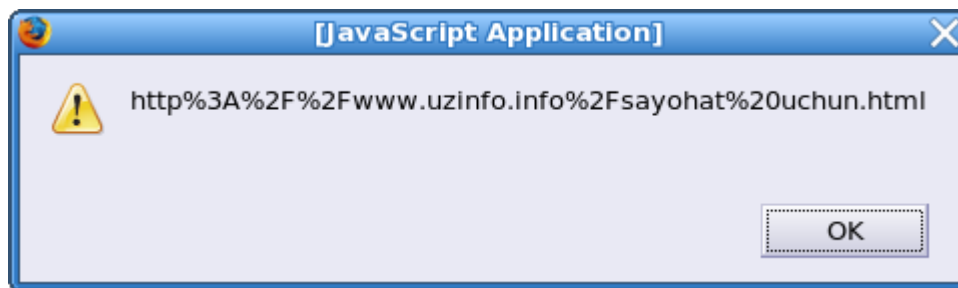
Bu ikta operatorning ishi shundan iboratki, URI(Universal Resource Identifier) larni yani web adreslarni tog'ri yozdirish uchundir. Misol uchun web browserlar probellik yani bosh joylik adreslarni qabul qilmaydi va uni hato chiqaradi. Shularni tuzatish uchun ishlatiladi. Bir birdan farqi encodeURIComponent() hamma belgilarni UTF8 ga ogiradi encodeURI() esa faqatgina probellarni yani bo'sh joylarnigina holos.

Misol:

```
var adres="http://www.uzinfo.info/sayohat uchun.html";
alert(encodeURIComponent(adres));
alert(encodeURIComponent(adres));
Shunda oynamizga quyidagilar chiqadi.
```



## JavaScript Hammaga



Mana ko'rib turibsizki `encodeURIComponent()` faqatgina bo'sh joyni encode qilib berdi, `encodeURIComponent()` esa hamma belgilarni UTF8 ga ogirdi va buni web browserlar tushuna oladigan qildi.

Bu operatorlarni teskarisi esa **`decodeURI()`**, va **`decodeURIComponent()`** lardir

### **`eval()`**

Bu operatorimizning ishi string ECMAScriptni script kabi ishlatishdir.

Misol

```
var sText = "alert('Salom Halq')";  
eval(sText);
```

bu aynan `alert("Salom Halq")` kabidir. Buni bizga nima keragi bor? Deylik qandaydir javascript ko'dini biz tekst faylning ichida joylashtirsak yoki qandaydir script ko'di bilan azgina o'zgartirishlar kiritish kerak deylik vaqtga qarab shunda biz bularni string orqali qo'yib ular bilan kerakli o'zgartirishlarni qilib eval orqali ishlatsak bo'ladi.

### **Prototip yaratish**

Bu nima degani quyidagi misolga qarang.

```
var Somsa = new Object;  
Somsa.daraja = 110;  
Somsa.narh = 250;  
Somsa.masalig = "Gosht, Piyoz, Hamir";  
Somsa.showNarh = function () {  
  alert (this.narh);  
}
```

```
Somsa.showNarh();
```

Bu yerda bizga somsani narhini chiqarib beradi, ho'sh agarda bizda bir nechti hil somsamiz bo'lsachi? Unda nima qilamiz hammasini bittami bitta shunday qilib belgilab chiqishimi kerakdir. Yoki constructor hususiyatlaridan foydalanamiz.

Bitta kichik misol:

```
function Somsa (daraja, narh, masalig) {  
  this.daraja=daraja;  
  this.narh=narh;  
  this.masalig=masalig;  
  this.showNarh = function(){  
    alert(this.narh);  
  }  
}
```

## JavaScript Hammaga

```
var Somsa1=new Somsa(100, 300, "Hamir, Gosht, Dumba, Piyoz");
var Somsa2 = new Somsa(120, 250, "Gosht, Piyoz, Hamir");
Somsa1.showNarh();
Somsa2.showNarh();
```

Ishlatib ko'ring. Ha ha ishladi, ko'rdingizmi qanchalik foydali constructor hususiyatga ega ekan. Ishimizni eng kamida 2 marotaba qisqartirdi.

Lekin ahamiyat bergan bo'lsangiz bu yerda biz Somsani ko'pyasini yaratmoqdamiz yani ikki marotaba Somsa kopyasi yaratildi. Ho'sh biza uchun har gal funktsiyani boshidan yaratishi nima keragi bor? Qoshimcha hotira ketishi va hkz. Keling mana shu yera prototype va constructor ning bir birga qoshilgan turini ko'raylik.

Misol:

```
function Somsa (daraja, narh, masalig) {
this.daraja=daraja;
this.narh=narh;
this.masalig=masalig;
}
Somsa.prototype.showNarh = function(){
alert(this.narh);
}
var Somsa1=new Somsa(100, 300, "Hamir, Gosht, Dumba, Piyoz");
var Somsa2 = new Somsa(120, 250, "Gosht, Piyoz, Hamir");
Somsa1.showNarh();
Somsa2.showNarh();
```

Bu yerda biz oldin o'tgan prototype degan object hususiyatini ishlatdik, va constructor dan ham foydalandik.

2. O'zingiz yeching: Devor nomli object yarating unda balandligi, kengligi, uzunligi, nimadan qurilganligi bo'lsin, va shunday methodi bo'lsinki u qurilgan hom ashyodan kelib chiqib shunday uyga qancha shunday homashyodan ketishini ko'rsatib bersin.

Hom ashyolar:

- a. hom gisht(kengligi 10sm, uzunligi 30sm, balandligi 10 sm)
- b. pishiq gisht (kengligi 10sm, uzunligi 25sm, balandligi 10sm)
- shlaka blok (kengligi 20, uzunligi 35sm, balandligi 20sm)

c.

Echimi:

```
function Devor (balandlik, uzunlik, kenglik, hom) {
this.uzunlik=uzunlik;
this.balandlik=balandlik;
this.kenglik=kenglik;
this.hom=hom;
}
Devor.prototype.showHisob = function(){
if (this.hom=="pgisht") { alert( (this.balandlik*this.uzunlik*this.kenglik)/0.0025);}
else if (this.hom=="hgisht"){alert( (this.balandlik*this.uzunlik*this.kenglik)/0.003);}
else if (this.hom=="shlak") {alert( (this.balandlik*this.uzunlik*this.kenglik)/0.14);}
else alert("Bunday hom ashyo yoq");
}
var Devor1=new Devor(5, 10,1, "shlak");
Devor1.showHisob();
```

Mana qarab o'rganib chiqing.

## JavaScript Hammaga

### Yangi method yaratish

Ho'sh siz oyladizki shu bilan objectlar bilan oynashimiz tugadimi? Yoq, ahir ECMAScript sizga anchagina kuch bergan, bor bo'lgan object'lar ustida ish qilishingiz mumkindir. Misol uchun NUMBER objecti. Misolga qaraylik

```
Number.prototype.toHexString=function() {  
return this.toString(16);  
}  
var iNum=34;  
alert(iNum.toHexString());
```

Mana ko'rib turganingizdek biz sonlar uchun qo'shimcha method yaratib berib qo'ydik. Huddi shunday siz bor method'larni ham o'zgartira olasiz.

### Dars 7

Mana yuqorida biz ECMAScriptni asosini ko'rib chiqdik, keling endi TAGlar bilan ishlashni va funktsiyalarni chaqirishni ko'raylik.

Misol:

```
<html>  
<head>  
<title>Dars7</title>  
<script language="javascript">  
function alert_me() {  
alert ('Salom Siz knopkani bosdingiz');  
}  
</script>  
</head>  
<body>  
<input type="button" value="Meni bos" onclick="alert_me()">  
</body>  
</html>
```

Yuqorida biz funktsiyani <head>'ni orasiga yozdik va u yerda funktsiya yaratdik, va <body> da <input> tagini ochib u yerdan onclick hususiyati orqali funktsiyamizni chaqirdik. Oynamizdagi tugmani bosganda biz alert\_me() funktsiyasini chaqirmoqdamiz.

Bu yerda biz "event handling" ishlatdik, buni chiqurroq keyinroq qarab chiqamiz.

### Window object.

BOM yani browserimiz bilan qilinishi mumkin bo'lgan ishlardir. Window esa bizning hozirgi ochiq oynamiz, yoki dokumentimiz desak ham bo'ladi.

**moveBy(dx, dy), moveTo(x, y), resizeBy(dw, dh), resizeTo(x, y)**

bu methodlar window object i bilan birga ishlatiladi.

Misol:

```
<script language="javascript">  
function katalik() {  
window.resizeBy(10, 10);  
}  
function yurgiz() {  
window.moveBy(10, 10);  
}
```

## JavaScript Hammaga

```
</script>
</head>
<body>
<input type="button" value="Kattalik" onclick="katalik()">
<br>
<input type="button" value="Yurgiz" onclick="yurgiz()">
</body>
</html>
```

Bu yerda biz ikta funktsiyadan foydalandik bittasi kattalashtiradi bittasi esa yurgizadi. Ho'sh kichiklashtirmoqchi bo'lsangiz unda – sonidan foydalaning.

Bazi bir methodlar va funktsiyalar browserning turiga bog'liqdir, va istagan ishingizni qilishingiz uchun browseringiz nima ekanligini aniqlashingiz kerak, shuning uchun azgina oldinga sakrab kirchik bir object va uni methodi bilan tanishtirib ketaman.

### **navigator.appName**

Bu sizda qanday browser ishlayotgan bo'lsa shuni nomini qaytaruvchi method'dir.

Misol:

```
<html>
<head>
<title>Dars7</title>
<script language="javascript">
function nima() {
alert(navigator.appName);
}
</script>
</head>
<body>
<input type="button" value="Browserim" onclick="nima()">

</body>
</html>
```

bu yerda sizga browseringiz Internet Explorer bo'lsa “Microsoft Internet Explorer” deb qaytaradi agarda Mozilla yoki Netscape bo'lsa unda “Netscape” deb qaytaradi. Shunga qarab siz ko'dingiz o'zgartirishni mumkindir.

Ho'sh agarda biz oynamizni kattaligiga qarab turib kattalashtirishni yoki kichiklashtirishni qilmoqchi bo'lsakchi? Unda quyidagi method'lar yordam beradi.

Internet Explorer:

**window.screenLeft, window.screenTop (bular qayerda browser turganligi),  
document.body.offsetWidth, document.body.offsetHeight(kattaligi faqatgina html  
oynasigina holos)**

Mozilla:

**window.screenX, window.screenY (oynamiz joylashgan joy)  
window.innerWidth, window.innerHeight (oynamizning html bo'lgan joyi kattaligi)  
window.outerWidth, window.outerHeight (butun oynamizning kattaligi)**

Mana yuqorida ko'rganingizdek ikta browser uchun bir nechti hil methodlar.

1. O'zingiz ishlang: Shunday script yozingki, bu script browserga qarab turib oynamizni kattaligini aniqlasin va kattaligi agarda 400\*300 dan kichik bo'lsa unda 100\*100 ga kattalashtirsin.

## JavaScript Hammaga

Echimi:

```
<html>
<head>
<title>Dars7</title>
<script language="javascript">
function Kichik() {
if (navigator.appName=='Microsoft Internet Explorer'){
    if (document.body.offsetHeight<300 && document.body.offsetWidth<400) {
        window.resizeBy(100, 100); }
    }
else {
    if (window.innerWidth<400 && window.innerHeight<300) {
        window.resizeBy(100, 100);
    }
}
}
</script>
</head>
<body>
<input type="button" value="Browserim" onclick="Kichik()">

</body>
</html>
```

Ko'rib turganingizdek bu juda ham oson.

### **window.open()**

Bu methodimiz bizga yangi oyna ochish uchun kerakdir.

window.open(url, window\_name, window\_features);

yani window.open('test1.html', 'Meni oynam');

Misol:

```
function Yangi() {
window.open('http://www.uzinfo.info', 'Meni oynam');
}
```

bu yerda biz berilgan url'ni yangi oynada ochdik.

Ho'sh agarda siz shu oynada ochmoqchi bo'lsangizchi?

Unda quyidagi qo'shimchalarni window\_name o'rniga yozasiz:

\_self = shu oynada ochish

\_blank = bo'sh oyna ochish

\_parent = agarda frame ishlatilgan unda asos frameda ochadi

\_top = eng yuqoridagi frame'da ochadi.

