



# Java Script

dr Suzana Marković, dipl.ing. el.  
[suzana.markovic@bpa.edu.rs](mailto:suzana.markovic@bpa.edu.rs)

# Programski jezici

- **Kompajlerski jezici** - programski kôd konvertuje se u mašinski pre nego što se i pokrene, što znači da će se ta konverzija izvršiti samo jednom (Java);
- **Interpretatorski jezici** – konverzija kôda dešava se u momentu njegovog pokretanja i izvršava se svaki put kada se kôd pokrene (JavaScript);

# Istorijat Javascript-a

- ▶ Nastao je 1995. godine – kod je mogao da se izvršava u veb pretraživaču Netscape Navigator 2.
- ▶ Microsoft je na osnovu javne dokumentacije Netscape-a objavio skoro identičnu verziju svog jezika JScript (izvršavao se u Explorer-u 3)
- ▶ Udruženje ECMA (European Computer Manufacturers Association) je usvojilo svoj standard za ovaj jezik, pri čemu je JavaScript postao Netscape-ova implementacija, a JScript Microsoft-ova.
- ▶ Izveden je iz C jezika



# Skriptovanje na strani servera

- To je metoda programiranja na vebu koja pokreće softver na serveru, a ne pretraživač, ili služi za instaliranje plug-in-ova za kreiranje dinamičkih veb stranica.
- **Node.js** je serverska JavaScript platforma idealna za pravljenje „real-time“ veb aplikacija.



## Skriptovanje na strani klijenta

- Smešta se u HTML kod (ili kao posebna datoteka koja se poziva) i otvara u pretraživaču dajući veliku interaktivnost dokumentu.
- Npr. provera korisničke forme i otklanjanje eventualnih grešaka, pre slanja serveru.
- Omogućava promenu sadržaja veb stranice u zavisnosti od korisničkih unosa ili nekih drugih promenljivih.

# Gde pisati kod?

- JavaScript na *internetu* - Najlakši način za početak:
  - Sajtovi kao što su [jsbin.com](http://jsbin.com), [jsFiddle.com](http://jsFiddle.com) ili [js.do](http://js.do) služe za pisanje programa.
- JavaScript u *pregledačima* – Može se pisati u programu za pisanje teksta kao što su Notepad, Notepad++, Sublime, VSCode itd.
  - Program treba snimiti kao HTML fajl i otvoriti ga u pregledaču (npr. Chrome)



## Gde pisati kod?

- ▶ Javascript u *konzoli* - Instalirati Node.js, napisati JavaScript program, snimiti ga kao .js fajl i pokrenuti ga iz klasične konzole pomoću komande `node mojprogram.js`. Program će se izvršiti kao bilo koji program napisan u jezicima Pascal, Python, C, itd.



# Osobine Javascript-a

- ▶ Javascript je *platformski neutralan* – kod se izvršava u okviru pretraživača bez obzira na hardversko i softversko okruženje
- ▶ *Modularno programiranje* – za čuvanje i izvršavanje koda koriste se posebni dokumenti sa .js ekstenzijom
- ▶ *Integrisanost sa HTML-om* – u okviru veb stranice moguće je kombinovati html i js kod, ali je moguće i iz js generisati html kod.





# Gde se pojavljuje i kako se implementira js

- Pojavljuje se:
  - U vidljivom delu html-a --> body tag
  - U nevidljivom delu html-a --> head tag
  - kao zaseban .js fajl
  - Script tagovi su **inline elementi** – nemaju uticaja na formatiranje strane
- Implementira se:
  - Direktnim unosom koda na stranu
  - Dodavanjem reference na eksterni fajl



# Alati za unos js koda

- Notepad
- Notepad ++
- Visual Studio
- VS Code
- Eclipse
- DreamViewer
- Konzola veb pretraživača (F12)



# Izvršavanje koda

```
<script>  
alert("Hello World!")  
</script>
```

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

# JavaScript – korišćenje iz html – a i iz Js-a

```
<body>  
  <script>  
    document.writeln("Zdravo svete!");  
  </script>...
```

```
<script>  
  document.writeln("Zdravo svete!");  
</script>  
</head>
```

```
<script>  
  document.writeln("<h1>Zdravo svete!</h1>");  
</script>
```

# Eksterni java script

## Html kod

```
<html>
<head>
<title>JS kod</title>
<script src="javascriptKod.js">
</script>
</head>

<body onload="poruka()">

</body>
</html>
```

## Eksterni JS kod

```
// JavaScript Document
function poruka()
{
    alert("Ovo je neka poruka")
}
```

# Svojstva Java Script-a

- Java script je case sensitive
- Oznaka ; na kraju naredbe je opcionalna
- Sa ; se mogu napisati višestruke JavaScript naredbe u istoj liniji
- Bez oznake ; JavaScript naredbe se moraju pisati u posebnim linijama
- Blokovi se pišu unutar zagrada {...}
- Blok predstavlja celinu – ekvivalentnu jednoj naredbi

# Komandni blokovi

- Naredbe se grupišu u blokove naredbi, koji su izdvojeni velikim zagradama:

```
{  
document.writeln("Da li ovo radi? ");  
document.writeln("Radi! <BR>");  
}
```

