



OSNOVE GEOINFORMATIKE

JavaScript

Skript jezici



- Obezbeđuju interaktivnost na web stranicama
- "Jednostavni" programski jezici
- Izvršavaju se u čitaču
- Ugrađuju se u HTML stranice
- Interpretirani jezik
 - nema kompajliranja
 - izvršava se momentalno

Skript



- Tag `<script>` specificira Script kod koji se pokreće direktno u browser-u
- Browser sve između tagova `<script>` i `</script>` smatra elementima skripta
- Tag `<script>` se može javiti bilo gde u HTML dokumentu
 - postoji razlika između `<head>` i `<body>` sekcije
 - kod definisan u tagu `<script>` u `<body>` sekciji se izvršava prilikom crtanja stranice
 - kod definisan u tagu `<script>` u `<head>` sekciji se ne izvršava automatski već se poziva iz skripta u `<body>` sekciji
- Ne mora kod da se nalazi u HTML datoteci
 - može i u drugoj datoteci, a da se pozove iz HTML datoteke
- Ako atribut **type** ima vrednost “**text/javascript**”, tada se radi o JavaScript programskom jeziku

Primer



```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
...
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<script type="text/javascript">
```

```
...
```

```
</script>
```

```
</body>
```

Primer skripta u datoteci



```
<html>
```

```
<head>
```

```
<script src="skript.js"></script>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

Js_external.html

JavaScript



- Sintaksa slična programskom jeziku Java
 - nije programski jezik Java
- Nema tipove podataka
 - kod deklaracije promenljivih se ne stavlja tip (interpreter).
- Nema kreiranja novih klasa
 - ugrađene funkcije,
 - ugrađeni objekti
- Sistem događaja

Pozivanje JavaScript-a



- Kao reakciju na neki događaj.
- Unutar `<script>` taga bilo gde unutar HTML dokumenta
 - Ako koristimo JavaScript funkciju, nju moramo da definišemo unutar `<head>` taga da bismo mogli da je pozivamo iz bilo kog JavaScript koda.
- Kao adresu unutar `<a>` taga:

```
<a href="javascript:funkcija( 'parametar' ) ;">  
klikni</a>
```

Promenljive



- Promenljive sadrže informacije
- Deklaracija promenljivih upotrebom ključne reči var
- Primer:

```
var a;
```

```
var b = 5;
```

```
var c = "Pera";
```

- Jezik definiše 5 primitivnih tipova podataka:
 - number, string, boolean, undefined, null

Vrednost promenljive



Non-zero value



null



0



undefined



Promenljive



- Nakon deklaracije, varijabla se može inicijalizovati:

```
var x;
```

```
x = 5;
```

- Inicijalizacija može i uz deklaraciju:

```
var x = 5;
```

- Varijabla može i da promeni tip:

```
var x = 5;
```

```
x=5.3;
```

```
x = "Mika";
```

Js_variables.html



Aritmetički i operatori dodele

- Aritmetički: `+` `-` `*` `/` `%` `++` `--`

```
x = 5;
```

```
y = x * 4;
```

```
z = y % 5;
```

- Dodele: `=` `+=` `-=` `*=` `/=` `%=`

```
y += 5;            y=y+5;
```

- Operator `+` ima posebno značenje kada su operandi stringovi:

```
a = "Pera";
```

```
b = "Car";
```

```
c = a + " " + b;
```

- Kada sabiramo stringove i brojeve, rezultat je string



Aritmetički operatori

$y = 5;$

Operator	Rezultat
$x=y+2$	$x=7$
$x=y-2$	$x=3$
$x=y\%2$	$x=1$
$x=++y$	$x=6, y=6$
$x=y++$	$x=5, y=6$
$x=--y$	$x=4$

Operatori dodele



$x = 10;$

$y = 5;$

Operator	Isto kao	Rezultat
$x=y$		$x=5$
$x+=y$	$x=x+y$	$x=15$
$x-=y$	$x=x-y$	$x=5$
$x*=y$	$x=x*y$	$x=50$
$x/=y$	$x=x/y$	$x=2$
$x\%=y$	$x=x\%y$	$x=0$

Relacioni operatori



- Relacioni: `==` `===` `!=` `<` `<=` `>` `>=`

```
x = 5;
```

```
if (x == 5)
```

```
    document.write("x je jednako 5");
```

- Operator `===` će porediti i vrednost i tip:

