

Napredne tehnike JavaScripta

Napredni rad sa funckijama

- U osnovnom kursu smo naučili šta su funckije.
- Rekli smo da je to deo koda koji obavlja niz operacija kroz jednu logičku celinu.
- Mogu da budu sa parametrima ili bez njih i obično vraćaju neki rezultat koristeći ključnu reč **return**.
- Naziv funkcije predstavlja referencu na nju ako je dodeljenja nekoj promenljivoj.
- Takve promenljive se zovu **funkcionalne promenljive**.

Primer: ***funkcionalne-promenljive.html***

Napredne tehnike JavaScripta

Napredni rad sa funkcijama

- Funkcije možemo da koristimo kao argumente drugih funkcija i tada je promenljiva koja se definiše u nekoj funkciji koja prima takav argument tipa funkcija .
- Takođe funkcionalne promenljive mogu da budu argumenti drugih funkcija.
- Praktično na ovaj način jedna funkcija se prosleđuje drugoj i može da se izvrši unutar nje.
- **Obratiti pažnju** da funkcija koja se prosleđuje kao argument mora da vrati vrednost.

Primer: *[funkcionalne-promenljive-argument.html](#)*

Napredne tehnike JavaScripta

Objekti

- Objekat je referentni tip podataka.
- Objekti služe za čuvanje kolekcija različitih tipova.
- Prazan objekat možemo da napravimo koristeći jednu od dve sintakse:

`var korisnik = new Object(); //Object ključna reč`

`var korisnik = {}; // objekat literal sintaksa`

- Obično se druga sintaksa koristi, treću sintaksu spominjemo kasnije .
- Objekat pravimo sa uglastim zagradama `{...}` sa opcionom listom svojstva ili promenljivih
- Svojstva ili promenljive se navodi u obliku

`ključ:vrednost`

gde je `ključ` string tipa ili se još zove ime promenljive, a `vrednost` može biti bilo kojeg tipa.

Napredne tehnike JavaScripta

Objekti

-Primer objekta :

```
var korisnik = { // objekat korisnik  
  ime: "Pera", //u ključu "ime" čuvamo vrednost "Pera"  
  godine: 30 //u ključu "godine" čuvamo vrednost 30  
};
```

-Vrednosti promenljive objekta pristupamo koristeći dot notaciju (.)

Primer :

```
alert(korisnik.ime); // ispisaće nam Pera  
alert(korisnik.godine); // ispisaće nam 30
```

-Možemo dinamički dodati promenljivu na sledeći način

```
korisnik.daLiJeAdmin = true;
```

-Možemo i da obrišemo promenljivu sa

```
delete korisnik.godine;
```

Napredne tehnike JavaScripta

Objekti

- Ukoliko objekat nema promenljivu – properti i ukoliko želimo da pristupimo takvoj promenljivoj dobićemo *undefined* kao povratnu vrednost.
- Postoji specijalan operator *in* koji proverava da li promenljiva postoji u objektu
- PRIMER – za vežbu uraditi primer

```
var korisnik = {  
    ime: "Pera",  
    godine: 30  
};  
alert("prezime" in korisnik); //šta nam vraća ?  
alert("godine" in korisnik) //šta nam vraća ?
```

Napredne tehnike JavaScripta

Objekti

- Ispis svih promenljivih možemo da uradimo koristeći for...in petlju na sledeći način:
- PRIMER – za vežbu uraditi primer

```
var korisnik = { // objekat korisnik
  ime: "Pera",
  godine: 30
};
for(var x in korisnik){// promenjiva x čuva ključ.

  alert(x); // prikazujemo nazive ključeva-ime ,godine

  alert( korisnik[x] ); // prikazujemo vrednosti ključeva-Pera, 30
}
```

Napredne tehnike JavaScripta

Objekti

-Još jedna bitna razlika između objekata i primitivnih vrednosti je da se objekti čuvaju i kopiraju po referenci, dok se primitivne vrednosti dodeljuju ili kopiraju kao cele vrednosti.

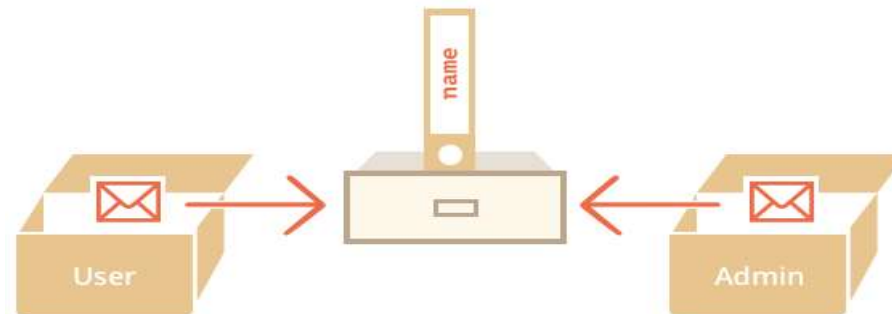
PRIMER: primitivne vrednosti

```
var message = "Hello!";  
var phrase = message;
```



