

# JavaScript Tutorijal

JavaScript je programski jezik za HTML i Web.

Date,html

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>Moj prvi JavaScript</h1>
```

```
<button type="button"
```

```
onclick="document.getElementById('taj').innerHTML = Date()"
```

```
klikni za datum i vreme.</button>
```

```
<p id="taj"></p>
```

```
</body>
```

```
</html>
```

## Uvod u JavaScript

---

JavaScript je najpopularniji programski jezik na svetu.

Evo nekoliko primera sta sve JavaScript moze da uradi.

---

## JavaScript moze da promeni HTML (Content) sadrzaj

Jedan od mnogih HTML method je **getElementById()**.

U ovom primeru je koriscen metod "nadj" HTML element (sa id="tu"), i promeni njegov sadrzaj unutra sa (**innerHTML**) to "Hello JavaScript":

### Example

```
document.getElementById("tu").innerHTML = "Ala sam se promenio ";
```

menja.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>Sta JavaScript moze da uradi?</h1>
```

```
<p id="tu">JavaScript moze da promeni HTML sadrzaj.</p>
```

```
<button type="button"
```

```
onclick="document.getElementById('tu').innerHTML = 'Ala sam se promenio!'">
```

```
stisni!</button>
```

```
</body>
```

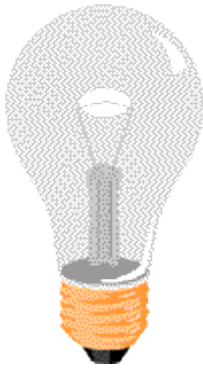
```
</html>
```

---

## JavaScript moze da promeni HTML atribut

U ovom primeru se menja HTML slika, tako sto se menja atribut `<img>` tag:

## Sijalica



Klikni da upalis/ugasis svetlo.

Bulb.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>JavaScript moze da promeni sliku</h1>
```

```

```

```
<p>Klikni da upalis/ugasis svetlo.</p>
```

```
<script>
```

```
function changeImage() {
```

```
    var image = document.getElementById('mojaslika');
```

```
    if (image.src.match("upaljenasijalica")) {
```

```
        image.src = "ugasenasijalica.gif";
```

```
    } else {
```

```
        image.src = "upaljenasijalica.gif";
```

```
}  
  
}  
  
</script>  
  
</body>  
  
</html>
```

Sujalica.html

```
<!DOCTYPE html>  
  
<html>  
  
<body>  
  
<h1>JavaScript moze da izmeni sliku</h1>  
  
  
  
<p>Klikni da upalis/ugasis svetlo.</p>  
  
<script>  
  
function promenaslike() {  
  
    var image = document.getElementById('slika');  
  
    if (image.src.match("upaljenasijalica")) {  
  
        image.src = "ugasenasijalic.gif";  
  
    } else {
```

```
        image.src = "upaljenasijalica.gif";  
    }  
}  
</script>
```

```
</body>
```

```
</html>
```

## JavaScript moze da promeni HTML Styles (CSS)

Menjanje stila nekog HTML element, je varijanta menjanja HTML atributa:

### Primer

```
document.getElementById("demo").style.fontSize = "25px";
```

```
menjastil.html
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>Sta sve moze JavaScript da uradi?</h1>
```

```
<p id="probni">JavaScript moze da promeni stil nekog HTML element.</p>
```

```
<script>
```

```
function myFunction() {
```

```
    var x = document.getElementById("probni");
```

```
    x.style.fontSize = "20px";
```

```
    x.style.color = "green";
```

```
}
```

```
</script>
```

```
<button type="button" onclick="myFunction()">Stilizuj!</button>
```

```
</body>
```

```
</html>
```

## JavaScript može da proverava ispravnost unosa podataka

JavaScript se često koristi za proveru ispravnosti unosa :

Molimo Vas da unesete broj između 1 i 10

```
validacija.html
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>JavaScript može da proverava ispravnost unetih podataka</h1>
```

```
<p>Molimo Vas da unesete broj između 1 i 10:</p>
```

```
<input id="broj">
```

```
<button type="button" onclick="myFunction()">taj broj</button>
```

```
<p id="to"></p>
```

```
<script>

function myFunction() {

    var x, text;

    // nadjí vrednost unosa sa id="broj"

    x = document.getElementById("broj").value;

    // Ako x nije broj (Not a Number) ili nije izmedju 1 i 10

    if (isNaN(x) || x < 1 || x > 10) {

        text = "Nisi ti to dobro nesto uradio";

    } else {

        text = "Input OK";

    }

    document.getElementById("to").innerHTML = text;