# Komentari u JavaScript-u

- Komentar na kraju reda – linije je `//`
- Proizvoljni komentar između znakova `/*` i `*/` – komentar u više redova;
- Kao i u drugim programskim jezicima, komentari služe za:
  1. Opis programa običnim govornim tekstom
  2. Privremeno isključivanje delova programa tokom razvoja programa



# JavaScript promenljive

- ▶ JavaScript nije strogo tipizirani jezik, tako da deklaracija promenljivih nije obavezna.
- ▶ Promenljiva `x` se PRVI put u programu može navesti i kao:  

```
x = 5;
```
- ▶ Tip promenljive se deklarira implicitno – preko trenutno dodeljene vrednosti - script jezik
- ▶ Deklaracija promenljive `x`:  

```
var x = 5;  
x = "Hello World";
```
- ▶ Tip podatka koji sadrži promenljiva *može da se menja* u toku programa
- ▶ Ime promenljive je *case sensitive* i sastoji se od slova (a – z, A – Z), cifara (0 – 9) i znaka “\_” – underscore. Prvi znak imena promenljive mora biti slovo , “\_” ili \$.



# Definicija tipa podataka

- Ne zahteva se i nije dozvoljeno deklarirati tip podataka promenljive
- JavaScript interpreter automatski prepoznaje koji tip podataka je smešten u okviru promenljive i
- Dodeljuje tip podatka promenljivoj **dinamički**

# Osnovni – ugrađeni tipovi podataka

- Undefined – nije definisana vrednost
- Boolean – logički tip
- Number
  - Celi brojevi sa osnovom 10, 8 ili 16.
  - Racionalni brojevi - 3.14, 314E-2 ili 314e-2.
- String - proizvoljan niz karaktera između navodnika ("neki tekst") ili apostrofa ('neki tekst')
- Osim ovih osnovnih tipova postoje i objekti – ugrađeni i korisnički definisani.

# Aritmetički operatori

- ➔ Služe za obavljanje aritmetičkih operacija između promenljivih.
- ➔ U donjoj tabeli korišćena je vrednost  $y=5$ .

Operator	Description	Example	Result
+	Addition	$x=y+2$	$x=7$
-	Subtraction	$x=y-2$	$x=3$
*	Multiplication	$x=y*2$	$x=10$
/	Division	$x=y/2$	$x=2.5$
%	Modulus (division remainder)	$x=y\%2$	$x=1$
++	Increment	$x=++y$	$x=6$
--	Decrement	$x=--y$	$x=4$

**Operatori ++ i - - mogu biti prefiksni i postfiksni**

# Operatori dodele

- Služe za dodeljivanje vrednosti promenljivama.

Operator	Example	Same As	Result
=	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

# Operator + u stringovima

- Sabiranje stringova svodi se na njihovo povezivanje
- Prazan string "".

```
txt1="What a very";  
txt2="nice day";  
txt3=txt1+" "+txt2;
```

# Sabiranje brojeva i stringova

```
<html>
<body>

<script type="text/javascript">
x=5+5;
document.write(x);
document.write("<br />");
x="5"+"5";
document.write(x);
document.write("<br />");
x=5+"5";
document.write(x);
document.write("<br />");
x="5"+5;
document.write(x);
document.write("<br />");
</script>

<p>The common rule is: If you add a number
and a string,
the result will be a string.</p>

</body>
</html>
```

10  
55  
55  
55

The common rule is: If you add a number and a string, the result will be a string.

# Relacioni operatori

- Ovi operatori se koriste u logičkim izrazima radi utvrđivanja jednakosti ili razlike između promenljivih.
- Korišćena vrednost  $x=5$ ;

Operator	Description	Example
<code>==</code>	is equal to	<code>x==8</code> is false
<code>===</code>	is exactly equal to (value and type)	<code>x===5</code> is true <code>x==="5"</code> is false
<code>!=</code>	is not equal	<code>x!=8</code> is true
<code>&gt;</code>	is greater than	<code>x&gt;8</code> is false
<code>&lt;</code>	is less than	<code>x&lt;8</code> is true
<code>&gt;=</code>	is greater than or equal to	<code>x&gt;=8</code> is false
<code>&lt;=</code>	is less than or equal to	<code>x&lt;=8</code> is true



