

WRD – skripta/podsjetnik sa predavanja

FIT Mostar

## Napomene

Sadržaj ove skripte predstavlja materijal koji je prezentiran na predavanjima iz predmeta WRD. To je skraćena verzija opšte literature i sadrži samo najvažnije elemente koje student treba naučiti iz ovog predmeta. Svrha kreiranja ove skripte je da studenti imaju sve potrebne reference na jednom mjestu u jednom dokumentu.

Ova skripta se treba koristiti zajedno sa code-ovima koji su objavljeni na stranici poslije svakog predavanja.

Osim ove skripte, studentima se preporučuje da koriste literaturu preporučenu u syllabusu za ovaj predmet:

- John Duckett, HTML and CSS: Design and Build Websites, 2011.
- Steve Suehring, JavaScript Step by Step, Third Edition, 2013.

kao i on-line reference za HTML, CSS i JavaScript:

- <https://www.w3.org/TR/html5/> - HTML 5

- <https://www.w3.org/TR/html51/> - HTML 5.1

- [https://www.w3.org/standards/techs/css#w3c\\_all](https://www.w3.org/standards/techs/css#w3c_all) – CSS – trenutni razvoj

- <https://www.w3.org/TR/2011/REC-CSS2-20110607/css2.pdf> - CSS - stabilno

## Sadržaj:

HTML .....	6
Šta je HTML? .....	6
Historija HTML .....	6
Preglednici/Browser .....	6
Verzije HTML-a .....	6
Struktura HTML dokumenta .....	7
Globalna struktura .....	7
Editori .....	8
Doctype deklaracije .....	8
Tagovi .....	8
Osnovni elementi .....	8
Semantika u HTML-u .....	9
Break i horizontal rule elementi .....	9
Link .....	9
Slika .....	10
Definisanje atributa lang .....	10
Meta podaci i meta tagovi .....	10
Još tagova .....	10
Formatiranje teksta .....	11
Tekst .....	12
Komentari .....	13
Specifični formati .....	13
HTML liste .....	13
Tabele .....	15
Block i inline elementi .....	16
Div i span elementi .....	16
Inframe element .....	17
HTML forme .....	17
Elementi HTML forme .....	17
Funkcionalnost .....	18
CSS .....	20

Inline .....	21
Interni CSS .....	21
Eksterni CSS .....	21
CSS – Best Practice .....	22
Prioriteti stilova – Kaskadni redoslijed (Cascading order) .....	22
Selektori.....	22
Element selektor .....	23
ID selektor .....	23
Class selektor .....	23
Grupisanje selektora.....	24
CSS komentari .....	24
Još neka pravila pisanja CSS-a .....	25
Boje.....	25
Borders, Margine, Padding, Veličina .....	25
Block i Inline elementi .....	25
Div i Span elementi.....	26
Pozicije, forsiranje block-inline.....	27
Parent>Child element selektor.....	27
CSS property – multiple values.....	27
CSS pseudo-elementi i pseudo-klase .....	27
Pseudo-klasa.....	27
Detaljno o CSS-u – pregled kompletne CSS specifikacije.....	31
Scripting jezici.....	31
Client-side scripting .....	32
Server-side scripting .....	32
JavaScript.....	32
Razlozi korištenja .....	32
Mjesto uključivanja JavaScripta .....	32
Načini uključivanja JavaScript .....	32
Varijable .....	34
Deklaracija varijabli .....	34
Komentari .....	36
Scope varijable .....	37
Operatori .....	37

Funkcije u JavaScript .....	38
Funkcije bez povratne vrijednosti .....	38
Funkcije sa povratnom vrijednošću .....	38
Deklaracija/Definicija Funkcije bez povratne vrijednosti .....	38
Deklaracija/Definicija Funkcije sa povratnom vrijednošću .....	38
Prilog 1 - HTML Cheatsheet .....	39

# HTML

## Šta je HTML?

HTML (HyperText Markup Language) – je jezik koji se koristi za kreiranje, opis i prikaz web stranica. On je osnovna gradivna struktura web stranica. HTML nije programski jezik. HTML je deklarativni (opisni) jezik i on je zapravo skup markup tagova (oznaka). Za razliku od programskih jezika, HTML ne može izvršiti code. On je pasivni jezik i služi samo za opis sadržaja.

## Historija HTML

HTML se prvi put spominje 1991. godine u dokumentu pod nazivom „HTML tags“ i taj dokument je sadržavao jednostavan HTML dizajn. HTML je kreiran od strane Tim Berners Lee-ja, koji je u to vrijeme bio zaposlen u CERN-u. Prvu uspješnu komunikaciju računara preko interneta Berners Lee je uspostavio 1990.

### World Wide Web

World Wide Web (WWW ili W3) je najkorištenija usluga Interneta. World Wide Web omogućava pristup hipertekstualnim dokumentima na internetu.

Hipertekstualni dokumenti su dokumenti koji sadržavaju slike, tekst ili multimedijalne sadržaje a povezani su hiperlinkovima.

Za preuzimanje i pristup sadržajima hiperlinkova koriste se preglednici.

Izumitelj World Wide Web-a je Tim Berners Lee i on je ujedno i čelnik World Wide Web Consortium-a. World Wide Web Consortium je organizacija koja se bavi standardizacijom tehnologija korištenih na Internetu.

## Preglednici/Browser

Programi koji omogućavaju korisnicima pregled HTML sadržaja. Preglednici se nalaze na svim uređajima koji imaju pristup internetu.

Neki od najpoznatijih preglednika su: Google Chrome, Internet Explorer, Mozilla Firefox i dr.

## Verzije HTML-a

**Prva verzija (1993.)** – poprilično ograničena verzija, koja čak nije omogućavala dodavanje slika u HTML dokumente.

**HTML 2.0** – prva imenovana verzija.

**HTML 3.0 (1995.)** – prva verzija koja je omogućavala definiciju tabela.

**HTML4 (1997.)** – nastavljeno je prihvaćanje oznaka nametnutih od strane proizvođača web preglednika.

**HTML5 (2014.)** – najnovija verzija HTML-a nastala u suradnji World Wide Web Consortium (W3C) i Web Hypertext Application Technology Working Group (WHATWG).

HTML – detaljan pregled može se naći na slijedećoj web stranici:

<https://www.w3schools.com/tags/default.asp>

## Struktura HTML dokumenta

Deklaracija `<!DOCTYPE html>` označava da je tip dokumenta na kojem radimo HTML 5.

Tri su osnovna elementa strukture HTML dokumenta i to su:

- `html`
- `head`
- `body`

## Globalna struktura

Globalna struktura HTML dokumenta

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title> Naslov </title>
```

```
</head>
```

```
<body>
```

```
    <h1> Najviši nivo </h1>
```

```
    <p> Paragraf </p>
```

```
</body>
```

```
</html>
```

Svaki tag koji otvorimo moramo i zatvoriti tako što poslije znaka `"<"` dodajemo znak `"/"`.

## Editori

Postoji veliki broj editora za HTML i dijele se na tekstualne i WYSIWYG (What You See Is What You Get). Mnogi old school web developeri koriste tekstualne editore. Većina početnika koristi vizualne editore. Neki developeri koriste editore koji omogućavaju oboje.

## Doctype deklaracije

Doctype deklaracija je jako važan dio HTML dokumenta i bez nje preglednik ne bi mogao ispravno obraditi stranicu. Web preglednici prikazuju web stranicu potpuno ispravno samo ukoliko tačno znaju koja **HTML verzija je korištena**

Kao što je već spomenuto, doctype deklaracija za HTML 5 je `<!DOCTYPE html>`. Međutim kod prehodnih verzija HTML-a doctype deklaracije su nešto komplikovanije.

Kod HTML 4.01 Strict deklaracija je:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
```

```
"http://www.w3.org/TR/html4/strict.dtd">
```

U XHTML 1.0 Transitional deklaracija je:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

## Tagovi

HTML tagovi su osnovni gradivni elementi svake stranice i pomoću njih su pisane HTML stranice i HTML dokumenti. Svaki tag opisuje poseban element sadržaja dokumenta odnosno opisuje kako će se nešto prikazivati u web browser-u.