```
if (x === "5")
```

```
    document.write("x je string sa sadržajem 5");
```

- Rezultat relacionih operatora je logička vrednost tačno (true) ili netačno (false)

Relacioni operatori



`x = 5;`

Operator	Rezultat
<code>==</code>	<code>x == 8</code> je netačno (false)
<code>===</code>	<code>x == 5</code> je tačno (true) <code>x === "5"</code> je netačno (false)
<code>!=</code>	<code>x != 8</code> je tačno (true)
<code>></code>	<code>x > 8</code> je netačno (false)
<code><</code>	<code>x < 8</code> je tačno (true)
<code>>=</code>	<code>x >= 8</code> je netačno (false)
<code><=</code>	<code>x <= 8</code> je tačno (true)



Logički operatori

- Logički: `&&` `||` `!`
- Rezultat logičkih operatora je tačno (true) ili netačno (false)
- Operandi logičkih operatora su logički izrazi

<code>&&</code>	0	1
0	0	0
1	0	1

<code> </code>	0	1
0	0	1
1	1	1

<code>!</code>	
0	1
1	0

Logički operatori



`x = 6;`

`y = 3;`

Operator	Objašnjenje	Primer
<code>&&</code>	konjukcija (and, i)	<code>(x < 10 && y > 1)</code> tačno (true)
<code> </code>	disjunkcija (or, ili)	<code>(x==5 y==5)</code> netačno (false)
<code>!</code>	negacija (not, ne)	<code>!(x==y)</code> tačno (true)

Uslovni operator



- Sintaksa

`promenljiva = (uslov) ? vrednost1 : vrednost2`

- To je kao:

`if (uslov)`

`promenljiva = vrednost1;`

`else`

`promenljiva = vrednost2;`

- Primer:

`x = (y > 3) ? 5 : 6;`

Kontrola toka



- `if else`
- `switch`
- `for`
- `while`
- `do while`
- `break`
- `continue`

if else



- Opšta sintaksa:

```
if(uslov_1)
```

```
    telo_1
```

```
else if(uslov_2)
```

```
    telo_2
```

```
else
```

```
    telo_3
```

Js_if.html

Js_if_else.html

Primer



```
if (poeni > 94)
    ocena = 10;
else if (poeni > 84)
    ocena = 9;
else if (poeni > 74)
    ocena = 8;
else if (poeni > 64)
    ocena = 7;
else if (poeni > 54)
    ocena = 6;
else ocena = 5;
```

Primer



```
<script type="text/javascript">
var d = new Date();
var time = d.getHours();

if (time < 10)
{
    document.write("Dobro jutro!");
}
else
{
    document.write("Dobar dan!");
}
</script>
```

switch



- Izraz u **switch()** izrazu mora da proizvede celobrojnu vrednost.
- Ako ne proizvodi celobrojnu vrednost, ne može da se koristi **switch()**, već **if()**!
- Ako se izostavi **break**; propašće u sledeći **case**.
- Kod **default** izraza ne mora **break** - to se podrazumeva.

Js_switch.html

Primer



```
switch (a)
{
    case 1:
    case 2: i = j + 6;
            break;
    case 3: i = j + 14;
            break;
    default: i = j + 8;
}
```


Primer



```
<script type="text/javascript">
//Nedelja=0, Ponedeljak=1, Utorak=2, itd.
var d=new Date();
theDay=d.getDay();
switch (theDay)
{
case 5:
    document.write("Petak");
    break;
case 6:
    document.write("Subota");
    break;
case 0:
    document.write("Nedelja");
    break;
default:
    document.write("Jos nije vikend!");
}
</script>
```

While i do.. while



Difference between while &
do...while... 😄

while



- Za cikličnu strukturu kod koje se samo zna uslov za prekid.
- Telo ciklusa ne mora ni jednom da se izvrši
- Opšta sintaksa:

```
while (uslov)
```

```
telo
```

- Važno: izlaz iz petlje na false!

Js_while.html

Primer



```
<html>

<body>

<script type="text/javascript">

var i=0;

while (i<=10)

{

    document.write("Trenutno je " + i);

    document.write("<br />");

    i=i+1;

}

</script>

</body>

</html>
```

Primer



```
<html>

<body>

<script type="text/javascript">

//racunanje a na n

i = 1; a = 2; n = 3;

stepen = 1;

while (i++ <= n)

    stepen *= a;

document.write("a na n je " + stepen);

</script>

</body>

</html>
```

do while



- Za cikličnu strukturu kod koje se samo zna uslov za prekid
- Razlika u odnosu na while petlju je u tome što se telo ciklusa izvršava makar jednom.
- Opšta sintaksa:

do

telo

while (uslov) ;

- Važno: izlaz iz petlje na false!

Primer



```
<html>

<body>

<script type="text/javascript">

var i=0;

do

{

    document.write("The number is " + i);

    document.write("<br />");

    i=i+1;

}

while (i<0);

</script>

</body>

</html>
```

Laboratorija za geoinformatiku

for



- Za organizaciju petlji kod kojih se unapred zna koliko puta će se izvršiti telo ciklusa.
- Petlja sa početnom vrednošću, uslovom za kraj i blokom za korekciju.
- Opšta sintaksa:

```
for (inicijalizacija; uslov; korekcija)  
telo
```

Js_for.html

for



```
for (i = 0; i < 10; i++)
```

```
    document.write(i + "<br/>");
```

- može i višestruka inicijalizacija i step-statement:

```
for(i = 0, j = 1; i < 10 && j != 11; i++, j++)
```

- oprez (može da se ne završi):

```
var x;
```

```
for (x = 0; x != 10; x+=0.1) ...
```

Primer



```
<html>
<body>
<script type="text/javascript">
var i=0;
for (i=0; i <= 10; i++)
{
    document.write("The number is " + i);
    document.write("<br />");
}
</script>
</body>
</html>
```

break i continue



- **break** – prekida telo tekuće ciklične strukture (ili **case** dela) i izlazi iz nje.
- **continue** – prekida telo tekuće ciklične strukture i otpočinje sledeću iteraciju petlje.

break i continue