Ho'sh yana bir qo'shimcha qo'shishingiz mumkindir yani:

window.open('http://www.uzinfo.info', 'Meni Oynam', 'nimaladir');

nimaladir quyidagilar bo'lishi mumkindir:

Nomi	qiymati	Tarifi
left	Number	Oynaning chap qismining joyi
top	Number	Oynaning yuqori qismining joyi
height	Number	Oynaning bo'yi (eng kam qiymat 100)
width	Number	Oynaning eni (eng kam qiymat 100)
resizable	yes,no	Oynamizni kattaligi o'zgartirila olinish olinmasligini belgilaydi (joriy NO)
scrollable	yes,no	Oynamizda scrollbar bo'lish bo'lmasligi (joriy NO)
toolbar	yes,no	Toolbar bor yoki yo'qligini belgilab beradi (joriy NO)

## JavaScript Hammaga

status	yes,no	Statusbar bor yoki yo'qligi (joriy NO)
location	yes,no	Location bar bor yoki yo'qligi (joriy NO).

Keling misolda ko'rib chiqamiz.

Misol:

```
<html>
<head>
<title>Ikkinchi Oyna</title>
<script language="javascript">
function Yangi() {
window.open('http://www.uzinfo.info', 'Oyna', 'left=100, top=100, height=600, width=400,
status=yes');
}
</script>
</head>
<body>
<input type="button" value="Yubor" onclick="Yangi()">
</body>
</html>
```

Ho'sh qanday qilib biz ochilgan oynani ochgan oynamiz orqali boshqarishimiz mumkinligini ko'rib chiqamiz.

Quyidagi misolga qarang:

### **test1.htm**

```
<html>
<head>
<title>Birinchi oyna</title>
<script language="javascript">
var fOyna;
function Yangi() {
fOyna=window.open('./test2.htm', 'Oyna', 'left=100, top=100, height=600, width=400,
status=yes');
}
function Yurgiz() {
fOyna.moveBy(100, 100);
fOyna.resizeBy(100, 100);
}
</script>
</head>
<body>
<input type="button" value="Och" onclick="Yangi()">
<input type="button" value="Yurgiz" onclick="Yurgiz()">
</body>
</html>
```

### **test2.htm**

```
<html>
<head>
<title>Ikkinchi Oynamiz</title>
</head>
<body>
<h1>Bu ikkinchi oynamiz</h1>
```

## JavaScript Hammaga

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

Mana ko'rganingizdek bu juda ham osondir. Huddi shunday siz birinchi oynadan ikkinchi oynaga ma'lumot yuborishingiz ham mumkindir.

### **close()**

Bu method oyna bilan birga ishlatiladi, va bu aktiv oyna yoki ko'rsatilgan oynani yopish uchun ishlatiladi.

`window.close();` //bilan ochiq oynani yopishingiz mumkindir.

### **focus()**

Bu method'imiz asosiy yoki unga tegishli bo'lgan oynalarni aktivlashtirish yan focus berish uchun ishlatiladi.

`window.focus();`

Misol:

#### **test1.htm**

```
<html>  
<head>  
<title>Birinchi oyna</title>  
<script language="javascript">  
var fOyna;  
function Yangi() {  
fOyna=window.open('./test2.htm', 'Oyna', 'left=100, top=100, height=600, width=400,  
status=yes');  
}  
function Focus_ber() {  
fOyna.focus();  
}  
  
</script>  
</head>  
<body>  
<input type="button" value="Och" onclick="Yangi()">  
<input type="button" value="Focus" onclick="Focus_ber()">  
</body>  
</html>
```

#### **test2.htm**

```
<html>  
<head>  
<title>Ikkinchi Oynamiz</title>  
</head>  
<body>  
<h1>Bu ikkinchi oynamiz</h1>  
  
</body>  
</html>
```



## JavaScript Hammaga

### Dars8 Dialog Box'lar

Bu alert kabi ochiladigan oynalardir. Buning ichiga alert, confirm, prompt dialog oynlari kiradi. alert()'ni biz yuqorida ko'rib chiqqanmiz.

#### **confirm()**

Bu oynamiz biz Yes/No (OK/Cancel) oynasizidir.

Misol:

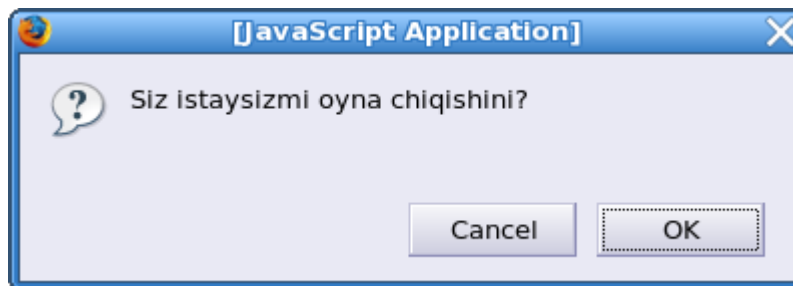
```
<html>
<head>
<title>Confirm</title>
<script language="javascript">
```

```
function Ist() {
if(confirm('Siz istaysizmi oyna chiqishini?')) {alert('Siz istadingiz va mana oynacha sizga');}
else {alert('Siz istamadingiz va mana oynacha baribir sizga');}
}
```

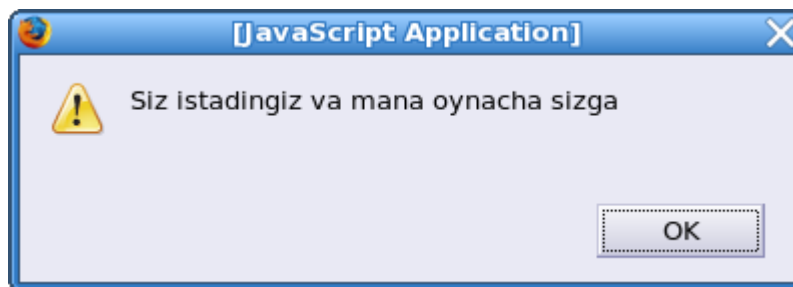
```
</script>
</head>
<body>
<input type="button" value="Istaysizmi?" onclick="Ist()">

</body>
</html>
```

Shunda oynangizga



chiqadi, agarda siz OK ni bossangiz unda



Quyidagi oyna chiqadi. Ko'rganingizdek bu juda ham qulay va kerakli oynadir.

Agarda siz tashqaridan qandaydir ma'lumot o'qimoqchi bo'lsangizchi? Unda nimani ishlatish kerak? Sizga yordamga prompt() oynasi keladi.

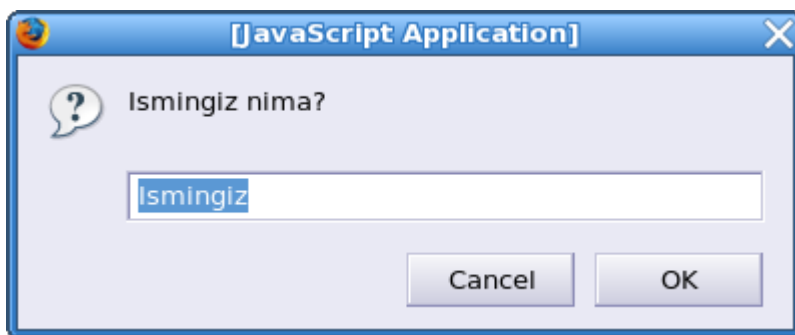
#### **prompt()**

```
var cName=prompt('Ismingiz nima?', 'Ismingiz');
```

## JavaScript Hammaga

```
alert ('Salom '+cName + ' hush kelibsiz!');
```

oynangizda



shu oynani ko'rishingiz mumkindir.

### Status Bar

Agar bilsangiz Internet Browser tagida ma'lumot ko'rsatib turuvchi joy status bar deb nomlanadi. Va siz javascript'da shu joyga istagan ma'lumotingizni yozishingiz mumkindir.

Buning uchun window.status va window.defaultStatus hususiyatlaridan foydalanasiz.

### window.defaultStatus

Bu sahifangiz ochilganda har doim yani joriy holatda ko'rsatuvchi ma'lumotdir.

Misol:

```
window.defaultStatus="Salom bu meni sakkizinchi darsim";
```

### window.status

Qandaydir vaqt uchun o'zgartirish.

```
<html>
<head>
<title>Status</title>
</head>
<body>
<a href="test2.htm" onmouseover="javascript:window.status='Salom bu meni sakkizinchi
darsim'">StatusBar</a>
</body>
</html>
```

Bu yerda biz statusbarga yozmoqdamiz. Shuni esdan chiqarmangki bazi bir Internet browser'lar buni ko'rsatmasligi mumkin, sababi ularni o'zi u yerga o'zining ma'lumotini yozib turadi.

### Vaqt kuttirish

Deylik sizning scriptingizda qandaydir funktsiya yoki qandaydir ish 3-4 sekundan keyin yoki qandaydir vaqtdan keyin ishlashi keraki, yoki qandaydir vaqt davomida qaytarilib turishi kerak. Unda **setTimeout()** kerak bo'ladi.

Bunda siz ikta narsani ko'rsatishingiz kerak yani:

```
setTimeout(funktsiya_nomi/ish, kutish_vaqti);
```

Misol:

```
<script language='javascript'>
var aID=setTimeout("alert('sinch')", 2000);
</script>
```

## JavaScript Hammaga

Bu yerda alert('sinch') ikki sekunddan keyin ishlaydi sababi 2000 bu millisecondlarda berilgandir.

### clearTimeout()

Timeout qo'yganimizda bizga timeout'nin ID'sini qaytaradi yani bu huddi prosess kabi o'ziga raqam oladi va shu raqam orqali siz unga murojat qilishingiz mumkin. Yuqoridagi misolda aID ga shu raqam berildi. ClearTimeout esa qo'yilgan timeout'ni olib tashlash uchun kerakdir. Nimaga qoyamiz keyin ob tashashimiz kerak? Deylik qandaydir holatda ish bajarilmasiligi kerak, boshqa holatda esa bajarilishi kerak.

Ishlatish: clearTimeout(aID);

Misol:

```
<html>
<head>
<title>Timeout_clear</title>
</head>
<body>
<script language='javascript'>
var ald=setTimeout("alert('Mana')", 3000);
if (!confirm('Timeoutishlatasizmi?')){fMy(true)}
```

```
function fMy(t1) {
clearTimeout(ald);
}
</script>
</body>
</html>
```

Bu yerda sizdan so'raydi agar ishlatmoqchi bo'lsangiz unda ishlatadi agarda istamasangiz ishlaytmaydi.

1. O'zingiz ishlang: agarda onmouseover <a href> da 10 soniya ichida bo'lmasa unda alert orqali buni bizga etsin va agarda bo'lsa onmouseover unda yana boshidan 10 soniya sanash boshlansin.

Echimi:

```
<html>
<head>
<title>Timeout_clear</title>
</head>
<body>
<script language='javascript'>
var ald;
function myFunc() {
clearTimeout(ald);
ald=setTimeout("alert('10 soniya otdi')", 10000);
}
myFunc();
</script>
<a href="#" onmouseover="myFunc()">Shu yerdan 10 soniya ichida o't</a>
</body>
</html>
```

Hozir biz bir marotaba qanadaydir vaqtdan keyin ishlatishni ko'rdik, agar bizga har 5 sekunda ishlashi kerak bo'lsachi? Unda **setInterval()** va **clearInterval()** ishlatishimiz kerak.

## JavaScript Hammaga

Bular ham setTimeout() va clearTimeout() kabi ishlaydi.

Misol:

```
<script language='javascript'>
var ald=setInterval("alert('salom')", 4000);
</script>
```

### History

Sizning saytingizga kirishdan oldin ziyoratchi bir qancha boshqa saytlardan kelgan bo'lishi mumkin, yoki deylik sizning sahifalaringizni o'zida aylangan, va siz **back** tugmasi orqali bir oldingi sahifaga yubormoqchisiz, nima qilasiz? Browser'ingizni history xususiyatidan foydalanasiz.

#### window.history()

Agarda bitta orqaga yubormoqchi bo'lsangiz unda window.history(-1) oldinga (forward) bitta yubormoqchi bo'lsangiz unda window.history(1) buyrug'ini ishlatishingiz kifoyadir. Misollarda o'zingiz ko'rib chiqing.

Agarda history'ida qancha sahifa borligini ko'rmoqchi bo'lsangiz unda window.history.length buyrug'idan foydalaning.

Misol:

```
<script language='javascript'>
alert(window.history.length);
</script>
```

### Document

Bu object'imiz ham DOM va BOM ga kiradi. Bu object'imiz ochilgan sahifamizning xususiyatlarini o'zgartirish yoki qayta berish uchun kerak bo'ladi. BOMda quyidagi xususiyatlari bor:

alinkColor	Aktive linklar uchundir yani: <body alink="color">*
bgColor	Oynamizning orqa fon rangi: <body bgcolor="color">*
fgColor	Sahifada tekstning yozilish rangi: <body text="color">*
lastModified	Sahifa qachon ohirgi marotaba o'zgartirilgani
linkColor	Link'larning default rangi: <body link="color">*
referrer	Bir oldingi sahifa URL adresi (history(-1))
title	Sahifaning: <title/> tagi ichidagi tekst
URL	Ochiq turgan URL adresi
vlinkColor	Kirilgan Link'lar rangi: <body vlink="color">*

Bu yerda ba'zi bir xususiyatlarga yozishingiz ham mumkin, bazilar faqat o'qish uchundir.