}

</script>
```

```
</body>
```

```
</html>
```

## Da li znate?

JavaScript i Java su potpuno dva razlicita jezika, po konceptu i dizajnu.

JavaScript je izmislio Brendan Eich 1995, i taj jezik je postao ECMA standard 1997.

ECMA-262 je zvanično ime. ECMAScript 6 (je realizovan u junu 2015) i to je najnovija verzija JavaScripta.

# JavaScript gde sve se stavlja

---

JavaScript može biti smesten u `<body>` i `<head>` sekciji neke HTML strane.

---

## `<script>` Tag

U HTML, JavaScript mora biti smesten između `<script>` i `</script>` tagova.

### primer

```
<script>
document.getElementById("demo").innerHTML = "neki moj JavaScript";
</script>
```

---

## JavaScript funkcije I događaji

JavaScript **function** je blok JavaScript coda, koji može biti izvršen kada bude "pozvan" .

Na primer, neka funkcija može biti izvršena kada se **event** desi, kao što je pritisak na dugme-button.

## JavaScript u `<head>` ili `<body>`

Možete da staviti I više I koliko hocete JavaScripta u jedan HTML document.

Script može biti smesten u `<body>`, ili u `<head>` section of nekog HTML page, ili u oba.

Ali vi držite ceo kod na jednom mestu.

---

## JavaScript in `<head>`

U narednom primeru, JavaScript function smestena je u `<head>` section HTML page.

Funkcija se poziva(called) kada se klikne na dugme:



## Probaj.html

```
<!DOCTYPE html>

<html>

<head>

<script>

function myFunction() {

    document.getElementById("taj").innerHTML = "A, jesam li ti rekla.";

}

</script>

</head>

<body>

<h1>Neki naslov</h1>

<p id="taj">Ako stisnes dugme ja cu se promeniti</p>

<button type="button" onclick="myFunction()">probaj</button>

</body>

</html>
```

## JavaScript u <body>

U ovom primeru JavaScript funkcija je stavljena u <body> sekciji HTML stranice.

Funkcija se poziva na izvršenje kada se klikne na dugme:

## Primer `fjabody.html`

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>Naslov ove web strane</h1>
```

```
<p id="promenime">Neki paragraf koji cemo da promenemo na klik</p>
```

```
<button type="button" onclick="fjapromeni()">Probaj da kliknes</button>
```

```
<script>
```

```
function fjapromeni() {
```

```
    document.getElementById("promenime").innerHTML = "Sad sam se promenio.";
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

Dobro je da se Javascript postavlja na dnu body-ja jer to dozvoljava da se cela stranica ucita. Mix codova mozedo uspori prikaz stranice.

---

## Spoljasnji JavaScript `moji sajtovi`

Skript moze da se napise na posebnoj stranici kao spoljasnji fajl.

Spoljasnji skript je praktican jer moze da se upotrebi na vise razlicitih stranica.

Tada JavaScript fajl sacuvate sa ekstenzijom `.js`.

Ako to radite, stavite ime skript fajla u `src` (source) atribut `<script>` tag:

#### Primer

```
<!DOCTYPE html>
<html>
<body>
<script src="nekispoljni.js"></script>
</body>
</html>
```

Linkovanje za taj Javascript fajl mozete da stavite u `<head>` ili `<body>` kako vam je volja.

Skript ce da se ponasa kao da je lociran tacno tamo gde je tag `<script>` lociran.

U spoljasmem JavaScriptu ne moze da stoji `<script>` tagovi.

---

#### External JavaScript prednosti:

- Tako se odvaja HTML i kod
- Tako se laces pise, menja I dopunjuju i HTML i JavaScript
- Kes JavaScript fajlovi mogu da ubrzaju loadovanje stranica

## JavaScript Output izlaz I prikaz

---

Nema nikakvu print ili displej funkciju za prikazivanje, on to radi drugacije.

---

#### JavaScript Display mogucnosti

JavaScript moze da "prikazuje" podatke na razlicite nacine:

- Pise u alert box, koristeci tag **window.alert()**.
  - Pise u HTML izlaz koristeci **document.write()**.
  - Pise u HTML elementu, koristeci **innerHTML**.
  - Pise u browser console, koristeci tag **console.log()**.
-

## Koriscenje window.alert()

Moze da se koristi alert box za prikaz podataka:

alertprozor.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>Sabiranje</h1>
```

```
<p>Sabiranje 5+6</p>
```

```
<script>
```

```
window.alert(5 + 6);
```

```
</script>
```

```
</body>
```

```
</html>
```

## Koriscenje document.write()

**document.write():**

sabiranje.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>Sabiranje</h1>
```