# Logički operatori

- Služe da odrede logiku između promenljivih
- Za  $x=6$  i  $y=3$

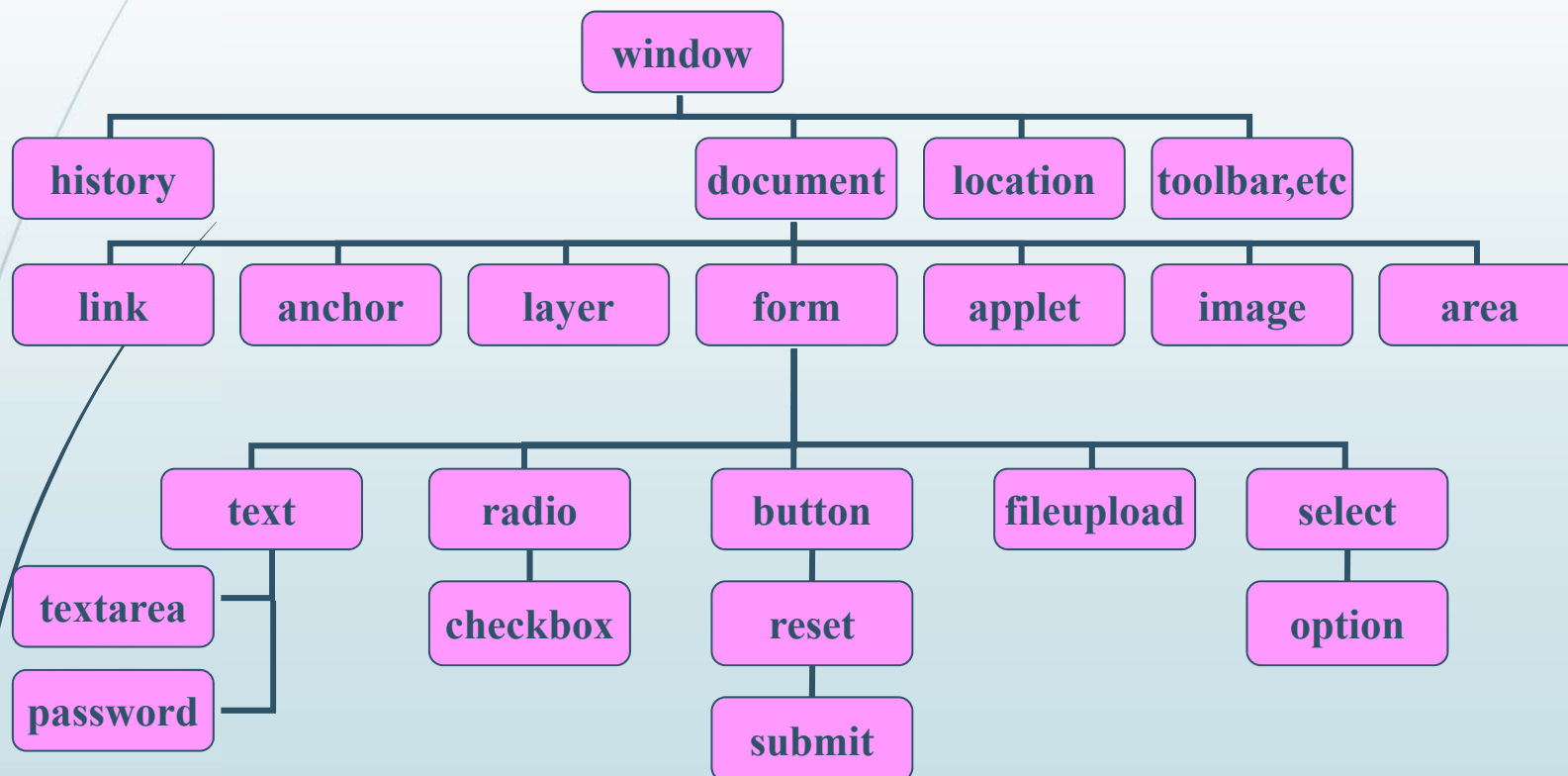
Operator	Description	Example
&&	and	$(x < 10 \ \&\& \ y > 1)$ is true
	or	$(x==5 \    \ y==5)$ is false
!	not	$!(x==y)$ is true

# Bitovski logički operatori

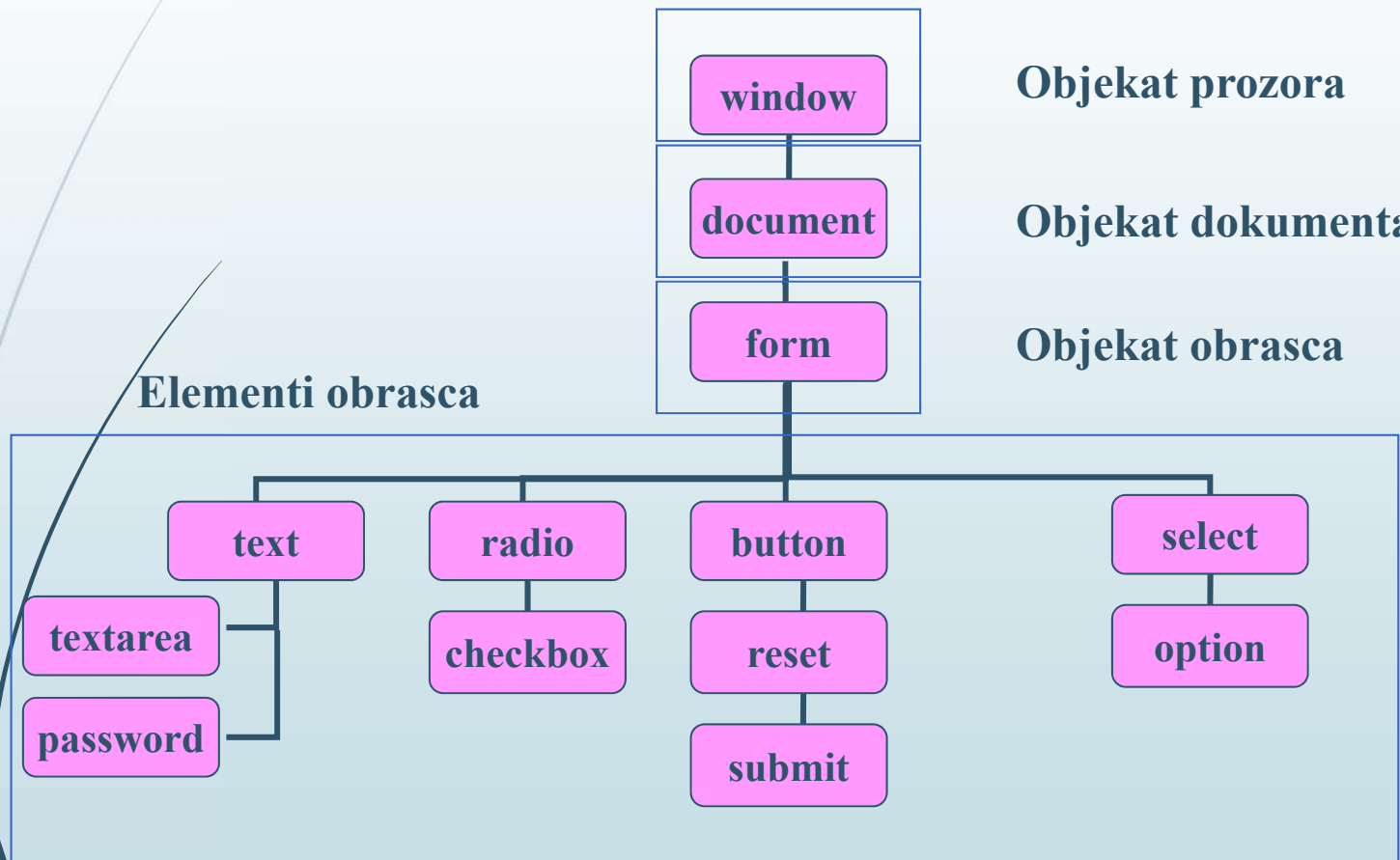
Operator	Description	Example
&	Logičko I	255 & 15 -- 00 001 111
	Logičko ILI	255   15 -- 11 111 111
^	Logičko Exkluzivno ILI	255 ^ 15 -- 11 110 000
<<	Pomeranje ulevo	010 << 2 -- 01000
>>	Pomeranje udesno	010000 >> 2 -- 0100
~	Logičko NE	~15 -- 11 110 000