Misol:

```
<html>
<head>
<title>History</title>
</head>
<body>
<script language='javascript'>
window.document.title="Tester";
</script>
</body>
</html>
```

Mana bu yerda boshida History edi title keyin esa Tester'ga aylandi.

Misol2:

```
<html>
<head>
<title>History</title>
```

## JavaScript Hammaga

```
</head>
<body>
<h1>Mana bu meni boshidagi tekstim</h1>
<script language='javascript'>
alert(window.document.lastModified);
alert(window.document.bgColor);
window.document.bgColor="red";
alert(window.document.fgColor);
window.document.fgColor="yellow";
</script>
<h1>Bu esa scriptdan keyingi tekstim</h1>
</body>
</html>
```

Misolni ko'rib chiqing va tahlil qiling.

Window so'zini ishlatmasligingiz ham mumkin, sababi hozirgi ochiq sahifa uchun document.xxx deb ketishingiz ham mumkindir.

### **Bulardan tashqari DOM uchun ham bir qancha document'ning hususiyatlari bordir:**

anchors	Sahifadagi anchorlar, (<a href="anchors"></a>)
applets	Sahifadagi appletlar
embeds	<embed> taglari
forms	Sahifadagi formalar
images	<img src> taglari yani sahifadagi rasmlardir
links	Sahifadagi linklar yani: <a href="somewhere.htm"></a>

Misol:

```
<html>
<head>
<title>Link</title>
</head>
<body>
<a href="/test3.htm"> test3 </a><br>
<a href="http://sinch.narzikulov.com"> UzInfo </a>
<script language='javascript'>
alert(document.links[0]);
document.links[0].href="/test2.htm";
alert(document.links[0]);
</script>
</body>
</html>
```

Bu yerda biz boshida ikta link yaratdik, songra esa ularni birinchisini alert bilan ko'rsatdik, undan so'ng esa shu link'ga document.links[0] orqali "href" iga yangi qiymat yozdik. Va oynaga yana uni qiymatini chiqardik.

Siz qolganlari bilan ham huddi shunday ishlashingiz mumkindir.

Qolgan hususiyat va methodlar bilan o'zingiz mustaqil ishlab ko'ring va tekshirib chiqing.

BOMda document'ning yana bitta methodi bordir, bu **write** va **writeln**dir yani html document'imizga nimanidir yozdirishdir. Bir biridan farqi writeln qatorma qator yozadi va qator ohiriga /n belgisini qo'yib ketadi.

## JavaScript Hammaga

Ho'sh nimaga kerak diyishingiz mumkin? Agarda siz DHTML yani Dynamic HTML yozmoqchi bo'lsangiz va ziyoratchini qilgan ishlariga qarab sahifani o'zgartirmoqchi bo'lsangiz unda bu method sizga kerak bo'lishi aniqdir.

Keling misollarda ko'ramiz.

Misol:

```
<html>
<head>
<title>write</title>
</head>
<body>
JavaScript Oson, <br>
va Juda qulay! <br>
<script language='javascript'>
var a=prompt("Ismingiz nima?");
document.write("<h1>Salom "+a+" sahifamga hush kelibsiz!</h1>");
</script>
</body>
</html>
```

Bu misolimizda biz ziyoratchidan ismini so'radik va uni ismini sahifaga yozib kutib oldik!

Ho'sh agarda siz o'zingiz turgan sahifani yangilamoqchi bo'lsangiz unda nima qilishingiz kerak?

```
<html>
<head>
<title>write</title>
<script language='javascript'>
function Yangilash(){
var a=prompt("Ismingiz nima?");
var a1=document.open();
a1.write("<html><head><title>Writing</title></head><body><h1>Salom "+a+" sahifamga hush
kelibsiz!</h1></body></html>");
a1.close();
}
</script>
</head>
<body>
JavaScript Oson, <br>
va Juda qulay! <br>
<input type=button onclick="Yangilash()" value="Yangila">
</body>
</html>
```

Yuqorida biz qanday turgan oynani boshidan yangilashni ko'rdik, Nimaga tugma orqali qildik? Sababi agarda body ichiga scriptni joylashtirganimizda oyna yangilanmasdan faqatgina ma'lumot qo'shilardi holos.

Keling qanday qilib yangi oyna ochib unga ma'lumot yozishni ko'ramiz.

Misol:

```
<html>
<head>
<title>write</title>
<script language='javascript'>
function Yangilash(){
var a1=window.open("about:blank", "yangi Oyna");
```

## JavaScript Hammaga

```
a1.document.write("<html><head><title>Writing</title></head><body><h1>Salom bu yangi  
oyna</h1></body></html>");  
a1.document.close();  
}  
</script>  
</head>  
<body>  
JavaScript Oson, <br>  
va Juda qulay! <br>  
<input type=button onclick="Yangilash()" value="Yangila Oyna">  
</body>  
</html>
```

Tahlil qiling!

### Dars 9 Location

Bu BOMning yana bir object'idir. Location hozir turgan sahifaning adresi bilan ishlaydigan objectdir. Uni quyidagi hususiyatlari bordir.

**hash** Agarda URL'da # belgisi bo'lsa shu belgi va undan keyingi qismini qaytaradi (misol: `http://www.somewhere.com/index#section1` hash="#section1").

**host** Serverning nomi (misol, `www.somewhere.com`)

**hostname** Hostga teng, farqi `www` so'zini olib tashlaydi.

**href** Hozir yuklangan sahifani to'la adresi

**pathname** host'dan keyingi qismi (`http://www.somewhere.com/pictures/index.htm` `pathname="/pictures/index.htm"`).

**port** Serverning http porti.

**protocol** `"/"` belgidan oldingi qism, yani qaysi protocolda ishlayotgani

**search** `"?"` belgisidan keyin kelgan qism

Misol:

```
<script language='javascript'>  
alert(location.href);  
</script>
```

Agarda sahifa adresini o'zgartirmoqchi bo'lsak unda nima qilish kerak?

**location.assign()** va **location.replace()** methodlaridan foydalanishimiz kerak.

Misol:

```
<html>  
<head>  
<title>Location</title>  
</head>  
<body>  
Meni sahifam  
<script language='javascript'>  
function change_me(){  
location.replace("http://www.sinch.narzikulov.com");  
}  
setTimeout(change_me, 5000)  
</script>  
</body>
```

## JavaScript Hammaga

</html>

Bu yerda biz replace orqali sahifamizni 5 soniyadan keyin o'zgartirdik. Iltisangiz location.href="nimadir" orqali ham o'zgartirishingiz mumkindir.

1. O'zingiz yeching: Shunday script yozingki uchta tugma bo'sin shu uchta tugmadan uktasi ikki hil URLni shu oynada ochsin va uchunchi tugma sizdan qaysi urlga boramiz deb so'rasin va o'sha url'ga borsin!

Echim:

```
<html>
<head>
<title>Location</title>
<script language='javascript'>
function change_me(nomer){
switch(nomer) {
case 1: location.replace("http://www.sinch.narzikulov.com");
break;
case 2: location.replace("http://uzinfo.info");
break;
case 3: {
var url3=prompt("URL adresini kiriting");
location.replace(url3);
}
}
}
}
</script>
</head>
<body>
<input type="button" value="birinchi" onclick="change_me(1)"> <br>
<input type="button" value="Ikkinchi" onclick="change_me(2)"> <br>
<input type="button" value="Uchinchi" onclick="change_me(3)"> <br>
</body>
</html>
```

### Location.reload()

Bu hozirgi sahifani boshidan yuklash uchun kerakdir, va u ikta qiymat ola oladi true va false, true bersak unda sahifamiz Server'dan yangilanadi, agarda false berak unda cache'dan yuklanadi!

Misol:

```
location.reload(true);
```

Serverdan yangilandi. Bu nimaga kerak? Agarda sizning sahifangiz qanadaydir vaqtda o'zgarib turishi kerak bo'lsa unda reload() ni ishlatib yangilab yangi ma'lumotlarni olishingiz mumkindir!

### Navigator Object

Bu object'imiz web browser haqida ma'lumot saqlaydi. Agarda siz web browser'ga qarab javascriptingiz ishlashini iltisangiz unda marhamat qilib navigator object'idan foydalaning. Quyida navigator objectining asosiy (IE va Mozilla) xususiyatlarini ko'rsatib va tushuntirib o'taman.

**appName**            Browserning kodoviy nomini beradi

**appName**            Browserning ofitsialniy nomini beradi



## JavaScript Hammaga

<b>appVersion</b>	Browser versiyasini beradi
<b>cookieEnabled</b>	Cookie yoqiq yoki o'chiqligini beradi
<b>javaEnabled()</b>	Java yoqiq yoki o'chiqligini beradi
<b>platform</b>	Browser ishlayotgan OS'ni nomini beradi
<b>plugins</b>	Qanday pluginlar borligini ko'rsatadi
<b>taintEnabled()</b>	Agarda taint' yoqqoq bo'lsa unda boshqa web serverlar sizning sahifangizning hususiyatlarini o'qiy oladilar, va siz qaysilari o'qilishi mumkin bo'lsa shularni ochiq qoldirib qolganlarini yopishingiz mumkindir.
<b>userAgent</b>	Ziyoratchining web browserining header'larini o'qiy olasiz. Yani web browser nomi, Osnomi va boshqa kerakli ma'lumotlarni qaytaradi.

Misol:

```
<html>
<head>
<title>Navigator</title>
<script language='javascript'>
function show_me(){
alert(navigator.userAgent);
}

</script>
</head>
<body>
<input type="button" value="userAgent" onclick="show_me()"> <br>
</body>
</html>
```

Oyangizga quyidagi chiqadi:



Ko'rib turganingizdek qaday operatsiyon sistema va qanday browser, qanday klaviatura va hkz ma'lumotlarini ko'rsatdi.

1. O'zingoiz yozing: Qanday operatsion sistema ishlatilayotganini ko'rsatuvchi (windows, linux, mac) script yozing.

Echimi:

O'zingiz ishleng!

## JavaScript Hammaga

2. O'zingiz yozing: Yuqoridagi hamma hususiyatlarga misol yozib chiqing!

### Screen object

Bu object'imiz ziyoratchi kompyuteri haqida yana azgina ma'lumot beradi, ammo havfsizlik bo'lishi uchun bu yerda siz oz ma'lumot olasiz.

Hususiyatlaril:

**availHeight** — Ishlatish mumkin bo'lgan, windows oynasining bo'yining uzunligi (pixel)

**availWidth** — Ishlatish mumkin bo'lgan, windows oynasining enining uzunligi (pixel)

**colorDepth** — qancha bit rangni berish uchun ishlatiladi. Asosan 32dir.

**height** — Bo'yining uzunligi (pixel)

**width** — Ening uzunligi (pixel)

Misol: Biz bu misolda ekranimizda qancha joy bo'lsa shunchaga web browser'ni kattalashtirishimiz kerak.

```
<html>
<head>
<title>Screen</title>
<script language='javascript'>
function katta(){
window.resizeTo(screen.availWidth, screen.availHeight);
}

</script>
</head>
<body>
<input type="button" value="userAgent" onclick="katta()"> <br>
</body>
</html>
```

3. O'zingiz yozing: 5 ta tugma bo'lsa shu tugmalar orqali siz sahifani orqa fon rangini o'zgartiring.

```
<html>
<head>
<title>Back</title>
<script language="JavaScript">

function backs(color) {
document.bgColor = color
}
</script>
</head>
<body>
<form>
<input type="button" value="Red" onclick="backs('red')">
<input type="button" value="Yellow" onclick="backs('yellow')">
<input type="button" value="Blue" onclick="backs('blue')">
<input type="button" value="White" onclick="backs('white')">
<input type="button" value="Gray" onclick="backs('gray')">
</form>
</body>
</html>
```

4. O'zingiz yeching: Sahifaga ikki marta sichqoncha bosilganda rang o'zgarsin, yani randomli!

## JavaScript Hammaga

Echimi:

```
<html>
<head>
<title>Back</title>
<script>
function makeRandom(range) {
    rand=Math.floor(range*Math.random())
    return rand
}

function makeHexa(thiscol) {
    var colhex_left = Math.floor(thiscol/16)
    var colhex_right= thiscol-(colhex_left*16)
    if (colhex_left == 10) {colhex_left="A"}
    if (colhex_left == 11) {colhex_left="B"}
    if (colhex_left == 12) {colhex_left="C"}
    if (colhex_left == 13) {colhex_left="D"}
    if (colhex_left == 14) {colhex_left="E"}
    if (colhex_left == 15) {colhex_left="F"}
    if (colhex_right == 10) {colhex_right="A"}
    if (colhex_right == 11) {colhex_right="B"}
    if (colhex_right == 12) {colhex_right="C"}
    if (colhex_right == 13) {colhex_right="D"}
    if (colhex_right == 14) {colhex_right="E"}
    if (colhex_right == 15) {colhex_right="F"}
    var colhex = ""+colhex_left+colhex_right
    return colhex
}

function dblclick() {
    var redcolor=makeRandom(255)
    redcolor=makeHexa(redcolor)
    var greencolor=makeRandom(255)
    greencolor=makeHexa(greencolor)
    var bluecolor=makeRandom(255)
    bluecolor=makeHexa(bluecolor)
    document.bgColor="" + redcolor+greencolor+bluecolor
}

</SCRIPT>

</head>
<body ondblclick="dblclick()">
</body>
</html>
```

5. O'zingiz yozing: Shunday script yozingki, sahifaga kirganda, vaqtga qarab ertalabmi kechki paytmi shunga qarab siz bilan salomlashsin!

