



Java Script_1

dr Suzana Marković, dipl.ing. el.
suzana.markovic@vps.ns.ac.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, jsFiddle.com ili 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

Zdravo svete!

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

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

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



JavaScript – korišćenje iz konzole

```
> console.log("Zdravo svete!")  
Zdravo svete!
```

```
<script>  
console.log("Zdravo svete!")  
</script>
```

```
Zdravo svete!
```



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 deklariše 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 "".

```
<script>  
var ime="Ana";  
var prezime="Marić";  
document.writeln(ime+" "+prezime);  
</script>
```

Ana Marić

```
<script>  
var ime="Ana";  
var prezime="Marić";  
console.log(ime+" "+prezime);  
</script>
```

Ana Marić

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.

Oduzimanje brojeva i stringova

```
<html>
<body>
  <script>
    x=5-4;
    document.write(x);
    document.write("<br>");
    x="7"-4;
    document.writeln(x);
    document.write("<br>");
    x="8"-4;
    document.writeln(x);
    document.write("<br>");
    x=6-"4";
    document.writeln(x);
  </script>
</body>
</html>
```

1
3
4
2

Relacioni operatori

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

Operator	Description	Example
==	is equal to	$x==8$ is false
===	is exactly equal to (value and type)	$x===5$ is true $x==="5"$ is false
!=	is not equal	$x!=8$ is true
>	is greater than	$x>8$ is false
<	is less than	$x<8$ is true
>=	is greater than or equal to	$x>=8$ is false
<=	is less than or equal to	$x<=8$ is true

Logički operatori

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

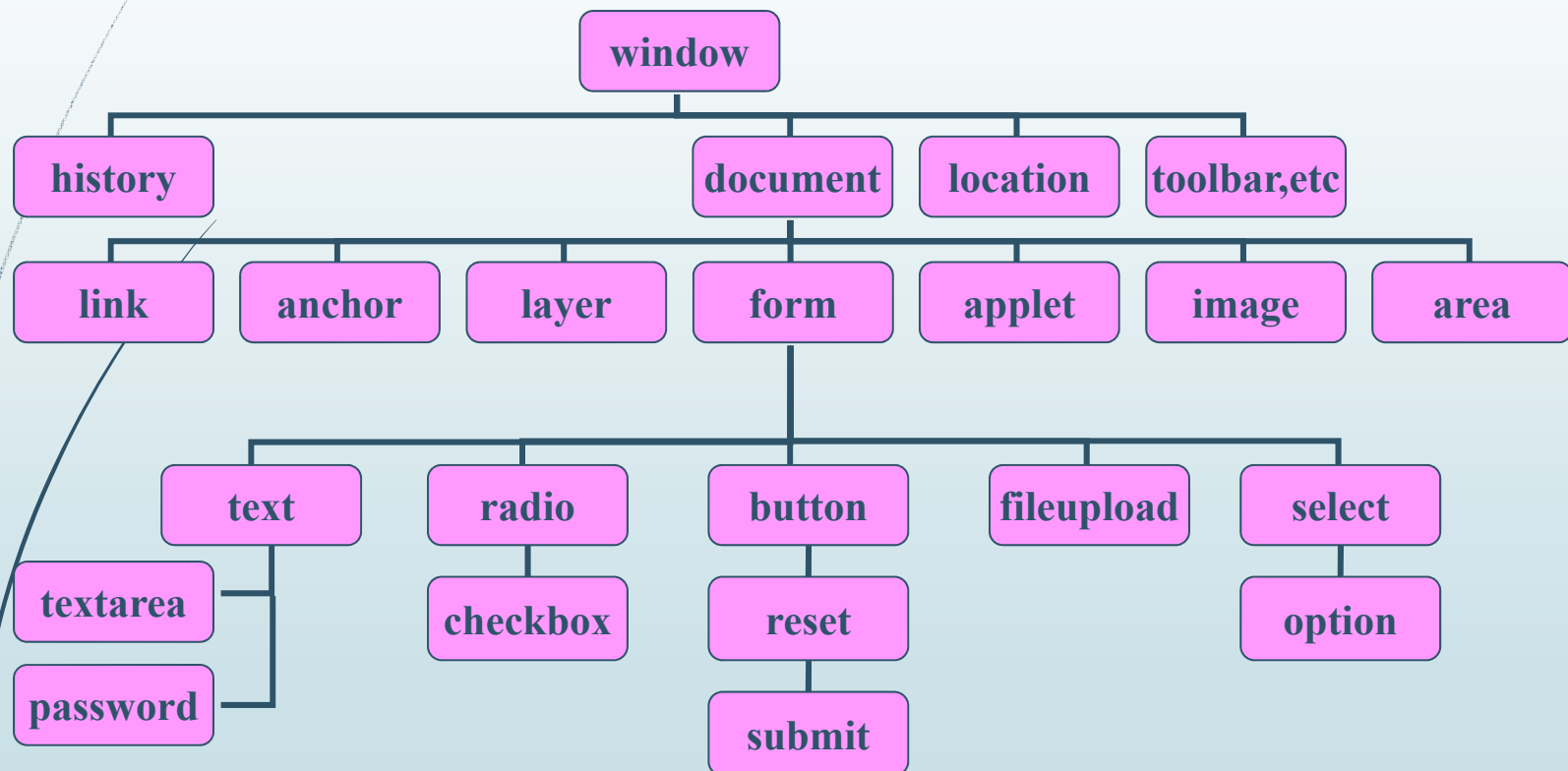
Operator	Description	Example
&&	and	$(x < 10 \ \&\& \ y > 1)$ is true
	or	$(x==5 \ \ y==5)$ is false
!	not	$!(x==y)$ is true