**255 --11 111 111**  
**15-- 00 001 111**

# Model objekata dokumenta

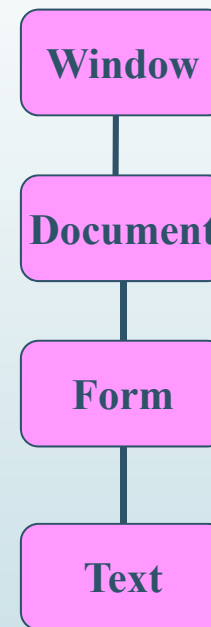


# Model objekata dokumenta



# Dodavanje elementa za unos teksta

```
<html>
<head><title>
Simple doc
</title></head>
<body>
<h1> Hello </h1>
<form>
<input type="text">
</form>
</body>
</html>
```



# Referenca objekta – polje za unos teksta

`window`

`window.document`

`window.document.formName`

`window.document.formName.textName`

Window

Document

Form

Text

# Definicija objekata

- ▶ U JavaScript-u objekat može biti: polje za unos teksta, dugme ili ceo HTML dokument
- ▶ Svaki od objekata je jedinstven
- ▶ Tri najvažnije odlike objekata definišu njegov izgled, ponašanje i kako ga skript kontroliše, to su:
  - ▶ **svojstva (properties)**
  - ▶ **metode**
  - ▶ **procedure za obradu događaja**

# Svojstva

- Svojstva HTML objekta najčešće se podešavaju atributima HTML oznaka – statički metod
- Prisustvo JavaScripta često dodaje opcione attribute čija se inicijalna vrednost može modifikovati kada se dokument učita – dinamički metod
- Veb strana se tada tretira kao dokument.
- Dokument ima svojstva (properties), npr. boja njegove pozadine (bgColor).
- Pozivanje svojstva obavlja se na sledeći način:
  - document. ime svojstva - Za pristupanje svojstvima objekata u sintaksi se koristi tačka





# Skript tagovi

```
<html>  
<body bgcolor="brown">  
  <p>paragraf 1</p>  
  <script>  
    document.bgColor="beige"  
  </script>  
</body>  
</html>
```

# Interakcija sa korisnikom

- JavaScript ima mogućnost da kreira male dijaloge za *ispis* ili *unos* teksta koji su potpuno nezavisni od sadržaja HTML dokumenta
- Najjednostavniji način da se prosledi kratka poruka korisniku je korišćenje *alert metoda*.
  - `alert("Kliknite na OK za nastavak.");` ili
  - `window.alert(...)`
- Window - najviši vrh hijerarhije prostora imena – globalni objekat sa metodom alert

# Primer

```
<html>
<body>
  <p>paragraf 1</p>
  <script>
    // 1.script blok
    alert("prvi skript blok");
  </script>
  <p>paragraf 2</p>
  <script>
    // 2.script blok
    document.backgroundColor = "brown";
    alert("drugi skript blok");
  </script>
  <p>paragraf 3</p>
</body></html>
```

// - komentar

alert() – funkcija

Poruka unutar message box-a specificira se unutar zagrada alert() funkcije i zove se funkcijski **parametar**.