Osnovni HTML tagovi su:

```
<html> </html>
```

```
<head> </head>
```

```
<body> </body>
```

## Osnovni elementi

Osnovni element koji obuhvata cijeli html dokument je **<html>** tag. Prvi element u **<html>** tag-u je **<head>** tag, koji obuhvata osnovne informacije o dokumentu u njemu se nalazi **<title>** tag u kojem se nalazi naslov koji će se vidjeti u tabu pokrenutog browser-a. Ispod **<head>** tag-a nalazi se **<body>** tag u kojem se nalazi sav sadržaj koji je vidljiv u browseru.

U **<body>** tagu najosnovniji elemnti su **<h1>** tag i **<p>** tag. **<h1>** tag je oznaka za naslov. Broj 1 označava naslov najvišeg nivoa



Postoji ukupno 6 nivo „h“ tag-a. U kodu razlika je samo u broju koji se nalazi iza ‘<h’, <h1> <h2> <h3>...

<p> tag je oznaka za paragraf i prikazuje obični tekst ispisan na stranici.

Primjer:

```
<!DOCTYPE html>
<html>
<head>
  <title> Naslov </title>
</head>
<body>
  <h1> Najviši nivo </h1>
  <h2> 2. nivo </h2>
  <h3> 3. nivo </h3>
  <h4> 4. nivo </h4>
  <h5> 5. nivo </h5>
  <h6> 6. nivo </h6>
  <p> tekst </p>
</body>

</html>
```

Tagovi obično dolaze u paru. Prvi tag se naziva početni ili startni (otvarajući) tag. Drugi tag je završni ili end (zatvarajući) tag. U završnom tagu se nalazi kosa crta prije imena taga.

Izuzev DOCTYPE taga, kod pisanja ostalih tagova nije bitno da li ih pišemo velikim ili malim slovima jer HTML 5 nije case sensitive.

## Semantika u HTML-u

Prilikom pisanja HTML koda moramo voditi računa o semantici. Semantika u kontekstu HTML-a predstavlja pravilnu upotrebu pravih elemenata u pravu svrhu. Npr. heading elemente koristimo za ispisivanje naslova jer im je to primarna svrha. Headinzi doprinose boljem pretraživanju i indeksiranju web stranica - SEO (search engine optimization).

### Break i horizontal rule elementi

Ukoliko želimo da započnemo novi red u tekstu dokumenta koristimo **break** elemenat čiji je tag <br/>.

Ovaj element spada u grupu tagova koji ne posjeduju završni tag.

Ukoliko želimo da vizualno odvojimo sadržaj HTML horizontalnom linijom koristimo **horizontal rule** <hr/>.

### Link

Hiperlink (skraćeno link) je poveznica, odnosno riječ, skupina riječi ili slika na koju možemo kliknuti kako bi nas povezala sa novim dokumentom ili web stranicom. Kada se pokazivač miša nalazi na linku on mijenja svoj izgled u ruku. Linkovi se prave pomoću <a> </a> taga, što je skraćenica od anchor (sidro).

Unutar početnog taga navode se atributi. Atributi (svojstva) su elementi u okviru tagova koji bliže određuju osobine tagova.

## Slika

Oni nose informacije o elementu i dati su u obliku: ime atributa = "njegova vrijednost omeđena navodnicima". U slučaju linka koristi se **href** atribut čija vrijednost određuje odredište na koje link vodi.

Za ubacivanje slike u HTML dokument koristi se **<img>** tag. Atributi **<img>** taga:

**src** (source atribut) određuje putanju do fizičkog fajla iz kojeg postavljamo sliku.

Atribut **alt** (alternative) - tekst koji će biti vidljiv u browseru ukoliko je učitavanje slike iz fizičkog dijela bilo neuspješno. Atributi **width** i **length** predstavljaju veličinu prikazane slike u pikselima. Element **img** nema svoj završni tag.

Primjer:

```
<!DOCTYPE html>
<html>
<head>
  <title> Naslov Web stranice </title>
</head>
<body>
  <a href="http://www.fit.ba">Kliknite ovdje da otvorite</a>
  <br/>
  
</body>

</html>
```

## Definisanje atributa lang

Definisanje jezika je veoma bitno zbog poboljšanja kvaliteta pretraživanja i bolje rangiranje stranice od strane web pretraživača. Koristimo atribut **lang** u **<head>** tagu.

## Meta podaci i meta tagovi

Meta podaci - podaci koji se ne vide u okviru browsera korisnika, ali su potrebni za pravilno ispisivanje sadržaja. Nisu dio sadržaja i nalaze se u **<head>** tagu. Meta tagovi - meta podaci koji sadrže određene informacije vezane za prikaz web sadržaja

Npr. svrha taga UTF-8 je pravilan prikaz svih aktivnih svjetskih jezika, pa tako i naših karaktera; č, ž, š, ć, đ. Kod za tag UTF-8: **<meta charset="UTF-8">**

## Još tagova

Tag **<pre>** **</pre>** umjesto **<p>** **</p>** će prikazati i novi red i razmaknice onako kako je to napisano u dizajnu stranice.

Primjer:

```
<!DOCTYPE html>
<html>
<head lang="en-US">
```

```
<title>Naslov </title>

<meta charset ="UTF-8">
</head>
<body>

<code>Tekst koda</code>

<br>

<kbd>Tekst sa tastature</kbd>

<br>

<samp>Primjer koda</samp>

<br>

<tt>Teletip test</tt>

<br>

<var>Tekst varijable</var>

<br>

<pre>Preformatirani tekst
za koji ne treba br tag
jer prikazuje nove redove
iz html-a
s a r a z m a c i m a
</pre>
</body>
</html>
```

## Formatiranje teksta

Formatiranje teksta se najčešće vrši pomoću CSS stilova uz pomoć **style** atributa. U html-u postoje elementi koji ovo rade i koji su se koristili u ranijim verzijama html-a.

Isprobajte slijedeći code u browseru

```
<p> Bold <b>bold</b> tekst </p>

<p> Strong <strong>Strong</strong> tekst </p>

<p> Naglašen <em>emphasized</em> tekst </p>

<p> Italik <i>italic</i> tekst </p>

<p> Markirani <mark>marked</mark> tekst </p>

<p> Umanjeni <small>small</small> tekst </p>
```

<p> Precrtani <del>deleted</del> tekst </p>

<p> Podvučeni ili ubaceni <ins>inserted</ins> tekst </p>

<p> Subscript <sub>subscript</sub> tekst </p>

<p> Superscript <sup>superscript</sup> tekst </p>

Primjer:

```
<!DOCTYPE html>
<html lang = "bs">
<head>
  <title> Naslov Web stranice </title>

  <meta charset =UTF-8>
</head>
<body>
  <p> Bold <b>bold</b> tekst </p>
  <p> Strong <strong>Strong</strong> tekst </p>
  <p> Naglašen <em>emphasized</em> tekst </p>
  <p> Italik <i>italic</i> tekst </p>
  <p> Markirani <mark>marked</mark> tekst </p>
  <p> Umanjeni <small>small</small> tekst </p>
  <p> Precrtani <del>deleted</del> tekst </p>
  <p> Podvučeni ili ubaceni <ins>inserted</ins> tekst </p>
  <p> Subscript <sub>subscript</sub> tekst </p>
  <p> Superscript <sup>superscript</sup> tekst </p>

</body>

</html>
```

## Tekst

Za prikaz teksta ili citata u okviru navodnika koristi se tag <q> (*engl. quote*). Za slučaj da treba kao citat, ili kao navedeni tekst da prikazemo neki blok tekst ili odlomak, koristi se <blockquote> tag.

