Web programiranje

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>
```

JavaScript

- Sintaksa slična programskom jeziku Java
  - nije programski jezik Java
- Nema tipove podataka
  - kod deklaracije promenljivih se ne stavlja tip (interpreter)
    - JIT (Just In Time compiler)
- 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";
```

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 = "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
==	x == 8 je netačno (false)
===	x === 5 je tačno (true) x === "5" je netačno (false)
!=	x != 8 je tačno (true)
>	x > 8 je netačno (false)
<	x < 8 je tačno (true)
>=	x >= 8 je netačno (false)
<=	x <= 8 je tačno (true)

Logički operatori

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

&&	0	1
0	0	0
1	0	1

||	0	1
0	0	1
1	1	1

!	
0	1
1	0

Logički operatori

x = 6;

y = 3;

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

- 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<10);
</script>
</body>
</html>
```

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

Js_for_loop_headers.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.

Js_loop*.html

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();
mycars[0] = "Saab";
mycars[1] = "Volvo";
mycars[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;
}
```

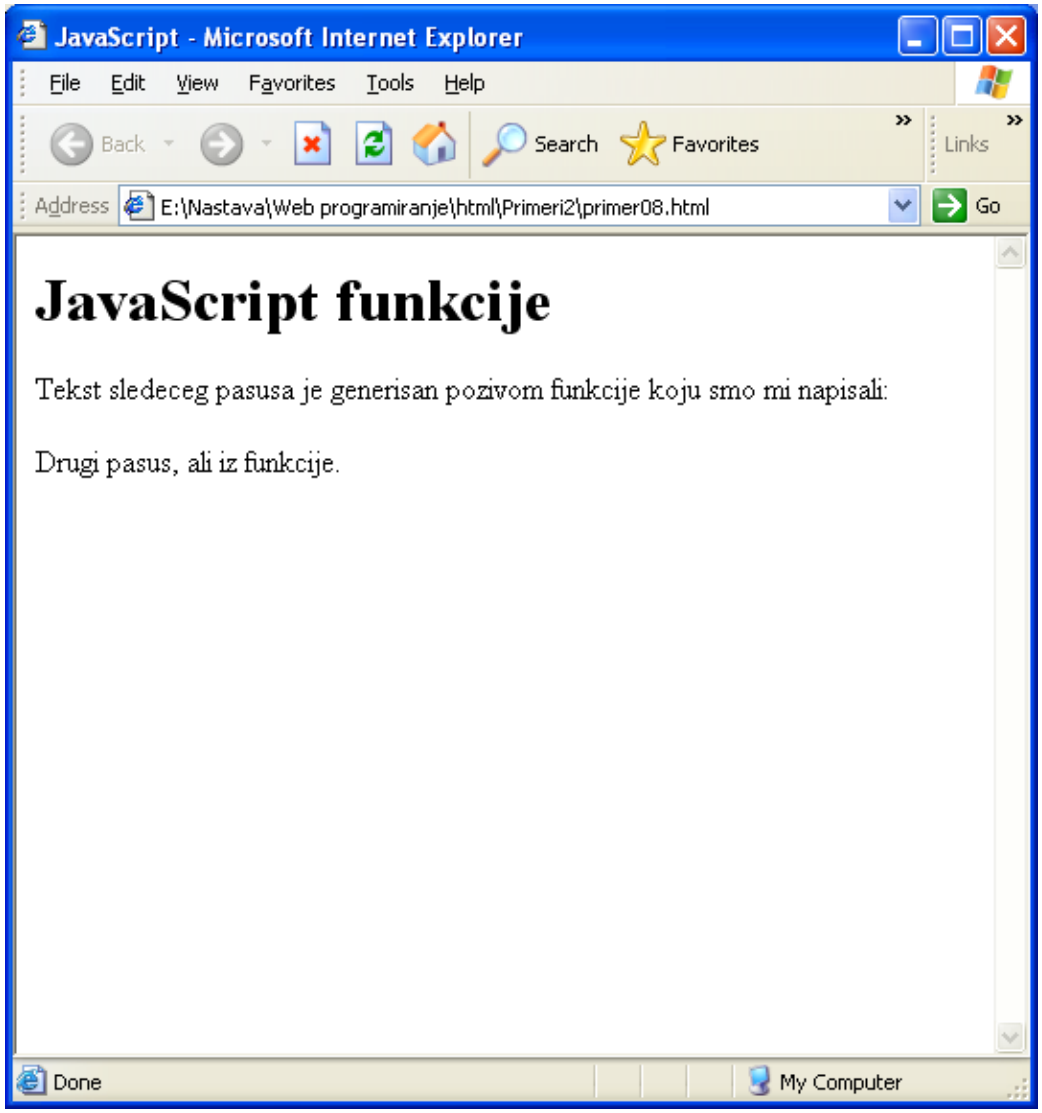
- Poziv funkcije iz tela HTML dokumenta (unutar <body> taga)

Js_call_function*.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


JavaScript - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Links
Address E:\Nastava\Web programiranje\html\Primeri2\primer08.html
Go
JavaScript funkcije
Tekst sledeceg pasusa je generisan pozivom funkcije koju smo mi napisali:
Drugi pasus, ali iz funkcije.
Done
My Computer

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 pormeni 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 učita
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 subit u formi
onunload	korisnik napusti stranicu