```
<html>
<body>
<script type="text/javascript">
var i=0;
for (i=0; i <= 10; i++)
{
    if (i==3)
    {
        break;
    }
    document.write("The number is " + i);
    document.write("<br />");
}
</script>
</body>
</html>
```

Primer – izlaz iz ugnježdene petlje



```
for (...)  
{  
    for (...)  
    {  
        ...  
        if (uslov)  
            break;  
    }  
}
```

for ... in petlja



- Za iteriranje kroz nizove
- Opšta sintaksa:

```
for (promenljiva in niz) {  
    ...  
}
```

Js_for_in.html

Primer



```
<html>

<body>

<script type="text/javascript">

var x;

var vozila = new Array();

vozila[0] = "Saab";

vozila[1] = "Volvo";

vozila[2] = "BMW";

for (x in vozila)

{

    document.write(vozila[x] + "<br />");

}

</script>

</body>

</html>
```

Funkcije



- Definicija funkcija unutar `<head>` taga:

```
function f(arg1, arg2) {  
    ...  
    return vrednost;  
}
```

- Poziv funkcije iz tela HTML dokumenta (unutar `<body>` taga)

`Js_call_fiunction*.html`

`Js_alert.html`

`Js_alert_multi.html`

Funkcije



```
<html>
  <head>
    <title>JavaScript</title>
    <script type="text/javascript">
      function ispis() {
        document.write("Drugi pasus, ali iz funkcije.");
      }
    </script>
  </head>
  <body>
    <h1>JavaScript funkcije</h1>

    <p>
      Tekst sledeceg pasusa je generisan pozivom funkcije koju smo mi napisali:
    </p>

    <p>
      <script language="JavaScript">
        ispis();
      </script>
    </p>
  </body>
</html>
```

Funkcije



Globalne promenljive



- Ako u java script-u postoji vise funkcija koje treba da rade sa istom promenljivom onda se definise globalna promenljiva

```
var globalVar="10";  
function dodaj()  
{ var zbir=globalVar+1;  
  alert(zbir);  
}  
function oduzmi()  
{ var razlika=globalVar-1;  
  alert(razlika);  
}
```

Pisanje skriptova za prozore



■ Otvaranje novog prozora

```
var  
childWindow=open(http://www.google.com,"googleWindow","height=400  
, width=400");
```

```
//pomeranje prozora
```

```
childWindow.moveTo(100,100);
```

```
//promena velicine prozora
```

```
childWindow.resizeTo(100,200)
```

Događaji



- Događaji se registruju i odrađuju *event handler*-ima
- U skoro svaki element se može staviti atribut tipa događaja koji ima kao vrednost ime funkcije koja će se aktivirati (*event handler*)
- Primer:

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

Događaji



Atribut	Događa se kada ...
onabort	se prekine učitavanje slike
onblur	element izgubi fokus
onchange	korisnik pomeni sadržaj polja
onclick	se klikne mišem na objekat
ondblclick	se dva puta klikne po objektu
onerror	se dogodi greška prilikom učitavanja dokumenta ili slike
onfocus	element dobije fokus
onkeydown	se pritisne taster
onkeypress	se pritisne, pa otpusti taster, ili se drži pritisnut
onkeyup	se otpusti taster
onload	se stranica ili slika učitava
onmousedown	se pritisne dugme miša
onmousemove	se miš pomera
onmouseout	miš izađe izvan zone elementa
onmouseover	miš pređe preko elementa
onmouseup	se otpusti dugme miša
onreset	se klikne na reset dugme
onresize	se prozoru ili frejmu promeni veličina
onselect	je tekst selektovan
onsubmit	se klikne na dugme submit u formi
onunload	korisnik napusti stranicu

Događaji



```
<html>
  <head>
    <title>JavaScript</title>
    <script type="text/javascript">
      function mis() {
        confirm("Da li ste sigurni?");
      }