Bitovski logički operatori

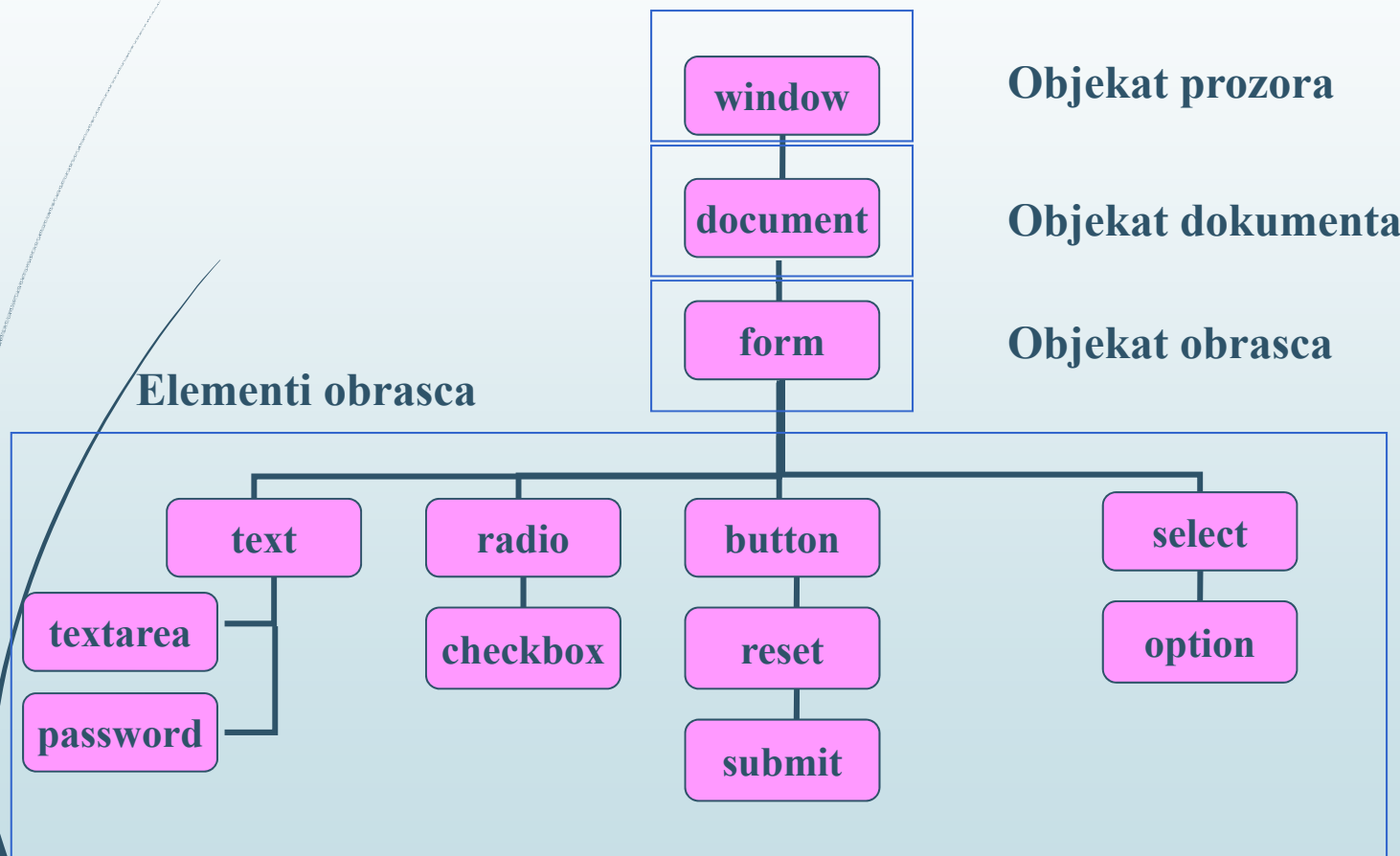
Operator	Description	Example
&	Logičko I	255 & 15 -- 00 001 111
	Logičko ILI	255 15 -- 11 111 111
^	Logičko Exkluzivno ILI	255 ^ 15 -- 11 110 000
<<	Pomeranje ulevo	010 << 2 -- 01000
>>	Pomeranje udesno	010000 >> 2 -- 0100
~	Logičko NE	~15 -- 11 110 000

