

ПРОГРАМИРАЊЕ ВЕБ АПЛИКАЦИЈА

Наставна јединица : Предавање 2

Теме: HTML, CSS



WEB PROGRAMIRANJE - PODELA JEZIKA

- ▶ Kako je web sajt je produkt programiranja potrebno je definisati programske jezike i njihovu podelu koji su neophodni za njegovu realizaciju.
- ▶ Prva podela:
 - ▶ Markup jezici
 - ▶ Skript jezici
 - ▶ Programski jezici

WEB PROGRAMIRANJE - PODELA JEZIKA

- ▶ Podel prema mestu izvršavanja, gde se konkretno realizuje ideja koju je programer osmislio:
 - ▶ klijentski
 - ▶ serverski
- ▶ **Klijentski** jezici : izvršavaju se na strani klijenta
 - ▶ Manje komplikovani i tehnički zahtevni, prostiji za upotrebu
 - ▶ Nezavisni od strukture hardvera, operativnog sistema, korisničkih privilegija
 - ▶ Za realizaciju neophodna grafička vizualizacija odnosno poseban softver – *browser*
 - ▶ **HTML, CSS, JavaScript, JQuery, Jscript....**

WEB PROGRAMIRANJE - PODELA JEZIKA

► **Serverski** jezici - izvršavaju se na serveru

- Kompleksnija sintaksa
- Složenija pravila za upotrebu
- Potreban web server za realizaciju
- Mogu da ostvare komunikaciju sa bazama podataka ili različitim tipovima tekstualnih ili drugih datoteka
- Najčešće se koriste za realizaciju određenih funkcionalnosti sajtova, različite vrste bezbedonosnih provera, generisanje dinamičkih sadržaja i slično.
- Ne mogu da prikažu rezultat svog rada bez upotrebe klijentskih jezika. Rezultat rada serverskih jezika se ugrađuje u klijentske i tek tada web sajt funkcioniše kao celina.
- Rezultat svakog serverskog jezika se na kraju mora predstaviti kao HTML i CSS kod.
- Korisnik nikada ne može da vidi na klijentskoj strani originalni serverski kod – vidi samo konačan rezultat.
- **PHP, ASP, Java, Python, Ruby, Perl, ASP.NET**

WEB MARKUP JEZICI

- ▶ *Markup languages (markap jezici) su kreirani sa ciljem da se standardizuje način kreiranja, opisa, obrade i prikaza teksta.*
- ▶ **Jezici označavanja**
- ▶ Unutar samog dokumenta se upisuju posebne sekvence karaktera poznate kao **tagovi** ili oznake kojim se **obeležavaju pojedini elementi** ili grupe elemenata radi prikazivanja ili identifikacije.
- ▶ Zasnovani na internacionalnom standardu ISO 8879 – **Standard Generalized Markup Language (SGML)**. Ovaj standard je nastao 1986, dakle pre pojave veba, sa idejom da se kreira jezik kojim će se definisati format u tekstualnim dokumentima.
- ▶ SGML dokument koristi posebnu datoteku koja se zove **Document Type Definition (DTD)**, koja definiše tagove koji se koriste za opis formatiranja teksta, attribute i entitete nekog markup jezika, kao i način na koji se oni zajedno koriste.

WEB MARKUP JEZICI

- ▶ Iz SGML-a razvila dva različita jezika: **HTML** i **XML**.
- ▶ SGML DTD je u početku definisao HTML na taj način da se HTML mogao koristiti samo za jednostavnu isporuku veb sadržaja. S jedne strane ovo je bila prednost, s obzirom da je na taj način HTML bio jednostavniji za korišćenje od SGML, ali s druge strane HTML je iz istog razloga ograničen sa svojom efikasnošću i fleksibilnosti.
- ▶ **HTML (HyperText Markup Language)** nastao je krajem 1990. godine kao naslednik SGML-a.
- ▶ Predstavlja vrlo striktnu formu SGML-a i ima tačno definisan broj tj. skup tagova koji su na raspolaganju programeru.
- ▶ Svaki tag ima striktnu sintaksu i definisan način delovanja na podatke.
- ▶ Većina tagova ima i grafičku interpretaciju, pa je HTML zamišljen sa ciljem da grafički uredi i struktura podatke

WEB MARKUP JEZICI

- ▶ XML (Extensible Markup Language) nastao je nakon 1996. godine a standardizovan tek 2008. godine (peto izdanje verzije 1.0). Za razliku od HTMLa, XML nije zamišljen da podržava grafički prikaz podataka.
- ▶ Njegov primarni cilj je da omogući opisivanje podataka, njihovo struktuiranje i da omogući proširivost u broju tagova.
- ▶ XML dokument može opisati samog sebe, organizaciono urediti veliki broj podataka (npr. kao baza podataka) i daje mogućnost da se definiše proizvoljno veliki broj tagova sa proizvoljnim imenima.
- ▶ Zajedničko i za HTML i XML je da su oni čisto tekstualni fajlovi, laki za učenje, čitanje i razumevanje a podjednako razumljivi i čoveku i mašini, što je i bio cilj markap jezika.