      function greeting() {
        alert("Dobrodosli na ovu stranicu");
      }
    </script>
  </head>
  <body onload="greeting()">
    <h1>JavaScript događaji</h1>
    <p>
      Događaji koje svojom akcijom izazove korisnik ili web browser mogu biti
      obradjeni JavaScript kodom. Prilikom učitavanja ove stranice automatski će biti
      pozvana jedna JavaScript funkcija. Pored toga, možete da predjete misem preko ovog
      <a href="primer09.html" onmouseover="mis()">linka</a>.
    </p>
  </body>
</html>
```

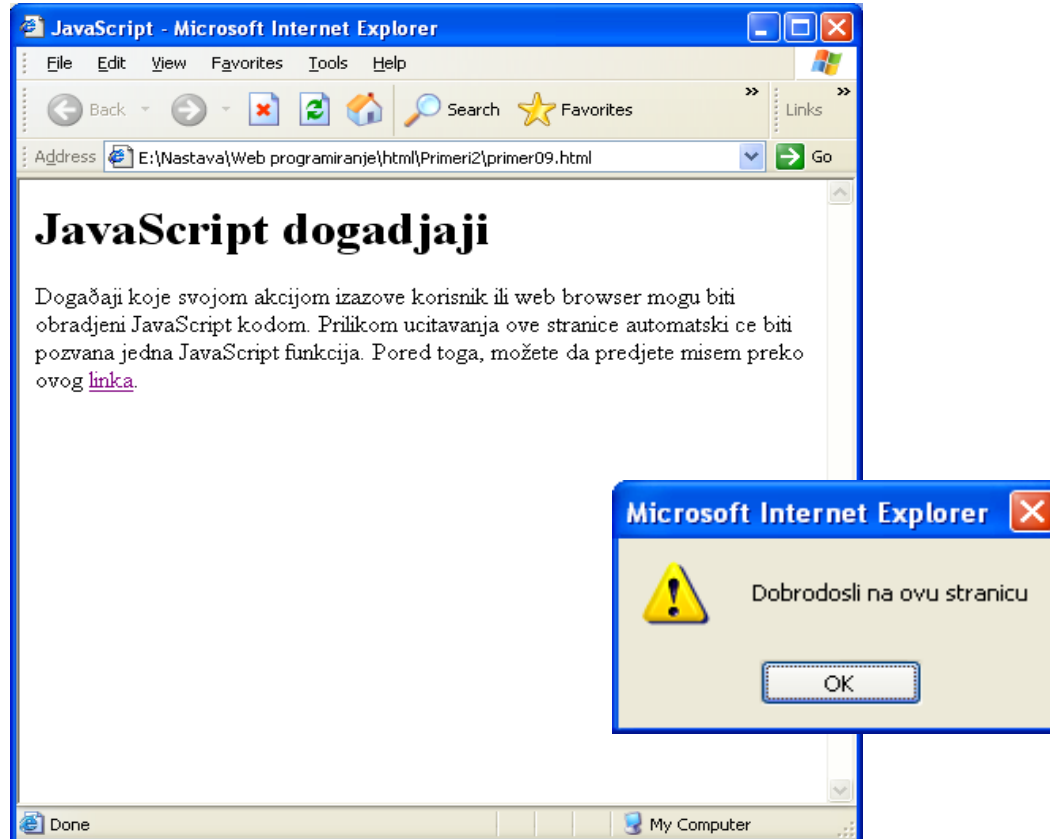
Js_confirm.html

Js_događaji.html

Js_head_alert.html

Js_call_function.html

Događaji



Pozivanje JavaScript-a



- Kao reakciju na neki događaj.
- Unutar `<script>` taga bilo gde unutar HTML dokumenta
 - ako koristimo JavaScript funkciju, nju moramo da definišemo unutar `<head>` taga da bismo mogli da je pozivamo iz bilo kog JavaScript koda.
- Kao adresu unutar `<a>` taga:

```
<a href="javascript:funkcija( 'parametar' ) ;">  
klikni</a>
```

Reakcija na neki događaj



```
<html>
  <head>
    <title>JavaScript</title>
    <script type="text/javascript">
      function greeting() {
        alert("Dobrodosli na ovu stranicu");
      }
    </script>
  </head>
  <body onLoad="greeting()">
    <h1>JavaScript događaji</h1>
    <p>
      Događaji koje svojom akcijom izazove korisnik ili web browser mogu
      biti obradjeni JavaScript kodom. Prilikom učitavanja ove stranice
      automatski će biti pozvana jedna JavaScript funkcija.
    </p>
  </body>
</html>
```

Reakcija na neki događaj



Preko `<script>` taga unutar `<body>` sekcije



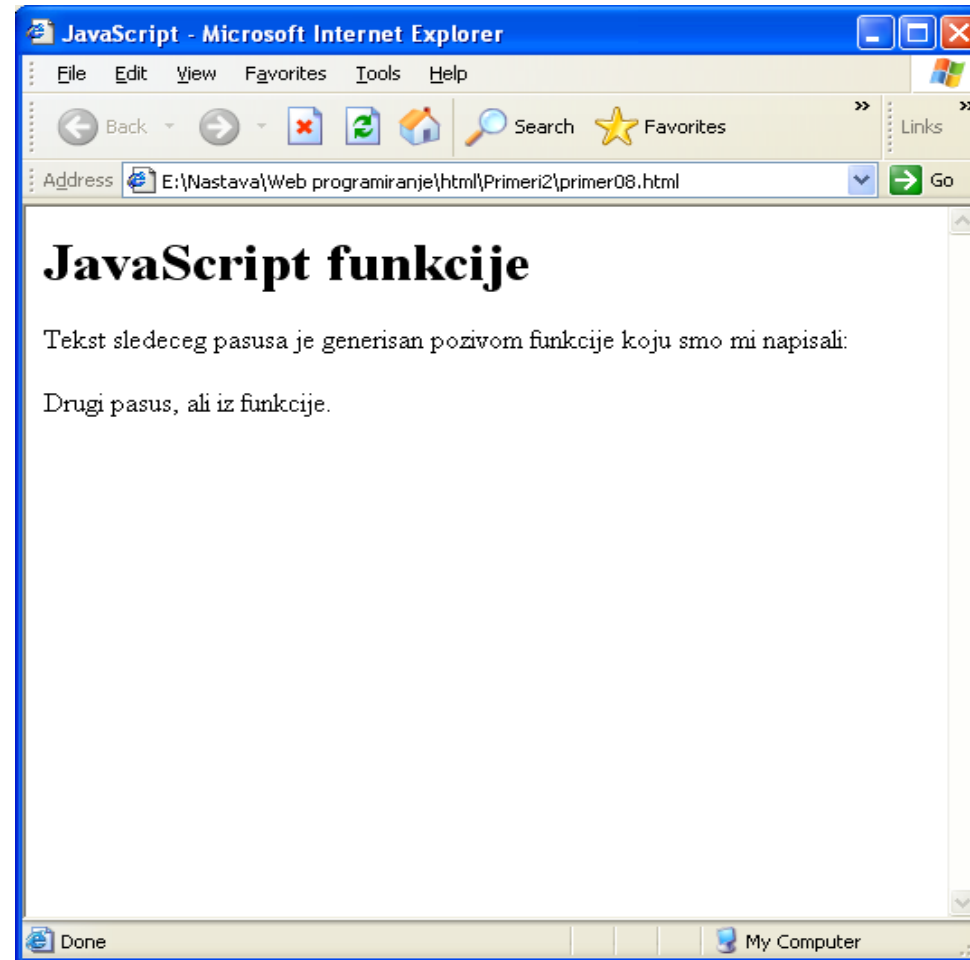
```
<html>
<head>
  <title>JavaScript</title>
  <script type="text/javascript">
    function ispis() {
      document.write("Drugi pasus, ali iz funkcije.");
    }
  </script>
</head>
<body>
  <h1>JavaScript funkcije</h1>

