

Veb programiranje

Tema: JS objekti

Šta je to objekat

- ❑ **Objekat:** drvo, kuća, auto, pas
- ❑ Svaki objekat ima **osobine** (za auto: težina, boja) i **metode** (kreće se, koči, staje)
- ❑ Svi automobili imaju te iste osobine, samo su vrednosti tih osobina različite
- ❑ Svi automobili imaju iste metode, ali ih izvršavaju u svoje vreme

Objekti - osnovni koncept

- ❑ **Prosti tipovi podataka** u JavaScript-u su brojevi, stringovi, boolean, null i undefined.
- ❑ Vrednosti prostog tipa su nepromenljive
- ❑ Sve ostalo su **objekti**.
- ❑ Promenljive prostog tipa imaju jednu vrednost
- ❑ I objekti su promenljive, samo što one sadrže više vrednosti

```
var car = {type:"Toyota", model:"Yaris", color:"red"};
```

JavaScript **objekat**
je kolekcija imenovanih
vrednosti

Objekti - osnovni koncept

- Objekti imaju **osobine** (properties)
- Svaka osobina ima **naziv i vrednost**.
- Pišu se kao **name:value** parovi
- Objekti u JavaScript-u su **besklasni**, to jest nemaju posebnu definiciju klase iz koje se izvode.

Objekti - kreiranje

- Objekti se kreiraju pomoću vitičastih zagrada {}.

```
prazan_objekat = {};
```

```
var ucenik = {  
    ime: "Jovan",  
    prezime: "Jovanovic",  
    razred: 3  
};
```

Objekti - kreiranje

```
var ucenik = new Object();  
ucenik.ime = "Jovan";  
ucenik.prezime = "Jovanovic";  
ucenik.razred = 3;
```

- Oba ova načina naprave isti objekat, pa se obično koristi prva metoda
- Prva metoda: jednostavnija, preglednija i brža

Objekti - osobine

```
var ucenik = {  
    ime: "Jovan",  
    prezime: "Jovanovic",  
    razred: 3  
};
```

➤ Pristup osobini objekta:

`imeObjekta.imeOsobine`

`ucenik.ime`

ili

`imeObjekta["imeOsobine"]`

`ucenik["ime"]`

Objekti - prolazak kroz objekat

- `for...in` petlja prolazi kroz sve proprijetije objekta

```
for (nekaPromenljiva in object) {  
    // kod  
}
```

Objekti - dodavanje u objekat

- Novi properti postojećem objektu dodajemo prosto tako što mu dodelimo vrednost

```
ucenik.skola = "Vuk Karadzic"
```

Objekti - brisanje iz objekta

- **delete** briše properti iz objekta:

```
var ucenik = {  
    ime: "Jovan",  
    prezime: "Jovanovic",  
    razred: 3  
};  
  
delete ucenik.razred;
```

- **delete** briše i vrednost i osobinu

Objekti

- Objekte pozivamo po referenci, ne po vrednosti

```
var ucenik = {ime: "Jovan", prezime: "Jovanovic",  
             razred: 3};  
var drugiUcenik = ucenik
```

- drugiUcenik **nike kopija** ucenik-a, drugiUcenik **jeste** ucenik.
- Sve promene objekta drugiUcenik **će** da promene i objekat ucenik, jer se radi o istom objektu

Objekti - metode

- Metode su akcije koje se izvršavaju nad objektima
- Metode su funkcije koje se definišu kao osobine objekta

```
var ucenik = {  
    ime: "Jovan",  
    prezime: "Jovanovic",  
    razred: 3,  
    punoIme: function() {  
        return this.ime + "" + this.prezime;  
    }  
};
```

- U definiciji funkcije reč **this** se odnosi na objekat u kom se funkcija nalazi.

Objekti - pristup metode

- Metodama pristupamo sledećom sintaksom:

```
imeObjekta.imeMetode()
```

- Primer:

```
var ucenik = {  
  ime: "Jovan",  
  prezime: "Jovanovic",  
  razred: 3,  
  punoIme: function() {  
    return this.ime + " " + this.prezime;  
  }  
};  
imeUcenika = ucenik.punoIme();
```



Hvala na pažnji!