ISTORIЈАТ

VERZIЈА	GODINA
HTML	1991
HTML+	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML 1.0	2000
HTML 5	2012
XHTML 5	2013

HTML - *HYPER TEXT MARKUP LANGUAGE*

- ▶ *HTML je jezik koji ima tačno definisan skup tagova koja su nam na raspolaganju i pomoću njih se kreira prikaz svake veb stranice.*
- ▶ Dokumenti, veb strane, mogu da se sastoje od teksta, tabela, lista, slika i elemenata za unos podataka.
- ▶ Svaki dokument može da sadrži i meta informacije koje ukazuju na to šta sadrži dokument. Unutar dokumenta se uobičajeno nalaze hiperlinkovi koji pomoću URI-ja daju vezu ka drugim veb stranicama.

HTML - *HYPER TEXT MARKUP LANGUAGE*

- ▶ Dokument može da sadrži i ugnježdene skriptove.
- ▶ Za interpretaciju HTML dokumenta se koriste različiti ***korisnički agenti***.
- ▶ Platformski neutralan jezik
- ▶ Fajlovi imaju ekstenziju .html ili .htm, i nalaze se u određenom direktorijumu servera koji je povezan na Internet.

HTML 5

- ▶ HTML5 donosi i promenu logike prema skript jezicima i skladištenju podataka
- ▶ HTML5 je nova generacija HTML, a ne samo nova verzija
- ▶ Formalno, tek 2008. godine je objavljen prvi nacrt predloga HTML5 od strane W3C
- ▶ Nezvanično $\text{HTML5} \approx \text{HTML 5} + \text{CSS 3} + \text{JavaScript}$
- ▶ Na ovaj način HTML 5 je alat za:
 - Označavanje - Markup (HTML 5)
 - Prikazivanje - Presentation (CSS 3)
 - Interakciju - Interaction (DOM, Ajax, APIs)
- ▶ Od 2015. godine poslednji HTML standard

OSNOVNE KOMPONENTE HTML JEZIKA

- ▶ Osnovna komponenta veb dokumenta je **HTML elemenat**.
- ▶ HTML elemenat obuhvata **otvarajući, zatvarajući tag** i **sadržaj** koji se nalazi između njih.

HTML elemenat: <IME_TAGA> Sadržaj </IME_TAGA>

- ▶ **Tag** predstavlja **oznaku** koja govori čitaču kako da prikaže određeni sadržaj opisane veb stranice.
- ▶ Otvarajući tag: **<IME_TAGA>** - označava početni tag elementa
- ▶ Zatvarajući tag: **</IME_TAGA>** - označava krajnji tag elementa

PODELA TAGOVA

- ▶ Prosti tagovi se koriste za opisivanje jednostavnih elemenata jezika HTML:
<IME_TAGA>
- ▶ Složeni tagovi (upareni) se pišu u paru, gde zatvoreni tag predstavlja tag sa znakom / i predstavlja kraj naredbe:
<IME_TAGA> Sadržaj.... </IME_TAGA>
- ▶ Tagovi sa atributima:
< IME_TAGA A1=a A2=5 ...> Sadržaj... </ IME_TAGA > koji pružaju dodatne informacije za tekst koji se prikazuje.

DEKLARACIJA <!DOCTYPE>

- ▶ DOCTYPE deklaracijom se definiše tip dokumenta.
- ▶ Deklaracija je prva oznaka, koja se navodi pre <html> oznake.
- ▶ Deklaracija nije HTML tag, već predstavlja oznaku koja veb čitaču kaže u kojoj verziji je napisana data HTML stranica, kako bi čitač korektno prikazao datu stranicu
- ▶ Ceo fajl se tumači kao XML fajl i samim tim mora da ukazuje na DTD fajl, u kome je opisana struktura tog “proizvoljnog XML dokumenta”.
- ▶ Sve ovo je sa ciljem da se pokuša prevazilaženje različitih tumačenja browsera pojedinih tagova ili atributa.