Echimi:

```
<html>
<head>
<title>Vaqt</title>
<script language="JavaScript" type="text/javascript">
```

## JavaScript Hammaga

```
today = new Date();
year = (today.getFullYear() ? today.getFullYear() : today.getYear());
month = today.getMonth();
switch (month) {
    case 0 : month = "Yan"; break;
    case 1 : month = "Fev"; break;
    case 2 : month = "Mar"; break;
    case 3 : month = "Apr"; break;
    case 4 : month = "May"; break;
    case 5 : month = "Iyun"; break;
    case 6 : month = "Iyul"; break;
    case 7 : month = "Avg"; break;
    case 8 : month = "Sen"; break;
    case 9 : month = "Oct"; break;
    case 10 : month = "Noy"; break;
    case 11 : month = "Dek"; break;
}
date = today.getDate();
day = today.getDay();
switch (day) {
    case 0 : day = "Yakshanba"; break;
    case 1 : day = "Dushanba"; break;
    case 2 : day = "Seshanba"; break;
    case 3 : day = "Chorshanba"; break;
    case 4 : day = "Payshanba"; break;
    case 5 : day = "Juma"; break;
    case 6 : day = "Shanba"; break;
}
hours = today.getHours();
if (hours<12) greeting = 'Ertalabki salomdan bebahra qolmang!!';
    if (hours<18 && hours>11) greeting = 'Ertalabki salomdan bebahra qolmang!';
    if (hours<20 && hours>17) greeting = 'Kechki salomdan bebahra qolmang!';
    if (hours>19) greeting = 'Tungi salomdan bebahra qolmang!';
hour = (hours>12) ? (hours-12) : hours;
halfDay = (hours>11 && hours<24) ? 'PM' : 'AM';
minutes = today.getMinutes();
dayStr = day+ ', ' +date+ '&nbsp;' +month+ ',&nbsp;' +year;
timeStr = (minutes<10) ? hour+ ':0' +minutes+ ' ' +halfDay : hour+ ':' +minutes+ ' ' +halfDay;
</script>

</head>
<body >
<script language="JavaScript" type="text/javascript">
document.writeln('Bugun ' +dayStr+ ' .<br>Hozir ' +timeStr+ ' .<br>');document.write(greeting+
'<br>');
</script>
</body>
</html>
```

6. O'zingiz yozing: Shunday script yozingki vaqtini ko'rsatib tursin, yani har bir o'tayotgan sekundlarni!

Echimi:  
<html>

## JavaScript Hammaga

```
<head>
<title>Vaqt</title>
<script Language="JavaScript">
function tickClock()
{
    // this tells the browser to call the "tickClock()" function every 1 second
    setTimeout("tickClock()", 1000);

    thisTime = new Date()
    hours = thisTime.getHours()
    minutes = thisTime.getMinutes()
    seconds = thisTime.getSeconds()

    // convert the digit to string
    // make sure there're at least 2 digits
    if (hours>=12)
    {
        hours-=12
    }

    hourString="0"+hours+""
    minuteString="0"+minutes+""
    secondString="0"+seconds+""

    // format the hour
    hour1=parseInt(hourString.charAt(hourString.length-1))
    hour2=parseInt(hourString.charAt(hourString.length-2))

    // format the minute
    minute1=parseInt(minuteString.charAt(minuteString.length-1))
    minute2=parseInt(minuteString.charAt(minuteString.length-2))

    // format the seconds
    second1=parseInt(secondString.charAt(secondString.length-1))
    second2=parseInt(secondString.charAt(secondString.length-2))

    document.clockForm.clock.value="" + hour2+hour1+":"+minute2+minute1+":"+second2+second1
}
</script>

</head>
<body onLoad="tickClock()">
<form name="clockForm">
    <p><input type="text" name="clock" size="8"> </p>
</form>
</body>
</html>
```

Mana bir qancha kerak bo'lishi mumkin bo'lgan scriptlarni ko'rib chiqdik va siz oldin o'tgan hamma mavzularni qaytarib chiqdingiz degan umiddaman.

## JavaScript Hammaga

### Dars 10 DOM

#### Nodes

Node'lar bu agardasiz body'ni oladigan bo'lsangiz unda body bu node, uni ichidagi boshqa tag'lar misol uchun <img> <a> va boshqalar bular uni child (yani bolalari) dir. Biz node'ni yaqol ko'rsatish bilan uni ichidagi yoki o'zini o'zgaritishimiz, ularga yozishimiz mumkindir. Misollarda yana ham aniq ko'rasiz va tushunib olasiz.

Hususiatlari va method'lari:

Hususiat	Tarifi
nodeName	node'ning nomi
nodeValue	agar bo'lsa unda uni qiymati
nodeType	Uni turi
ownerDocument	node'imiz tegishli bo'lgan document
firstChild	Node'ning ichidagi taglardan birinchisi
lastChild	Eng ohirgi tagi
childNodes	Node'ning ichidagi hamma bolalari
previousSibling	Node'ning tegishli bo'lgan oilasidan birinchisi
nextSibling	Hozirgi Node'dan keyingi node oilasidagi element
hasChildNodes	Agar Node'ning ichida bolalari bo'lsa True bo'lmasa False bo'ladi
attributes	Node'ning ichidagi elementlari(bolalari) dan tanlanganini hususiatlarini beradi
appendChild	Node'imizning eng pasiga yana bir element qo'shish
removeChild	Qandaydir elementni (bolasini) Node'dan olib tashlash
replaceChild (newnode, oldnode)	Eski elementni yangisiga almashtirish
insertBefore(newnode, refnode)	Qandaydir renode elemetdan oldingi joyga yangi element qo'shadi

Keling misolda ko'rib chiqaylik.

```
<html>
<head>
<title>Node</title>
</head>
<body>
<script type="text/javascript">
var oNode = document.documentElement;
var oHTML = oNode.firstChild;
alert (oHTML);
</script>
```

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

Va oynangizga quyidagi chiqadi. Bu degani, Node'ning birinchi bolasi bu HEAD dir, oynada yozganidek bu obyektidir va HTMLHeadElement dir.

agar:  
<html>

## JavaScript Hammaga

```
<head>
<title>Misol </title>
</head>
<body>
<script type="text/javascript">
var oNode = document.documentElement;
var oHTML = oNode.firstChild;
alert (oNode.childNodes.length);
</script>

</body>
</html>
```

Bu misolda bizga 2 ko'rsatadi, sababi bizda documentimizda ikta element bor bular head va body'dir.

Node' turlari yani type'ni bilish uchun nodeType ni ishlatish.

Misol:

```
<script type="text/javascript">
alert (document.nodeType);
</script>
```

Bu yerda biz document ning node turini ko'ramiz yani bu 9 soniga tengdir, agar,

```
<script type="text/javascript">
alert (document.documentElement.nodeType);
</script>
```

Bu yerda esa 1 soni chiqadi, chunki bu node turi yani elementniki 1 dir. mana bu yerda hamma node turlari.

ELEMENT\_NODE: 1  
ATTRIBUTE\_NODE: 2,  
TEXT\_NODE: 3,  
CDATA\_SECTION\_NODE: 4,  
ENTITY\_REFERENCE\_NODE: 5,  
ENTITY\_NODE: 6,  
PROCESSING\_INSTRUCTION\_NODE: 7,  
COMMENT\_NODE: 8,  
DOCUMENT\_NODE: 9,  
DOCUMENT\_TYPE\_NODE: 10,  
DOCUMENT\_FRAGMENT\_NODE: 11,  
NOTATION\_NODE: 12

Mana yuqoridagi listdan ko'rishingiz mumkinki bu yerda documentning node'i 9 a uni elementini node'i 1 va hkz.

Yana shular bilan birga siz attributelar bilan ham ishlashingiz kerak. Buning uchun quyidagi qo'shimcha methodlar sizga yordam beradi.

**getItem(name)** – Bu yerda nodeni bizga beradi, "name" to'g'ri kelgan

**removeNamedItem(name)** – Bu esa "name" li nodeni o'chirib tashlash

**setNamedItem(node)** – Node'ni spiskaga qo'shadi

**item(pos)** - "pos" da joylashgan nodeni beradi.

Keling misolda ko'raylik, ancha tushunarli bo'ladi.

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

## JavaScript Hammaga

```
<title>Misol </title>
</head>
<body id="test">
<p id="d1"> Hello Misha </p>
<script type="text/javascript">
var oP = document.getElementsByTagName("p");
var iP = oP[0].attributes.getNamedItem("id").nodeValue;
alert (iP);
</script>
</body>
</html>
```

Bu misolda biz oP degan obyekt hosil qildik bu BODY'ning ichidagi hamma P largadir, agar bizaga birinchi P ning ID'si kerak bo'lsa unda yuqoridagicha nodeValue'ni olishimiz mumkindir. Huddi shunday qilib siz setNamedItem orqali ID ni o'zgartishingiz mumkindir.

Yuqoridagiga o'shagan yana bir qancha hususiat va/yoki method lar ham bor bular;

- 1. getAttribute(name)**
- 2. setAttribute(name, newname)**
- 3. removeAttribute(name)**

Misol:

```
<script type="text/javascript">
var oP = document.getElementsByTagName("p");
var iP = oP[0].getAttribute("id");
alert (iP);
</script>
```

Bu yerda siz yuqoridagi ishni tezroq va qulayroq yo'l orqali bajardingiz.

Biz yuqoridagi ikkta misolda getElementsByTagName komandasini ishlatdik. Bu komanda BODY ning ichidagi istalgan taglarga ishlatilishi mumkin va uni o'zgartish uchun ishlatish mumkindir.

Misol:

```
<html>
<head>
<title> Misol </title>
<script type="text/javascript">
function stat_time(){
var oName = document.getElementsByTagName("input");
oName[0].setAttribute("value", "Test");
}
</script>

</head>
<body>
<form name="my_frm">
<input name="text1" type="text" value="12">
<input name="text2" type="text">
<a href="#" onclick="stat_time()">Boshqa sahifa</a>
</form>
</body>
</html>
```

Mana ko'rganingizdek bu ham juda oson va qulaydir.



## JavaScript Hammaga

### Dars 11

Ho'sh agarda siz qandaydir element (<p>, <img>, ...) yaratmoqchi bo'lsangizchi? Unda quyidagilar sizga yordam beradi.

Operator	Tarifi
CreateAttribute(name)	Documentimizda «name» li attribute yaratadi
CreateComment(text)	«text» li komentariya joyini yozadi
CreateDocumentFragment()	Document fragmentini yozadi
CreateElement(tag)	«tag» li element yaratadi
CreateTextNode(text)	«text» lik faqat tekst ma'lumot yozadi.

Keling misolda ko'raylik, shunda yana ham tushunarli bo'ladi.

Misol:

```
<html>
<head>
<title> Misol </title>
<script type="text/javascript">
function stat_time(){
var oP = document.createElement("p");
var oB = document.createElement("b");
var oText = document.createTextNode("Tekst yozdik");
oB.appendChild(oText);
oP.appendChild(oB);
document.body.appendChild(oP);
}
</script>
</head>
<body onLoad="stat_time()">
</body>
</html>
```

Mana yuqorida qilgan ishlarimiz:

1. biz documentimizda <p> elementini yaratdik
2. keyin esa documentda yana <b> elementini yaratdik
3. so'ngra esa faqat tekst yozdik
4. so'ngra <b> </b> orasiga tekstni joylashtirdik
5. <p> ning ichiga esa <b> ni joylashtirdik
6. va shularni hammasini <body> tagini orasiga qo'ydik

Bu bazan bizga kerak bo'ladi sababi siz oldindan qanday ma'lumotlar bo'lishini bilmaysiz, shunda siz oldindan formatlab yozib qoyasiz va yangi ma'lumotlarni o'zi avtomatik ravishda qo'yib ishlata boshlaydi.

