

**TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI
AXBOROT TEXNOLOGIYALARI FAKULTETI
“MULTIMEDIA TEXNOLOGIYALARI” KAFEDRASI**

REFERAT

MUALLIF: X.A.Ishanxodjayeva (TATU, e-mail adish_adisha@ mail.ru)

JAVA SCRIPT TILINI O'RGANAMIZ

Toshkent 2012

JAVA SCRIPT TILINI O'RGANAMIZ

Reja:

1. Java Script tiliga kirish. Java Script imkoniyatlari
2. Java Scriptda hodisalar bilan ishlash
3. Java Scriptda o'zgaruvchilar bilan ishlash
4. Java Scriptda funksiyalar

JAVA SCRIPT TILIGA KIRISH

JavaScript -bu interaktiv va dinamik HTML Web sahifa yaratishga mo'ljallangan dasturlash tili hisoblanadi.

Java Script Netscape Communications va Sun kompaniyalarining birgalikdagi mahsuloti bo'lib, u 1995 yilda birinchi versiyasi yaratilgan (Java Script 1.3).

Java Scriptning dastlabki nomi LiveScript deb atalgan va u Netscape tomonidan yaratilgan.

Java Script klient va server tomonidan dasturlash tili bo'lib, u HTML tilining teglarini va Internet brauzer(Microsoft Internet Explorer,Opera,Mozilla, Netscape) oynasini boshqaradi. Bunda brauzerning nimasini boshqaradi degan savol tug'ilishi mumkin. Bu savolga keyingi bo'limlarimizda to'xtalib o'tamiz.

Boshqacha qilib aytganda HTML tili va JavaScript dasturlash tili birgalikda DHTML ni tashkil qiladi.

Java Script tili ECMA-262 (Evropa standarti) va ISO-162 62 (Xalqaro standarti) standartlari orqali standartlashtirilgan.

Java Script nisbatan olganda ob'jektga mo'ljallangan dasturlash tili bo'lib, Internet uchun klient va server tomonida ishlovchi ilovalar (senariylar) yaratiladi.

Java Scriptda yozilgan dastur kodini oldindan kompilyasiya qilish shart emas va qilinmaydi. Faqatgina dastur kodi interpretatsiya qilinadi. Shuning uchun Java Scriptni "interpretatorli til" deb ataladi.

Java Scriptda yozilgan dastur to'g'ridan-to'g'ri HTML formatida saqlanib bajariladi.

HTML fayli yuklangan vaqtda dastur matni brauzer tomonidan interpretatsiya qilinadi va u brauzerda bajariladi. Shuning uchun brauzerni interpretator deb ham nomlash mumkin.

Kompilyasiya va interpretatsiya tushunchalariga to'xtalib o'taylik.

Translyatsiya-bu ma'lum bir dasturlash tilida yozilgan dastur matnini (kodi) elektron hisoblash mashina tiliga o'tkazilish jarayoni tushuniladi.Shunday vazifani bajaruvchi maxsus dasturlar translyatorlar deb ataladi.

Translyatsiya ikkiga bo'linadi:

1. Interpretatsiya.
2. Kompilyasiya

Interpretatsiya-bu dastur matnidagi har bir qator buyruqlarini qatorma-qator mashina tiliga o'girilish jarayoni.

Kompilyasiya-bu dastur matnidagi har bir qator buyruqlarini mashina tiliga o'tkazishi bilan birgalikda ularning ketma-ketligini ta'minlash va sturktura bo'yicha mashina tiliga o'tkazilish jarayonidir.

JAVA SCRIPT IMKONIYATLARI

Java Scriptning asosiy imkoniyatlari quyidagilar:

- HTML hujjatning ko'rinishini dinamik boshqarish.

- Internet brauzer ishini boshqarish.
 - Foydalanuvchi bilan interaktiv muloqotni o'rnatish imkoniyati.
 - Matematik hisoblashlarni bajarish.
- Java Script bilan tizim(OS) o'rtasida hech qanday fizik yoki dasturiy vositalar mavjud emas. Natijada Java Script tilining kamchiliklari paydo bo'ladi.
- Java Scriptning asosiy kamchiliklari:
- Printer va boshqa qurilmalardan komp'yuter va local tarmoqda foydalanib bo'lmaydi.
 - HTML bilan bevosita birgalikda ishlaydi.
 - Tizimli fayllarni yaratishda imkoniyatining kamligi.
 - Ko'p oqimlilik hisoblashlarni tashkillashtirish mumkin emas.
- Java Script tili asosan Java va C++ dasturlash tillarining imkoniyatlaridan foydalangan. Bunday imkoniyatlarini " JavaScript kodi bilan ishlash " bo'limida ko'rib o'tamiz.

JAVA SCRIPT TILINING STRUKTURASI

Java Script dasturi quyidagi qismlardan tashkil topgan:

1. Konteynerlar. <SCRIPT>
2. Hodisalarni qayta ishlovchi(EventHandler).
3. Gipermatn ko'rinishi

JAVA SCRIPT KONTEYNERI.

HTML faylida ikkita oraliq teg <SCRIPT>...</SCRIPT> teglari ishlatiladi.

Masalan:

```
<SCRIPT language="javascript" src="java.js">...</SCRIPT>
```

Konteynerdagi language qaysi tilda yozilganligini ko'rsatadi. Odatda bu shart emas. Undan tahsqri Internet Explorer VBSriptni ham qo'llab quvatlaydi va bunda language=VBScript ko'rsatiladi.

Demak, language parametri yordamida uning qaysi dasturlash tilida tuzilganligini ko'rsatish maqsadga muvofiqdir.

<SCRIPT>...</SCRIPT> juft teg asosan <HEAD> va <BODY> seksiyalarida ishlatilishi mumkin.

<HEAD> seksiyada aralash funksiyalar, bunda o'zgaruvchilar va massivlarning boshlang'ich qiymatlarini o'zlashtiradi.