  <p>
    Tekst sledeceg pasusa je generisan pozivom funkcije koju smo mi napisali:
  </p>

  <p>
    <script language="JavaScript">
      ispis();
    </script>
  </p>
</body>
</html>
```

Js_dobar_dan.html

Preko `<script>` taga unutar `<body>` sekcije



Ugrađene funkcije



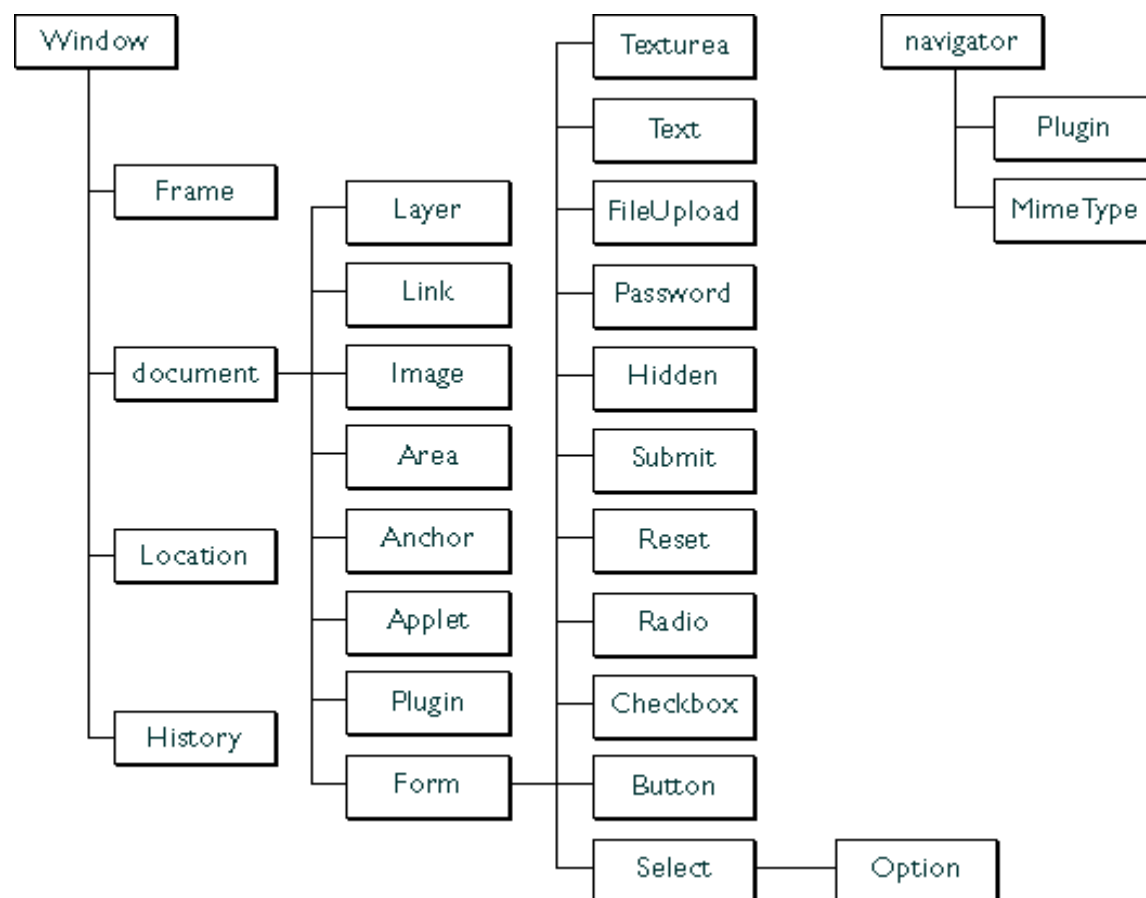
■ Sistemske funkcije:

- *isNaN()* – vraća true ako prosleđeni string nije broj,
- *eval()* – interpretira prosleđeni string kao JavaScript kod,
- *parseInt(string)* – parsira string u intidžer,
 - Bitno kod formi jer vrednosti koje se unose u polja se tretiraju kao stringovi, ovu funkciju koristimo ako nam treba broj koji cemo dalje da koristimo u aritmetickim funkcijama
- *parseFloat(string)* – parsira string u float promenljivu,
- *alert()* – ispis poruke u MessageBox-u
- *escape()*, *unescape()* – kodira/dekodira URL-ove (npr. zamenjuje razmak simbolom '+' i sl.)

Js_alert.html

Js_random.html

Objekat prozora -hijerarhija



Window objekt



- Omogućuje manipulaciju prozorima
- Sadrži informacije o tekućem prozoru
- Metode:
 - *alert()*, *confirm()*, *prompt()* - poruka u prozoru (MessageBox)
 - *back()*, *forward()* - povratak na prethodnu stranicu/odlazak na sledeću (iz istorije)
 - *moveBy()*, *MoveTo()* - pomera prozor
 - *open()* - otvara nov prozor
 - *setTimeout("kod", timeout)/clearTimeout()* – podešava/isključuje kod koji će se izvršavati kada istekne timeout
 - *setInterval("kod", perioda)/clearInterval()* – zadaje funkciju koja će se periodično izvršavati
- Atributi:
 - *history* - istorija odlazaka na stranice,
 - *document* - tekući HTML dokument,
 - *frames* - niz svih frejmova u prozoru,
 - *location* – kompletan URL tekuće stranice,
 - *statusbar* - statusna linija na dnu ekrana

Js_prompt.html

Confirm dijalog



- Dijalog za potvrdu, ako se klikne potvrdno izvršava se prvi alert, a ako se klikne odricno izvršava se drugi alert u else delu