Događaji

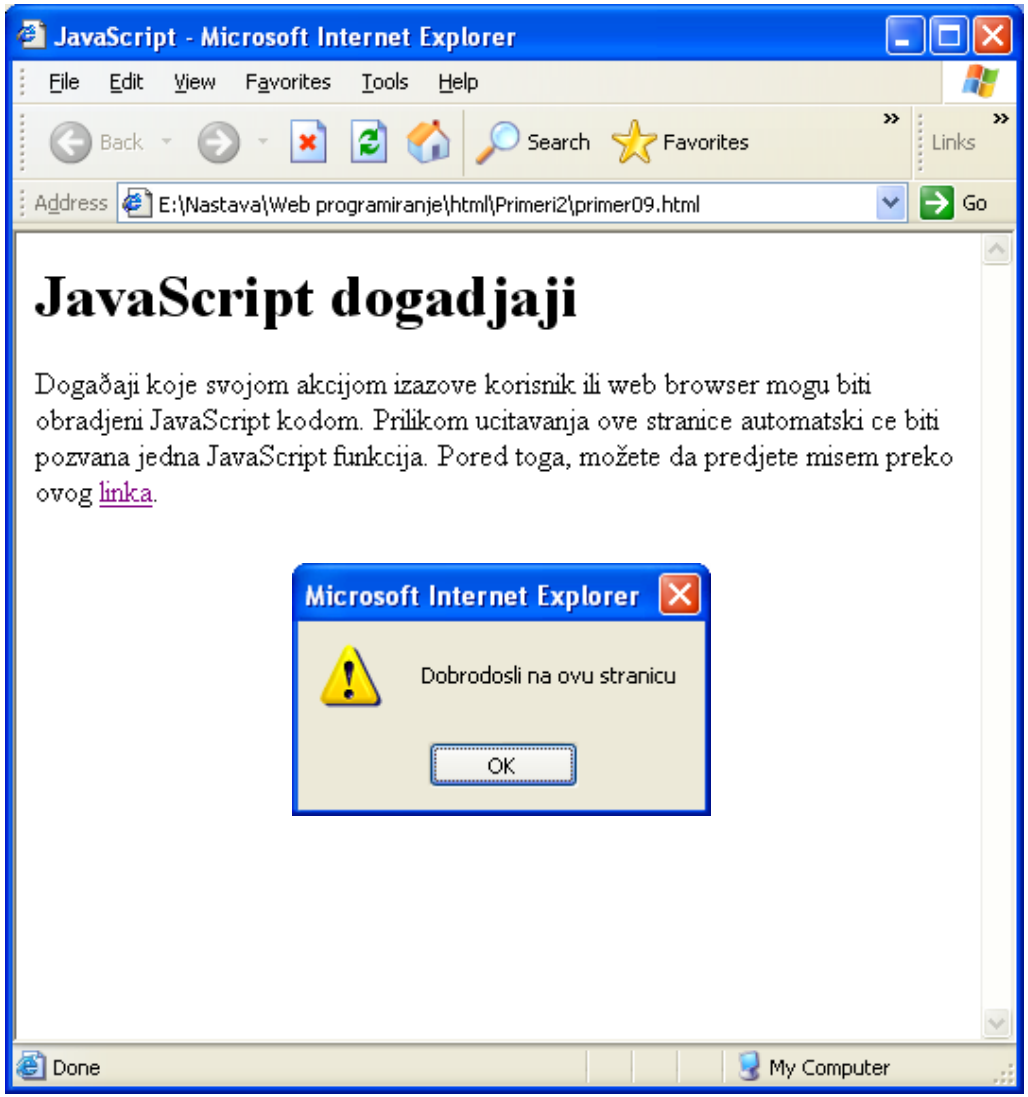
```
<html>
  <head>
    <title>JavaScript</title>
    <script type="text/javascript">
       function mis() {
         confirm("Da li ste sigurni?");
       }

       function greeting() {
         alert("Dobrodosli na ovu stranicu");
       }
    </script>
  </head>
  <body onload="greeting()">
    <h1>JavaScript dogadjaji</h1>
    <p>
      Dogadjaji koje svojom akcijom izazove korisnik ili web browser mogu biti
    obradjeni JavaScript kodom. Prilikom ucitavanja ove stranice automatski ce
    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_dogadjaji.html

Js_head_alert.html

Js_call_function.html

Događaji


JavaScript - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Links
Address E:\Nastava\Web programiranje\html\Primeri2\primer09.html
Go
JavaScript dogadjaji
Događaji koje svojom akcijom izazove korisnik ili web browser mogu biti obradjeni JavaScript kodom. Prilikom ucitavanja ove stranice automatski ce biti pozvana jedna JavaScript funkcija. Pored toga, možete da predjete misem preko ovog linka.
Microsoft Internet Explorer
Dobrodosli na ovu stranicu
OK
Done
My Computer

Pozivanje JavaScript-a

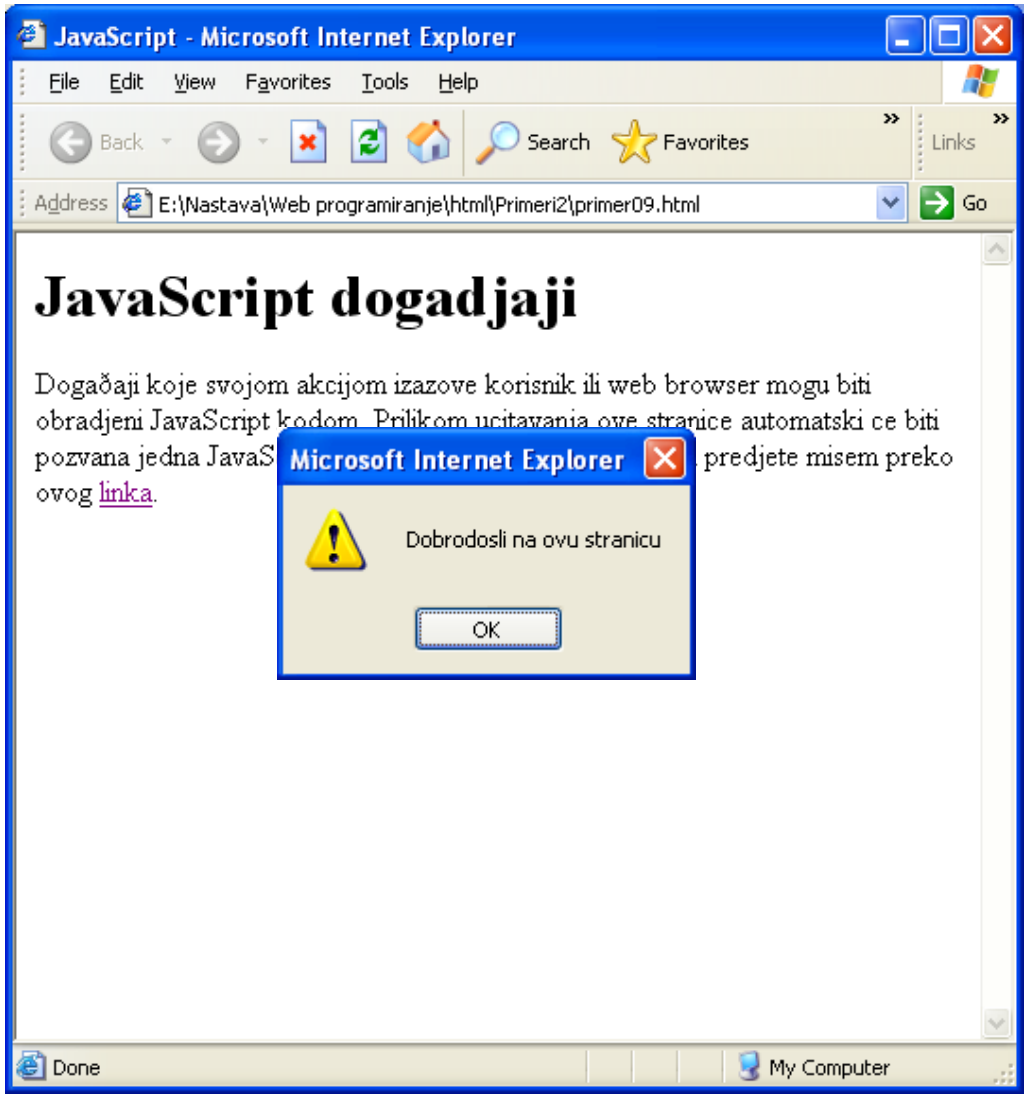
- Kao reakciju na neki događaj.
- Unutar <script> taga bilo gde unutar HTML dokumenta
  - ako koristimo JavaScript funkciju, nju moramo da definišemo unutar <head> taga da bismo mogli da je pozivamo iz bilo kog JavaScript koda.
- Kao adresu unutar <a> taga:

```
<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 dogadjaji</h1>
    <p>
      Dogadjaji koje svojom akcijom izazove korisnik ili web
   browser mogu biti obradjeni JavaScript kodom. Prilikom
   ucitavanja ove stranice automatski ce biti pozvana jedna
   JavaScript funkcija.
    </p>
  </body>
</html>
```

Reakcija na neki događaj

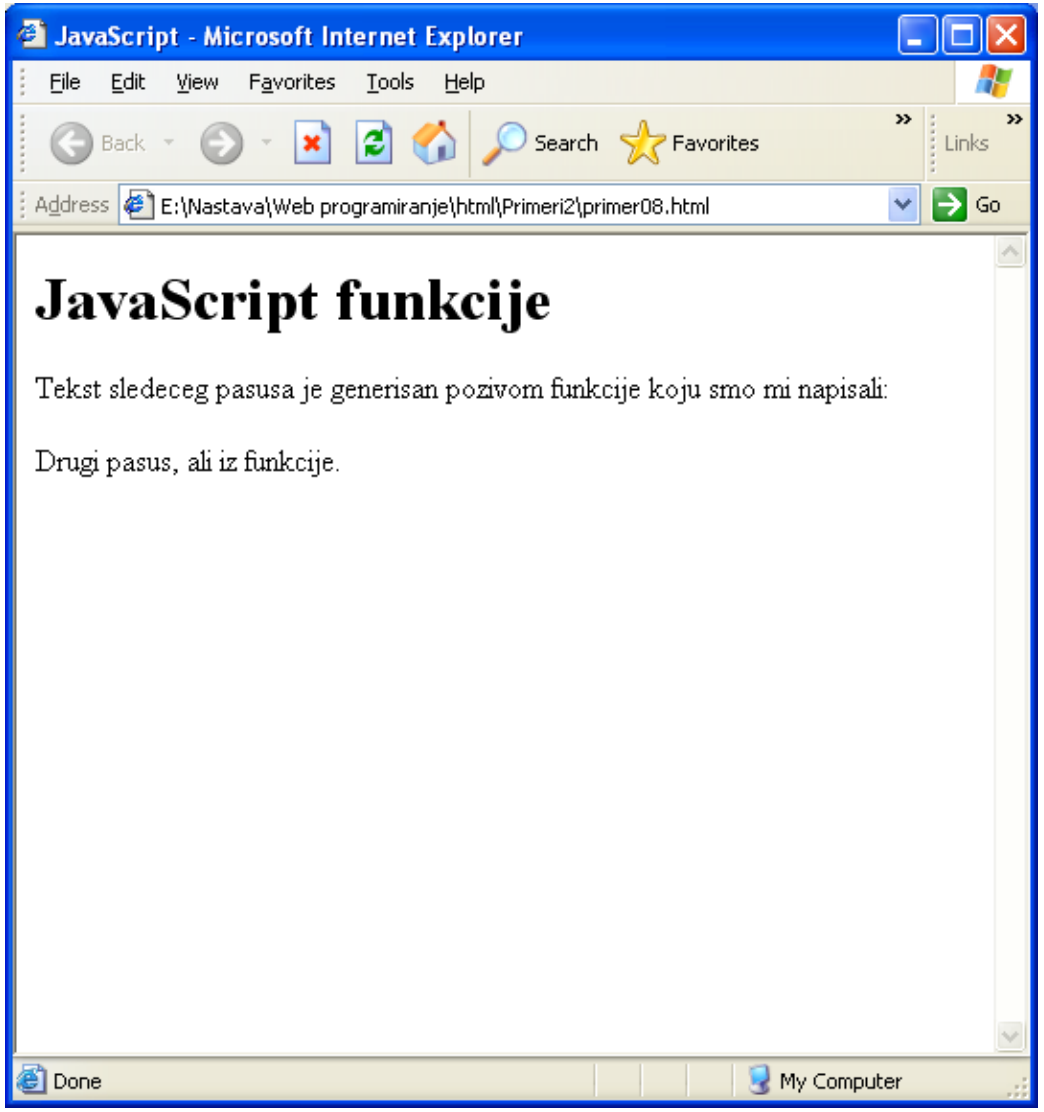


Preko <script> taga unutar <body> sekcije