<BODY> seksiyasida esa **document.write()** operatoridan tashkil topgan dastur fragmentiga ruxsat beriladi.

"javascript:" manzili (gipermatnli ko'rinish)

```
<A HREF="javascript:operator yoki funksiya">matn</A>
```

Masalan quyidagi konstruksiyani ko'raylik :

```
<A HREF="javascript:alert('SALOM')">SALOM</A>
```

Bunda gipermatn "SALOM" ustiga sichqoncha kursori olib borib bosilgan vaqtda **alert("SALOM")** funsiyasi ishga tushadi. Java Scriptni ishlatishni bunday usuli **Gipermatnli uzatkich** usuli deyiladi. Odatda bunday usuladan foydalanish qulaylik tug'diradi.

JAVA SCRIPTDA HODISALARNI QAYTA ISHLOVCHI QISM.

Hodisalarni qayta ishlovchi bo'lim Java Scriptning asosiy qismini tashkil etadi. Bu asosiy qismdan ko'piroq Web dasturchilar foydalanadilar.

Hodisalarni qayta ishlovchi qismda biz ko'proq bevosita HTML teglari bilan ishlaymiz. Bunda <FORM> elementlari bizga yordam beradi. Undan tashqari Java Scriptda hodisalar turkumi mavjud bo'lib, ular Down, Press, Up va Interaktiv jarayonlarni amalga oshirishimizga yordam beradi. Hodisalarni bajarishni tugma elementiga yoki gipermatn uzatkichiga biriktirish mumkin.

Quyida button elementiga biriktirilgan hodisani qanday bajarilishini ko'rish mumkin:

```
<FORM>
```

```
<INPUT type="button" value="Hodisa" onClick="alert(Salom JavaScript)">
```

</FORM>

Bunda onClick hodisalarni qayta ishlovchisidan foydalaniladi.
onClick->EventHandler

GIPERMATN UZATKICHGA BIRIKTIRILGAN HODISALAR

```
<A HREF="#examle_2" onMouseOver="aler('onmouseover!'); return true"
onClick="alert('click!'); return true">Gipermatn</A>
```

Yuqorida ikki xil hodisalarni qayta ishlovchisi ko'rsatilgan.

onMouseOver-sichqoncha ko'rsatkichini gipermatn ustiga olib borilishi bilan amalga oshadi.

onClick-sichqonchani bosish bilan.

JAVA SCRIPTDA JARAYONLAR:

- onClick: sichqonchani bosgan vaqt;
- onMouseOver: sichqoncha ko'rsatkichini yo'naltirgan payt;
- onMouseOut: sichqoncha ko'rsatkichini tashqariga olgan payt;
- onLoad: HTML hujjati ishga tushgan payt, ya'ni oyna ochilganda;
- onUnload: HTML hujjati ishni yakunlagan payt, ya'ni oyna yopilganda;
- onBlur: sichqoncha ko'rsatkichi oyna yoki freymdan tashqarida bo'lishi;
- onChange: kontent o'zgargan payt;
- onDragDrop: ob'ektni dragdrop qilgan payt;
- onFocus: sichqoncha ko'rsatkichini oyna yoki freymda ushlagan payt;
- onKeyDown: ishchi tugmani bosib turgan payt;
- onKeyPress: ishchi tugmani bosgan payt;
- onKeyUp: ishchi tugmani bosib qo'yib yuborgan payt;
- onMouseDown: sichqonchani tugmasi bosilgan payt;
- onMouseMove: sichqoncha ko'rsatkichini qimirlatgan payt;
- onMouseUp: sichqonchani bosib qo'yib yoborgandan keyingi payt;
- onSubmit: submit bo'lgan payt;
- onSelect: tanlangan payt.

Undan tashqari onAbort, onError, onReset, onResize kabi jarayonlar ham mavjud. Ularni o'zingiz mustaqil o'rganing.

Event Handlers uchun Forms

- onBlur --> select, text, textarea, password, file;
- onClick --> button, checkbox, radio, reset, submit;
- onFocus --> select, text, textarea, password, file;
- onSelect --> select, text, textarea;
- onSubmit --> form.

Event Handlers uchun Links

- onClick --> links (a href="...");
- onMouseOver --> links (a href="...").

Event Handlers uchun Windows

- onLoad --> window, frame;
- onUnload --> window, frame.

Masalan:

```
<BODY onLoad="alert('Salom JavaScript')">
```

```
<FRAMESET onLoad="alert('Salom Java Script')">
```

KOMMENTARIYA (IZOH)

Barcha dasturlash tillarida izoh juda zarur bo'ladi. Chunki dasturni tekshirish va har bitta qatorini analiz qilish, uni tushuna olish qulay bo'ladi.

Interpretator JavaScript dasturida(<SCRIPT>...</SCRIPT> teglari orasida) uchta turdagi izoh mavjud:

1. `“//”` -bir qator uchun izoh belgisi.
 2. `“/* */”`-ko'p qator uchun izoh belgisi.
 3. `<!-- -->`.z
- `“//”` va `“/* */”` kommentariya C++ va JAVA tillarida ,shuningdek DELPHI da ham qo'llaniladi.

O'GARUVCHILAR,UNING NOMI VA TOIFASI

Java Script dasturlash tilida i=3 ko'rinishdagi operatorida i-o'zgaruvchi 3 sonini o'zlashtiradi. O'zgaruvchilarga son yoki bosh harfi raqam bilan boshlanuvchi nom berib bo'lmaydi. Internet brauzerning yangi versiyalarida oz'garuvchining nomi sifatida `“$”` belgisini ishlatish mumkin. Variable(O'zgaruvchi).

O'zgaruvchilarni yozgan vaqtda harflarning katta-kichikligiga e'tibor beriladi. Ularning farqi bor.