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

Model objekata dokumenta

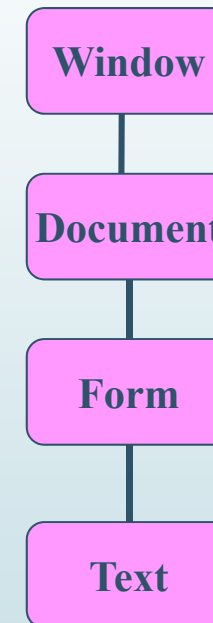


Model objekata dokumenta

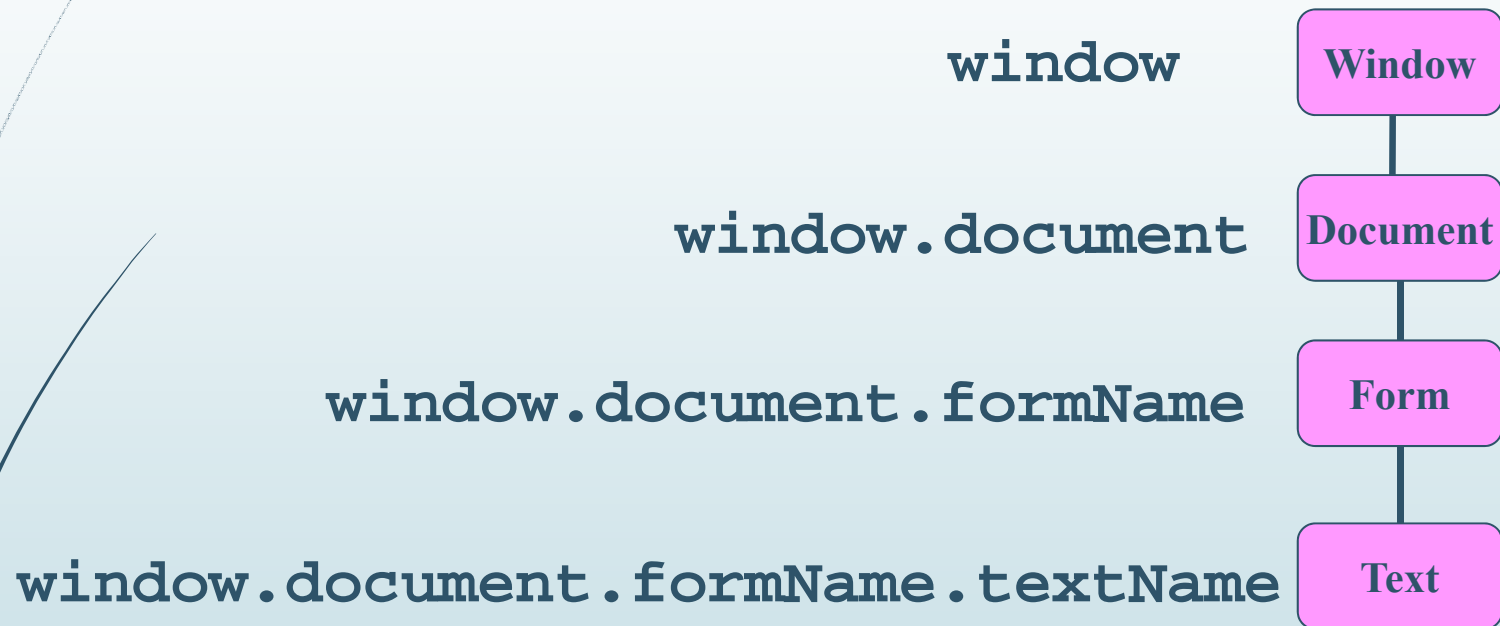


Dodavanje elementa za unos teksta

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



Referenca objekta – polje za unos teksta



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 učitava – **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.bgColor = "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');`