Ho'sh agarda qandaydir elementini olib tashlash niyatimiz bo'lsachi?

Unda yuqorida ko'rgan ba'zi bir komandalar **removeChild**, **replaceChild** bizga kerak bo'ladi.

Misol:

```
<html>
<head>
<title> Misol 17</title>
<script type="text/javascript">
function rem_p(){
var oB = document.getElementsByTagName("b");
document.body.removeChild(oB[0]);
}
```

## JavaScript Hammaga

```
</script>
</head>
<body >
<a href="#" onclick="rem_p()">Test</a>
<b>
Bu birinchi B dir
</b>
<br>
<b>
Bu esa ikkinchi B dir
</b>
</body>
</html>
```

Bu yerda biz hamma <b> larni oB ga yozdik va body'da birinchi uchragan <b> o'chirib tashladik. Ko'rganingizdek bu juda ham osondir.

### Tablitsalar

Tablitsalar bilan ishlash ham yuqoridagilar kabidir, ammo qo'shimchalari ham bor. Quyida keltirilgan:

Komandalar	Tarifi
caption	<table> dagi <caption> elementi
tBodies	<tbody> elementi
tFoot	<tfoot> elementi
tHead	<thead> elementi
rows	hamma qatorlar
createTHead()	<thead> elementi yaratish
createTFoot()	<tfoot> elementi yaratish
createCaption()	<caption> elementi yaratish
deleteTHead()	<thead> elementini o'chirish
deleteTFoot()	<tfoot> elementini o'chirish
deleteCaption()	<caption> elementini o'chirish
deleteRow(joyi)	«joyi» dagi qatorni o'chirish
insertRow(joyi)	«joyi»ga bitta qator qo'shadi

### <tbody> elementlari

Komanda	Tarifi
rows	qatorlar to'plami
deleteRow(joyi)	«joyi»dagi qatorni o'chirish
insertRow(joyi)	«joyi»ga qator qo'shadi

### <tr> elementlari

Komanda	Tarifi
cells	Hujralar to'plami
deleteCell(joyi)	«joyi»dagi hujrani o'chirish
insertCell(joyi)	«joyi»ga hujra qo'shadi

Keling bularni hammasini bir misolda ko'rib chiqaylik.

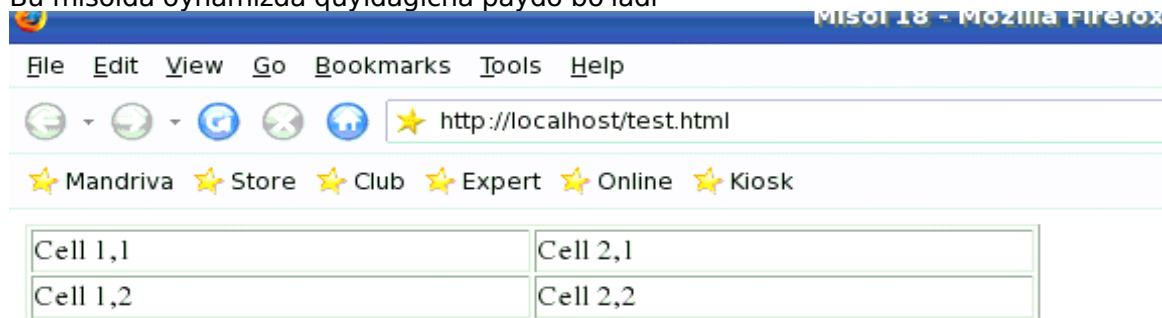
Misol:

```
<html>
<head>
<title>Misol </title>
</head>
<body id="test">
```

## JavaScript Hammaga

```
<script type="text/javascript">
//Tablitsa yaratamiz, chetlari 1 qalinlikda va oynamizni yarmisigacha
var oTable = document.createElement("table");
oTable.setAttribute("border", "1");
oTable.setAttribute("width", "50%");
//uning ichida tbody elementini yaratdik va uni oTable ning ichiga yerlashtirdik
var oTBody = document.createElement("tbody");
oTable.appendChild(oTBody);
//tbody ning ichida avval birinchi qatorni yaratamiz va uning ichida
// yani birinchi qator ichida bitta hujra yaratdik, va ichiga "cell 1, 1" deb yozdik
oTBody.insertRow(0);
oTBody.rows[0].insertCell(0);
oTBody.rows[0].cells[0].appendChild(document.createTextNode("Cell 1,1"));
//ikkinchi hujra yaratdik va ichiga "cell 1, 2" deb yozdik
oTBody.rows[0].insertCell(1);
oTBody.rows[0].cells[1].appendChild(document.createTextNode("Cell 2,1"));
//ikkinchi qator yaratdik va yuqoridagi kabi hujralar yaratdik
oTBody.insertRow(1);
oTBody.rows[1].insertCell(0);
oTBody.rows[1].cells[0].appendChild(document.createTextNode("Cell 1,2"));
oTBody.rows[1].insertCell(1);
oTBody.rows[1].cells[1].appendChild(document.createTextNode("Cell 2,2"));
//eng songida bodyning ichiga oTable ni yerlashtirdik.
document.body.appendChild(oTable);
</script>
</body>
</html>
```

Bu misolda oynamizda quyidagicha paydo bo'ladi



Mana bu ham juda oson va kelajakda judayam kerakli bo'lishi mumkindir.

## Dars 12 Regular Expression

Regular Expression bu qandaydir matndan, qanadaydir iborani qidirish, yoki borligini tekshirish uchun ishlatiladi. Buning uchun siz **RegExp()** komandasi ishlatiladi.

Misol:

```
var reTek = new RegExp("asa");
```

yoki

```
var reTek = new RegExp("asa", gi);
```

## JavaScript Hammaga

Bu yerda g=global yani butun matn bo'yicha qidiradi, birinchi topilgandan so'ng to'htab qolmaydi  
i= insensitive yani harflarni katta kichikligiga ahamiyat bermaydi.  
Yuqoridagi misolni biz Perl tilidagi kabi ham yozsak bo'ladi.

Misol:

```
var reTek = /asa/gi;  
var tText = "Asalomu aleykum";  
alert (reTek.test(tText));
```

Bu misolda biz tText ning ichida asa so'zi bor yo'qligini tekshirdik, va oynamizga



Ho'sh agarda bizga tekshirish emas, mos kelganlarni bizga qaytarishini hohlasakchi?  
Unda **exec()** ko'mandasidan foydalanamiz.

Misol:

```
<script type="text/javascript">  
var sText = "Sandjar ham shu yerda ular, qayerda?";  
var sAr = /ar/gi;  
var aTop = sAr.exec(sText);  
alert (aTop[0]);  
</script>
```

Bu yerda bizning oynaga «ar» so'zini chiqaradi. yani «ar» so'zi mos kelganligin bilish mumkindir.

**match()/search()** ham yuqoridagilar kabidir, farqi siz yozilgan matn orqali qidirasiz.

Yani:

```
<script type="text/javascript">  
var sText = "Sandjar ham shu yerda ular, qayerda?";  
var sAr = /ar/gi;  
var aTop = sText.match(sAr);  
alert (aTop[0]);  
</script>
```

Bunda ham «ar» oynamizga chiqadi, **search()** da esa nechitasi mos kelganligini raqam orqali qaytaradi.

**replace()**

Matnning ichidagi so'zlarni o'zgartirish uchundir.

Misol:

```
<script type="text/javascript">  
var sText = "Sandjar ham shu yerda ular, qayerda?";  
var sAr = /ar/gi;
```

## JavaScript Hammaga

```
var aTop = sText.replace(sAr, "aka");  
alert (aTop);  
</script>
```



Sandjaka ham shu yerda ulaka, qayerda?

OK

Mana ko'rganingizdek matndagi hamma «ar» larni «aka» bilan o'zgartirdik.

### **split()**

Bu ko'mandamiz siz bergan joylardan matnni bo'lib tashlaydi.

Misol:

```
<script type="text/javascript">  
var sText = "Sandjar ham shu yerda ular, qayerda?";  
var aTop = sText.split(",");  
alert (aTop[0]);  
alert (aTop[1]);  
</script>
```

Oynamizga



Sandjar ham shu yerda ular

OK



qayerda?

OK

Bu yerda siz ko'rganingizdek ikkiga bo'lib tashladi, yani «,» belgisidan ikkiga bo'ldi.

Belgilar

\t

\n

\r

Tarifi

«TAB» belgisi

Yangi qator belgisi

Enter yani boshiga qaytish belgisi

## JavaScript Hammaga

\f	Formaga mos kelish belgisi
\a	Alert belgisi
\e	«ESC» belgisi
\cX	Boshqaruvchi X belgisi
\b	Orqaga bitta o'chirish belgisi
\v	Bo'yiga «TAB» belgisi
\0	Null belgisi (yani yo'qlik belgisi)

Yuqoridagi belgilar sizga kelajakda kerak bo'ladi ma'nosida berib ketmoqdamiz, agar ishlatishga kelsak oddiy tekst kabi ishlataverasiz.

Misol:

```
<script type="text/javascript">
var sText = "Salom Sinch, \n qalesan?";
alert (sText);
</script>
```



Mana Bu yerda biz *Salom sinch*dan keyin qator tashadik va yengi qatordan *qalesan* so'zini yozdik. Shu kabi boshqalari bilan ham ishlashingiz mumkindir.

Agarda [] qovuslarni ishlatsangiz unda bu degani qavslar ichidagilardan bittasiga mos kelishi shart degani.

Misol:

```
<script type="text/javascript">
var sText = "Sem, Dem, rem, eem";
var sEm = /[sr]em/gi;
var aTop = sText.match(sEm);
alert (aTop[0]);
alert (aTop[1]);
</script>
```

Shunda oynamizga faqat «Sem» va «rem» chiqadi qolganlari to'g'ri kelmaganligi uchun hech qaysisi chiqmaydi.

Agarda yuqorida [sr] ning o'rniga [^sr] yozganimizda unda «sr» dan tashqari hammasi degan bo'lardik va «Sem», «rem» dan tashqari hammasi oynamizga chiqardi.

### Dars 13

Ho'sh bizda yana bu yerda qo'shimcha Class lar yani tayor shablonlar bor bular:

Belgi	Nimaga tengligi	Nimani bildirishi
.	[^\n\r]	Hamma belgilar faqat yangi qator va «ENTER» belgisidan tashqari
\d	[0-9]	Hamma sonlar
\D	[^0-9]	son bo'lmaganlar
\s	[\t\n\x0B\f\r]	belgilar
\S	[^\t\n\x0B\f\r]	shu belgilar emas

## JavaScript Hammaga

\w	[a-zA-Z_0-9]	hamma harf va sonlar va ost chiziq
\W	[^a-zA-Z_0-9]	hamma harf va sonlar va ost chiziqdan tashqari bo'lganlari

Va shular bilan bir qatorda jokerlarni ham berib o'taylik keyin umumiy ravishda misol keltiramiz

Belgi	Tarifi
?	Hech yoki bir marotaba qaytarilishi
*	hech yoki bir qancha marotaba qaytarilishi
+	Bir yoki bir necha marotaba
{n}	n marotaba
{n,m}	Eng kamida n va eng ko'pida m marotaba
{n,}	Eng kamida n marotaba.

Misol:

```
var sSon = «569»;  
var sTek = /\d\d\d/;  
alert (sTek.test(sSon));
```

Oynangizga «true» chiqadi sababi biz uch marotaba son kelishini tekshirdik va uch marotaba son keganligini ko'rdik.

yoki:

```
var sText = «soms»;  
var sTop = /s?omsa?/;          //Bu yerda biz s harfi hech yoki bir marotaba bo'lishi kerak, va a  
harfi ham  
var aTop = sText.match(sTop);  
alert (aTop);
```

Quyidagi kabi oyna bizga chiqadi.



Qolganlari bilan ham o'zingiz misollar yozib praktika qilib ko'ring.

Mana endi RegExp'ning yordamchi xususiyatlarini ko'rib chiqishga. Agar ahamiyat bergan bo'lsangiz yuqorida biz bir necha marotaba so'zdan keyin g yoki gi harflarini ham qoygan edik, unda shu ikkalasini nima ekanligini tushuntirdik. Mana endi esa qolganlariga ham vaqt yetib keldi.