```
<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

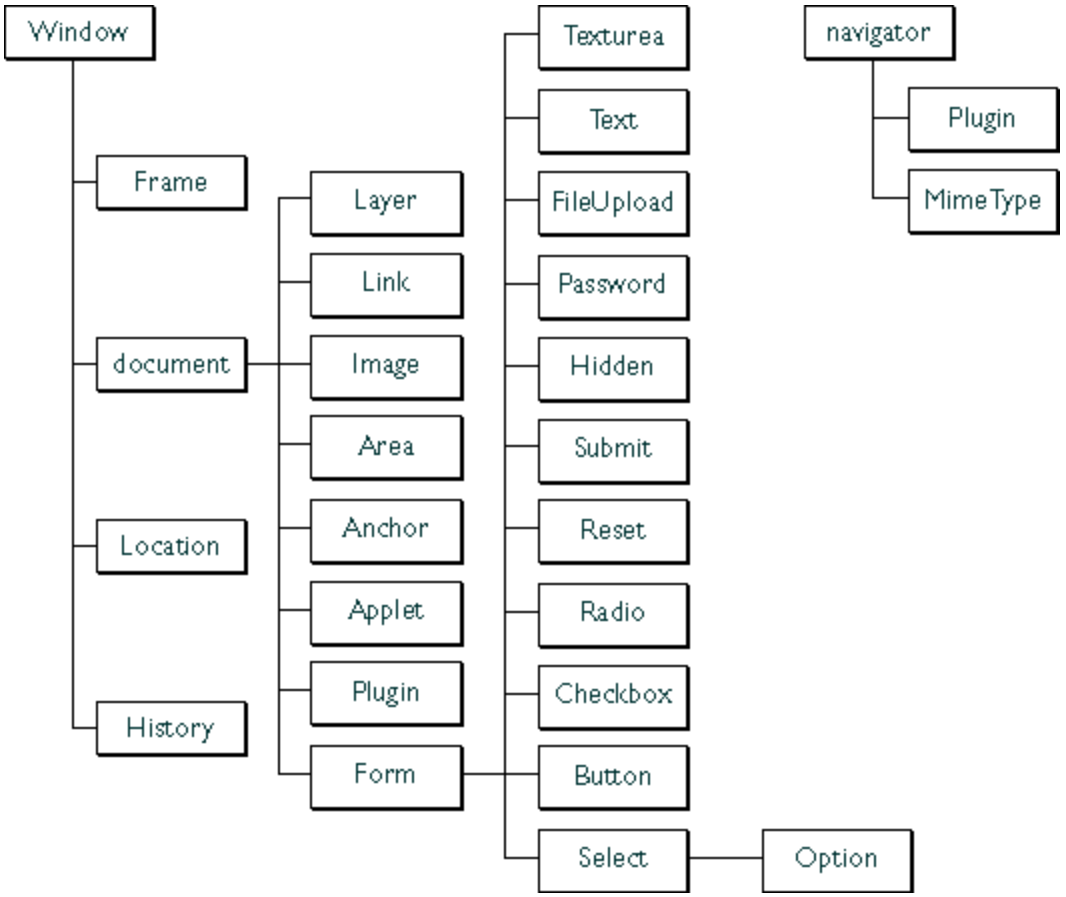


Ugrađene funkcije

- Sistemske funkcije:
  - *isNaN*() – vraća true ako prosleđeni string nije broj,
  - *eval*() – interpretira prosleđeni string kao JavaScript kod,
  - *parseInt*() – parsira string u intidžer,
  - *parseFloat*() – 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

Hijerarhija objekata



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

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
  - *replace*() - učitava novi URL
- Atributi:
  - *href* – pun URL do stranice:

`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* - sadrzaj **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.