```
if (confirm('Da li je lepo vreme?'))
```

```
    alert('Idemo napolje!');
```

```
else
```

```
    alert('Ostacemo unutra!');
```

Location objekt



- Reprezentuje URL stranice koja je učitana u navigator:

```
location = "http://www.google.com"
```

- Sadrži informacije o tekućem dokumentu
- Metode:
 - *reload()* - ponovno učitavanje tekućeg prozora primer: `location.reload(true);`
 - *replace()* - učitava novi URL
- Atributi:
 - *href* – pun URL do stranice:
 - Ovako se može pristupiti drugoj stranici:

```
location.href="http://www.google.com"
```

- *protocol* – protokol iz URL-a
- *host* – adresa servera iz URL-a
- *port* – port iz URL-a
- *pathname* – putanja do resursa
- *search* – parametri forme

History objekt



- Omogućuje kontrolu pristupa već viđenim stranicama
- Sadrži listu adresa posećenih stranica
- Metode:
 - *back()* - učitava prethodnu stranicu iz liste
 - *forward()* – učitava sledeću stranicu iz liste
 - *go()* - učitava zadatu adresu iz liste
- Atributi:
 - *current* – trenutno učitana adresa
 - *length* – broj stavki u history listi
 - *next* – zadavanje sledećeg elementa
 - *previous* – zadavanje prethodnog elementa

Document objekt



- Omogućuje ispis HTML-a na ekran
- Sadrži informacije o tekućem dokumentu
- Metode:
 - *write()* - ispisuje na ekran tekst
- Atributi:
 - *forms* - niz svih formi u dokumentu
 - *links* - niz svih linkova u dokumentu
 - *applets* - niz svih apleta u dokumentu
 - *title* - sadržaj **title** taga

String objekt



■ Reprezentuje string

- string konstanta "tekst" reprezentuje string

■ Metode:

- *substring()* – vraća deo stringa
- *split('?')* – vraća niz stringova kao rezultat "razbijanja" stringa
- *indexOf()*, *lastIndexOf()* – vraća poziciju nekog podstringa
- *charAt()* – vraća karakter sa zadate pozicije

■ Atributi:

- *length* – dužina stringa

Forme



- Reprezentovane **form** objektom.
- Metode:
 - *submit()* - šalje podatke iz forme na odredište definisano **action** atributom **form** taga.
 - *reset()* - simulira pritisak na Reset dugme forme.
- Atributi:
 - *elements* - niz elemenata forme. Svaki element ima **value** atribut za pristup sadržaju,
 - *length* - broj elemenata na formi.
 - *action* - sadržaj action atributa.

Js_forme.html

Forma



- Preuzimanje vrednosti polja u javascriptu