Za skraćenice se koristi abbr tag (*engl. abbreviation*).

Ovaj tag je veoma koristan prilikom pretraživanja u okviru web pretraživača i koristan je za davanje više informacija o nekim skraćenicama.

Primjer:

```
<!DOCTYPE html>
<html>
<head lang="en-US">
  <title> Naslov Web stranice </title>

  <meta charset ="UTF-8">
</head>
<body>
```

```

<p>Neki tekst sa citatom:<q> Ovo je citat. </q>
<br>

<br>
</p>
<hr>
<p> Neki drugi citat:<p>
<blockquote>
Neki drugi citat.
Neki drugi citat.
Neki drugi citat.
</blockquote>
<hr>
<p><abbr title="Fakultet informacijskih tehnologija">FIT</abbr> "Još neko objašnjenje."
</p>
<hr>
<p>Adresa FIT-a</p>
<address>
Univerzitet "Džemal Bijedić" Mostar <br>
Sjeverni Logor br.12 <br>
88000 Mostar, BiH <br>
TEL/FAX: +38736811130<br>
</adress>
</body>

</html>

```

## Komentari

Komentare možemo unositi bilo gdje unutar HTML dokumenta. Komentari služe za dodatno pojašnjenje code-a. Možemo privremeno isključiti dio code-a za potrebe debug-iranja. Za ubacivanje komentara potrebno je na početku komentara napisati: `<!--` a na kraju: `-->`. Na ovaj način, sve što se nalazi između navedenih znakova bit će ignorisano od strane preglednika.

## Specifični formati

Elementi `<code>`, `<samp>` i `<kbd>` prikazuju neproporcionalna slova.

Element `<kbd>` prikazuje podebljana slova.

Element `<samp>` prikazuje nakošena slova.

Element `<var>` prikazuje nakošena slova

Element `<pre>`, čuva razmake i prelazak u novi red.

## HTML liste

Liste organizuju popis stavki u sadržaju.

Tri su vrste lista u HTML-u:

- uređena lista: `<ol> ... </ol>`

- neuređena lista: `<ul> ... </ul>`
- definicijska lista: `<dl> ... </dl>`

Unutar liste su elementi/item koji se navode pomoću taga `<li>` (engl. *list item*).

Item-i mogu sadržavati druge HTML elemente.

### *Uređene liste*

Elementi uređene liste su:

uređena lista (Ordered list) : `<ol> ... </ol>`

stavka liste (List item): `<li> ... </li>`

### *Neuređene liste*

Elementi neuređene liste su:

neuređena lista (Unordered list) : `<ul> ... </ul>`

stavka liste (List item): `<li> ... </li>`

Često se koriste u kombinaciji sa script jezicima za menije, albume slika i sl.

### *Definicijske liste*

Definicijske liste za kreiranje koriste dva elementa:

`<dt>` - pojam koji se kreira i

`<dd>` - definicija pojma kojeg smo kreirali.

Elementi definicijske liste:

- definicijska lista (Definition list) : `<dl> ... </dl>`
- nazivi definicije: `<dt> ... </dt>`
- Definicija/navođenje definicije: `<dd> ... </dd>`

Moguće je kaskadno ugnjezditi liste.

Primjer:

```
<!DOCTYPE html>
<html>
<head lang="en-US">
  <title> Naslov</title>

  <meta charset ="UTF-8">
</head>
<body>

<!--Neuređena lista-->
<p>Predmeti:</p>
```

```

</ul>
<li>PR I</li>
<li>PR II</li>
<li>WRD</li>
<li>Engleski</li>
<li>Matematika</li>
</ul>
<!--Uređena lista-->
<p>Predmeti:</p>
<ol>
<li>PR I</li>
<li>PR II</li>
<li>WRD</li>
<li>Engleski</li>
<li>Matematika</li>
</ol>
<!--Primjer za definicijsku listu-->
<dl>
<dt>Numericki popis</dt>
<dd>- PR I.</dd>
<dt>Programiranje I</dt>
<dd>- PR II.</dd>
</dl>
<!--Kombinovanje listi i ugnježdavanje-->
<ul>
<li>Prva godina</li>
<ol>
<li>PR I</li>
<li>PR II</li>
</ol>
<li>Druga godina</li>
<ul>
<li>WRD</li>
<li>PSR</li>
</ul>
</ul>
</body>
</html>

```

## Tabele

Kreiraju se **<table>** tagom. Redove definišemo **<tr>**. Čelije definišemo **<td>** tagom.

Čelije mogu sadržavati bilo koji tipa podatka: tekst, sliku, link, formu, druge tabele i sl.

Najčešće korišteni atributi su: border,width, align.

```

<!DOCTYPE html>
<html>
<head lang="en-US">
  <title> Naslov </title>

  <meta charset ="UTF-8">

```

```

</head>
<body>

<table border="1">
<tr>
<th>Godina</th>
<th>Naziv <br>predmeta</th>
<th>Broj <br>casova</th>
<th>ECTS</th>
</tr>
<tr>
<td>Prva</td>
<td>PR I</td>
<td>60</td>
<td>10</td>
</tr>
<tr>
<td>Prva</td>
<td>PR II</td>
<td>60</td>
<td>10</td>
</tr>
<tr>
<td>Druga</td>
<td>WRD</td>
<td>60</td>
<td>10</td>
</tr>
<tr>
<td colspan="3">Total:</td>
<td>30</td>
</tr>
</body>
</html>

```

## Block i inline elementi

HTML elemente generalno možemo podijeliti na blok elemente i inline elemente.

Blok elementi - prikazuju se tako što počinju novom linijom i zauzimaju cijelu širinu web stranice.

**heading, paragraf, ul lista, div, table,** itd.

Inline elementi - prikazuju se tako da ne lome red.

**b tag, td, a tag, span, img** itd.

## Div i span elementi

### *div*

Može se koristiti kao kontejner za sve ostale elemente. Nema direktno semantičko značenje i direktne attribute. Može imati dodatne attribute: style, class i id. Div je blok element.



## **span**

Kontejner kratkih tekstova sa zadatim stilom. Ne lomi red, odnosno span je inline element.

## **Iframe element**

Iframe je html element koji sadrži drugu html stranicu. Atribut src je adresa web stranice koja se prikazuje u frame-u.

Atributi width i height mogu biti i u procentima.

Ime iframe-a može biti vrijednost atributa target u okviru linka.

```
<!DOCTYPE html>
<html>
<head lang="en-US">
  <title> Naslov </title>

  <meta charset ="UTF-8">
</head>
<body>

<iframe width="700" height="500"
name="iframe_FIT">/iframe>
<p > Kliknite <a href="http://www.fit.ba/"
target="iframe_FIT">Link za pristup</a> službenoj stranici FIT-a</p>
</body>
</html>
```

## **HTML forme**

HTML forme su se pojavile 1992. godine kao dio specifikacije HTML 2.0. Služe za kreiranje različitih vrsta korisničkog unosa i prikaza podataka. Koriste se za dostavljanje podataka serveru, kao i za prikaz podataka dobijenih od servera.