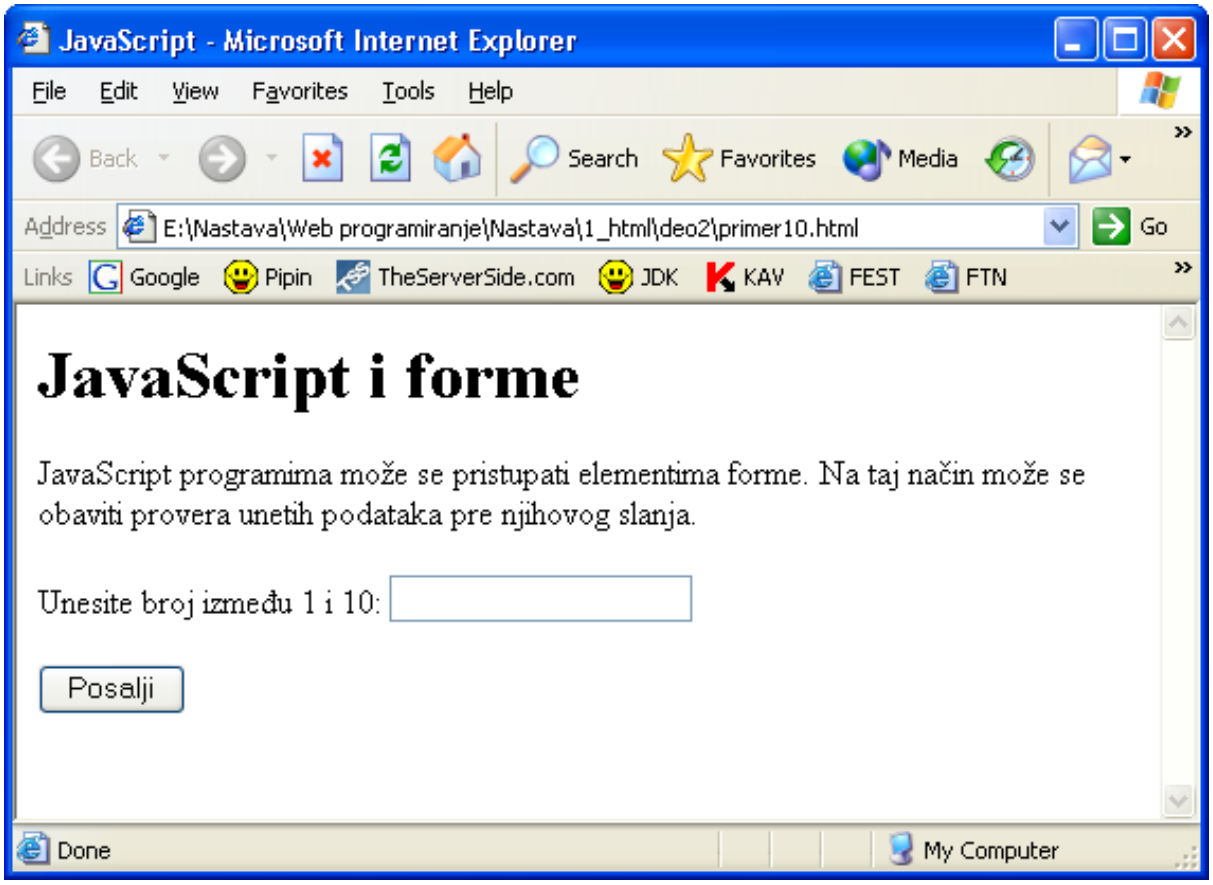
Js_forme.html

HTML\3_Forme.html

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
    nacin  moze 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>
```

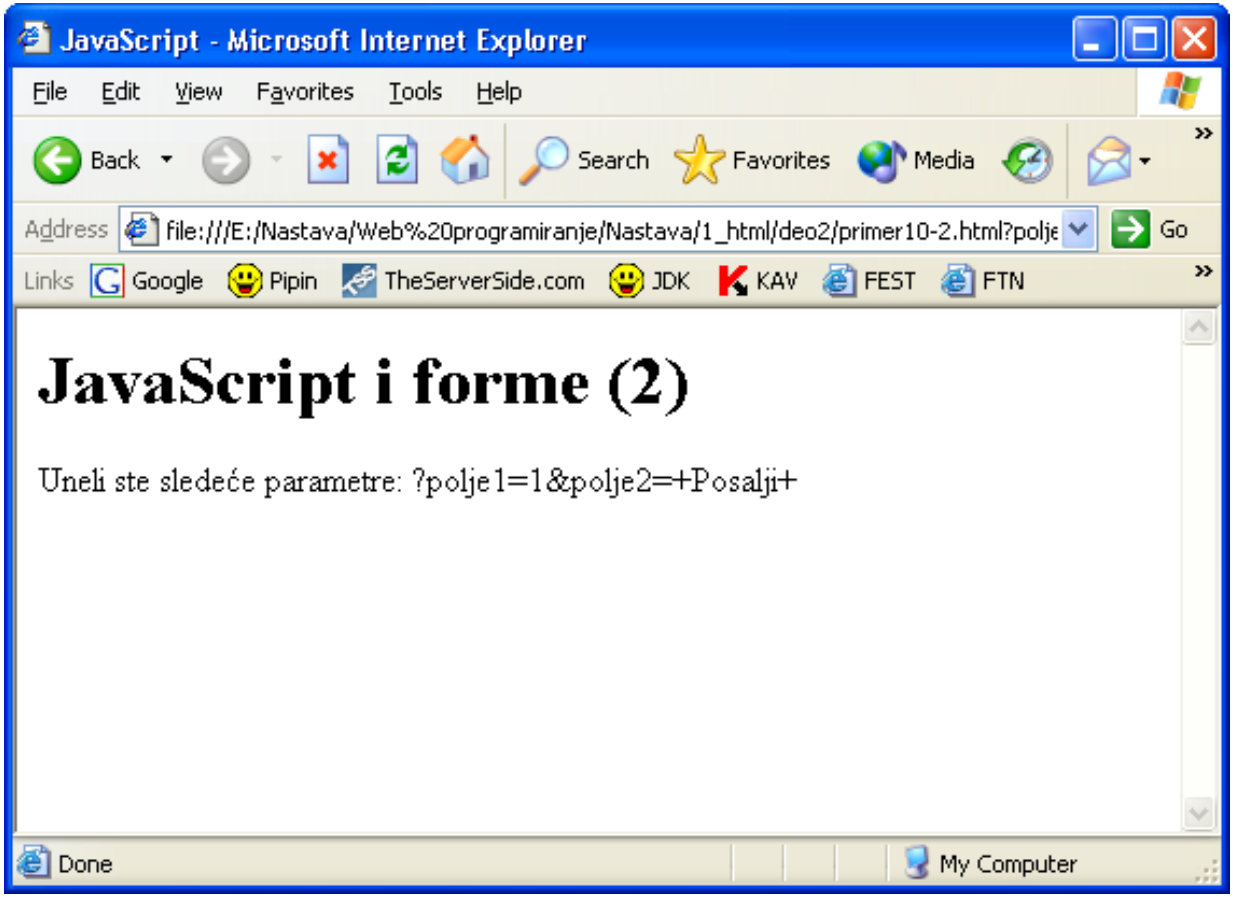
Forme


JavaScript - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address E:\Nastava\Web programiranje\Nastava\1_html\deo2\primer10.html
Go
Links Google Pipin TheServerSide.com JDK KAV FEST FTN
JavaScript i forme
JavaScript programima može se pristupati elementima forme. Na taj način može se obaviti provera unetih podataka pre njihovog slanja.
Unesite broj između 1 i 10:
Posalji
Done
My Computer

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

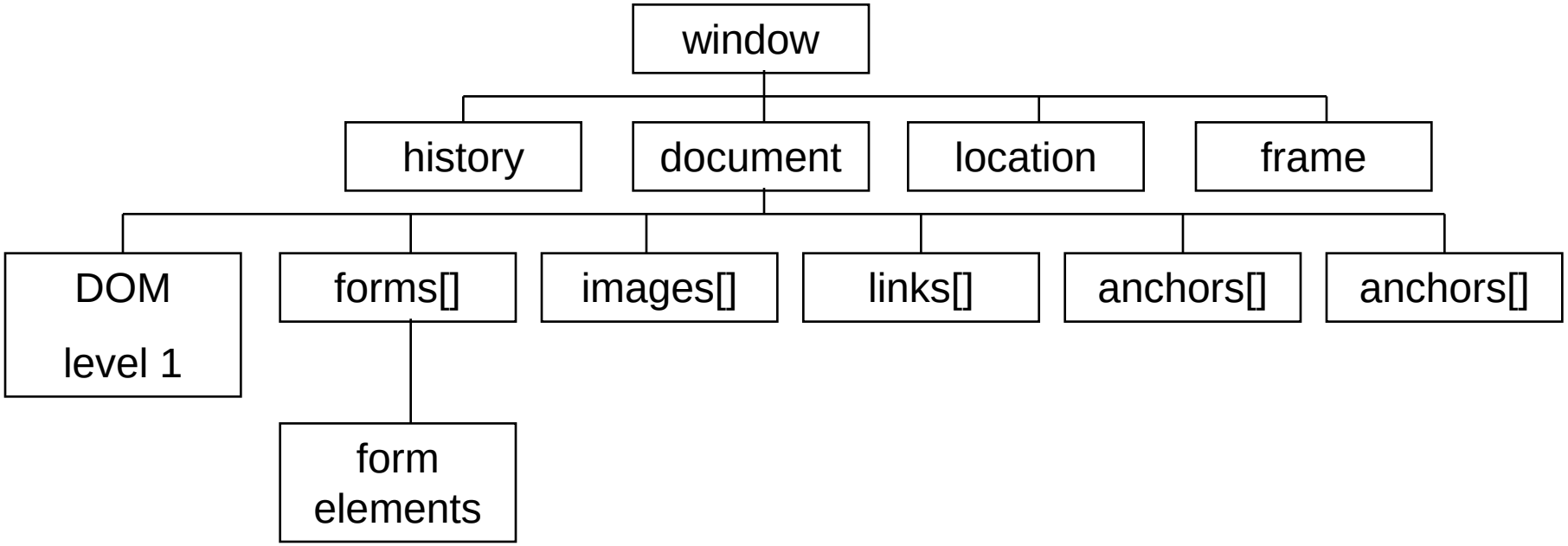