**global** – g belgisidir, agar buni qoysangiz butun matnni hammasini tekshiradi

**ignoreCase** – i belgisidir, yani harflarni katta kichikligini hisobga olmaydi

**lastIndex** – Agar test yoki exec() bilan qidirtirsangiz unda bu yerda sizda keyingi mos kelish joyini sonini beradi

**multiline** – m belgisidir. Bu agar bizning matn bir necha qator bo'lsa unda uni ishlatish kerak, shunda qancha qator bo'lsa hammasidan qidiradi.

**source** – regular expressionni yozuvidir, yani agar regexp /[ba]\*/ bo'lsa source [ba]\* bo'ladi.

## JavaScript Hammaga

Keling yuqoridagilarni ishlatishga yana bir misol keltiraylik.

email adress to'g'ri yozilib yozilmaganini tekshirish uchun

```
var reEmail = /^(?:\w+\.?)*\w+@(?:\w+\.?)*\w+$/;
```

Bu yerda ? belgisidan keyin : nuqta turibti bu degani bu faqat bitta gruppaga degani agarda qoymaganimizda buni javascript bir qancha gruppaga ham tekshirib ko'rishi mumkin edi.

Misol uchun «abs dfs hrj» bular 3 ta gruppadir bizga shulardan birinchisi kerakdir.

A «^» belgimiz esa bu yerda qatorni boshlanishi ekanligini bildiradi.

\w+\.? - bu esa sozlar yoki sonlar va ular bilan birga . belgisi ham ishlatilishi mumkin degani.

\*\w+@ bu esa bir nechta yozuvlardan keyin «@» belgisi bo'lishi kerak deganidir.

va hkz.

## Events

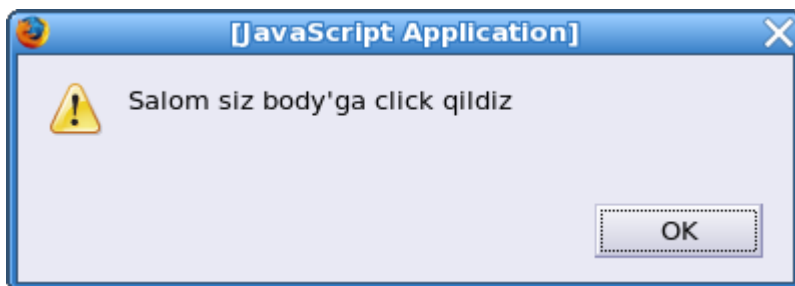
### Dars 14

Event'lar bular sizning nima ish qilganingizga qarab yani sichqonni chap tugmasini bosdingizmi yoki enter'ni bosdingizmi shuni aniqlash va kerak bo'lsa qandaydir ko'dni ishlatishdir.

Deyarli hamma teglarda event'lari bordir, misol uchun body tagida onclick event'i

Misol:

```
<html>
<head>
<title>Events</title>
<script Language="JavaScript">
function alert_me(){
alert("Salom siz body'ga click qildiz");
}
</script>
</head>
<body onclick="alert_me()">
</body>
</html>
```



Mana bu yerda biz body'ga bosishimiz bilan yuqoridagi habar chiqadi. Va shu kabi qolgan taglarga ham qo'yishingiz mumkin.

Misol

```
<html>
<head>
<title>Events</title>
<script Language="JavaScript">
function alert_me(){
alert("Salom siz H1 tagiga click qildiz");
}
</script>
```



## JavaScript Hammaga

```
</head>
<body >
<H1 onclick="alert_me()"> Bu H1 tagidir</H1>
</body>
</html>
```

Mana bu yerda biz H1'ga event berdik siz huddi shunday HTML tagiga ham onclick yoki shu kabi eventlarni berishingiz mumkindir (qolgan eventlar quyida ko'rib chiqiladi).

### Listeners

Listener'lar bular event bo'lsihini kutib o'tirivchi va agarda shu event bo'lsa qandaydir ko'dni ishlatishni kutib o'tiradi. Listener qo'yish quyidagicha bo'ladi.

addEventListener va removeEventListener.

Misol:

```
<html>
<head>
<title>Events</title>
</head>
<body >
<div id="h11"> Bu H1 tagidir</div>
<script Language="JavaScript">
var oH1=document.getElementById("h11");
oH1.addEventListener("click", alert_me, false);
function alert_me(){
alert("Salom siz div tagiga click qildiz");
}
</script>
</body>
</html>
```

Har qanday event'dan keyin bir qancha event xususiyatlari qandaydir ma'lumot oladi.

Mana ular:

Hususiyat	Tarifi
altKey	Agar «ALT» bosilgan bo'lsa unda true bo'ladi
button	Agar sichqonchani bossangiz unda qaysi tugamligiga qarab qiymat oladi 0 – bosilmagan 1 – chap tugma 2 – o'ng tugma 3 – chap va o'ng bosilgan 4 – o'rta tugma bosilgan 5 – chap va o'rta tugma bosilgan 6 – o'ng va o'rta tugma bosilgan 7 – chap, o'rta va o'ng tugmalar bosilgan
clientX	sichqoncha qaysi joyda (x kordinata bo'yicha) (toolbar, statusbar va hkz lardan tashqari).
clientY	Y ko'r dinata bo'yicha
ctrlKey	«CTRL» tugmasi bosilganligi
fromElement	Qaysi tagdan event ishlatib kelganligi
keyCode	keypress event uchun, qaysi tugma bosilganligi (enter, backspace, ...)
charCode	bosilgan tugma ko'di

## JavaScript Hammaga

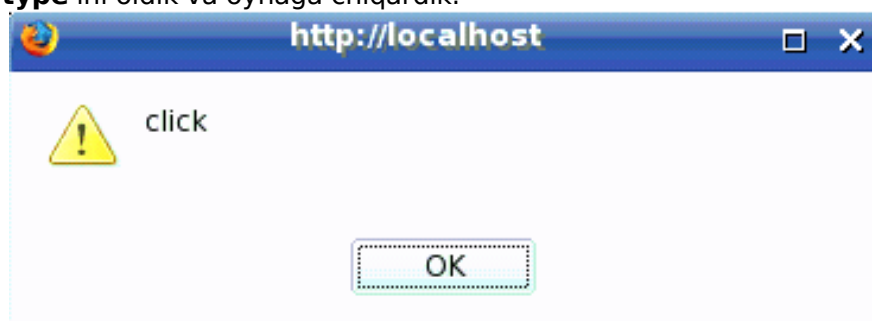
offsetX	x kordinata event ishlatilgan elementga nisbatan
OffsetY	y kordinata
repeat	Agar qandaydir tugma bosilib turgan bo'lsa
screenX	x kordinata monitoring/ekranning qaysi joyida ekanligi sichqonchani.
screenY	y kordinata
shiftKey	«SHIFT» bo'silganligi
srcElement	Qaysi element eventni ishlatganligi
toElement	Qaysi elementga eventdan kelganligi
type	Event ning turi
x/y	qaysi oynada event bo'lsa shu oynaga nisbatan x va y kordinatalar

Keling bir kichik misolda ko'rib chiqamiz, shunda yana ham aniq tushunasiz.

Misol:

```
<html>
<head>
<title>Misol 21 </title>
<script type="text/javascript">
function handle_click(oEvent){
var oType = oEvent.type;
alert (oType);
}
</script>
</head>
<body>
<a href="#" onclick="handle_click(event)">Testing</a>
</body>
</html>
```

Mana bu misolda biz **onclick()** bilan **handle\_click()** ni chaqirdik va unga event so'zi bilan event'ni yubordik. va uni **type** ini oldik va oynaga chiqardik.



ko'rganingizdek.

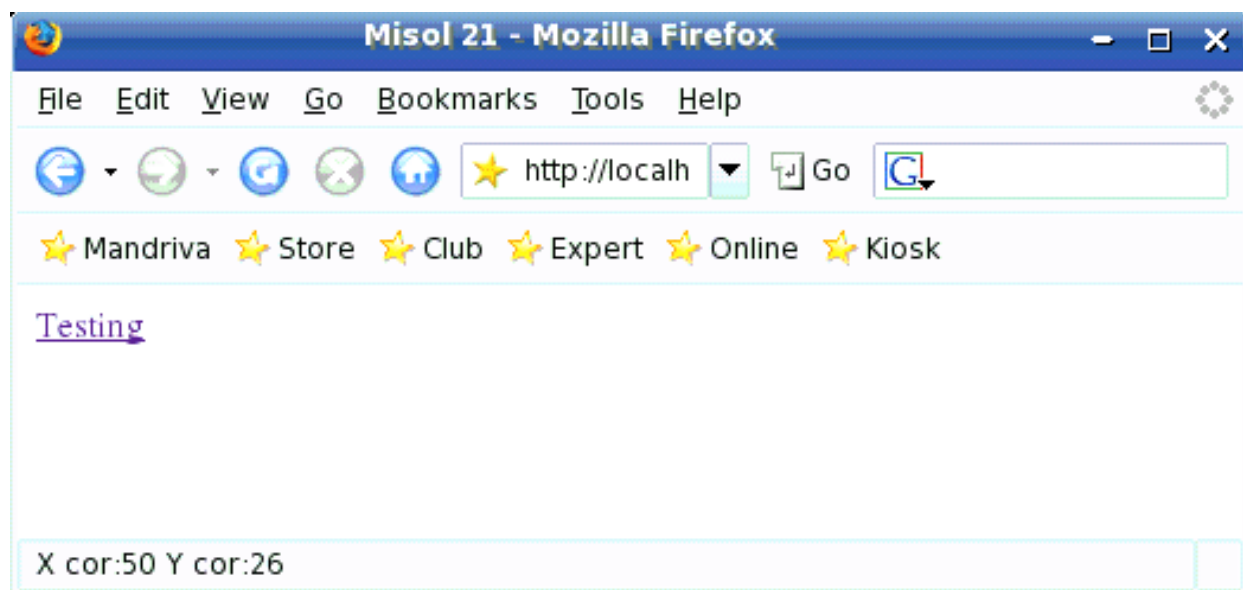
Ho'sh yana bir misolga qaraylik

Misol:

```
<html>
<head>
<title>Misol 22 </title>
<script type="text/javascript">
function handle_click(oEvent){
```

## JavaScript Hammaga

```
var oType = oEvent.type;
alert (oType);
}
function stat_chiq(oEvent){
var oX = oEvent.clientX;
var oY = oEvent.clientY;
window.status="X cor:" + oX+ " Y cor:" + oY;
}
</script>
</head>
<body>
<a href="#" onclick="handle_click(event)" onmouseover="stat_chiq(event)">Testing</a>
</body>
</html>
```



Biz bu yerda yana bir event'ni ko'rdik bu **onmouseover()** yani bu sichqonchani shu joyning ustiga boshqa joydan kelishidir. .

Va unga biz **stat\_chiq()** funksiyasini belgilab berdik, bu funksiya oddiy ish qiladi, qayerda sichqoncha bo'lsa shu joyni kordinatlarini chiqarib beradi.

### Dars 15

Mana tablitsada hamma event'lar berilgan:

<b>Event nomi</b>	<b>Tarifi</b>
click	qachonki sichqonchani bir marotaba bossangiz
dblclick	sichqonchani ikki marta bossangiz
mousedown	sichqonchani tugmasini bosishingizda ishlaydi
mouseup	sichqoncha tugmasini bosib/qo'yvorganizda ishlaydi
mouseout	sichqoncha element ustidan tashqariga chiqib ketganida ishlaydi
mousemove	sichqoncha element ustida harakatlanganida

## JavaScript Hammaga

Bular hammasi sichqoncha bilan aloqali bo'lgan eventlardir, yana klaviatura bilan aloqali eventlar ham bor.

Mana bular:

<b>Event nomi</b>	<b>Tarifi</b>
keydown	qachonki klaviatura bosilsa ishlaydi
keyup	bosilgan tugma qoyib yuborilganda ishlaydi
keypress	klaviatura bosilib qoyib yuborilgandan keyin ishlaydi

Misol:

```
<html>
<head>
<title>Misol 24 </title>
<script type="text/javascript">
function stat_chiq(oEvent){
var oCode = oEvent.charCode;
window.status="bosilgan:" + oCode
}
</script>
</head>
<body>
```

Nimanidir bosing:

```
<br>
<textarea id="txt1" cols="15" rows="5" onkeypress="stat_chiq(event)"> </textarea>
</body>
</html>
```

Mana bu misoldan siz ko'rganingizdek qaysi tugma bosilsa shuni ASCII ko'dini statusbarda chiqardik.