Yana shuni aytib o'tish kerakki, Java Scriptda o'zgaruvchilarni e'lon qilishda hech qanday toifa ko'rsatilmaydi. Shuning uchun uning toifasi avtomatik ravishda **string** deb olinadi. Agar bizga son qiymat kerak bo'lsa, biz uni maxsus funksiyalar yordamida son qiymatiga aylantirishimiz mumkin.

Ko'p hollarda o'zgaruvchilarni e'lon qilishda **var** xizmatchi so'zidan foydalaniladi. Lekin **var** xizmatchi so'zini har doim ham ishlatish talab qilinmaydi. Ko'plab Web dasturchilar funksiyaning ichki qismida, ya'ni lokal o'zgaruvchi sifatida har bir o'zgaruvchini e'lon qilishda **var** xizmatchi so'zidan foydalanishadi. Yana shuni aytib o'tish kerakki, Java Script kodidagi har bir qator `“;”` belgisi bilan yakunlanadi yoki bu belgini qo'ymasa ham bo'ladi.

E'lon (declaration) :

var nom=birinchi qiymat(value)

Masalan: **var** x1=4.4, x2="5"; //to'g'ri

var 1x=4.4; //noto'g'ri

OPERATORLAR

1. Matematik operatorlar:
+, -, *, /, %.
2. Munosabatli operatorlar:
>, >=, <, <=, ==, != .
3. Mantiqiy operatorlar:
!(antonim), &&(va), ||(yoki).

Yoqorida ko'rsatilgan operatorlarning barchasi C++,Java va C# dasturlash tillarida ishlatiladi.

BA'ZI QONUN QOIDALAR

Matematik barobar `“=”` belgisi Java Scriptda ikkita barobar `“==”` ga teng.

Misol:

```
<html>
<body>
<script>
document.write("Salom bolalar");
</script>
</body>
</html>
```

Natijada HTML Web sahifa yuklanganda brauzer asosiy oynasida `“Salom bolalar”` matni hosil bo'ladi.

FUNKSIYALAR

Funksiya- bu maxsus yozuv bo'lib bu yozuvda dastur qismi joylashgan va dasturning ixtiyoriy qismidan turib uni chaqira olish mumkin, yani prosedura deb ataluvchi qism dasturga o'xshash blokdir. Java Scriptda funksiya yagona nomga ega. U bir yoki yoki bir necha parametrlarni qabul qiladi va natijalarni qaytaradi. Java Scriptda funksiyadan foydalanishdan maqsad dasturdagi ba'zi bir qismlar takrorlanuvchan bo'ladi. Ana shu takrorlanuvchi qismlarni qayta-qayta yozish natijasida dasturning hajmi ortib ketadi va uning ishlash tezligi kamayadi.

Undan tashqari Java Scriptda barcha jarayonlar funksiya tanasi ichida bajariladi. Dasturning ixtiyoriy joyidan turib uning nomi orqali chaqiriladi.

Demak qism dastur funksiyada joylashgan bo'lib, u funksiya tanasi deb ataladi.

Funksiyani e'lon qilishda **function** operatoridan foydalaniladi:

function {funksiya nomi}([argumentlar ro'xati])

Funksiya:

- Maxsus faoliyat(activity) yoki maqsad uchun.
- Bir paytni o'zida ko'p ishlarni bajaradi.
- Funksiyani e'lon qilish.

function nom(parm1, parm2)

```
{  
  Java Script kodi  
}
```

- Funksiyani chaqirish.
- Event Handler orqali HTML ichida funksiya chaqirish:
 - Event Handler="funksiya nomi (parm1ning value)";
- Java script ichida funksiya chaqirish

Misol 1:

```
function salom()  
{  
  alert('Salom');  
}
```

Funksiyaga ixtiyoriy nom berishingiz mumkin. Faqatgina funksiya nomida probel ishlatmang.

Misol 2:

```
<html>  
<body>  
<script language="JavaScript">  
  function welcome() {  
    document.write("Уй саҳифага хуш келибсизлар.<br>");  
  }  
  function thanks() {  
    document.write("Уй саҳифага кирганингиз учун раҳмат!<br>");  
  }  
  welcome();  
  thanks();  
</script>  
</body>  
</html>
```

Misol 3:

```
<html>  
<head>  
<script language="JavaScript">  
  function sum() {
```

```

var x = 10;
var y = 20;
var жавоб = x + y;
alert(жавоб)
}
</script>
</head>
<body>
<form>
<input type="button" value="жавоб" onClick="sum()">
//Event handler onClick
</form>
</body>
</html>

```

STANDART FUNKSIYALAR

-eval;
-escape;
-unescape;
-parseInt;
-parseFloat

Yuqoridagi funksiyalarni ishlatish uchun quyidagi ob'ektlardan foydalaniladi:

Math-matematik funksiyalarni qo'llash uchun kerak bo'ladigan ob'ekt.

String-Qatorlarni qayta ishlash ob'ekti.

Window-Muloqot(dialog) oynasini ochish ob'ekti.

1.eval('ifoda')

Misol:

```

Val=2; Ret=eval("2+Val"); alert(Ret); //natija 4 soni
eval('1+2+3') // natija 6 soni
eval('1'+4') //natija 14 soni
eval('1'+'+4') //natija 5 soni
eval('1'+'-4') // natija 3 soni

```

2.escape(ifoda | 'qator')

Funksiya argumenti qator yoki ifoda bo'lishi mumkin.Bu funksiya raqamlar, lotin alifbosi va "+", "-", "*", "_", "@" belgilarni o'zgartirmasdan qolgan simvollarni %XX kombinasiyasi bilan o'zgartiradi.

XX-o'n oltilik kod.