JavaScript - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address file:///E:/Nastava/Web%20programiranje/Nastava/1_html/deo2/primer10-2.html?polje
Go
Links Google Pipin TheServerSide.com JDK KAV FEST FTN
JavaScript i forme (2)
Uneli ste sledeće parametre: ?polje1=1&polje2=+Posalji+
Done
My Computer

Document Object Model (DOM)

- DOM predstavlja objektnu reprezentaciju XML dokumenta.
- JavaScript poseduje skup funkcija za rad sa DOM objektima.
- Postoji više nivoa reprezentacije:
  - DOM Level 0 i
  - DOM Level 1,
  - DOM Level 2,
  - DOM Level 3.

DOM Level 0

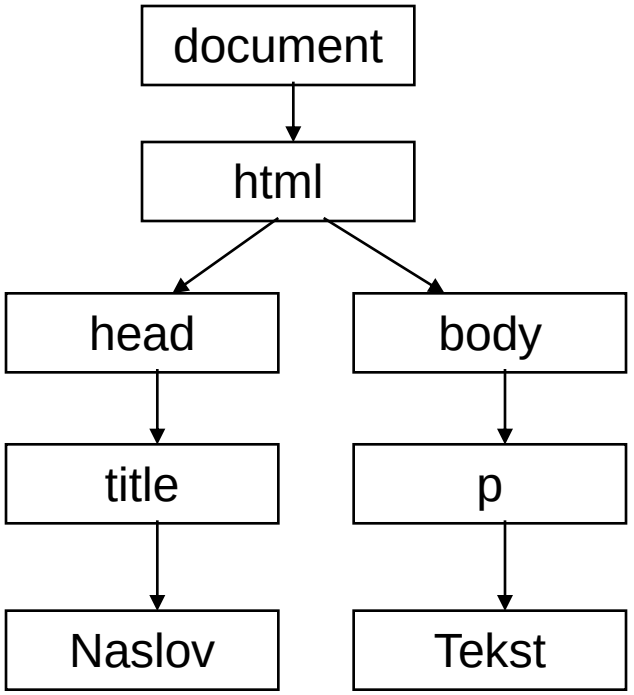
- DOM Level 0 omogućuje pristup elementima stranice preko predefinisanih objekata.



DOM Level 1-3

- DOM nivoi 1-3 predstavljaju objektnu reprezentaciju sadržaja HTML dokumenta
- Primer:

```
<html>
    <head>
        <title>Naslov</title>
    </head>
    <body>
        <p>Tekst</p>
    </body>
</html>
```



DOM reprezentacija

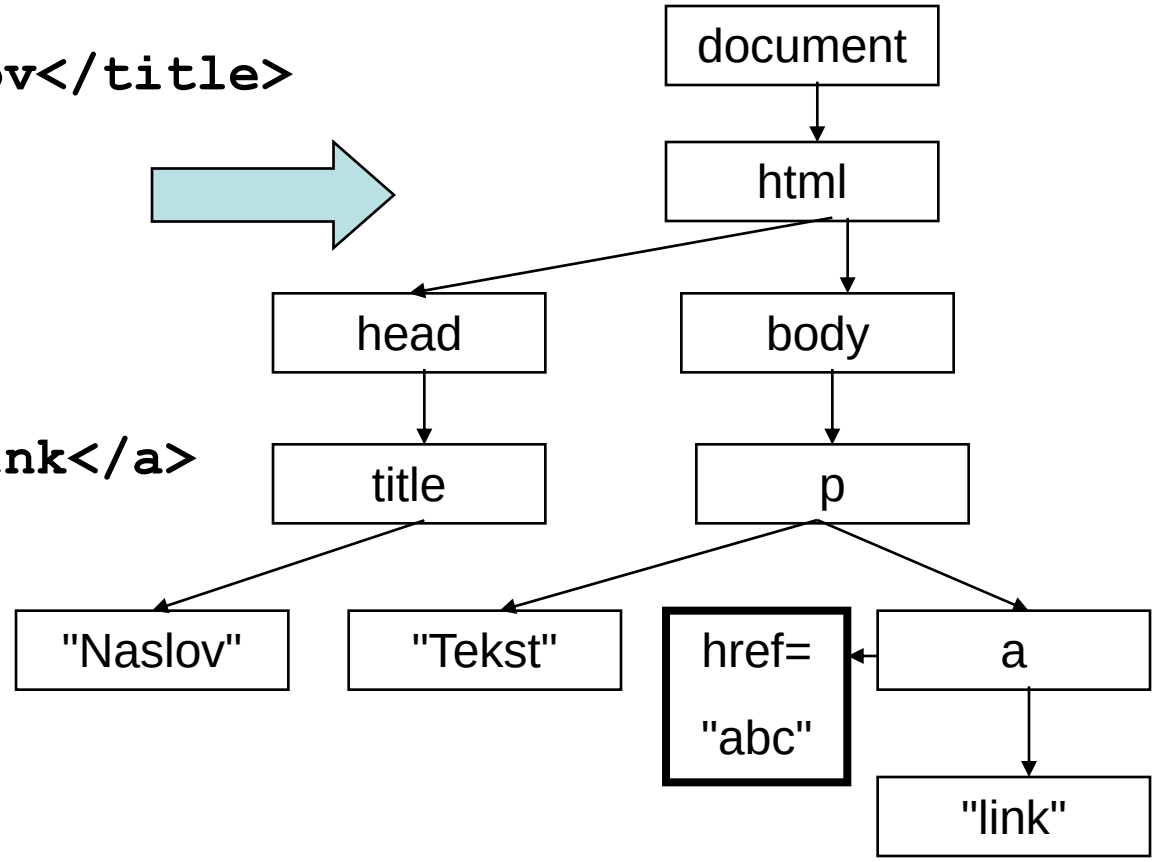
- HTML dokument se posmatra kao stablo koje se sastoji iz elemenata
- Koren stabla je <html> tag
- Svaki HTML tag je čvor tipa element u stablu
- Svaki atribut je čvor tipa atribut u stablu
- Svaki tekst je čvor tipa tekst (tekstualni čvor) u stablu
- Svaki komentar je čvor tipa komentar u stablu

DOM Stablo

- Svaki HTML dokument se posmatra kao DOM stablo
- Čvor na vrhu se zove korenski čvor
- Svaki čvor osim korenskog ima jednog roditelja (čvor iznad)
- Svaki čvor može da ima potomke (decu – čvorovi ispod)
- List je čvor bez dece
- Čvorovi istog nivoa (sibiling) su čvorovi sa istim roditeljem

DOM Stablo

