

# Osnove Javascript-a

Javascript je:

- Slabo tipizirani programski jezik
- interpretirani programski jezik
- Standardni jezik koga podržavaju svi web browser-i

Javascript omogućava "oživljavanje" stranica i predstavlja veći deo funkcionalnosti samih stranica. Zajedno sa CSS-om i HTML-om predstavlja suštinu svih stranica na web-u. Da nema javascript-a ne bi bilo ni popularnih frameworka, koje ćemo raditi kasnije na kursu.

## Definisanje

Prilikom definisanja javascript koda moguće je to uraditi:

1. Unutar zasebnog javascript fajla (sa ekstenzijom .js) koji se uključuje u html stranice preko *script* taga unutar *src* atributa kao za čiju vrednost ćemo staviti putanju do samog javascript fajla.

Primer:

```
<script src="path/to/file.js"/>
```

2. Unutar html-a između otvarajućeg i zatvarajućeg *script* taga.

Primer:

```
...  
<script>  
    alert('hello!')  
</script>  
...
```

## Promenljive

Definisanje javascript promenljivih se može raditi na četiri načina:

1. Upotrebom *var* rezervisane reči: promenljiva će biti vidljiva u okviru cele funkcije
2. Upotrebom *let* rezervisane reči: Promenljiva će biti vidljiva u okviru bloka u kojem je definisana
3. Upotrebom *const* rezervisane reči: Promenljiva se vidi u okviru bloka, i neće menjati vrednost.
4. Bez upotrebe reči: Ovim dobijamo globalne promenljive koje su pristupačne svuda. (Neki alati za rad sa frontendom ovo ne dopuštaju)

## Poruke

Postoje tri vrste dijaloga koji se koriste prilikom interakcija sa veb stranicom:

1. `alert(tekst)` - predstavlja obaveštenje za korisnika da se nešto desilo
2. `confirm(tekst)` - tera korisnika da potvrdi akciju koju je započeo
3. `prompt(tekst)` - daje mogućnost korisniku da unese neku vrednost kroz popup dijalog

## Funkcije

Javascript stavlja veliki akcenat na funkcije i dolazi sa velikim brojem funkcija koje možete koristiti u svakodnevnom radu. Funkcije delimo na:

1. Korisnički definisane
  - Definišu se početnom rečju *function*
  - Kako je javascript funkcionalni jezik, samo funkcije možete “sačuvati” (njihovu referencu) u promenljivu i tako je poslati nekoj drugoj funkciji kao parametar
2. Ugrađene ili predefinisane
  - isNaN
  - eval
  - parseInt
  - ... (Opširnija lista se može naći [ovde](#))

## Objekti

Objekti predstavljaju proširenje koncepta promenljivih. Kao što promenljive mogu da imaju prostu vrednost, na primer neki broj, slovo, niz slova ili *string*, tako mogu za vrednost da imaju i neki objekat. Vrednost objekata definišemo u vitičastim zagradama “{}” i koristimo **key:value** sintaksu. Tako na primer možemo imati objekat koji predstavlja osobu:

```
let person = {  
  name: "Petar",  
  surname: "Petrović",  
  age: 20,  
  eyeColor: "green"  
}
```

Objekte možemo podeliti na:

1. Ugrađene
  - Nizovi (var niz = new Array())
  - String (var tekst = "tekst")
2. Korisnički definisani
  - Object
  - JSON
  - Pozivom konstruktora

## Grananje

Javascript podržava dva načina kondicionala. To su:

1. If, else if, else  
If (uslov) {  
 ...  
} else if (drugi uslov) {  
 ...  
} else {

```

    ...
}
2. Switch case
switch(uslov) {
    case x:
        //block x
        break;
    case y:
        //block y
        break;
    default:
        //block
}

```

Dodatne napomene vezane za if/switch:

- Provera jednakosti:
  - “==” nije isto kao i “===”
    - 2 == “2” (true)
    - 2 === “2” (false)
- Postoje “*truthy*” i “*falsy*” vrednosti:
  - Falsy: sve što se evaluira na **false, 0, “”, null, undefined, NaN**
  - Truthy: sve što nije falsy