Misol:

```
escape("Salom dunyo!") // natija Salom%20dunyo%21
```

3.unescape(ifoda | 'qator')

Misol:

```
unescape("Salom%20dunyo%21") natija Salom dunyo!
```

4. parseInt

Qatorni butun songa aylantiruvchi usul yoki funksiyadir. **string --> integer**

Misol 1:

```

parseInt('.3') // natija 0.3
parseInt('0.1e6') // natija 100000

```

Misol 2:

```

var a="3"
var b="4";
var c1=a+b; --> c1="34";
var c2=parseInt(a) + parseInt(b);

```

5. parseFloat

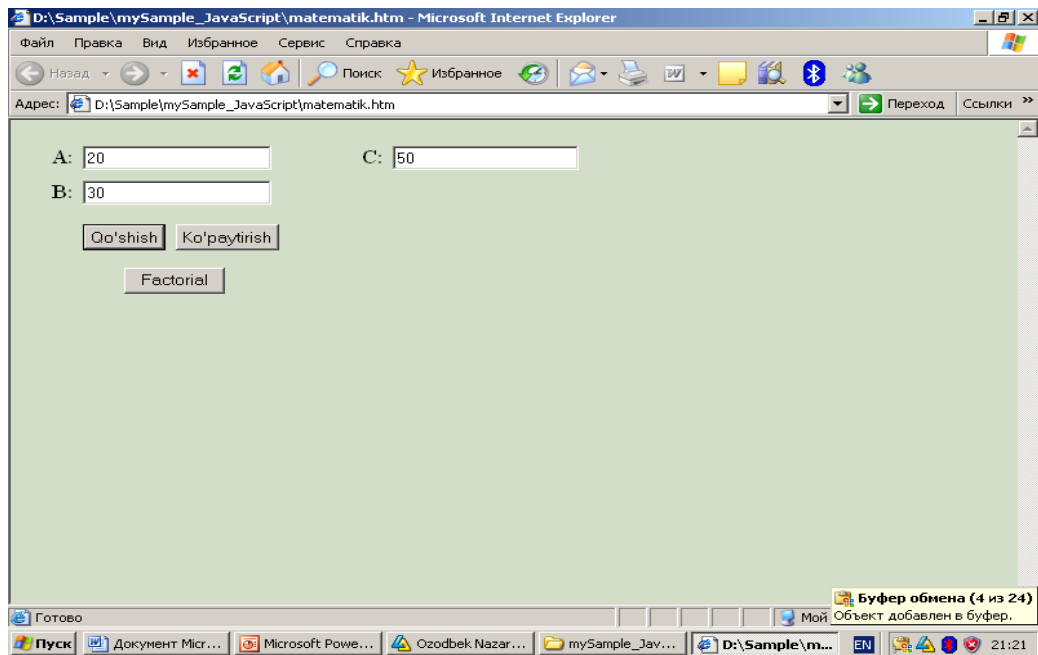
Qatorni haqiqiy songa aylantiruvchi usul. **string --> float**

Misol 1. Qo'shish va ko'paytirish amallarini bajaruvchi dastur kodini ko'raylik:

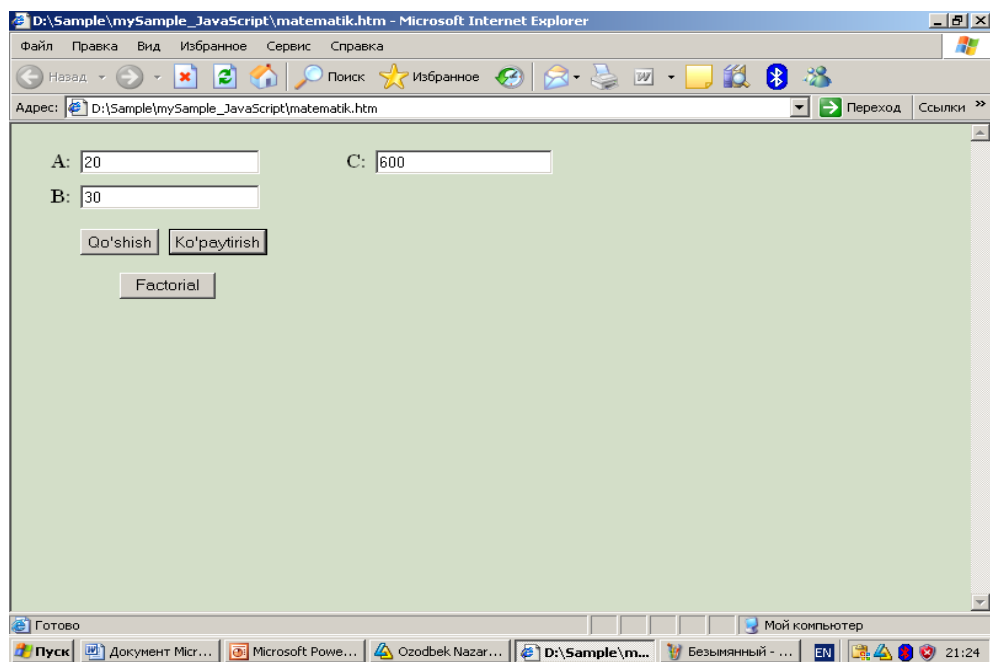
```

<html>
<head>
<title></title>
<script id="clientEventHandlersJS" language="javascript">-
function Add()
{
var x=myForm.qabul1.value;
var y=myForm.qabul2.value;
if ((x=="") || (y==" "))
alert("Qiymatlarini kiriting:");
else
{
var z=parseFloat(x)+parseFloat(y);
myForm.natija.value=z;
}
}
function Kopaytirish()
{
var x=myForm.qabul1.value;
var y=myForm.qabul2.value;
var z=parseFloat(x)*parseFloat(y);
myForm.natija.value=z;
}
</script>
</head>
<body bgColor="#d3dec8" ms_positioning="GridLayout">
<form name="myForm">
<INPUT id="Button1" type="button" name="Button1" onclick="Add()">
<INPUT id="Button2" type="button" name="Button2" onclick="Kopaytirish()">
<INPUT id="qabul2" type="text" name="qabul2">
<INPUT id="qabul1" type="text" name="qabul1">
<INPUT id="natija" type="text" size="18" name="natija">
</form>
</body>
</html>

```

Qo'shish jarayoni natijasi.