- U konzoli:
 - `console.log("Test");`

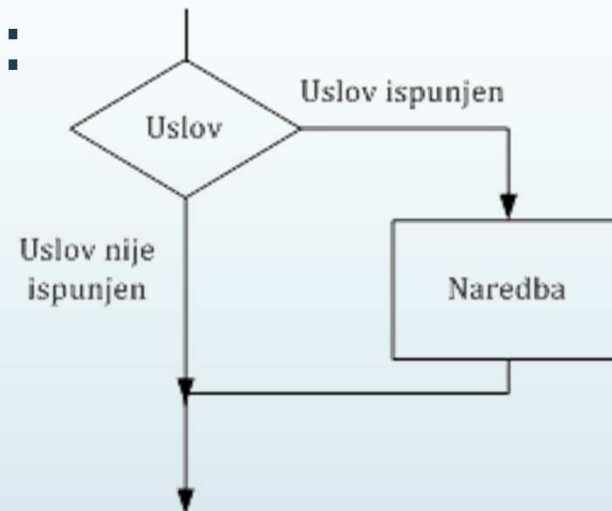
Interakcija sa korisnikom - primer

```
<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>
```


Kontrola toka programa

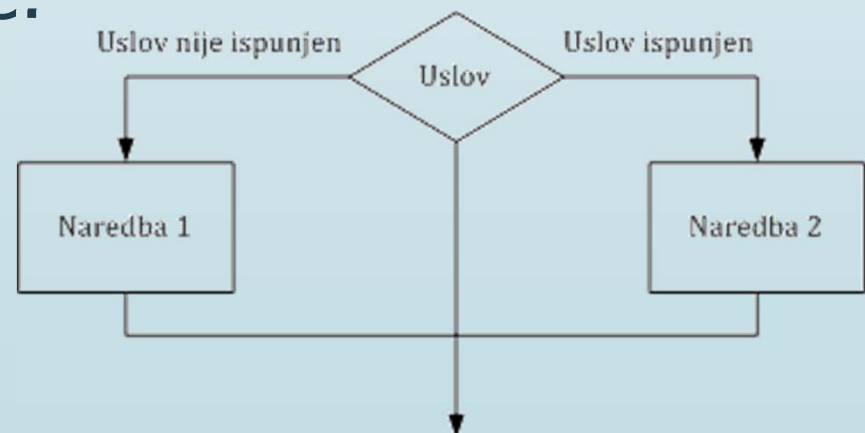
➔ Jednosmerno testiranje:

```
if (uslov)  
{  
  naredba  
}
```



➔ Dvosmerno testiranje:

```
if (uslov)  
*{  
  naredbe2  
}  
else  
{  
  naredbe1  
}
```