# Interakcija sa korisnikom

- `alert()` metoda omogućava slanje poruka korisniku, ali ne i preuzimanje odgovora od njega.
- Metod `prompt()` omogućava jednostavnu interakciju sa korisnikom.
- `prompt()` kreira dijalog prozor u kome se može proslediti poruka korisniku, ali se formira i polje u koje korisnik može uneti proizvoljan tekst.

# Izlaz iz JavaScripta

- Ostvaruje se metodama
  - write (šalje tekst u prozor WWW čitača bez pomeranja) i
  - writeln (isto kao write(), s tim što se posle ispisa teksta kursor pomera u sledeći red objekta)
- `document.write("Test");`
- `document.writeln('Test');`

# Interakcija sa korisnikom - primer

```
<html>
<head>
  <title>Primer</title>
</head>
<body>
  <script>
    document.write("<h1>Dobrodošli ");
    document.writeln(prompt("Unesite Vaše ime:"));
    document.writeln(prompt("Unesite Vaše prezime:"));
    document.write(" u svet JavaScripta!</h1>");
    document.write('<h2>Čestitamo!</h2>');
  </script> </body></html>
```

## Uslovni (ternarni) operator

- Služi za dodeljivanje vrednosti promenljivoj u zavisnosti od nekog uslova

```
ime_promenljive=(uslov)?vrednost1:vrednost2;
```

```
greeting=(visitor=="Pres")? "Dear President":  
"Dear";
```

- Ako je promenljiva *visitor* jednaka „Pres“ (ako je uslov tačan) promenljiva će dobiti vrednost „Dear President“ a u protivnom će biti (uslov nije tačan) „Dear“.

# Primer – uslov

```
<html><body><script>
    var degFaren = prompt("Unesite stepene u
farenhajtima",212);
    var degCel;
    degCel = 5 / 9 * (degFaren - 32);
    document.write(degFaren + "\xB0 F je " + degCel +
"\xB0 C<br>");
    if (degCel < 0) {
        document.write("Ovo je ispod nule!");
    }
    if (degCel == 100)
        document.write("Ovo ključa!");
</script> </body> </html>
```



# Kontrola toka programa

```
<script>
var age = prompt("What is your age?");

if (Number(age) < 18) {
    alert("Sorry, you are too young to drive this car. Powering off");
} else if (Number(age) > 18) {
    alert("Powering On. Enjoy the ride!");
} else if (Number(age) == 18) {
    alert("Congratulations on your first year of driving. Enjoy the ride!");
}
</script>
```



# Primer

```
var job = 'teacher';

job = prompt('What does john do?');

▼ switch (job) {
  case 'teacher':
    console.log('John teaches kids.');
```

break;

```
  case 'driver':
    console.log('John drives a cab in
Lisbon.');
```

break;

```
  case 'cop':
    console.log('John helps fight
crime.');
```

break;

```
  default:
    console.log('John does something
else.');
```


}



# Uslovni ciklusi u JavaScriptu (while)

```
while ([izraz])  
{ [naredba];  
  [naredba]; ... }
```

```
i=1  
while(i<=10){  
  document.write(i+ "<br>");  
  i=i+1;  
}
```



# Uslovni ciklusi u JavaScriptu (do-while)

[inicijalizacija]

**do** {

telo\_petlje

[iteracija]

} **while** (uslov);

```
i=1  
do {  
  document.write(i+ "<br>");  
  i=i+1;  
} while(i<=10)
```



# Uslovni ciklusi u JavaScriptu (for)

**for** (inicijalizacija; uslov; iteracija)

```
{  
telo_petlje;  
}
```

```
for (i=1; i<=10; i++){  
document.write(i+ "<br>");
```

# Prekidi

```
for(i=0; i<10; i++){  
  document.write(i+ ` `);  
  if (i%2 ==0)  
    continue;  
  document.writeln(`<br>`);  
}
```

```
0 1  
2 3  
4 5  
6 7  
8 9
```

```
for(i=0; i<10; i++){  
  document.write(i+ " ");  
  if (i%2 !==0)  
    break;  
  document.writeln(" ");
```

```
0 1
```



# JavaScript ciklus enumeracije (for-in)

- Ciklus enumeracije je ciklus pomoću koga se mogu pobrojati svi članovi (svojstva i metode) nekog objekta.
- Ciklus će se izvršiti po jednom za svakog pripadajućeg člana.
- Ovaj ciklus se još naziva i "for-in" petlja. Zadaje se na sledeći način:

```
for ([promenljiva] in [objekat]) [naredba];
```

# Funkcije

- Funkcija predstavlja skup naredbi koje izvršavaju neki specifičan zadatak.