Ko'paytirish jarayoni natijasi.

QATORLAR BILAN ISHLOVCHI FUNKSIYALAR

Undan tashqari Java Script dasturlash tilida qatorlar bilan ishlash uchun tayyor funksiyalar mavjud. Ular quyidagilar:

1. **length** -bu funksiya qator uzunligini aniqlaydi.

```
var str='baxtiyor_007'
```

```
var sleng=str.length; // natija sleng=12
```

2. **charAt(son qiymati)** -bu funksiya o'zining parametri yordamida berilgan qatordagi ixtiyoriy simvol tanlab olinadi. Funksiya parametri 0 qiymatidan boshlanadi.

```
var str="javascript_3.0"
```

```
var myChar=str.charAt(2); // natija myChar="v" (index from 0)
```

3. substr(qiymat1,qiymat2)-bu funksiya berilgan string toifadagi ma'lumotdan nechanchi simvoldan boshlab nechta simvol tanlab olishni ta'minlaydi.

```
<html>
<head>
<script>
function test1()
{
var s, ss;
var s = "baxtiyor";
ss =s.substr(0,4); // natija ss="baxt"
alert(ss);
}
</script>
</head>
<body>
<input type="button" onclick="test1()">
</body>
</html>
```

5. indexOf('simvol')- bu funksiya mavjud qatordagi bizga kerak bo'lgan simvollar ketma-ketligi yoki bitta simvol joylashgan joyning son qiymatini aniqlaydi. Aniqlash jarayoni qatorning birinchi simvolidan boshlanadi.Index 0 dan boshlanadi (index from 0).

Agar berilgan qatordan kerakli simvol topilmasa natija -1 ga teng bo'ladi.

Oxirgi simvol qiymati **length-1** ga teng bo'ladi.

Misol 1:

```
var str="Java_texnologiyasi"
var myIndex=str.indexOf("t"); //natija myIndex=5
var myIndex=str.indexOf("h"); //natija -1
var myIndex=str.indexOf('v',3); //natija -1 (chunki,"v"simvol qiymati 2 )
var myIndex=str.indexOf('J',0); // natija 0
```

Misol 2:

```
<html>
<head>
<script>
function test()
{
var str1="baxtiyor"
var s =str1.indexOf("yor"); // natija s=5
alert(s);
}
</script>
</head>
<body bgcolor="#ccfcf">
<input type="button" value="tugma" onClick="return test()">
</body>
</html>
```

6. lastIndexOf('simvol') -bu funksiya ham mavjud qatordagi bizga kerak bo'lgan simvollar ketma-ketligi yoki bitta simvol joylashgan joyning son qiymatini aniqlaydi. Faqatgina indexOf

dan farqi shundaki, aniqlash jarayoni qatorning oxirgi simvolidan boshlanadi. Index 0 dan boshlanadi (index from 0).

Agar berilgan qatordan kerakli simvol topilmasa natija -1 ga teng bo'ladi.

Misol 1;

```
function test()
{
var str1="baxtiyor"
var s=str.lastIndexOf("yor"); // natija 5
alert(s);
}
```

Mosol 2:

```
var str1="Hello"
var s=str.lastIndexOf("l"); // natija 3
var s=str.lastIndexOf("l",2); // natija 2
var s=str.lastIndexOf("l",1); // natija -1
var s=str.lastIndexOf("Hell",1); // natija 0
var s=str.lastIndexOf('Hell'); // natija 0
```

7. toLowerCase() – bu funksiya rus yoki lotin alifbosidagi katta harflarni kichik harflarga almashtiradi.

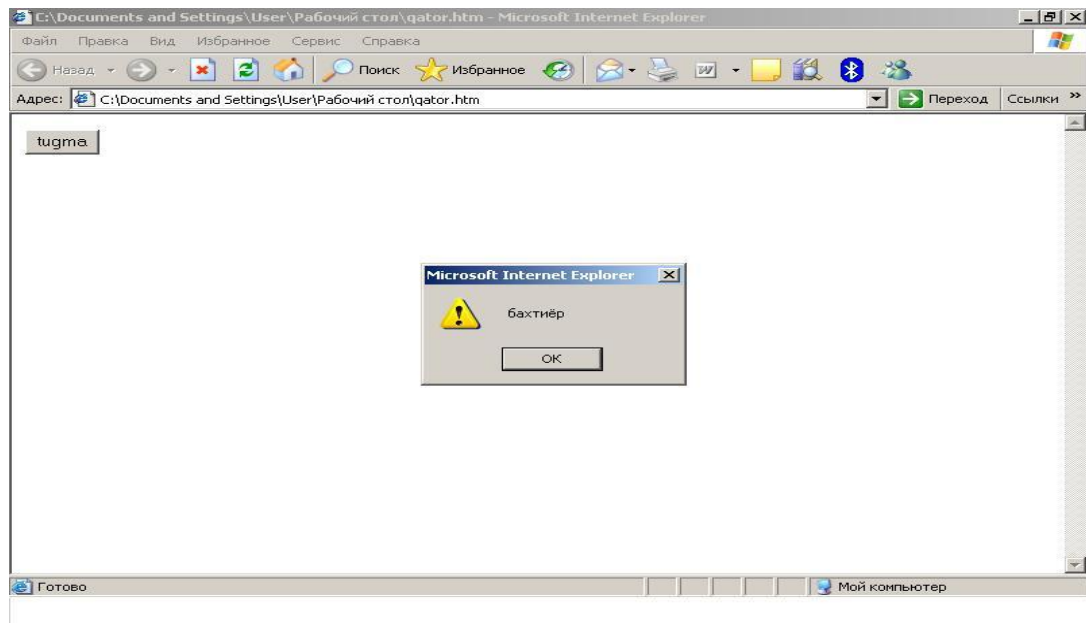
Misol 1:

```
'SAlom!'.toLowerCase(); // natija 'salom!'
```

Misol 2:

```
<html>
<head>
<script>
function my()
{
var str='БАХТИЁР';
var s=str.toLowerCase();
alert(s);
}
</script>
</head>
<body>
<input type="button" value="tugma" onClick="my()">
</body>
</html>
```

Natija:



8.toUpperCase()- bu funksiya rus yoki lotin alifbosidagi kichik harflarni katta harflarga almashtiradi.