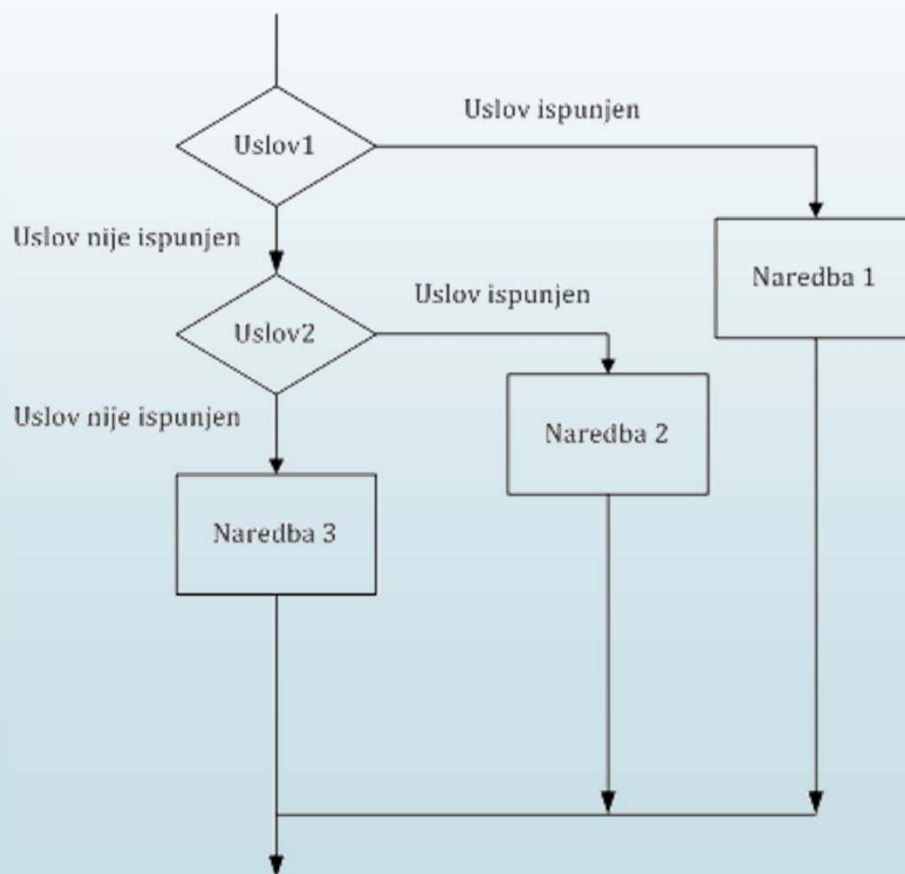


Primer

```
<script>  
  var x=prompt("Unesi x:");  
  if (Number(x) < 0)  
    document.write("Uneli ste negativan broj!");  
  else  
    document.write("Uneli ste pozitivan broj!");  
</script>
```

Višestruko grananje

```
if (uslov1)
{
    naredbe1
}
else if (uslov2)
{
    naredbe2
}
.....
else
{
    naredbe3
}
```



Primer

```
var a = prompt("Unesite neki broj od 1 do 5:");  
switch (Number(a))  
  {  
  case 1:  
    document.writeln("Uneli ste broj jedan");  
    break;  
  case 2:  
    document.writeln("Uneli ste broj dva");  
    break;  
  case 3:  
    document.writeln("Uneli ste broj tri");  
    break;  
  case 4:  
    document.writeln("Uneli ste broj cetiri");  
    break;  
  case 5:  
    document.writeln("Uneli ste broj pet");  
    break;  
  default:  
    document.writeln("Niste uneli odgovarajuci broj!");  
    break;  
  }
```

Uslovni (ternarni) operator

- Služi za dodeljivanje vrednosti promenljivoj u zavisnosti od nekog uslova

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

```
punoletstvo=(godine >=18)? „Vi ste punoletni“:  
„Vi niste punoletni“;
```



Primer – uslov

```
<script>  
var punoletstvo;  
var godine=prompt("Unesite broj godina:");  
punoletstvo=(Number(godine) >=18)? "Vi ste  
punoletni": "Vi niste punoletni";  
console.log(punoletstvo);  
</script>
```



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];
```