Same forme ne omogućuju nikakvu funkcionalnost. Preko njih se prikupljaju i prikazuju podaci. Na jednoj stranici može biti više formi.

### **Elementi HTML forme**

Najčešći elementi HTML forme su ulazni elementi:

- textboxovi za unos teksta,
- textboxovi za unos lozinki,
- polja potvrde i kružići sa opcijama,
- padajući popisi,
- dugme za submitiranje,
- ostala dugmad (cancel, back) itd.

## Funkcionalnost

Forme predstavljaju neku vrstu obrasca za prikupljanje podataka. Prikupljeni podaci se proslijeđuju skripti na serveru koja obavlja određenu funkciju. Na jednoj stranici može biti više formi.

Forma je HTML element koji sadrži razne atribute i podelemente. Sve što se proslijeđuje serveru, stavlja se u sadržaj ovog taga.

Obavezni atributi svake forme su:

- Action
- Method
- Name
- ID

Neobavezni atributi su:

- Enctype
- Target

## Atribut Method

Definiše na koji način će podaci biti proslijeđeni. Moguća su dva načina:

- **GET method** – Podaci iz forme se šalju server kroz URL (Nije preporučljivo jer su informacije nezaštićene i podložne hakovanju)
- **POST method** – Pakuje podatke iz forme unutar HTTP zahtjeva (Podaci nisu šifrirani, pa su, iako sigurni negu putem GET method, i dalje podložni napadima hakera)

## Atributi Name i ID

- Atribut **ID** – koristi se za jedinstveno određivanje HTML elementa na stranici ili za stilizaciju u CSS-u. ID mora biti jedinstven.
- Atribut **Name** – određuje ime forme i proslijeđuje se serveru.

Preporučljivo je postavljati oba atributa sa istim vrijednostima

## Atributi Enctype i Target

- Atribut **Enctype** – određuje na koji način će podaci biti enkodirani prilikom slanja na server.
- Atribut **Target** – određuje na koji način će se pokrenuti stranica iz Action atributa

## Kontrole HTML forme

Zadužene za prikupljanje i proslijeđivanje podataka.

Postoji više tipova:

- Tekstualni tip
- Radio / Check buttons

- Submit buttons itd.

### *Input*

Najčešći tag u formama je **<input>**. Input je samozatvarajući tag jer se svi njegovi parametri podešavaju kroz attribute. Koristi se za različite kontrole, a na šta se tačno odnosi određuje obavezni atribut **type**.

### *Input (text)*

Da bi unijeli obični textbox, unijet ćemo input tag sa type="text" atributom. Atribut **name** u elementu **<input>** omogućava skripti da prepozna podatak. Atribut **maxlength** je također moguće dodati i njime određujemo maksimalan broj karaktera. Osim toga, ukoliko želimo da neko polje ima početnu vrijednost, koju korisnik obriše pa unese vlastitu, koristimo atribut **value**.

### *Label tag*

Da bi korisnik znao šta treba da unese u dato polje koristimo **<label>** tag koji sadrži u sebi atribut **for** koji mu omogućava da se poveže sa određenim **<input>** tagom.

### *Input (password)*

Za razliku od unosa teksta, ukoliko želimo da omogućimo korisniku da unese password koristimo **<input>** tag ali sa type-om password.

### *Radio dugmad (radio button)*

Koristi se za predefinisane opcije. Ne unosi se vrijednost nego se bira samo jedna od ponuđenih opcija. Svaki izbor zahtijeva poseban **<input>** tag.

Da bi browser znao da je više tagova grupisano zajedno, moraju imati isto ime u **name** atributu (ID mora biti razlícit)

### *Checkbox*

Koristi se također za predefinisane opcije. Razlika između ove kontrole i radio button kontrole je u tome što korisnik može označiti više polja

### *Submit dugme*

Služi za prosljeđivanje podataka iz forme serveru na daljnu obradu. ID i name nisu obavezni, a value definiše ono što će pisati na dugmetu. Također umjesto teksta može se nalaziti i slika na dugmetu.

### *Button (dugme)*

Novija kontrola koja može obuhvatiti više elemeneta npr. i sliku i tekst. Nije obavezno ali se jako često koristi. Koristi se za brisanje svega što je uneseno u sva polja.

### *Upload fajla*

Za uploadovanje raznih fajlova (dokumenata, slika, video klipova itd.) koristi se atribut type="file". Forma mora biti podešena na POST method

Enctype mora biti *enctype="multipart/form-data"*. Ništa nije moguće stilizovati CSS-om, ni dugme, ni polja, ni ostalo, već to zavisi od browsera

### *Textarea*

Kreira veće polje za tekst i podržava više redova. Nije samozatvarajući tag (zahtijeva početak i kraj), između se unosi tekst koji će biti prikazan u browseru.

## ***Datumska polja***

Sadržava podatke o datumu, vremenu, mjesecu itd.

## ***Padajući izbornik***

Omogućava korisniku da odabere jednu od vrijednosti iz padajućeg menija.

Primjer:

```
<!DOCTYPE html>
<html>
<body>

<form action="negdje_saljemo_ovo.asp" >

Ime:<br>
<input type="text" name="firstname" value="Elmir" required>
<br>
Prezime:<br>
<input type="text" name="lastname" required>
<br>
Kanton: <br> <input type="text" name="kanton" pattern="[A-Za-z]{5}" title="Unesi 5 slova skracenice kantona">
<br>
Izaberi dokumente: <input type="file" name="dokumenti" multiple>
<br>

Datum rođenja:
<input type="date" name="bday" max="dd-mm-yyyy"><br>
<br>

Datum zaposlenja:
<input type="date" name="zaposlenje" min="31-12-2000"><br>
<br>

Unesi broj predmeta 1-10:
<input type="number" name="brpredmeta" min="1" max="10"><br>
<br>
<input type="range" name="neki_rang" min="100" max="1000" value="200">
<br>
E-mail: <input type="email" name="email" autocomplete="on"><br>

<input type="submit">

</form>

</body>
</html>
```

## **CSS**

CSS – Cascading Style Sheet

Predstavlja opis načina formatiranja. Nime razdvajamo HTML sadržaj od pravila formatiranja.

### **Načini uključivanja u HTML**

- Inline
- Interni
- Eksterni

#### **Inline**

Piše se u HTML code-u Ne mijenja stil kompletnog taga. Važi za trenutnu liniju.

Primjer:

```
<!DOCTYPE html>
<html>
<body>
<h1> H1 je ovakav</h1>
<h1 style="color:blue; background:red;">Plavi tekst</h1>
<h1> Proba teksta</h1>
<h1> H1 se nije promijenio</h1>

</body>
</html>
```

#### **Interni CSS**

Piše se u head sekciji HTML. Važi u domenu tog HTML file-a.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1 {color: blue; background-color: #00ff00;}
p {color: red; background-color: rgb(0,0,255);}
</style>
</head>
<body>
<h1> H1 je sada ovakav</h1>
<h1> Proba teksta</h1>
<h1> H1 se promijenio</h1>
<p>I p tag se promijenio</p>

</body>
</html>
```

#### **Eksterni CSS**

Uključuje se u HTML. Više HTML file-ova mogu koristiti isti CSS. Jedan HTML može koristiti više CSS. Olakšava stiliziranje kompletnog site-a. Pogodan je za „Corporate look“ site-a jer omogućava da se jedan stil lakše primjeni na sve html dokumente.