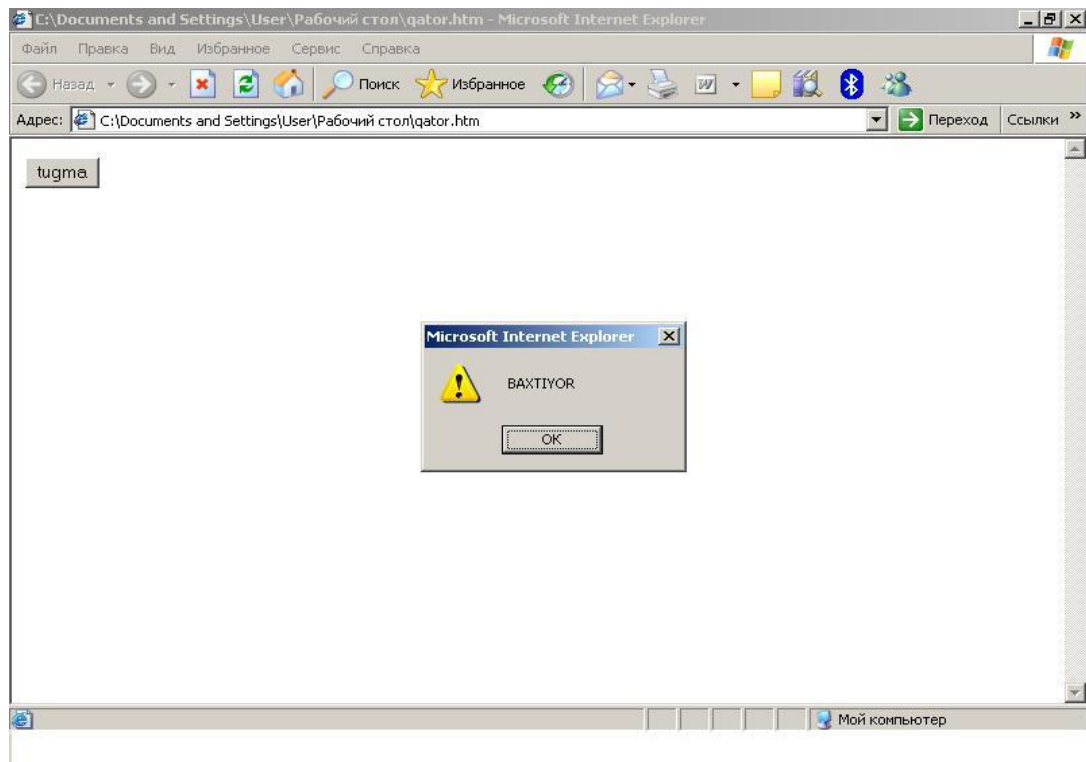
Misol 1:

`'Salom!'.toUpperCase();` // natija 'SALOM!'

Misol 2:

```
<html>
<head>
<script>
function my()
{
var str='baxtiyor';
var s=str.toUpperCase();
alert(s);
}
</script>
</head>
<body>
<input type="button" value="tugma" onClick="my()">
</body>
</html>
```

Natija:



9.charCodeAt(son qiymati) – bu funksiya yordamida 0-127 oraliqqa tushadigan simvol kodini aniqlash mumkin (ASCII).

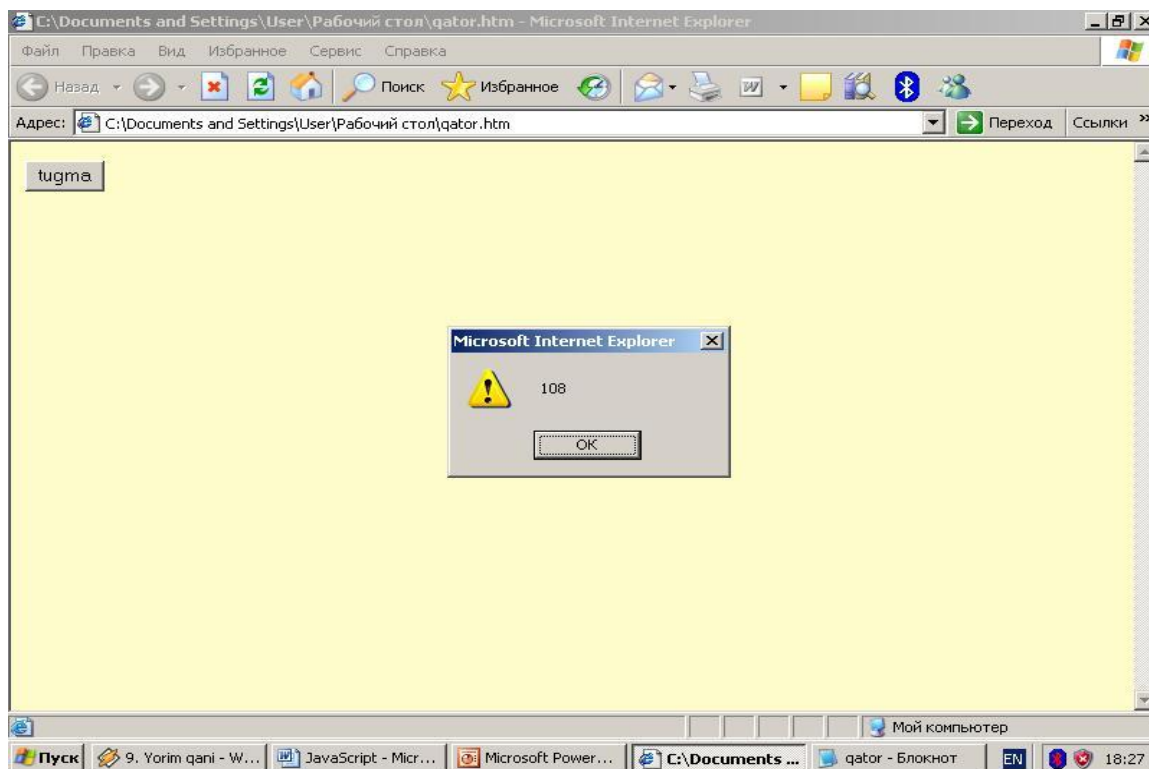
Misol 1:

```
var str='copy';
var myChar=str.charCodeAt(0); // natija myChar=99("c")
```

