



Java Script_2

dr Suzana Marković, dipl.ing. el.
suzana.markovic@vps.ns.ac.rs

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 4

```
function najveca_vrednost(x, y) {  
    return x > y ? x : y;  
}  
maksimum = najveca_vrednost(-10, 15);  
console.log('Veca vrednost je ' + maksimum);
```

```
ispisi_argument = function (arg) {  
    console.log(arg);  
}; // Tačka-zarez je obavezan znak jer je ovo naredba dodeljivanja  
rezultat = ispisi_argument(7);  
console.log('Povratna vrednost funkcije koja nema return naredbu je '  
+ rezultat );  
console.log('Tip povratne vrednosti funkcije koja nema return naredbu  
je ' + typeof rezultat );
```

Nizovi

- Niz se može deklarirati bilo gdje 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 **konstruktora 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 1

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

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

```
for(i = 0; i < br.length; i++)
```

```
console.log("Broj[" + i + "]: " + br[i])
```

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);  
/console.log(zaposlen.ime+" "+zaposlen.prezime);
```

*"This" je ključna reč
koja se odnosi na
objekat kome pripada.*



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 (2021 - 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=2021-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() + "." +  
(sada.getMonth()+1) + "." + sada.getFullYear() + "<br>");
```

Predefinisani objekat - Math

- Objekat Math nema konstruktor
- Sve metode i svojstva se kreiraju bez prethodnog kreiranja Math objekta
- Sintaksa je Math.property
- Postoji 8 konstanti: Math.E, Math.PI, Math.SQRT...
- Postoje 4 metode koje vraćaju broj:

Math.round(x)	Returns x rounded to its nearest integer
Math.ceil(x)	Returns x rounded up to its nearest integer
Math.floor(x)	Returns x rounded down to its nearest integer
Math.trunc(x)	Returns the integer part of x (<u>new in ES6</u>)

https://www.w3schools.com/js/js_math.asp



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

Događaji

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

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>
```