```
<p>Koliko je 5+6?</p>
```

```
<script>
```

```
document.write(5 + 6);
```

```
</script>
```

```
</body>
```

```
</html>
```

Ako se koristi `document.write()` nakon nekog HTML dokumenta koji je učitao skroz, bice **izbrisani svi postojeći HTML-ovi**:

```
nemamevise.html
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>Mnozenje</h1>
```

```
<p>mnozenje prirodnih brojeva 5 i 6.</p>
```

```
<button type="button" onclick="document.write(5 * 6)">rezultat</button>
```

```
</body>
```

```
</html>
```

Naredbu `document.write()` method trebalo bi da koristite samo za testiranje

## Koriscenje innerHTML

Za pristup nekom HTML elementu, JavaScript moze da koristi **`document.getElementById(id)`** method.

**id** atribut definise HTML element, a **innerHTML** property definise HTML sadrzaj:

deljenje.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>Aritmeticke operacije</h1>
```

```
<p>Mnozenje</p>
```

```
<p id="demo"></p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = 'Mnozenje prirodnih brojeva';
```

```
</script>
```

```
</body>
```

```
</html>
```

Da bi ste prikazali podatke "display data" u HTML, (u najvećem broju slučajeva) vi treba da dodelite vrednost nekom innerHTML elementu.

## JavaScript Syntax

---

JavaScript **syntaksa** je skup pravila.

---

## JavaScript Programi

**Komputerski program** je spisak "instrukcija" koje kompjuter treba da se izvrši.

U programskim jezicima ove programske instrukcije se zovu **naredbe**.

JavaScript naredbe se medjusobno odvajaju **tacka-zarezom**.

### Primer

```
var x = 5;  
var y = 6;  
var z = x + y;
```

U HTML-u, JavaScript program se izvrsavaju u brauzeru.

---

## JavaScript Naredbe

JavaScript naredbe se sastoje od:

Vrednosti, Operatora, Izraza, Rezervisanih reci i Komentara.

---

## JavaScript Vrednosti

Pomocu JavaScript sintaksi definisu se dva tipa vrednosti: Fiksne vrednosti I promenljive vrednosti.

Fiksne vrednosti se zovu konstante (**literals**). Promenljive vrednosti se zovu promenljive (**variables**).

---

## JavaScript konstante (Literals)

Najvaznija pravila za pisanje fiksnih vrednosti su:

**Brojevi** se pisu sa I bez decimala:

10.50

1001

**Stringovi** su tekstovi, pisu se izmedju duplih Ili jednostrukih navodnika:

"Zorica Jelic"

'Bane, Bojan I Zarko'

## Tačka-zarez ;

Tačka-zarez u JavaScript-u odvaja naredbe.

Dodajte tačku-zarez na kraju svake izvršne naredbe.

```
a = 5;  
b = 6;  
c = a + b;
```

# JavaScript Funkcije

---

JavaScript funkcije je blok kodova napravljen da uradi nesto is a block of code designed to perform a particular task.

JavaScript funkcija ce biti izvršena kada je "nesto" pozove.

---

## fjaJs.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>Ovo je primer za funkciju mnozenje:</p>
```

```
<p id="ovde"></p>
```

```
<script>
```



```
function mnozenje(a, b) {  
    return a * b;  
}  
  
document.getElementById("ovde").innerHTML = mnozenje(4, 3);  
  
</script>  
  
</body>  
</html>
```

# JavaScript Objekti

---

## Objekti iz realnog zivota, osobine, metode

U realnom zivotu, kola su realni **objekt**.

Kola imaju neke **osobine** kao npr. Tezina, boja, i **metode** na primer start i stop:

Object	Properties	Methods
	kola.marka = Opel	kola.start()
	kola.model = corsa	kola.drive()
	kola.tezina = 1200kg	kola.brake()
	kola.boja = zelena	kola.stop()

Sva kola imaju ove osobine (**properties**), ali su vrednosti razlicite od kola do kola.

Sva kola imaju mogucnost( **methods**), ali se one izvode razlicit broj puta.

---

## JavaScript Objekti

Ovim kodom dodeljujete prostu vrednost (Opel) promenljivoj koja se zove kola:

```
var kola = "Opel";
```

[Try it Yourself »](#)

Objekat je promenljiva takodje. Ali objekat moze da sadrzi u sebi vise promenljivih.

Sledecim kodom zadajemo **vise vrednosti** (Opel, Corsa, zelena) **promenljivoj** koja se zove **kola**:

```
var kola = {tip:"Opel", model:"Corsa", boja:"zelena"};
```

[Try it Yourself »](#)

Vrednosti se zadaju na sledeci nacin **ime:vrednost** kao par (ime I vrednost su odvojeni dvema tackama).

JavaScript objekti su kontejneri za imenovane vrednosti .

---

## Objektne osobine (Properties)

ime:vrednost parovi (u JavaScript objectima) se zovu **properties**.