```
<html>
   <head>
       <title>Naslov</title>
   </head>
   <body>
       <p>
   Tekst
   <a href="abc">link</a>
       </p>
   </body>
</html>
```



DOM i JavaScript

- DOM objektima se može pristupiti jedino iz skripta.
- JavaScript poseduje atribute i metode za pristup DOM elementima.
- Osnovni element je document objekat.
  - On sadrži sve čvorove DOM stabla koji reprezentuju HTML stranicu
- Obično se elementima HTML stranice doda atribut ***id*** da bi se lakše pronašli u DOM stablu:

Tekst

Objekat tipa čvor (node)

- Atributi:
  - *nodeName* – ime čvora
  - *nodeType* – tip čvora (1 za HTML tagove, 2 za atribute, 3 za tekstualne čvorove, 8 za komentar, 9 za dokument)
  - *nodeValue* – sadržaj tekstualnog čvora
  - *innerHTML* – sadržaj čvora kao HTML
  - *id* – ID čvora
  - *firstChild*, *lastChild* – prvi/poslednji čvor ispod u hijerarhiji
  - *childNodes* – niz čvorova koji su u prvom nivou ispod, u hijerarhiji
  - *parentNode* – objekat koji sadrži tekući čvor
- Atributi stila – svaki čvor ima atribut stila – *style*:
  - *cvor.style.top=10* – stil {top:10}
  - *cvor.style.visibility*=“visible” – stil {visibility:visible}
- Ako je naziv stila sa crticom, u JavaScriptu se spaja i koristi veliko slovo:
  - *cvor.style.borderWidth = 0* – stil {border-width:0}

Objekat tipa čvor (node)

- Metode:
  - *appendChild(čvor*) – dodaje tekućem čvoru novi čvor, na kraj prvog nivoa ispod u hijerarhiji
  - *insertBefore(čvor*, drugi) – ubacuje zadati čvor ispred drugog čvora
  - *removeChild(čvor*) – uklanja zadati čvor iz stabla
  - *getAttribute(ime*) – vraća vrednost zadatog atributa
  - *setAttribute(ime*, vrednost) – postavlja vrednost atributa
  - *removeAttribute(ime*) – uklanja zadati atribut
  - *hasAttributes*() – vraća true ako tekući čvor ima atribute

Veza između JavaScript-a i stilova

- Sloju se dodeli ID:

```
<div id="aName" style="position...”>...</div>
```

- Registruje se JavaScript funkcija za neki tip događaja:

```
<div id=“file” onmouseover="showmain(this)" >
```

- Ako je potrebno da se *event handler* definiše na nivou dokumenta, radi se ovako:

```
function init() {
document.onmousedown=engage;
document.onmousemove=dragLayer;
document.onmouseup=disengage;
}
```

Veza između JavaScript-a i stilova

- Iz JavaScript-a se elementima HTML stranice pristupa preko DOM modela:

```
target = document.getElementById(neki_id);
```

- Atributi stila proizvoljnog elementa se menjaju preko atributa style elementa:

```
target.style.display = "none";
```

- Ako atribut ima '-' u imenu, izbaci se '-' i stavi veliko slovo:

```
target.style.borderWidth = 0;
```

Primer

- Listanje svih <p> tagova u dokumentu:

```
<html>
<body>
<p>Jedan paragraf</p>
<div>
<p>Drugi paragraf</p>
<p>Treći paragraf</p>
</div>
<script type="text/javascript">
x=document.getElementsByTagName("p");
document.write("<ul>");
for (i=0;i<x.length;i++)
{
  document.write("<li>" + x[i].innerHTML + "</li>");
}
document.write("</ul>");
</script>
</body>
</html>
```

Primer

Jedan paragraf

Drugi paragraf

Treći paragraf

- Jedan paragraf
- Drugi paragraf
- Treći paragraf

Primer

- Ispis prvog potomka zadatog <p> taga:

```
<html>
<body>
<p id="intro">Jedan paragraf</p>
<div>
<p>Drugi paragraf</p>
<p>Treći paragraf</p>
</div>

<script type="text/javascript">
x=document.getElementById("intro");
document.write("Sadržaj prvog paragrafa: " +
   x.firstChild.nodeValue);
</script>

</body>
</html>
```

Primer

Jedan paragraf

Drugi paragraf

Treći paragraf

Sadržaj prvog paragrafa: Jedan paragraf