```
var vrednost = document.forms['forma'].polje1.value;
```

- Ako se proverava checkbox ili radiobutton

```
var remember= document.forms['forma'].polje2;
```

```
if (remember.checked)
```

```
{.....}
```

```
else
```

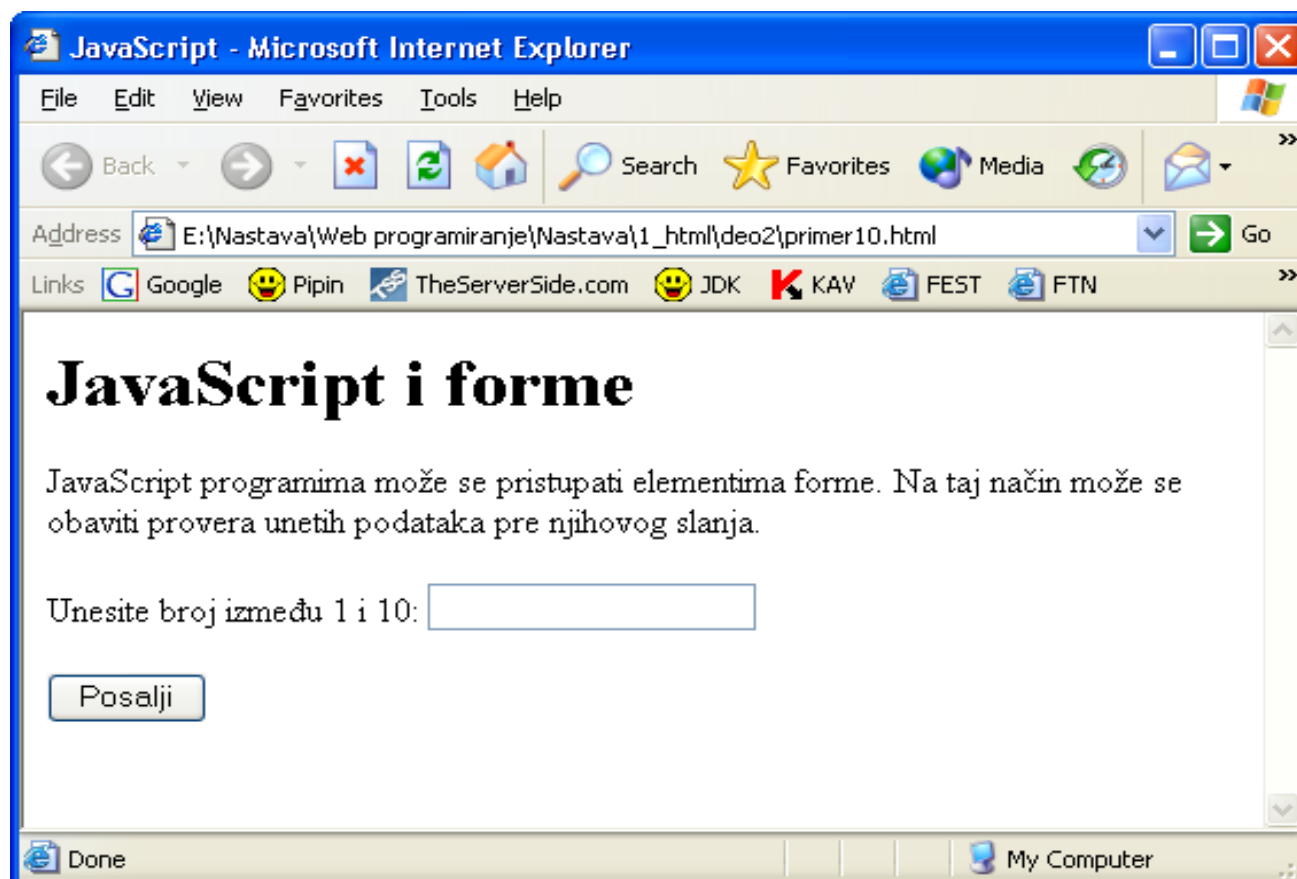
```
{..... }
```

Forme



```
<html>
<head>
  <title>JavaScript</title>
  <script type="text/javascript">
    function provera() {
      vrednost = document.forms['forma'].polje1.value;
      if (isNaN(vrednost)) {
        alert("Niste uneli broj");
        return false;
      } else if (vrednost >= 1 && vrednost <= 10) {
        return true;
      } else {
        alert("Niste uneli broj u opsegu od 1 do 10");
        return false;
      }
    }
  </script>
</head>
<body>
  <h1>JavaScript i forme</h1>
  <p>
    JavaScript programima može se pristupati elementima forme. Na taj način može se obaviti provera unetih podataka pre njihovog slanja.
  </p>
  <form name="forma" action="primer10-2.html" onSubmit="return provera()">
    <p>
      Unesite broj između 1 i 10:
      <input type="text" name="polje1"> <br><br>
      <input type="submit" name="polje2" value=" Posalji ">
    </p>
  </form>
</body>
</html>
```


Forme



Forme