Yana shular birlan bir qatorda boshqa tag'lar uchun ham bir qancha eventlar bor.

<b>Event nomi</b>	<b>tarif</b>
load	qachonki bir element yuklanganda ishlaydi
unload	load'ning teskarisi
error	agarda qandaydir hato chiqsa ishlaydi
select	bu asosan <input> <textarea> uchundir, qachonki 1 yoki undan ko'p harf tanlansa
change	<input> <textarea> uchundir, qachonki focus olgandan so'ng o'zgarish bo'lsa va shu o'zgarish bilan focus boshqa elementga o'tsa
submit	submit tugmasi bosilganda
reset	agarda reset tugmasi bosilsa
resize	qachonki oyna yoki frame hajmini o'zgartirsa
scroll	scrollbar'i bor elementda scroll ishlatilsa
focus	qandaydir element aktivlashtirilsa
blur	aktiv elementdan chiqib ketilsa

Ho'sh bularga ham bir ikta misol keltiraylik.

Misol:

```
<html>
<head>
<title>Misol 25 </title>
</head>
<body onload="alert('Sahifa yuklandi')">
```

Nimanidir bosing:

## JavaScript Hammaga

```
<br>
<textarea id="txt1" cols="15" rows="5" onchange="alert('Ichidagi tekst ozgarganku')" >
Test</textarea>
</body>
</html>
```

Mana bu yerda biz **onchange()** eventini ishlatdik, va shu bilan bir qatorda **<body>**'ning yuklanishiga ham **onload()** ga alert qo'ydik. Qolganlari bilan ham siz o'zingiz tekshirib misollarda ko'rib chiqing.

Misollar:

**Yer qimilashi** Juda ham chiroyli script!

```
<html>
<head>
<title>earth</title>
<SCRIPT language="JavaScript">
function earth(n) {
  if (self.moveBy) {
    for (i = 15; i > 0; i--) {
      for (j = n; j > 0; j--) {
        self.moveBy(0,i);
        self.moveBy(i,0);
        self.moveBy(0,-i);
        self.moveBy(-i,0);
      }
    }
  }
}
</SCRIPT>
</head>
<body>
<center>
<form>
<input onclick="earth(3)" type="button" value="Yer qimilashi">
</form>
</center>
</body>
</html>
```

**Tablitsani sichqoncha ustiga kelganda aktivlashtirish.**

```
<html>
<head>
<title>Tablitsa</title>
<script language="JavaScript">
function makevisible(cur,which){
strength=(which==1)? 1 : 0.2

if (cur.style.MozOpacity)
cur.style.MozOpacity=strength
else if (cur.filters)
cur.filters.alpha.opacity=strength*100
}
```

## JavaScript Hammaga

```
</script>
</head>
<body >
<table border="1" width="100%" style="filter:alpha(opacity=20);-moz-opacity:0.2"
onmouseover="makevisible(this,1)" onmouseout="makevisible(this,0)">
<tr>
<td>
Tablitsa aktiv yoki pasiv
</td>
</tr>
</table></body>
</html>
```

### Style Dars 16

Ho'sh agar siz style'lar (CSS) bilan ishlamoqchi bo'lsangiz unda nima qilish kerak? Qaysi komandalarni ishlatish kerak?

Buning uchun bizga yordamga quyidagi komandalar keladi:

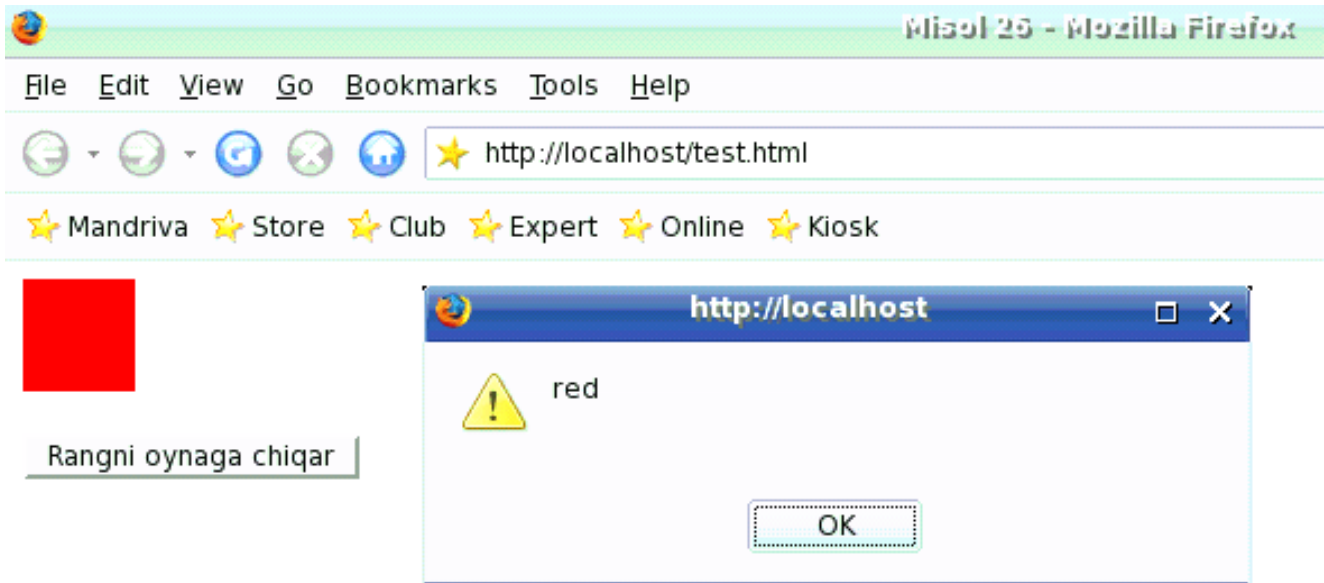
getPropertyValue(property name)	hususiatning qiymatini bizga olib beradi
item(index)	index'da berilgan hususiatni nomini beradi
removeProperty(property name)	hususiatni olib tashlaydi
setProperty(property_name, value, priority)	hususiat qo'shadi.

Misol:

```
<html>
<head>
<title>Misol 26</title>
<script type="text/javascript">
function sayStyle() {
var oDiv = document.getElementById("div1");
alert(oDiv.style.getPropertyValue("background-color"));
}
</script>
</head>
<body>
<div id="div1" style="background-color: red; height: 50px; width:
50px"></div>
<br>
<input type="button" value="Rangni oynaga chiqar" onclick="sayStyle()">
</body>
</html>
```

Mana ko'rganingizdek bu yerda biz style'ning ichidan background-color ni olib oynaga chiqardik.

## JavaScript Hammaga

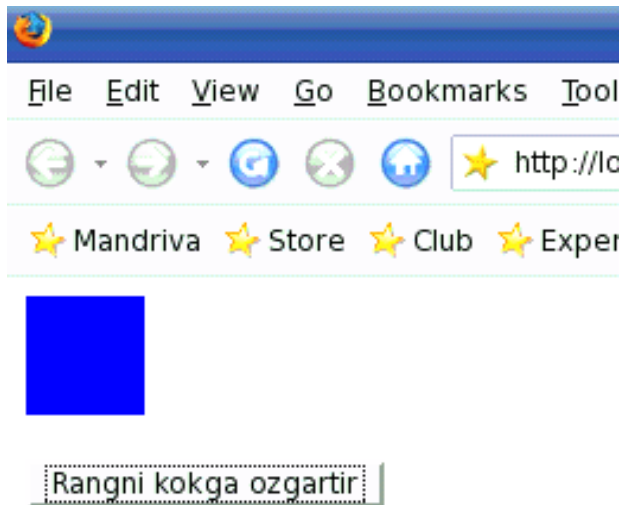


Ho'sh agar o'zgartirmoqchi bo'lsakchi?

```
<html>
<head>
<title>Misol 27</title>
<script type="text/javascript">
function putStyle() {
var oDiv = document.getElementById("div1");
oDiv.style.setProperty("background-color", "blue", "");
}
</script>
</head>
<body>
<div id="div1" style="background-color: red; height: 50px; width:
50px"></div>
<br>
<input type="button" value="Rangni kokga ozgartir" onclick="putStyle()">
</body>
</html>
```

Mana bu misoldan ko'rganingzdek bu ham juda osondir. :)

## JavaScript Hammaga



Mana yana bir misol style lar bilan aloqali:

```
<html>
<head>
<title>Misol 28</title>
<script type="text/javascript">
function showTip(oEvent) {
var oDiv = document.getElementById("divTip1");
oDiv.style.visibility = "visible";
oDiv.style.left = oEvent.clientX + 5;
oDiv.style.top = oEvent.clientY + 5;
}
function hideTip(oEvent) {
var oDiv = document.getElementById("divTip1");
oDiv.style.visibility = "hidden";
}
</script>
</head>
<body>
<p>Qizil joyning ustiga sichqoncha bilan boring.</p>
<div id="div1"
style="background-color: red; height: 50px; width: 50px"
onmouseover="showTip(event)" onmouseout="hideTip(event)"></div>
<div id="divTip1"
style="background-color: yellow; position: absolute; visibility:
hidden; padding: 5px">
<span style="font-size:13">Qizil oyna<br>
Bu yerga siz o'zingizni <br> komentariyangizni yozishingiz mumking
</span>
</div>
</body>
```



## JavaScript Hammaga

</html>

Bu yerda style bilan ozgina oynadik va uni huddi tooltip kabi qilib o'zimiz hohlagan yozuvni va hohlagan joyda chiqarib ko'rdik. Bu ba'zan juda ham kerakli bo'lishi mumkindir. Siz ham shu kabi ba'zi bir misollarni o'zingiz ko'ring.

Ho'sh agarda biz bir nechta CSS larimiz bo'lsa unda qanday bittasini o'chirib ikkinchisini yoqishimiz mumkin?

Buning uchun,

```
document.styleSheets[1].disabled=true;
```

yoki

```
document.styleSheets[1].disabled=false;
```

yo'li orqali o'chirib yoqishingiz mumkindir.

## Errors

### Dars 17

Bu mavzuyimizda biz hatolarni ko'rib chiqamiz, qanday hatoni qanday topish, yoki aylanib o'tish mumkinligini.

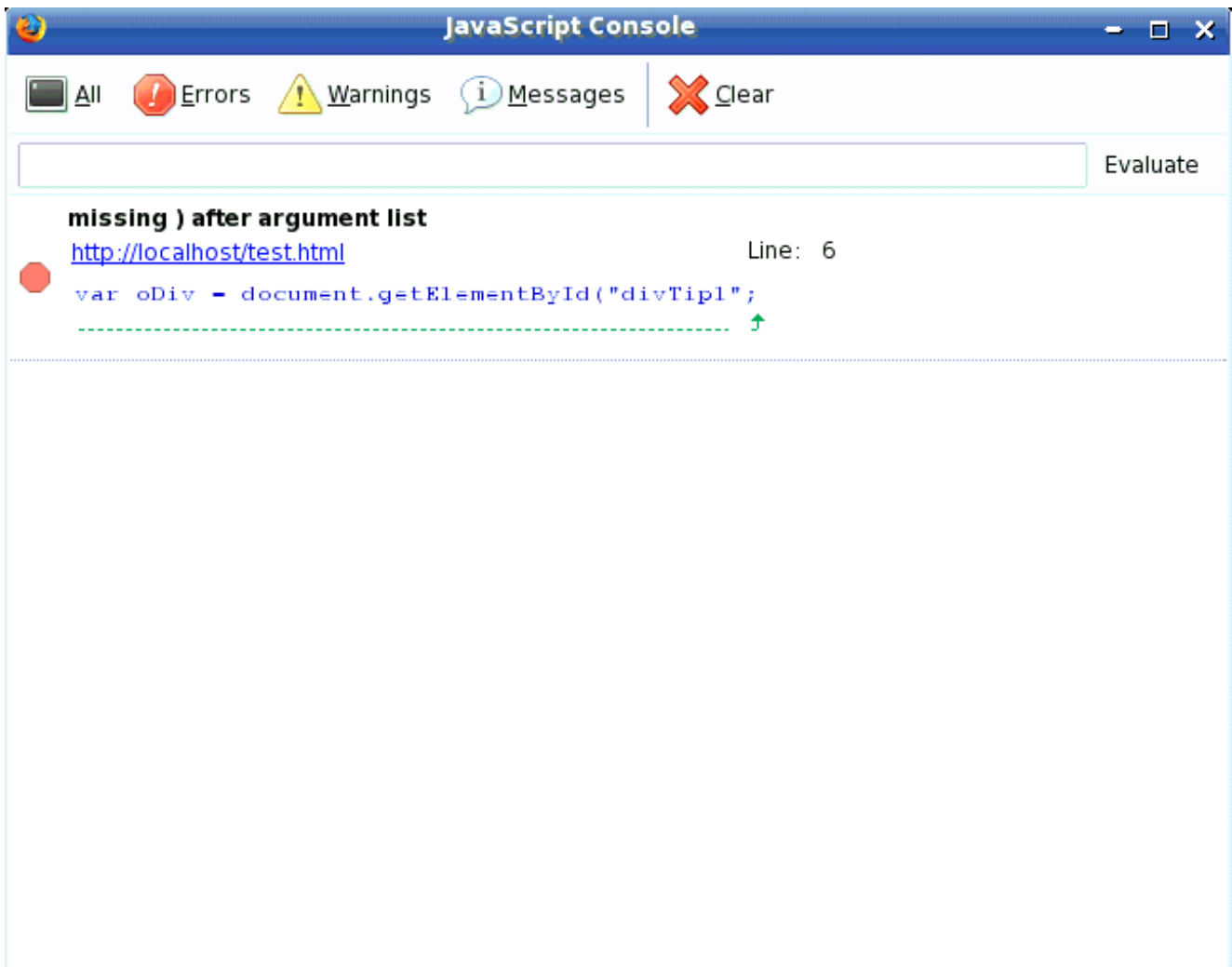
Mozilla, yoki shu browser turlarida javascript hatosini siz javascript console'dan ko'rishingiz mumkindir.