## Petlje

Javascript podržava nekoliko tipova petlji:

### 1. **FOR** petlje

- a. Petlje koje prolaze kroz svaki *property* objekta

```

for(let key in object){
    ...
}

```

- b. Petlje koje prolaze kroz svaki element niza

```

for(let key of iterable) {
    ...
}

```

- c. Petlje koje se ponavljaju neki broj puta

```

for(let i = 0; i < niz.length; i++) {
    ...
}

```

### 2. **WHILE** petlje

- a. Prolaze kroz blok koda dok je uslov zadovoljen i mogu se izvršiti ni jednom ili više puta

```

while(uslov){
    ...
}

```

- b. Prolaze kroz blok koda dok je uslov zadovoljen i mogu se izvršiti jednom ili više puta
- ```
do {  
    ...  
} while(uslov)
```

## Događaji

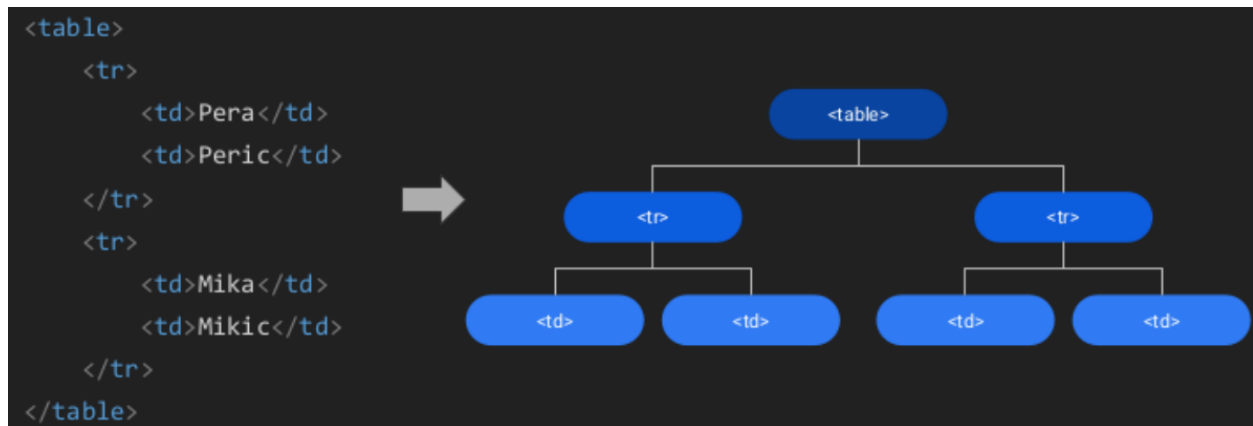
Javascript dolazi sa velikim brojem događaja na koje možete reagovati. Praktično, kad god korisnik nešto uradi ovi događaji se “okidaju” (odnosno browser pozove neku funkciju) i možete da izvršite neki kod kad se desi neki događaj. Primeri događaja su:

1. onload
2. onclick
3. onsubmit
4. onfocus (opširniju listu možete naći [ovde](#))

## DOM (Document Object Model)

Opisuje HTML element u obliku stabla. Pored toga sadrži liste predefinisanih objekata koji se mogu javiti na nekoj stranici kao što su forme, slike, itd. Kroz javascript možemo da pristupamo elementima, da dodajemo nove ili da stare uklanjamo. Postoje tri vrsta čvora:

1. Html tag
2. Html atribut
3. Tekst



Moguće je pristupiti elementima na nekoliko načina:

1. Preko id-a elements - `document.getElementById(id)`
2. Vrednosti atributa class - `document.getElementsByClassName(classname)`
3. Imena elementa - `document.getElementsByName(name)`
4. Po tagu - `document.getElementsByTagName(tagName)`
5. Css selektor - `document.querySelector(selector)`,  
`document.querySelectorAll(selector)`

Primeri interakcije sa DOM objektima:

1. `let imeEl = document.getElementsByName('ime')[0]`

2. `let ime = imeEl.value`
3. `imeEl.style.background = 'red'`
4. ...