- ▶ HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

- ▶ HTML 5

- ▶ <!DOCTYPE html>

STRUKTURA HTML STRANICE

```
<html>
```

```
  <head>
```

```
    <title>
```

ovde je naziv stranice

```
  </title>
```

```
</head>
```

```
<body>
```

ovde se unosi sve ono sto se prikazuje u okviru stranice

```
</body>
```

```
</html>
```

MINIMALNA STRUKTURA HTML DOKUMENTA OBUHVATA TAGOVE:

Početni	Završni	Opis taga
<code><html></code>	<code></html></code>	Tag koji definiše HTML stranu (početak i kraj)
<code><head></code>	<code></head></code>	Tag zaglavlja, sadrži meta-definicije HTML-dokumenta i informacije koje se ne pojavljuju eksplicitno u okviru dokumenta
<code><title></code>	<code></title></code>	Tag za naziv HTML-dokumenta (prikazuje se u browseru)
<code><body></code>	<code></body></code>	Tag koji definiše telo dokumenta (ono što se prikazuje korisniku)

<HTML>

- ▶ Tag <HTML> je okvir u kome se nalaze svi ostali tagovi;
- ▶ HTML dokument uvek počinje tagom <HTML>, a završava se tagom </HTML>

<HEAD>

- ▶ Tag <HEAD> uokviruje zaglavlje u kome se nalaze informacije o samom dokumentu (naslov, opis, ključne reči, ime autora, itd.) i gde se pišu svi viši programski script jezici (npr. Java Script)
- ▶ Upućuje čitač gde da nađe stilove za učitane stranice, pruža meta-informacije, itd.
- ▶ Sledeće oznake se mogu dodati u head sekciju:
 - ▶ <title>
 - ▶ <style>
 - ▶ <link>
 - ▶ <meta>
 - ▶ <script>
 - ▶ <noscript>
 - ▶ <base>

<BODY>

- ▶ Sve ono što vidimo u prozoru browser-a, tj. sadržina stranice, nalazi se u telu dokumenta koje uokviruje element <BODY>;
- ▶ U dokumentu sme da postoji samo jedan par tagova <BODY></BODY>

<STYLE>

- ▶ Tag <style> se koristi za definisanje stila.
- ▶ Njegovim korišćenjem utičemo na izgled pojedinih HTML elemenata na stranici.

Primer:

```
<style type="text/css">  
body { background-color: yellow; }  
p { color: blue; }  
</style>
```

<LINK>

- ▶ Tag <link> se koristi za definisanje veze (relacije) između HTML dokumenta i eksternog resursa.
- ▶ Najčešće se koristi za definisanje šablona za stilove (npr. CSS).

Primer:

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

<META>

- ▶ Metapodaci se ne prikazuju na stranici, ali su pogodni za parsiranje.
- ▶ Meta elementi se obično koriste za specifikaciju opisa stranice, ključne reči, informacije o autoru stranice, poslednju izmenu na stranici i druge metapodatke.
- ▶ Metapodatke mogu koristiti pretraživači na Internetu.

UPOTREBA <META> TAGA

- ▶ Definisanje kodnog rasporeda

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

- ▶ Definisanje ključnih reči za opis veb stranice:

```
<meta name="keywords" content="RT, NRT, AVT, Računarska tehnika, Nove računarske tehnologije">
```

- ▶ Definisanje opisa veb stranice

```
<meta name="description" content="Sajt Visoke škole elektrotehnike i računarstva strukovnih studija u Beogradu">
```

- ▶ Definisanje autora veb stranice

```
<meta name="author" content="Laza Lazarević">
```

- ▶ Osvežava dokumenat svakih 30s

```
<meta http-equiv="refresh" content="30">
```

<BASE>, <SCRIPT>

- ▶ Tag <base> definiše podrazumevanu adresu ili podrazumevano odredište za sve linkove na stranici.
- ▶ Tag <script> se koristi za skript jezike (npr. učitavanje JavaScript koda).

KOMENTAR U HTML KODU

- ▶ HTML omogućava pisanje komentara u okviru stranice.
- ▶ Tag koji se koristi je
<!-- za početak komentara i --> za kraj komentara.
 <!-- ovaj tekst je komentar
 koji se neće prikazati na stranici -->

ENTITETI

- ▶ Entiteti su specijalno definisane grupe karaktera, koje se kucaju u HTML kodu, i imaju unapred definisan način interpretacije u browser-u.
- ▶ Koriste se da bi se prikazali neki karakteristični simboli koji se regularno najčešće ne mogu drugačije prikazati ili uneti sa tastature.
- ▶ Obzirom da se ne prikazuju kao običan tekst, počinju simbolom & a završavaju se sa ; pa ih tako browser prepoznaje.

ENTITETI

space

< <

> >

€ €

© ©

® ®

& &

" "

' '

@ @

. .