```
<html>

  <head>
    <title>JavaScript</title>
  </head>

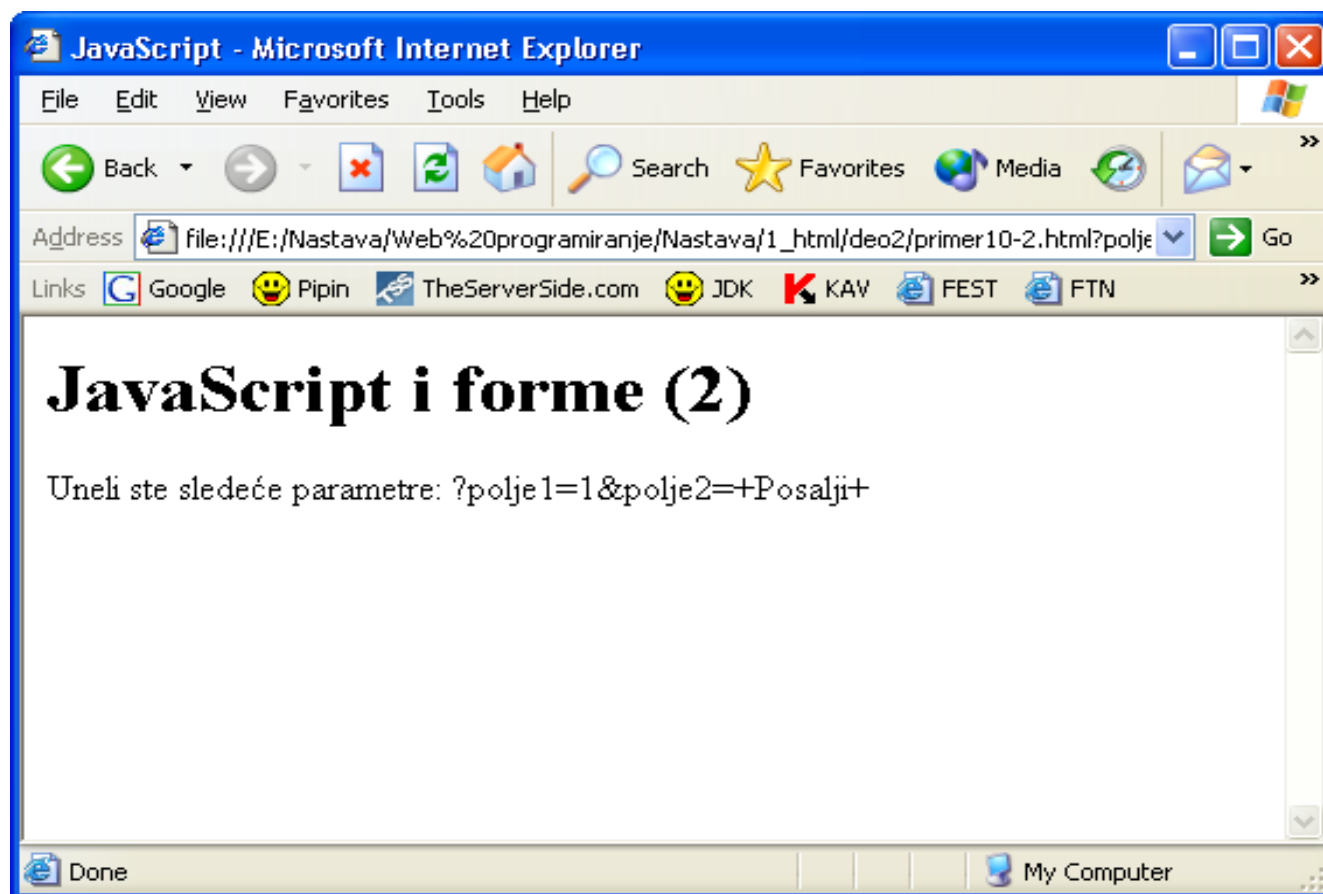
  <body>
    <h1>JavaScript i forme (2)</h1>

    <p>
      Uneli ste sledece parametre:
      <script type="text/javascript">
        document.write(window.location.search) ;
      </script>
    </p>

  </body>

</html>
```

Forme



Forme



```
<html>
```

```
<head>
```

```
  <title>JavaScript</title>
```

```
</head>
```

```
<body>
```

```
  <h1>JavaScript i forme (2)</h1>
```

```
  <p>
```

Uneli ste sledece parametre:

```
  <script type="text/javascript">
```

```
    var url=window.location.search;
```

```
    var parovi = url.split('?')[1];
```

```
    var param = parovi.split('&');
```

```
    for(var i = 0; i < param.length; i++) {  
      /* document.write(param[i].split('=')[0]);
```

```
        document.write('\n');
```

```
        document.write(param[i].split('=')[1]);
```

```
        document.write('\n');
```

```
        document.write('\n');*/
```

```
        if (param[i].split('=')[0]=='polje1')
```

```
        {
```

```
          document.write(param[i].split('=')[1]);
```

```
        }
```

```
    }
```

```
  </script>
```

```
</p>
```

```
</body>
```

```
</html>
```

Forme



JavaScript i forme (2)

Uneli ste sledece parametre: 6