Misol 2:

```
<html>
<head>
<script>
function charAt()
{
var str='salom';
var s=str.charCodeAt(2);
alert(s);
}
</script>
</head>
<body>
<input type="button" value="tugma" onClick="charAt()">
</body>
</html>
```

Natija:



. **alert, prompt, confirm** -(Habar qutisi) bu funksiyalar habar beruvchi funksiyalar deyiladi.

alert: alert('string');

home

prompt: prompt('message', 'initial_value');

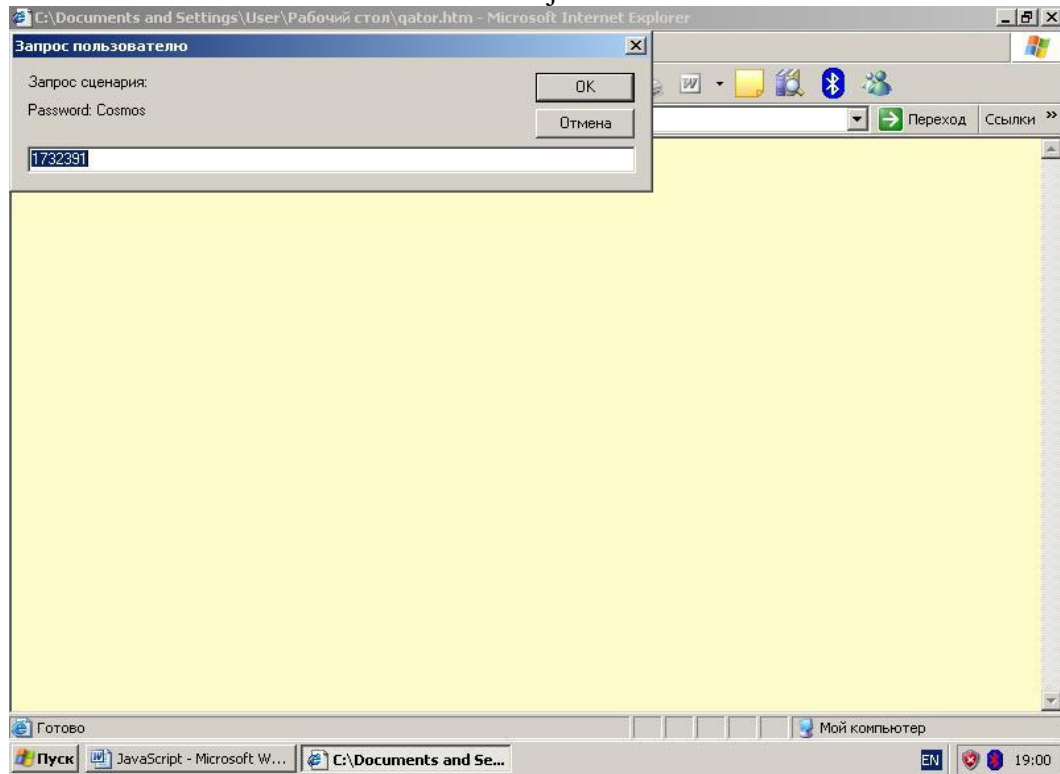
confirm: confirm('string');

Misol 1:

```
<html>
<head>
<script>
function habar()
{
    pass = prompt('Password: Cosmos', '1732391');
    if (pass == "1234")
        location.href = "matematik.htm";
    else
        alert("Password: Xato!");
}
</script>
</head>
<body bgcolor="ffcc">
<input type="button" value="tugma" onClick="habar()">
</body>
</html>
```

Natijada browserni yuklab tugma bosilgan vaqtda foydalanuvchi so'rovi oynasi hosil bo'ladi. Bunda ixtiyoriy senariy so'rovlarini amalga oshirish va boshqa sahifalarni chaqirish mumkin. Yoqoridagi misolda so'rov oynasiga kerakli kalit so'zini kiritib **"matematik.htm"** sahifasini yuklash jarayoni amalga oshirilgan.

Natija:



Misol 2:

```
function men()
{
  if (confirm('OK'))
    alert('Rahmat!')
  else
    alert('Hayr!')
}
```

Арифметик ifodalar

Арифметик ifodalarning qiymati sonlardan iborat bo'ladi va ularning natijasi ham faqat sonlardir.

1.Unar amal.

- -a // o'rin almashtirish amali;
- ++a // bittaga oshirish;
- a++
- --a // bittaga kamaytirish;
- a--

Misol 1:

```
function increment()
{
  var x=5;
  var y=5;
  alert(++x+" "+y++);
}
```



Misol 2:

```
function dicrement()
{
var x=5;
var y=5;
alert(++x+" "+y++);
}
```



Misol 3:

```
function arifmetik ()
{
var a=3,b=3;
var x=1,y=1;
x++;
++y;
a--;
--b;
alert("X="+x+" "+"Y="+y+" "+"A="+a+" "+"B="+b);
}
```



2.Binar amal.