```

<!DOCTYPE html>
<html>
<head>

<link rel="stylesheet" type="text/css" href="novi_stil.css">

</head>
<body>
<h1> H1 je sada ovakav</h1>
<h1> Proba teksta</h1>
<h1> H1 se promijenio</h1>
<p>I p tag se promijenio</p>
<neki_stil> Neki tekst nekog novog stila</neki_stil>

</body>
</html>

```

CSS za navedeni primjer koji treba snimiti kao: novi\_stil.css

```

body {background-color: rgb(0,10,10);}
h1 {display: inline; color: blue; background-color: #00ff00;}
p {color: red; background-color: rgb(0,10,255);}
neki_stil{display: block; color: blue; background-color: rgb(20,10,50);}

```

## CSS – Best Practice

Uključivati CSS kao eksterni file. Po potrebi modifikovati stil inline.

### Prioriteti stilova – Kaskadni redoslijed (Cascading order)

1. Inline stil u html elementu
2. Eksterni i interni CSS
3. Default HTML stil

**Pravilo: Ako je stil istog selektora definisan na više mjesta, za element se uzima stil prema navedenom redoslijedu prioriteta**

Moguće je uključiti više različitih eksternih CSS fajlova. Ukoliko ima više eksternih CSS stilova koji definišu istu osobinu istog selektora, uzima se ona osobina od zadnjeg učitanoog stila.

## Selektori

Pomoći CSS selektora se identificira element u html code-u na koji se treba primjeniti stil

Pronalaženje elemenata se bazira na njihovom:

- Imenu – name
- ID
- Klasi – class

- Atributu
- ...

### Element selektor

Element selektor – koristi ime elementa – name.

Npr.:

```
h1 {color: blue; background-color: #00ff00;}
```

```
lli
```

```
h1
```

```
{
```

```
color: blue;
```

```
background-color: #00ff00;
```

```
}
```

### ID selektor

ID selektor – koristi ID atribut html elementa.

Npr.:

```
#nekiStilID
```

```
{
```

```
color: yellow;
```

```
background-color: rgb(0,0,255);
```

```
text-align: center;
```

```
font-size:250%;
```

```
}
```

### Class selektor

Class selektor kreira klasu nekog stila i kao takav može se koristiti u tag-u.

Npr.:

```
.nekaKlasa
```

```
{
```

```
color: red;
```

```
background-color: rgb(0,0,255);
```

```
text-align: right;

font-size: 150%;

font-weight: bold;

}
```

Ili definicija klase koja se odnosi samo na jedan element npr. p

```
p.nekaDrugaKlasa

{

color: yellow;

background-color: rgb(255,0,50);

}
```

### Grupisanje selektora

Ukoliko imamo nekoliko različitih selektora sa istim stilom kao npr.:

```
h1 {
    text-align: right;
    color: blue;
}

h2 {
    text-align: right;
    color: blue;
}

p {
    text-align: right;
    color: blue;
}
```

mogu se grupisati i napisati kao:

```
h1, h2, p {
    text-align: right;
    color: blue;
}
```

### CSS komentari

Identični kao blok komentar u C++.

```
/* ovo je komentar u CSS code-u */
```

```
/* komentar se može
```



*pisati u više*

*linija\*/*

### Još neka pravila pisanja CSS-a

Property vrijednosti i jedinica se ne smiju razdvajati space-om. Npr:

- *font-size: 250%; - OK*
- *font-size: 250 %; - nije ispravno*

### Boje

U CSS postoji tri načina identificiranja boje:

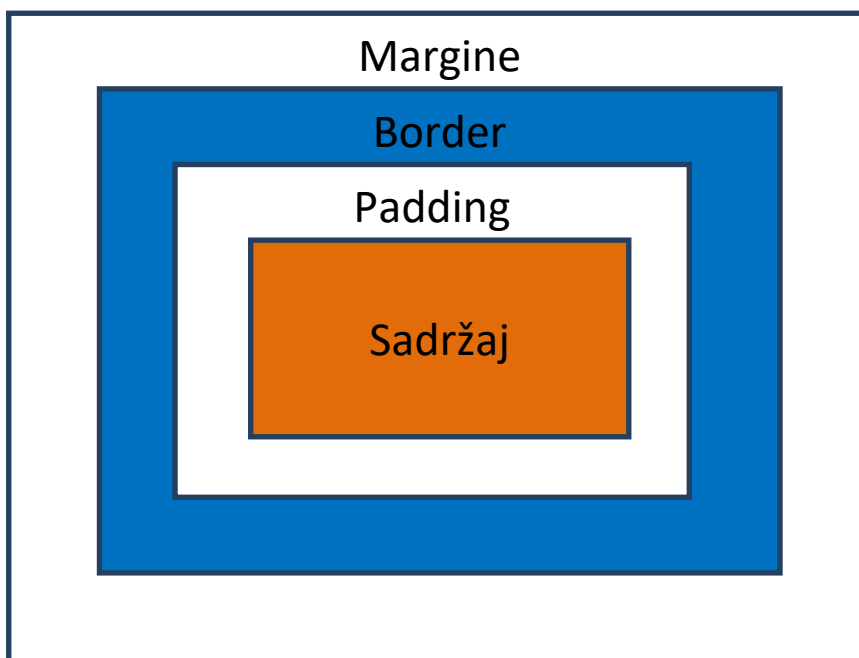
- Nazivom boje: **color: red;**
- RGB paletom: **color: rgb(0,0,255);**
- Hexadecimalnom vrijednošću: **color: #00ff00;**

Nazivi boja nisu case-sensitive. Paleta od 140 boja se može naći na slijedećem linku:

<http://www.cssportal.com/css3-color-names/>

### Borders, Margine, Padding, Veličina

Box model



### Block i Inline elementi

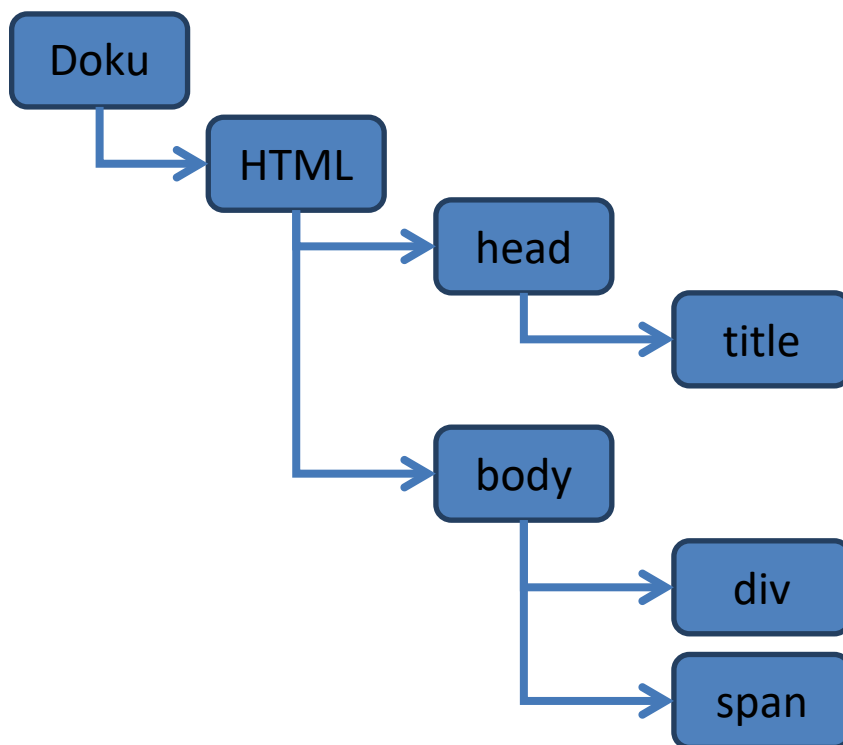
- Blok elementi zauzimaju cijeli red
  - Div elementi < div >
  - Paragrafi < p >
  - Liste < ul >

- Stavke liste < li >
- Naslovi < h1 > – < h6 >
- Tabele < table >
- Inline elementi zauzimaju prostor prema sadržaju
  - Span elementi<span>,
  - Linkovi<a>,
  - Bold formatiranje<strong> ili <b>,
  - Italic formatiranje<em> ili <i>,
  - Slike<img>,
  - Form oznake<label>.

### Div i Span elementi

- div – blok element koji obično obuhvata dio stranice
- span – inline element koji obično ima neki sadržaj

DOM – Document Object Model



## Pozicije, forsiranje block-inline

Moguće je promijeniti defaultno ponašanje elementa tako da block element postane inline i obratno.

## Parent>Child element selektor

Koristi se za definiciju stila elementa u slučaju specifične kombinacije parent>child. Npr. `div > ul` ili `span > p`. U ovom slučaju stil se odnosi isključivo ako je direktni parent element onaj koji je naveden. Inače element ostaje nepromjenjen.

Ostala parent>child pravila:

- `div p` U ovom slučaju odnosi se na bilo koji p pod div
- `div + p` Imaju isti parent a p slijedi odmah iza div
- `div ~ p` Imaju isti parent a ne mora odmah slijediti

## CSS property – multiple values

Moguće je za jedan property postaviti više vrijednosti:

*font: italic bold 12px/30px Georgia, serif;*

**Property mora biti naveden** jer CSS **ne omogućava** automatsko prepoznavanje vrijednosti i dodjeljivanje „jedinom mogućem“ property-ju

## CSS pseudo-elementi i pseudo-klase

- Pseudo-element: koristi se da formatira dio elementa