- Deklaracija:

```
function mojaFunkcija (param1,param2,... ){  
neki kod koji se izvršava...  
}
```

- Kod u funkciji se izvršava:

- Kada se desi neki događaj (npr. klik miša)
- Pozivom funkcije iz JS koda
- Automatski - samopozivanje



# Primer

```
<html><head>  
  <title> Funkcije JavaScripta </title>  
  <script>  
function pozdravPosetiocu()  
{  
  alert("Dobrodošli na sajt");  
  return;  
} </script>  
</head><body> <p> Tekst koji se ispisuje pre izvršavanja  
iskaza JavaScript-a.</p>  
  <script>  
    pozdravPosetiocu();  
  </script> <p> Tekst koji se ispisuje nakon  
izvršavanja iskaza JavaScript-a.</p>  
</body></html>
```

# Primer

```
<html><head>
  <title> Funkcija zbira </title>
  <script>
function FindTotal(a,b,c)
{
var total = a + b + c;
document.write ("Rezultat je: " + total+".");
return; }
</script></head>
<body>   <script>
  FindTotal(2,3,4);
  </script></body></html>
```



# Primer

```
<html><head><title> Funkcija zbira </title>
  <script>
function dodajBroj(a,b,c)
{
a = Number(a);
b = Number(b);
c = Number(c);
document.write ("Rezultat sabiranja brojeva je: " + (a +
b + c) + ".");
return;      }
</script></head>
<body><script>
a = prompt('Unesite broj a: ');
b = prompt('Unesite broj b: ');
c = prompt('Unesite broj c: ');
dodajBroj(a,b,c);
</script></body></html>
```

# Nizovi

- Niz se može deklarirati bilo gde u programu.
- Nigde se ne zadaje tip elemenata niza - svaki član niza je kao i bilo koja druga promenljiva - može biti bilo kog tipa.
- Prvi element niza je uvek sa indeksom 0.
- Niz se može kreirati preko **konstruktor objekta Array** (kreiranjem instance klase Array):
  - `a = new Array()` ili `a = new Array(N)` ili `a = new Array(c1, c2, ...)`
- korišćenjem **literala** ("nizovne konstante"):
  - `a = []` ili `a = [c1, c2, ...]`
- Npr. `br = [10, 8, 26, 5, 3, 12]`



## Primer

```
br = [10, 8, 26, 5, 3, 12]
```

```
console.log("Broj elemenata u nizu je: " +  
br.length)
```

```
i = 0
```

```
while ( i < br.length ) {  
  console.log("Broj["+i+"]: " + br[i])  i = i + 1}
```

# Objekti

- Skupovi promenljivih kojima se može pristupati po imenu.
- Npr. promenljiva sa podacima o nekom učeniku, kao što su njegovo ime, starost, broj telefona, mejl itd.
- Objekat može biti:
  - prosti podatak (npr. ime klijenta,...),
  - drugi objekat (npr. Klijent, Vrsta klijenta itd...) ili
  - funkcije.
- JavaScript poseduje svoje objekte (ugrađene tipove) ali ima mogućnost i deklarisanja sopstvenih tipova podataka.

# Kreiranje objekta

```
var zaposlen = {  
  ime: "Petar",  
  prezime: "Petrović",  
  zanimanje: "Programer"  
};  
console.log(this.zaposlen);
```

*"This"* nije promenjiva već specijalna reč programskog jezika (operator) i predstavlja **vezu** koja ukazuje na neku referencu (*objekat*).

# Svojstva i vrednosti objekata

- Vrednost svojstva objekta pored stringa i broja može biti **niz**:

```
var zaposlen = {  
    ime: "Petar",  
    prezime: "Petrović",  
    zanimanje: "Programer",  
    godinaRodjenja: 1987,  
    pol: ['muški', 'ženski']  
}; console.log(this.zaposlen.pol);
```



# Svojstva i vrednosti objekata

- Vrednost svojstva objekta pored stringa i broja može biti **funkcija**:

```
var zaposlen = {  
  ime: "Petar",  
  prezime: "Petrović",  
  zanimanje: "Programer",  
  godinaRodjenja: 1987,  
  izracunajGodina: function (godinaRodjenja) {  
    return (2019 - godinaRodjenja);  
  }  
};  
console.log(zaposlen.izracunajGodina(1987));
```



# Kreiranje novog svojstva objekta u funkciji

```
var zaposlen = {  
  ime: "Petar",  
  prezime: "Petrović",  
  zanimanje: "Programer",  
  godinaRodjenja: 1987,  
  pol: ['muški', 'ženski'],  
  izracunajGodina: function (godinaRodjenja) {  
    this.godina = 2019 - this.godinaRodjenja;  
  } };  
zaposlen.izracunajGodina();  
console.log(zaposlen);
```

## Predefinisani objekat - Date

- JavaScript poseduje izvestan skup predefinisanih objekata: Date, Array, Math.
- Novi objekat se kreira upotrebom operatora **new**:  
**var danas = new Date();**
- **Date** poseduje *metode* koje se mogu koristiti pri manipulaciji objektom:
  - `getHours()`, `getMinutes()`, `getMonth()`

```
var sada = new Date();  
document.write("Vreme je:" + sada.getHours() + ":" +  
sada.getMinutes() + "<br>");
```

```
var sada = new Date();  
document.write("Vreme je:" + sada.getDate() + "." +  
parseInt(sada.getMonth()+1) + "." + sada.getFullYear() + "<br>");
```