- $a * b$ // ko'paytirish;
- a / b // bo'luv;
- $a \% b$ // a ni b ga bo'lgandan keyingi qoldiq;
- $a \& b$ // "VA" ("И") amali (поразрядное);
- $a | b$ // "YOQI" ("ИЛИ") amali (поразрядное).

3. Sdvigi(siljish , o'zgarish)

- $a \ll b$ // chapga siljish;
- $a \gg b$ // o'ngga siljish;

MATEMATIK FUNKSIYALAR

Matematik funksiyalar **Math** ob'yekti yordamida bajariladi. Bu ob'yekt tarkibida barcha matematik amallar shuningdek sin, cos, tan, abs, max va min, log, atan, asin, acos, exp, random, round sqrt kabi funksiyalar mavjud va ulardan Math ob'yekti yordamida foydalaniladi. Java Script dasturlash tilida ular quyidagicha ifodalanadi:

- Math.abs(x) - qiymatning modulini hisoblash;
- Math.acos(x) - x ning arkkosinusini hisoblash;
- Math.asin(x) - x ning arksinusini hisoblash;
- Math.atan(x) - x ning arktangensini hisoblash;
- Math.atan2(a,b) - arktanges a / b bo'yich hisoblash;
- Math.ceil(x) - x ga juda yaqin sonni topish;
- Math.cos(x) - x o'zgaruvchining kosinusini hisoblash;
- Math.sin(x) - x o'zgaruvchining sinusini hisoblash;
- Math.exp(x) - e^x ni hisoblash;
- Math.log(x) - natural logarifmni hisoblash;
- Math.max(a,b) - a va b sonlar ichidan eng kattasini topish;
- Math.min(a,b) - a va b sonlar ichidan eng kichigini topish;
- Math.pow(a,b) - (a^b) a ning b chi darajasini hisoblash;
- Math.random() ;
- Math.round(x) - x sonini butun qismini topish;
- Math.sqrt(x) - \sqrt{x} ni hisoblash;
- Math.tan(x) - x o'zgaruvchini tangansini hisoblash;

Misol :

```
<html>
<head>
<title></title>
<script id="clientEventHandlersJS" language="javascript">
function math_max()
{
var x=form1.Text1.value;
var y=form1.Text2.value;
var z=Math.max(x,y); // funsiya max
form1.Text3.value=z;
//alert(z);
}
function math_min()
{
var x=form1.Text1.value;
var y=form1.Text2.value;
var z=Math.min(x,y); /funksiya min
alert(z);
}
function math_pow()
{
var x=form1.Text1.value;
var y=form1.Text2.value;
var z=Math.pow(x,y);
var z1=Math.pow(x,2);
form1.Text3.value=z+" "+z1;
}
function math_sqrt()
```

```

{
var x=form1.Text1.value;
var z=Math.sqrt(x);
form1.Text3.value=z;
}
function math_sin()
{
var x=form1.Text1.value;
var y=form1.Text2.value;
var z=Math.sin(x/y);
alert(z);
}
function math_log()
{
var x=form1.Text1.value;
var y=form1.Text2.value;
var z=Math.log(x);
var z1=Math.log(y)/Math.log(x) //  $\log_x y$  funksijsi
alert(z+" " +z1);
}
function math_modul()
{
var x=form1.Text1.value;
var y=form1.Text2.value;
var z=Math.abs(x+y);
form1.Text3.value=z;
}
function math_round()
{
var x=3.296, y=4.571;
var z=Math.round(x); // natija 3
var z1=Math.round(y); // natija 5
form1.Text3.value=z+" "+z1;
}
function math_exp()
{
var x=form1.Text1.value;
var y=form1.Text2.value;
var z=Math.exp(x*y); // e darajasi x*y
form1.Text3.value=z;
}
function math_ceil()
{
var y=form1.Text2.value;
var z=Math.ceil(y);
form1.Text3.value=z;
//alert(z);
}
</script>
</head>

```

```

<body bgcolor="#99cccc" ms_positioning="GridLayout">
<form id="form1" name="form1">
<INPUT id="Text2" type="text" name="Text2" language="javascript" size="21">
<INPUT id="Text3" type="text" size="4" name="Text3">
<INPUT id="Text1" type="text" name="Text1" size="21">
<INPUT id="Button1" type="button" onclick="funksiya nomi">
</form>
</body>
</html>

```

MANTIQUIY IFODALAR

Mantiqiy ifodalarning qiymati ikkita: true va false qiymatlarini qabul qiladi.

1. **Mantiqiy amallar.**

- !a //mantiqiy “EMAS” (“HE”)
- a && b //mantiqiy “VA” (“И”)
- a || b // mantiqiy “YOKI” (“ИЛИ”)

2. **Solishtirish amali.**

- a < b //kichik
- a <= b // kichik yoki teng
- a > b // katta
- a >= b // katta yoki teng

Yuqoridagi solishtirish amali faqat son va qatorlarga nisbatan ishlatiladi.

Undan tashqari hamma toifadagi(tip) ma'lumotlar uchun qo'llaniladigan solishtirish amallari ham mavjud:

- a==b // teng (qiymatlat bo'yicha)
- a===b // teng (qiymatlar va toifalar bo'yicha)
- a!=b //teng emas

Mantiqiy ifodalarga doir misollarni hozir ko'rib o'tmaymiz.Shart va sikl operatorlaridan keyin mantiqiy ifodalarni misollar yordamida tushunib olasiz.

ADABIYOTLAR

1. JavaScript [Текст] : справочник / Рик Дарнелл. - СПб; М.; Харьков: Питер, 2008. - 191 с.
2. Бранденбау. JAVA SCRIPT. Сборник рецептов для профессионалов. СПб. 2010.
3. www.ZiyoNet.uz