```
selector::pseudo-element {  
  property:value;  
}
```

- Pseudo-klasa: koristi se da formatira element u specifičnom stanju

```
selector:pseudo-class {  
  property:value;  
}
```

Pseudo-element formatiranje je primjenjivo na blok elementne. Stil se primjenjuje čak i ako je osobina elementa promjenjena

## Pseudo-klasa

Primjeri:

Primjer promjene izgleda linka.

*a:link, a:visited*

```
{
```

*color:yellow;*

```
}
```

Primjer pseudo-klase u kombinaciji sa parent-child selektorima u ovom slučaju pseudo-klasa se odnosi na div element koji je parent span elementu u stanju pointera postavljenog preko elementa

```
div:hover span {  
  
    display: block;  
  
}
```

Primjer za pseudo-element i pseudo-klasu:

```
<!DOCTYPE html>  
<html>  
<head>  
  
<link rel="stylesheet" type="text/css" href="novi_stil_5.css">  
  
</head>  
<body>  
<div id="vrh">  
<h1> Ovo je H1 tekst. Korištenje pseudo-elementa omogućava dizajneru da formatira specifični dio  
elementa kao npr. prvu liniju, prvo slovo, selekciju itd. Pseudo-element može biti jako koristan.</h1>  
</div>  
<div id="dno">  
<p> Ovo je tekst na dnu</p>  
</div>  
</br>  
<div id="sadržaj">  
  <div id="lijevi">  
    <ul class = "izbor">  
      <li class = "izborLinija"><a href="http://www.fit.ba">Programiranje I</a></li>  
      <li class = "izborLinija"> <a href="./EksterniCSS_inline_block.html">WRD</a></li>  
      <li class = "izborLinija">PSR</li>  
      <li class = "izborLinija">SP</li>  
    </ul>  
  </div>  
  <div id="srednji">  
<p>Pređi pointerom preko ovog teksta za demonstraciju pseudo-klase u kombinaciji sa parent-child  
selektorima</p>  
<span class = "negdjeSmjestamo">Span tekst Span tekst Span tekst Span tekst Span tekst Span tekst  
Span tekst Span tekst Span tekst Span tekst Span tekst Span tekst Span tekst Span tekst  
Span tekst Span tekst Span tekst </span>  
</div>  
  <div id="desni">  
    <ul class = "izbor">  
      <li class = "izborLinija"><a href="http://www.fit.ba">Programiranje I</a></li>  
      <li class = "izborLinija"> <a href="./EksterniCSS_inline_block.html">WRD</a></li>  
      <li class = "izborLinija">PSR</li>  
      <li class = "izborLinija">SP</li>  
    </ul>
```

```
</div>
</div>
</body>
</html>
```

CSS file za navedeni primjer (treba ga snimiti kao novi\_stil\_5.css)

```
body
{
background-image: url("slika.jpeg");
background-size: cover;
background-repeat: no-repeat;
background-position: left top;
background-attachment: fixed;
}
h1::first-line {color: blue; background-color: #00ff00;}
/*primjer pseudo-elementa - formatiranje prve linije h1 elementa-selektora*/
::selection {
color: red;
background: yellow;
}
/*primjer pseudo-elementa koji se odnosi na sve blok elemente*/
p
{
font-size:150%;
border-bottom: 2px dotted blue;
background-color: lightgrey;
margin-top: 3px;
margin-bottom: 3px;
margin-right: 1%;
margin-left: 1%;
padding-top: 1px;
padding-right: 1px;
padding-bottom: 1px;
padding-left: 1px;
}
.negdjeSmjestamo
{
position:relative;
top: 50px;
width: 90%;
height: auto;
font-weight: bold;
font-size:120%;
}

/*
div > ul
{
background-color: yellow;
}
```

```
*/  
/*primjer parent-child pseudo klase – ukloniti komentar po potrebi*/
```

```
.izbor{  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
}  
.izborLinija{  
    box-sizing: border-box;  
    padding: 8px;  
    margin-bottom: 8px;  
    background-color: blue;  
    color: #ffffff;  
}  
.izborLinija:hover {  
    background-color: magenta;  
}
```

```
a:link, a:visited  
{  
    color:yellow;  
}
```

```
/*primjer promjene izgleda linka*/
```

```
span {  
    display: none;  
    color:yellow;  
    background-color: blue;  
    padding: 1px;  
}
```

```
div:hover>span {  
    display: block;  
}
```

```
/*primjer pseudo-klase u kombinaciji sa parent-child selektorima  
u ovom slucaju pseudo-klasa se odnosi na div element koji je parent span elementu  
u stanju pointera postavljenog preko elementa*/
```

```
/*Raspored elemenata na stranici*/
```

```
#sadrzaj {  
    width:100%;  
    text-align:center;  
}
```

```
#vrh {  
    position: relative;  
    top: 2px;  
    width:100%;  
    text-align:center;  
    border: 2px solid red;  
}
```

```
#dno {  
    position: fixed;  
    bottom: 10px;  
    width:100%;  
    text-align:center;  
border: 2px solid red;  
}  
  
#lijevi {  
    display: inline-block;  
    float:left;  
    width:10%;  
border: 2px solid red;  
}  
  
#srednji {  
    display: inline-block;  
    width:78%;  
    height: auto;  
border: 2px solid red;  
}  
  