# Broj dana do kraja godine

```
<hr> <script>
    danas = new Date(); // danasnje vreme i datum
    kraj_god = new Date(); // kraj godine
    kraj_god.setMonth(11); // je u decembru
    kraj_god.setDate(31); // 31. dana u mesecu
    // Koliko ima milisekundi izmedju 31.12. i danasnjeg dana
    razlika = kraj_god.getTime() - danas.getTime();
    // A koliko je to dana?
    razlika = Math.floor(razlika / (1000 * 60 * 60 * 24));
    document.write("<h2 align=center>Broj dana do kraja
godine je " + razlika + "!!!</h2>");
</script> <hr>
```

```
<html>
<head>
  <title> Primer uslovnog izvršavanja </title>
</head>
<body>
  <script>
    var d = new Date();
    var time = d.getHours();
    var sada = new Date();
    document.write("Vreme je: " + sada.getHours() + ":" + sada.getMinutes()
+ "<br>");
    if (time < 12) {
      document.write("<b>Dobro jutro!</b>");}
    else if (time < 17) {
      document.write("<b>Dobar dan!</b>");
    }
    else document.write("<b>Dobro večer!</b>");
  </script> </body>
</html>
```

# Primer



# Šta su to događaji?

- Događaji su signali koji se generišu kada se odigra određena akcija.
- Događaji u veb čitaču su kada korisnik klikne na hipertekst link, kada se promeni podatak u ulaznom polju forme ili kada se završi učitavanje dokumenta.

# Događaji

Događaj	Opis	Obradivač događaja
<b>blur</b>	Događa se kada se ulazni fokus prebaci sa elementa forme (kada korisnik klikne van polja forme).	onBlur
<b>click</b>	Događa se kada korisnik klikne na formu ili na link.	onClick
<b>change</b>	Događa se kada korisnik promeni vrednost u <i>text</i> , <i>textarea</i> ili <i>select</i> elementima forme.	OnChange
<b>focus</b>	Događa se kada polje forme dobije ulazni fokus.	OnFocus
<b>load</b>	Događa se kada se učitava HTML dokument u WWW čitač.	OnLoad

# Događaji

Događaj	Opis	Obrađivač događaja
<b>mouseover</b>	Događa se kada korisnik pomera miša preko objekta.	OnMouseOver
<b>mouseout</b>	Događa se kada miš pređe preko objekta.	OnMouseOut
<b>select</b>	Događa se kada korisnik izabere ulazno polje forme.	OnSelect
<b>submit</b>	Događa se kad je forma prosleđena HTTP serveru (kada korisnik klikne na dugme za slanje sadržaja forme).	OnSubmit
<b>unload</b>	Događa se kada korisnik napušta HTML dokument, pre nego što počne sa učitavanjem novog HTML dokumenta.	OnUnload



# Primer

```
<html>
  <head>
    <title>Primer menjanja pozadine</title>
    <script>
      function boja()
      { var datum = new Date(); /* dodeljivanje tekućeg datuma
      promenljivoj */
        var sati = datum.getHours(); /* dodeljivanje tekućeg casa
      promenljivoj */
        if (sati > 4 && sati < 7) document.backgroundColor="#0000FF";
        else if (sati > 6 && sati < 9) document.backgroundColor="#0088FF";
        else if (sati > 8 && sati < 18) document.backgroundColor="#808080";
        else if (sati > 17 && sati < 20) document.backgroundColor="#FFA500";
        else if (sati > 19 && sati < 22) document.backgroundColor="#FF55FF";
        else if (sati > 21 || sati < 5) document.backgroundColor="#FF0000";
      }
    </script>
  </head>
  <body onload="boja()">
    Primer izbora boje podloge (BODY BGCOLOR) zavisno od
doba dana...
  </body>
</html>
```

# Poslednja izmena dokumenta

```
<html>
<head>  <title>Primer</title>
</head>
<body>
Dokument: <em>
<script>document.write(document.title);</script></em><br>
  URL: <em>
<script>document.write(document.URL);</script></em><br>
  Poslednja izmena:
  <em>
<script>document.write(document.lastModified);</script></em><br>
</body>
</html>
```

# HTML DOM Document

- Kada se HTML dokument učita u veb pretraživač, on postaje **objekat dokumenta**.
- Objekt dokumenta je koreni čvor HTML dokumenta.
- Pristup bilo kom elementu HTML stranice mora da započne pristupom document objektu.
  - Pronalaženje HTML elemenata
  - Promena HTML elemenata
  - Dodavanje i brisanje elemenata
  - Pronalaženje HTML objekata
  - Dodavanje Events Handler-a

# Pronalaženje HTML elemenata

Metoda	Opis
<code>document.getElementById(<i>id</i>)</code>	Pronalazi element po id-ju
<code>document.getElementsByTagName(<i>name</i>)</code>	Pronalazi elemente po imenu taga
<code>document.getElementsByClassName(<i>name</i>)</code>	Pronalazi elemente po nazivu klase

[https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_intro\\_lightbulb](https://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_lightbulb)



# Pronalaženje HTML elemenata - primeri

```
<p id="prf1"><b>Zdravo</b></p>
<p>Primer<i> getElementsById</i> metode</p>
<p id="prf2"></p>
<script>
var mojElement = document.getElementById("prf1");
document.getElementById("prf2").innerHTML =
"Tekst iz prvog paragrafa je " + mojElement.innerHTML+" ali piše u drugom
paragrafu";
</script>
```

```
<p><b>Zdravo</b></p>
<p>Primer<i> getElementsByTagName</i> metode.</p>
<p id="info"></p>
<script>
var x = document.getElementsByTagName("p");
document.getElementById("info").innerHTML =
'Tekst u prvom paragrafu (index 0) je: ' + x[0].innerHTML+ ", a u trećem
nema ničega "+x[2].innerHTML;
</script>
```