Bu Tools-- menyusining ichida va ko'rinishi



Shu kabidir. Va agar sizda hato bo'lsa (scriptda) unda shu yerda yozib ko'rsatadi. Mana kichik bir hato.

## JavaScript Hammaga



Mana ko'rganingizdek hato qayerda ekanligini ko'rsatib sizga yordamga keladi bu oynamiz. Ho'sh agar biz hato bo'lsa qandaydir bir yo'l bilan bizga habar berishini hohlasak unda nima qilamiz? Juda oson, onerror degan event' orqali buni aniqlashtiramiz.

Misol:

```
<html>
<head>
<title>Misol 28</title>
<script type="text/javascript">
window.onerror=function() {
alert("Hato, hato, hato chiqdi");
}
</script>
</head>
<body>
<a href="#" onclick="alerter()">Meni bos</a>
Bu yerda hato chiqishi kerak. Sababi yo'q funktsiyani chaqirtirdik
</body>
</html>
```

## JavaScript Hammaga

Bu yerda biz yo'q bo'lgan funktsiyani chaqirtirdik va hatoga yo'l qo'ydik, bunda hato avtomatik ravishda bizga alert orqali habar berildi.



No lekin, biz javascript console'ga kirganda ko'rdikki u yerda ham hato borligini ko'rsatib yozib qo'ygan. Buni yozmasligini hohlasek funktsiyamizga return true; yozishimiz kerak.

```
<script type="text/javascript">
window.onerror=function() {
alert("Hato, hato, hato chiqdi");
return true;
}
```

Shu kabi.

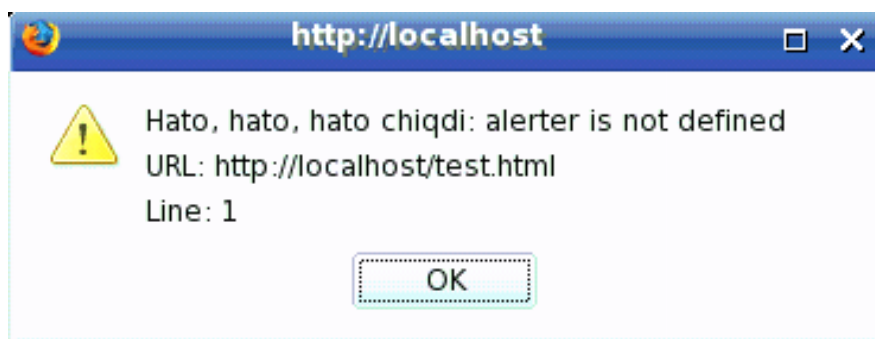
Ho'sh hato nimaligini bilish uchun nima qilish kerak?

Bu ham juda oson, misolga qarang.

```
<html>
<head>
<title>Misol 28</title>
<script type="text/javascript">
window.onerror=function(sMessage, sUrl, sLine) {
alert("Hato, hato, hato chiqdi: " +sMessage+ " \nURL: " + sUrl + " \n Line: " +sLine);
return true;
}
</script>
</head>
<body>
<a href="#" onclick="alerter()">Meni bos</a>
Bu yerda hato chiqishi kerak. Sababi yo'q funktsiyani chaqirtirdik
</body>
</html>
```

Mana shu kabi oynamiz chiqadi, va bu oynadan ko'rganingizdek qanday hato, qaysi faylda, va qaysi qatordaligini ko'rish mumkindir.

## JavaScript Hammaga



### Dars 18

Biz bular bilan bir qatorda **try ... catch** ishlatishimiz mumkin. Bu funktsiyalarni qiladigan ishi, sizning qandaydir ko'ringizni ishlatib, agar unda hato bo'lsa catch orqali ilib olib obrobotka qilish mumkindir.

```
try {  
  //ishlatish ko'di  
  [break]  
}  
catch ([exception]){  
  //agar hato chiqsa shu yerdagi ko'd ishlatiladi  
  [break]  
} [finally]{  
  //hato chiqsa ham, chiqmasa ham ishlatiladigan ko'd shu yerga yoziladi.  
}]
```

Misol:

```
try {  
  window.alerter(); //bizda yoq funktsiyani chaqirmoqdamiz  
  alert («Funktsiya ishlatildi»);  
}  
catch (exception){  
  alert («Hato chiqdi»);  
}  
finally {  
  alert («Try va catch ishlab tugadi :\)»);  
}
```

Bu yerda hatoni ko'rish uchun,

**message – hato**  
**name – hato turi**

Hato turlari

Hato turi

EvalError

RangeError

Paydo bo'lishi

eval() funktsiyasida hato bo'lsa.

JavaScript da katta va kichik hodisasi ishlatilganda (Number.MAX\_VALUE and Number.MIN\_VALUE).

## JavaScript Hammaga

ReferenceError	no'to'g'ri reference bo'lsa.
SyntaxError	Yozilishda hato bo'lsa
TypeError	O'zgaruvchan turi noto'g'ri belgilangan.
URIError	encodeURIComponent() yoki decodeURI() funktsialarida hato.

Misol:

```
try {
  eval («a ++ b»);
} catch (oException){
  if (o.Exception.name == «SyntaxError»){
    alert («Syntax Error: »+ oException.message);
  }
  else {
    alert («Hato chiqdi » + oException.message);
  }
}
```

Yuqorida ko'rilganidek bu juda ham oson, va javascript'ni kuchini yana ham oshiradi.

### **Agarda siz o'zingiz qandaydir hatoni yaratmoqchi bo'lsangiz unda nima qilish kerak?**

Buning uchun throw funktsiyasidan foydalaning.

misol:

```
throw new Error («Yana bitta hato chiqdi»);
throw new SyntaxError("I don't like your syntax.");
throw new TypeError("What type of variable do you take me for?");
throw new RangeError("Sorry, you just don't have the range.");
throw new EvalError("That doesn't evaluate.");
```

Mana shu kabi ishlatishingiz yani hato hosil qilishingiz mumkindir.

Nima uchun bu bizga kerak? Deylik agarda sizga qandaydir kiritish ma'lumoti kerak, va agarda shu ma'lumot kiritilmasa unda siz oynaga hato yuborishingiz mumkin, (alert'siz).

XML va JavaScript bilan ishlashni o'zingiz mustaqil o'rganib chiqing bu mavzu bizning kitobda yoritilmaydi.

## Cookies

### Dars 19

Cookies: bu har bir sahifaga kirgan odam uchun alohida va boshqa odamda unday cookie bo'lmaydi, yani o'zining unique key bilan saqlab qo'yiladi. Cookie server'da emas client (sahifaga kirgan inson) kompyuterida saqlanib qoladi. Agarda kimdir sahifaga kirib, o'ziga moslab olsa, misol uchun ranglarni va hkz, keyingi gal yana bir marotaba kirganda hamma o'zgartishlar saqlanib qoladi, va qandoq bo'lgan bo'lsa shu shaklda ko'rsatadi. Bu juda ham kuchli quroldir, siz har bir odam uchun alohida ma'lumot saqlab o'tirmaysiz, buni cookie'lar siz uchun qiladi, va sizning serveringga og'riligi tushmaydi. Lekin ohirgi vaqtlarda ko'p internet browserlar cookie'larni o'chirib qo'ymoqda sababi, ular faqat foydali emas, balki ularni «hacking» da ham ishlatish mumkindir.

Cookie'ning hususiatlari:

**Name** – cookie'ning ismi, har bir cookie har hil (unique) bo'lishi shart, cookie name harflar, sonlardan iborat bo'lishi mumkindir.

**Value** – cookie'da saqlanadigan ma'lumot. encodeURIComponent() orqali encoding qilish kerak.

## JavaScript Hammaga

**Domain** – hafsizlik uchundir, siz yaratgan cookie'ga boshqa domain'dan hech kim kira olmaydi

**Path** – aniq bir url yo'li, boshqa yo'llar kira olmaydi, misol: <http://www.uzinfo.com/books> bo'lsa unda <http://www.uzinfo.com> unga kira olmaydi.

**Expiration** – Qachon bu cookie o'chirib tashlanishi.

**Secure Flag** – true/false bu degani SSL, yoki https lardangina kira olinadi.

Cookie'ni client kompyuteriga yozish uchun,

```
function setCookie(sName, sValue, oExpires, sPath, sDomain, bSecure) {
var sCookie = sName + "=" + encodeURIComponent(sValue);
if (oExpires) {
sCookie += "; expires=" + oExpires.toGMTString();
}
if (sPath) {
sCookie += "; path=" + sPath;
}
if (sDomain) {
sCookie += "; domain=" + sDomain;
}
if (bSecure) {
sCookie += "; secure";
}
document.cookie = sCookie;
}
```

Mana shu funktsiyadan foydalaning. Bu oddiy va qulay yo'lidir.

Misol:

```
setCookie («Sinch», «Javascript bu oson?», new Date(Date.parse(«Dec 18, 2006 »)))
```

Bu bilan biz cookie yozib qo'ydik.

endi o'qish esa qanday?

```
function getCookie(sName) {
var sRE = "(?::; )?" + sName + "=([^;]*)?";
var oRE = new RegExp(sRE);
if (oRE.test(document.cookie)) {
return decodeURIComponent(RegExp["$1"]);
} else {
return null;
}
}
```

mana shu funktsiya bilan hohlagan cookie'ni olishingiz mumkindir.

Misol:

```
var sKitob = getCookie(«Sinch»);
```

Mana keling butun bir misolda ko'raylik:

```
<html>
<head>
<title>Misol 29</title>
<script type="text/javascript">
function setCookie(sName, sValue, oExpires, sPath, sDomain, bSecure) {
var sCookie = sName + "=" + encodeURIComponent(sValue);
if (oExpires) {
sCookie += "; expires=" + oExpires.toGMTString();
}
if (sPath) {
```

## JavaScript Hammaga

```
sCookie += "; path=" + sPath;
}
if (sDomain) {
sCookie += "; domain=" + sDomain;
}
if (bSecure) {
sCookie += "; secure";
}
document.cookie = sCookie;
}
function getCookie(sName) {
var sRE = "(?::; )?" + sName + "=([^\;]*)?";
var oRE = new RegExp(sRE);
if (oRE.test(document.cookie)) {
return decodeURIComponent(RegExp["$1"]);
} else {
return null;
}
}
function alertCookie(sName){
var sBook = getCookie(sName);
alert (sBook);
}

</script>
</head>
<body onload="setCookie ('Sinch', 'Javascript bu oson?')">
<a href="#" onclick="alertCookie('Sinch')">Meni bos</a>
</body>
</html>
```

Ishlatib ko'ring o'zingiz har hil misollarda ham ishlatib, iloji boricha ko'proq ishlatib o'rganing

Mana shu bilan JavaScript bo'yicha kichik darsligimizni tugataman, va niyat qilamanku bu kitob sizning JavaScript olamiga bo'lgan ilk qadamingizda katta yordam beradi. Bu yerda men eng kerakli deb bilgan hamma mavzularni ko'rib chiqdim. Agarda savollaringiz bo'lsa, yoki takliflar unda lutfan [kitob@uzinfo.com](mailto:kitob@uzinfo.com) yoki [uzinfo@inbox.ru](mailto:uzinfo@inbox.ru) ga hat yozing.

Sizga omad tilab qoluvchi Sandjar Alimov (aka Sinch)

*PS: aka = also know as :)*