#desni {  
    display: inline-block;  
    float:right;  
    width:10%;  
border: 2px solid red;  
  
}
```

## Detaljno o CSS-u – pregled kompletne CSS specifikacije

Za više informacija o CSS definiciji posjetite slijedeće web stranice:

- CSS – <https://www.w3.org/Style/CSS/current-work.en.html>
- <https://www.w3.org/TR/2011/REC-CSS2-20110607/css2.pdf>
- <https://drafts.csswg.org/css-syntax/>

## Scripting jezici

Omogućavaju dinamičnost web stranice

- Razlikujemo dva principa rada:
  - Client-side scripting
  - Server-side scripting

## Client-side scripting

U većini slučajeva radi se o JavaScript jeziku. Princip rada je da server "pošalje" source-code zajedno sa html i css code-om. Klijent – browser izvršava primljeni code na klijentskom računaru. Može imati sigurnosne rizike pa mnogi browser-i ali i antivirusi imaju ugrađenu zaštitu. Obično se treba omogućiti (enable) scripting na browseru.

## Server-side scripting

Postoji više različitih jezika koji se aktivno koriste i služe za server-side scripting kao npr.:

- ASP, ASP.NET,
- PHP,
- ColdFusion
- ...

U slučaju server-side scripting, code se izvršava na serveru, a klijentskom računaru se šalje samo "generisani" html koji se generiše prilikom izvršenja code-a na serveru.

Koristi se mahom za web-bazirane aplikacije sa bazom podataka

## JavaScript

Pojavio se 1995 u sklopu Netscape Navigatora v. 2.0 (LiveScript). Od 1996 ga podržavaju i ostali browseri. Standardiziran 1997 (ECMA Script).

JavaScript je client-side scripting jezik. On nema veze sa Java jezikom. Za JavaScript se može reći da je **programski jezik**. On obezbjeđuje interaktivnost i dinamičnost web stranice. JavaScript ima objektno orjentisane mogućnosti. Sintaksa mu je bazirana na C jeziku.

## Razlozi korištenja

HTML služi za opis sadržaja i ima ograničene mogućnosti. CSS služi sa pasivno formatiranje stranice

JavaScript se koristi za:

- Kreiranje dinamičkih web stranica
- Interakciju sa korisnikom
- Napredne multimedijalne mogućnosti

## Mjesto uključivanja JavaScripta

- u HEAD tagu
- u BODY tagu
- kombinovano

## Načini uključivanja JavaScript

1. Interni code u HTML-u (preko script tag)



2. Eksterni file (preko Javascript biblioteke)
3. *Kroz atribut (preko html tag atributa)*

### ***Interni code u HTML-u***

Sintaksa:

```
<script language="JavaScript"> ... </script>
```

Npr.

```
<head>
```

```
    <title>Neki naslov </title>
```

```
    <script language="JavaScript">
```

```
        alert("Poruka");
```

```
    </script>
```

```
</head>
```

### ***Eksterni file***

Sintaksa:

```
<script language="JavaScript" src="myLib.js" />
```

Npr.

```
<head>
```

```
    <title> Neki naslov </title>
```

```
    <script language="JavaScript" src="myLib.js"/>
```

```
    <script>
```

```
        //Ovdje se može pisati code da se koriste funkcije iz myLib.js
```

```
    </script>
```

```
</head>
```

### ***Kroz atribut***

Neki HTML tag atributi imaju posebne osobine. Te osobine podsjećaju na event objekta u klasičnim programskim jezicima. To im omogućava da imaju neku vrstu svog internog script-a.

Primjer: **onclick** ili **onload**

```
<input type="button" onclick="javascript:" />
```

Npr.

```
<body>

    <input type="button" onclick="alert('Poruka');" />

</body>
```

Primjer tri navedena načina uključivanja:

```
<!DOCTYPE html>
<html>
<head>
    <script type="text/javascript" src="mojSkript.js"></script>
</head>
<body>
    <script type="text/javascript">
        document.write("Primjer internog JavaScript code-a");
    </script>
<br/>

    <button onclick="getElementById('vrijeme').innerHTML=Date() + '<br/>' + ' Ovo je primjer JavaScript-a u tag-u' ">Vrijeme?</button>

    <h1 id="vrijeme"></h1>

    <button onclick="poruka()">Poruka?</button>

</body>

</body>
</html>
```

Slijedeći code snimiti kao mojSkript.js

```
function poruka() {
    alert("Ovo je poruka koju je generisao JavaScript iz eksterne biblioteke")
}
```

## Variable

JavaScript je **weakly typed language**. Tip varijabli se ne navodi prilikom deklaracije. Tip varijable se može promijeniti prilikom alokacije vrijednosti – dinamičko pisanje code.

JavaScript je zato:

- **weakly typed**
- **dinamički jezik**

## Deklaracija varijabli

Deklaracija sa inicijalizacijom

- Sintaksa: **var name = initialization\_value;**

- Primjer: var d = 4.567;

Deklaracija bez inicijalizacije

- Sintaksa: **var name;**
- Primjer: var b;

Višestruka deklaracija

**var name1, name2, name3, etc.;**

Npr.

var i, j, k, l;

### *Osnovni tipovi*

i= 123; - numerički

i= "Neki string"; - string

i= true; - boolean

Zbog weak typing osobine, ista varijabla može poprimiti različite tipove podataka. To je jedna od bitnih razlika u odnosu na C jezik.

Primjer:

```
<!DOCTYPE html>
<html>
<head>
  <script type="text/javascript" src="mojSkriptVar.js"></script>

</head>
<body>
  <script type="text/javascript">
i= 123;
document.write("<h1>");
document.write(i);
document.write("<br>");
i= "Neki string";
document.write(i);
document.write("<br>");
i= true;
document.write(i);
document.write("</h1>");
document.write("<h1>NAPOMENA: JavaScript se koristi i za dinamičko generisanje HTML
tagova</h1>");
  </script>

  <button onclick="poruka()">x===5</button>
  <button onclick="poruka2()">x==='5'</button>
<br><br><br>
```

```

<button onclick="document.getElementById('proba').innerHTML =sabiranje (2,3)
">Sabiranje</button>
<br><br>
<button onclick="document.getElementById('proba2').innerHTML = result_1; ">Prikaži varijablu
1</button>
<br><br>
<button onclick="document.getElementById('proba3').innerHTML =result_2; ">Prikaži varijablu
2</button>
<br><br>
<h1 id="proba"></h1>
<h1 id="proba2"></h1>
<h1 id="proba3"></h1>

</body>
</html>

```

Slijedeći code za potrebe primjera snimiti kao mojSkriptVar.js

```

function poruka() {
var x = 5;
    alert(x===5)
}

function poruka2() {
var x = 5;
    alert(x==="5")
}

function sabiranje(value1, value2){
    result_1 = value1+value2; //globalna varijabla
    var result_2 = value1+value2; //lokalna varijabla
return result_2;
}

```

## Komentari

Identični su kao u C jeziku:

*// komentar u jednoj liniji*

*/\**

*Blok komentar*

*\*/*

## Varijable

Imena varijabli moraju početi

- Slovom

- Znakom \$
- Znakom \_

ime varijable ne može sadržavati space ili specijalne karaktere. JavaScript je case-sensitive tako da Varijabla, varijabla, vaRijabla predstavljaju različite varijable.

### Scope varijable

Varijabla se može koristiti i ako ne navedemo ključnu riječ **var**. Ukoliko nije navedena ključna riječ var, varijabla je globalna i vidljiva je iz svih dijelova stranice. Ukoliko je navedena ključna riječ var, varijabla postaje lokalna i vidljiva je samo toj funkciji

Zbog osobine da se navođenjem **var** svodi varijabla na lokalni domen može se koristiti sljedeće:

```
<script>
```

```
var nekitekst = "ovo je globalna varijabla";
```

```
function domenVarijabla() //ova deklaracija ima domen skripta
```

```
{
```

```
var nekitekst = "ovo je lokalna varijabla";
```

```
//ova deklaracija ima domen funkcije
```

```
document.writeln(nekitekst);
```

```
}
```

```
document.writeln(nekitekst);
```

```
</script>
```

### Operatori

- Aritmetički operatori: + - \* % /
- Boolean (logički) operatori: && || !
- Operatori poređenja: == != > < <= >=
- Alokacijski operatori: =, +=, -=, \*=, /=, %=, ++, --
- Operator + u slučaju varijabli tipa string služi za konkatenciju stringova
- Operator ===
- Operator === poredi tipove varijabli i vrijednosti
- ""==0 // true

- `""===0 // false`

## Funkcije u JavaScript

- Funkcije bez povratne vrijednosti
  - Ekvivalent void funkciji u C jeziku
- Funkcije sa povratnom vrijednošću
  - Ekvivalent funkciji sa tipom u C jeziku

### Funkcije bez povratne vrijednosti

Sintaksa poziva: **functionName(arg1,arg2,arg3, etc.);**

Primjer: `alert("Neka poruka");`

### Funkcije sa povratnom vrijednošću

Sintaksa poziva: **result = functionName(arg1, ... );**

Primjer: `ime = prompt("Kako se zoveš?");`

### Deklaracija/Definicija Funkcije bez povratne vrijednosti

*function functionName(arg1,arg2,arg3, ... ){*

*// instructions*

*}*

### Deklaracija/Definicija Funkcije sa povratnom vrijednošću

*function functionName(arg1,arg2,arg3, ... ){*

*// instructions*

*return result;*

*}*

## Prilog 1 - HTML Cheatsheet

### Osnovni tagovi

`<html> </html>`

Creates an HTML document

`<head> </head>`

Sets off the title & other info that isn't displayed

`<body> </body>`

Sets off the visible portion of the document

`<title> </title>`

Puts name of the document in the title bar; when bookmarking pages, this is what is bookmarked

### Text Tagovi

`<pre> </pre>`

Creates preformatted text

`<h1> </h1> --> <h6> </h6>`

Creates headlines -- H1=largest, H6=smallest

`<b> </b>`

Creates bold text (should use `<strong>` instead)

`<i> </i>`

Creates italicized text (should use `<em>` instead)

`<tt> </tt>`

Creates typewriter-style text

`<code> </code>`

Used to define source code, usually monospace

`<cite> </cite>`

Creates a citation, usually processed in italics

`<address> </address>`

Creates address section, usually processed in italics

`<em> </em>`

Emphasizes a word (usually processed in italics)

`<strong> </strong>`

Emphasizes a word (usually processed in bold)

`<font size=?> </font>`

Sets size of font - 1 to 7 (should use CSS instead)

`<font color=?> </font>`

Sets font color (should use CSS instead)

`<font face=?> </font>`

Defines the font used (should use CSS instead)

### Linkovi

`<a href="URL">clickable text</a>`

Creates a hyperlink to a Uniform Resource Locator

`<a href="mailto:EMAIL_ADDRESS">clickable text</a>`

Creates a hyperlink to an email address

`<a name="NAME">`

Creates a target location within a document

`<a href="#NAME">clickable text</a>`

Creates a link to that target location

### Formatiranje

`<p> </p>`

Creates a new paragraph

`<br>`

Inserts a line break (carriage return)

`<blockquote> </blockquote>`

Puts content in a quote - indents text from both sides

`<div> </div>`

Used to format block content with CSS

`<span> </span>`

Used to format inline content with CSS

### Liste

`<ul> </ul>`

Creates an unordered list

`<ol start=?> </ol>`

Creates an ordered list (start=xx,  
where xx is a counting number)

`<li> </li>`

Encompasses each list item

`<dl> </dl>`

Creates a definition list

`<dt>`

Precedes each definition term

`<dd>`

Precedes each definition

### Grafički elementi

`<hr>`

Inserts a horizontal rule

`<hr size=?>`

Sets size (height) of horizontal rule

`<hr width=?>`

Sets width of rule (as a % or absolute pixel length)

`<hr noshade>`

Creates a horizontal rule without a shadow

``

Adds image; it is a separate file located at the URL

``

Aligns image left/right/center/bottom/top/middle (use CSS)

``

Sets size of border surrounding image (use CSS)

``

Sets height of image, in pixels

``

Sets width of image, in pixels

``

Sets the alternate text for browsers that can't  
process images (required by the ADA)

### Forme

`<form> </form>`

Defines a form

`<select multiple name=? size=?> </select>`

Creates a scrolling menu. Size sets the number of  
menu items visible before user needs to scroll.

`<select name=?> </select>`

Creates a pulldown menu

`<option>`

Sets off each menu item

`<textarea name=? cols="x" rows="y"> </textarea>`

Creates a text box area. Columns set the width;  
rows set the height.

`<input type="checkbox" name=? value=?>`

Creates a checkbox.

`<input type="checkbox" name=? value=? checked>`

Creates a checkbox which is pre-checked.

`<input type="radio" name=? value=?>`

Creates a radio button.

`<input type="radio" name=? value=? checked>`

Creates a radio button which is pre-checked.

`<input type="text" name=? size=?>`

Creates a one-line text area. Size sets length, in



characters.

`<input type="submit" value=?>`

Creates a submit button. Value sets the text in the submit button.

`<input type="image" name=? src=? border=? alt=?>`

Creates a submit button using an image.

`<input type="reset">`

Creates a reset button

#### **Table**

`<table> </table>`

Creates a table

`<tr> </tr>`

Sets off each row in a table

`<td> </td>`

Sets off each cell in a row

`<th> </th>`

Sets off the table header (a normal cell with bold, centered text)

`<table border=?>`

Sets the width of the border around table cells

`<table cellspacing=?>`

Sets amount of space between table cells

`<table cellpadding=?>`

Sets amount of space between a cell's border and its contents

`<table width=?>`

Sets width of the table in pixels or as a percentage

`<tr align=?>`

Sets alignment for cells within the row

(left/center/right)

`<td align=?>`

Sets alignment for cells (left/center/right)

`<tr valign=?>`

Sets vertical alignment for cells within the row

(top/middle/bottom)

`<td valign=?>`

Sets vertical alignment for cell (top/middle/bottom)

`<td rowspan=?>`

Sets number of rows a cell should span (default=1)

`<td colspan=?>`

Sets number of columns a cell should span

`<td nowrap>`

Prevents lines within a cell from being broken to fit

#### **HTML5 input tag attributi**

`<input type="email" name=?>`

Sets a single-line textbox for email addresses

`<input type="url" name=?>`

Sets a single-line textbox for URLs

`<input type="number" name=?>`

Sets a single-line textbox for a number

`<input type="range" name=?>`

Sets a single-line text box for a range of numbers

`<input type="date/month/week/time" name=?>`

Sets a single-line text box with a calendar

showing the date/month/week/time

`<input type="search" name=?>`

Sets a single-line text box for searching

`<input type="color" name=?>`

Sets a single-line text box for picking a color