# Pronalaženje HTML elemenata - primeri

```
<p>Primer getElementsByClassName metode.</p>
<div id="mojDIV">
  <p class="child">1. p element u divu koji ima class="child" (index 0).</p>
  <p class="child">2. p element u divu koji ima class="child" (index 1).</p>
  <p class="child">3. p element u divu koji ima class="child" (index 2).</p>
</div>
<p>Klikni da vidiš kako će se promeniti svi paragrafi unutar diva!</p>
<button onclick="mojaFunkcija()">Klikni</button>
<script>
function mojaFunkcija() {
  var x = document.getElementById("mojDIV");
  x.getElementsByClassName("child")[1].style.backgroundColor = "lightgreen";
  x.getElementsByClassName("child")[0].style.color = "green";
  x.getElementsByClassName("child")[2].style.border = "1px solid green"; } </script>
```

```
<p>Zdravo</p>
<p class="info">Prvi paragraf.</p>
<p class="info">Primer querySelectorAll metode.</p>
<p id="demo"></p>
<script>
var x = document.querySelectorAll("p.info");
document.getElementById("demo").innerHTML =
'Prvi paragraf (index 0) u klasi "info" je: ' + x[0].innerHTML; </script>
```

# Promena HTML elemenata

Svojstva	Opis
<i>element.innerHTML = new html content</i>	Menja HTML unutar elementa
<i>element.attribute = new value</i>	Menja vrednost atributa HTML elementa
<i>element.style.property = new style</i>	Menja stil HTML elementa
Metoda	Opis
<i>element.setAttribute(attribute, value)</i>	Menja vrednost atributa HTML elementa

# Promena HTML elemenata - primeri

```
<p id="demo">Klikni na dugme!</p>
<button onclick="mojaFunkcija()">Klikni</button>
<script>
function mojaFunkcija() {
  var x = document.getElementById("demo");
  x.style.color = "orange";
}
</script>
```

```
x.style.border = "1px solid red";
```





# Dodavanje i brisanje elemenata

Metoda	Opis
<code>document.createElement(<i>element</i>)</code>	Kreira HTML element
<code>document.removeChild(<i>element</i>)</code>	Uklanja HTML element
<code>document.appendChild(<i>element</i>)</code>	Dodaje HTML element
<code>document.replaceChild(<i>new, old</i>)</code>	Vrši zamenu HTML elementa
<code>document.write(<i>text</i>)</code>	Upisuje u HTML izlazni strim

# Dodavanje i brisanje elemenata - primeri

```
<ul id="mojaLista"><li>Kafa</li><li>Čaj</li><li>Mleko</li></ul>
<p>Klikni da vidiš promene!</p>
<button onclick="mojaFunkcija()">Klikni</button>
<script>
function mojaFunkcija() {
  var textnode = document.createTextNode("Voda");
  var item = document.getElementById("mojaLista").childNodes[2];
  item.replaceChild(textnode, item.childNodes[0]);
  document.getElementById("mojaLista").type = "square"; } </script>
```

```
<ul id="myList"><li>Kafa</li><li>Čaj</li><li>Mleko</li></ul>
<p>Klikni da vidiš promene!</p>
<button onclick="myFunction()">Klikni</button>
<script>
var array = ["circle", "square", "disc"];
var count = 0;
function myFunction() {
  document.getElementById("myList").type = array[count];
  count = (count + 1) % 3; } </script>
```

# Dodavanje i brisanje elemenata - primeri

```
<p>Klikni na dugme da kreiraš novi BUTTON element.</p>  
<button onclick="mojaFunkcija()">Klikni</button>  
<script>  
function mojaFunkcija() {  
  var novo_dugme = document.createElement("BUTTON");  
  novo_dugme.innerHTML = "Pritisni";  
  document.body.appendChild(novo_dugme);  
}  
</script>
```

# Dodavanje Events Handler-a

Metoda	Opis
<code>document.getElementById(<i>id</i>).onclick = function(){<i>code</i>}</code>	Dodavanje event handler koda na događaj „onclick“

```
<p id="demo" onclick="myFunction()">Click me.</p>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "YOU
CLICKED ME!"; } </script>
```

```
<p id="demo">Klikni me!</p>
<script>
document.getElementById("demo").onclick = function()
{mojaFunkcija()};
function mojaFunkcija() {
  document.getElementById("demo").innerHTML = "Hvala!";
}
</script>
```

# Linkovi

- Svojstva i metode Document objekta:

- [https://www.w3schools.com/jsref/dom\\_obj\\_document.asp](https://www.w3schools.com/jsref/dom_obj_document.asp)

- Svojstva i metode Element objekta:

- [https://www.w3schools.com/jsref/dom\\_obj\\_all.asp](https://www.w3schools.com/jsref/dom_obj_all.asp)

- Događaji

- [https://www.w3schools.com/jsref/dom\\_obj\\_event.asp](https://www.w3schools.com/jsref/dom_obj_event.asp)

- [https://www.w3schools.com/js/js\\_events.asp](https://www.w3schools.com/js/js_events.asp)

- Svojstva Style objekta

- [https://www.w3schools.com/jsref/dom\\_obj\\_style.asp](https://www.w3schools.com/jsref/dom_obj_style.asp)