```
var persona = {ime:"Zorica", prezime:"Jelic", visina:172,tezina:65, bojaOciju:"crna"};
```

# JavaScript oblast važenja

---

Gde sve vase I gde mozete da pristupite promenljivim .

---

## JavaScript oblast vazenja

U JavaScriptu, objekti I funkcije su takodje promenljive.

**U JavaScript, oblast vazenja se odnosi na promenljive, objekte I funkcije I gde sve one vase I mozete d aim pristupite.**

Oblast vazenja se menja unutar funkcije

---

## Local JavaScript promenljive

Promenljive deklarisanе unutar JavaScript funkcije, imaju **LOCAL** vaznost I vidljivost: njima moze da se pristupi samo u okviru funkcije.

### Primer

// code ne moze da koristi `carName`

```
function myFunction() {  
  var carName = "Volvo";  
  
  // code moze da koristi carName  
}
```

lokalno.html

<html>

<body>

<p>Lokalnoj promenljivoj moze da se pristupi samo unutar funkcije gde je deklarisanа.</p>

<p id="pokazni"></p>

<script>

myFunction();

document.getElementById("pokazni").innerHTML =

"Tip mog auta je " + typeof carName;

```
function myFunction() {  
    var carName = "Opel";  
}  
  
</script>  
  
</body>  
  
</html>
```

Buduci das u lokalne promenljive prepoznatljive samo unutar funkcije gde su deklarisanе, promenljive sa istim imenom mogu da budu upotrebljene u razlicitim funkcijama. Lokalne promenljive se startuju kad se funkcija izvrsava I nakon toga se brisu.

---

## Globalne JavaScript promenljive

Promenljive deklarisanе van funkcija su **GLOBAL**.

Globalna promenljiva ima **global oblast vazenja I vidljivosti**: All scripts and functions on sve skripte I funkcije na web page mogu im pristupiti.

### Example

```
var carName = "Volvo";  
  
// code here can use carName  
  
function myFunction() {  
    // code here can use carName  
}  
  
globalno.html  
  
<html>  
  
<body>
```

<p>Globalnim promenljivim moze da se pristupa iz bilo kog skripta ili funkcije.</p>

<p id="auto"></p>

<script>

var markaKola = "Opel";

prikaziKola();

function prikaziKola() {

document.getElementById("auto").innerHTML =

"Moja kola su marke " + markaKola;

}

</script>

</body>

</html>

## Automatically Global

---

Ako dodelite vrednost nekoj promenljivoj koja nije deklarisana u okviru funkcije ona automatski postaje **GLOBALNA** promenljiva.

Ovaj kod proglašava imeKola za GLOBALNU, iako se izvrsava unutar funkcije

ipakglobalna.html

<!DOCTYPE html>

<html>

<body>

<p>

Ako dodelite vrednost nekoj promenljivoj koja nije deklarisana u okviru funkcije ona automatski postaje GLOBALNA :

</p>

<p id="auto"></p>

<script>

prikaziKola();

document.getElementById("auto").innerHTML =

"Bojanova kola su " + markaKola;

function prikaziKola() {

markaKola = "Fiat";

}

</script>

</body>

</html>

## Životni vek JavaScript promenljivih

Životni vek jedne JavaScript promenljive počinje od momenta kad je deklarisana.

Lokalne promenljive se brišu kada je funkcija kompletno izvršena.

Globalne promenljive se brišu kada je strana zatvorena.

## Argumenti funkcije

Argumenti funkcije (parameteri) rade kao lokalne promenljive unutar funkcije.

## Globalne promenljive u HTML-u

With JavaScript, the global scope is the complete JavaScript environment.

U HTML-u sve globalne promenljive su window objekti.

### Example

```
// code here can use window.carName
```

```
function myFunction() {  
    carName = "Volvo";  
}
```

[Try it Yourself »](#)

# JavaScript događaji

---

HTML događaji su **"stvari"** koje se dešavaju HTML elementima.

Kada se JavaScript koristi na HTML stranama, JavaScript može da **"reaguje"** na događaje.

---

## HTML događaji

HTML događaj može da bude nešto što browser radi, ili nešto što korisnik radi.

Nekoliko primera HTML događaja:

- HTML web strana je završila sa loadingom
- HTML input polje se promenilo was changed
- HTML dugme je pritisnuto