PRIMER: objekti

```
var user = { name: "John" };  
var admin = user;  
// kopira referencu
```



Napredne tehnike JavaScripta

Objekti

PRIMER: uraditi primer za vežbu

```
var korisnik = { ime: 'Pera' };  
var admin = korisnik;  
admin.ime= 'Ana';//promenjeno je ime preko "admin" ref.  
alert(korisnik.ime); šta će biti rezultat ?
```

-Možemo da kažemo da su dva objekta ista samo ako imaju referencu na isti objekat

PRIMER :

```
var a = { x: 1, y: 2 };  
var b = { x: 1, y: 2 };  
var c = a;  
var ab = (a === b); //false (jesu istog tipa ali ne referenciraju isti objekat)  
var ac = (a === c); //true
```


Napredne tehnike JavaScripta

Objekti

-Objektima možemo da dodamo neku akciju tj. funkciju.

-To možemo da uradimo koristeći funkcije kao izraze, tko što funkciju dodelimo promenljivoj tipa objekat na sledeći način :

PRIMER :

```
var korisnik = {  
  ime: "Pera",  
  godine: 30  
};  
korisnik.ispisi = function(){- funkcija kao izraz  
  alert("Pozvao sam metod iz objekta");  
}  
korisnik.ispisi();<- poziv funckije / metod
```

- Svaka funkcija koja je dodeljena promenljivoj naziva se metoda

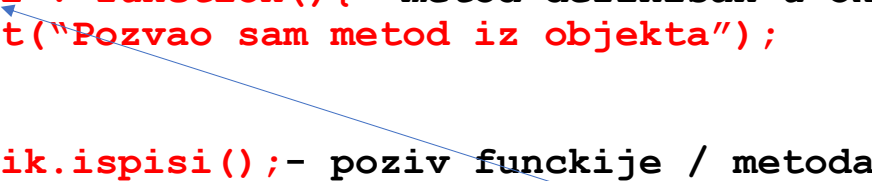
- Ovu metodu smo mogli definisati direktno u okviru objekta.

Napredne tehnike JavaScripta

Objekti

PRIMER 2:

```
var korisnik = {  
  ime: "Pera",  
  godine: 30,  
  ispisi : function() {- metod definisan u okviru objekta  
    alert("Pozvao sam metod iz objekta");  
  }  
};  
korisnik.ispisi(); - poziv funckije / metoda
```



Funkcija u objektu moze da bude samo anonimna, moze da prima parametre, njen kljuc je u stvari naziv funkcije

Napredne tehnike JavaScripta

Objekti

- Često je potrebno da metod koji je definisan u objektu treba da pristupi promenljivoj u okviru objekta kako bi mogla da nešto odradi.
- Npr. Ako bi u prethodnom primeru hteli da naša metoda ispiše ime korisnika potrebno je da pristupi promenljivoj ime.
- Da bi metoda pristupila objektu, metoda može da koristi ključnu reč *this*

PRIMER 3:

```
var korisnik = {  
  ime: "Pera",  
  godine: 30,  
  ispisi : function(){  
    alert("Ime korisnika je "+this.ime);  
  }  
};  
korisnik.ispisi();  
this vazi samo u okviru objekta
```

Napredne tehnike JavaScripta

Objekti

ZADATAK:

Napraviti objekat po imenu student i definisati mu sledeće promenljive
Ime, prezime ,broj indeksa, prosek ocena , godina studija.

Promenljivama dodeliti sledeće vrednosti :

Ime-Petar

Prezime-Petrović

Broj indeksa – 1123

Prosek ocena – 8.12

Godina studija - 4

Napisati metodu koja ispisuje trenutnu godinu studija i prosek ocena.

Napisati metodu koja mu dodaje novu ocenu i racuna novi prosek ocena.

Napisati metodu koja ispisuje sve podatke o studentu (ovu metodu pozvati poslednju)

Pozvati sve ove metode i rezultat prikazati u konzoli

Napredne tehnike JavaScripta

Objekti

-Podsetimo se prve sintakse :

```
var k = new Object();
```

-Sa gore navedenim izrazom **new Object** možemo da napravimo samo jedan objekat.

-Ako želimo da imamo tip objekat koji može da pravi više objekata istog tipa pravimo **konstruktor funkcije** koji pravi objekat, a nakon toga objektu možemo da definišemo promenjive i funkcije.

- Konstruktor je funkcija samo što se njen naziv podudara sa nazivom objekta.

- Primer konstruktora objekta :

```
function Korisnik(ime, prezime, godine) {  
    this.imeKorisnika = ime;  
    this.prezimeKorisnika = prezime;  
    this.godineKorisnika = godine;  
}
```

```
var k = new Korisnik('Pera', 'Perić', '22') - pravimo objekat
```

Konstruktor se poziva sa ključnom rečju new , tada se kreira objekat kojem se može pristupiti korišćenjem this reference.

Napredne tehnike JavaScripta

Objekti

Razlika između literal objekta i objekt konstruktora :

- ako hoćemo jednu instancu koristimo literal objekat
- ako hoćemo pravi objekat sa ponašanjem i više instanci koristimo konstruktor objekta