& &

ć Ć

č č

đ đ

TIPOVI HTML ELEMENATA

- ▶ Većina HTML elemenata se može svrstati u jedan od sledeća dva tipa:
 - ▶ block element i inline element.
- ▶ Block elementi obično počinju i završavaju se novim redom. Neki od njih su: `<h1>`, `<p>`, `<ul>`, `<table>`.
- ▶ Inline elementi se prikazuju bez novih redova. Primeri inline elemenata su: `<b>`, `<td>`, `<a>`, `<img>`.

PRAZNI ELEMENTI

- ▶ Elementi bez sadržaja predstavljaju prazne elemente .
- ▶ `<br>`, `<hr>` su prazni elementi bez završnog taga.
- ▶ `<br>` tag definiše prelazak u novi red bez otvaranja novog paragrafa
- ▶ `<hr>` tag se koristi za prikaz horizontalne linije na veb stranici

HTML ATRIBUTI

- ▶ HTML elementi mogu imati attribute koji obezbeđuju dodatne informacije u vezi elemenata.
- ▶ Atributi se navode u početnom, otvarajućem, tagu.
- ▶ Navode se u paru ime/vrednost
 - ▶ npr. name= " value"
- ▶ HTML link je definisan <a> tagom. Adresa na koju pokazuje link se specificira u href atributu:
 - ▶ Ovo je link
- ▶ Vrednost atributa treba da bude uvek u okviru znaka navoda, mada su dozvoljeni i apostrofi.

HTML ATRIBUTI ZA SVE ELEMENTE

Atribut	Opis
class	Navodi jedno ili više ime klase za element (odnosi se na definisanje klase u stylesheet-u)
id	Predstavlja jedinstveni identifikator elementa
style	Definiše CSS stil za element
title	Predstavlja dodatne informacije o elementu

NASLOVI $\langle H_N \rangle$.. $\langle /H_N \rangle$

- ▶ U HTML-u naslovi (eng. headers) prikazuju se prema relativnom nivou ciframa od 1 do 6.
- ▶ Opšti oblik taga za naslov je:
 $\langle H_n \rangle$ Tekst za naslov nivoa n $\langle /H_n \rangle$ – gde n dobija vrednosti od 1 do 6

(Primer.html):

```
<h1> Naslov H1 </h1>
```

```
<h2> Naslov H2 </h2>
```

```
<h3> Naslov H3 </h3>
```


PARAGRAF <P></P>

- ▶ Tekst se deli na logičke celine. Osnovne celine u HTML jeziku su paragraf i odeljak
- ▶ Novi paragraf (pasus) zadaje se <p> tagom.
- ▶ Ovaj tag odvoji tekst sa razmakom između redova.
- ▶ Na kraju paragrafa se preporučuje da se umetne i njegov završni tag (ali nije neophodno).

<p> Ovo je paragraf.</p>

<p> Ovo je drugi paragraf

PARAGRAF - ATRIBUT ZA PORAVNANJE

► ALIGN je atribut i može imati tri vrednosti:

- levo poravnanje (left)
- desno poravnanje (right)
- centriranje (center)

```
<p align="left">
```

```
<p align="center">
```

```
<p align="right">
```

FONT

- ▶ Ako se drugačije ne navede, čitač koristi font familije Times New Roman
- ▶ HTML елемент:
`<font>Odgovarajući sadržaj</font>`
- ▶ Mogući atributi:
 - ▶ face
 - ▶ size
 - ▶ color
- ▶ Redosled navođenja atributa nije bitan!

ATRIBUT FACE ZA FONT

- Могуће је навести и неколико врста фонтова у алтернативи, један за другим, тако што их одвојимо зarezима.

```
<font face="Comic Sans MS, Arial, Courier New">
```

ATRIBUT SIZE ZA FONT

- ▶ U okviru stranice može se definisati i veličina slova.
- ▶ Za ove potrebe koristi se size atribut, sa vrednostima od 1 do 7.
- ▶ Podrazumevana veličina teksta je 3.

ATRIBUT COLOR ZA FONT

- ▶ Ako se želi dobiti sledeći tekst ispisan crvenom bojom:

```
<font color="#ff0000"> Primer texta u boji. </font>
```

ili

```
<font color=„yellow"> Primer texta u boji.</font>
```

Primer texta u boji.

STILIZOVANJE TEKSTA

- ▶ Stilizovanje teksta u formi Bold, Italic, Underline:
 - ▶ pisanje podebljanim slovima (bold ili strong)
`<b>Podebljan odnosno boldovan tekst</b>`
 - ▶ pisanje iskošenim slovima (italic)
`<i>Iskošen tekst</i>`
 - ▶ podvlačenje teksta (underline)
`<u>Podvučen tekst</u>`

STILIZOVANJE TEKSTA

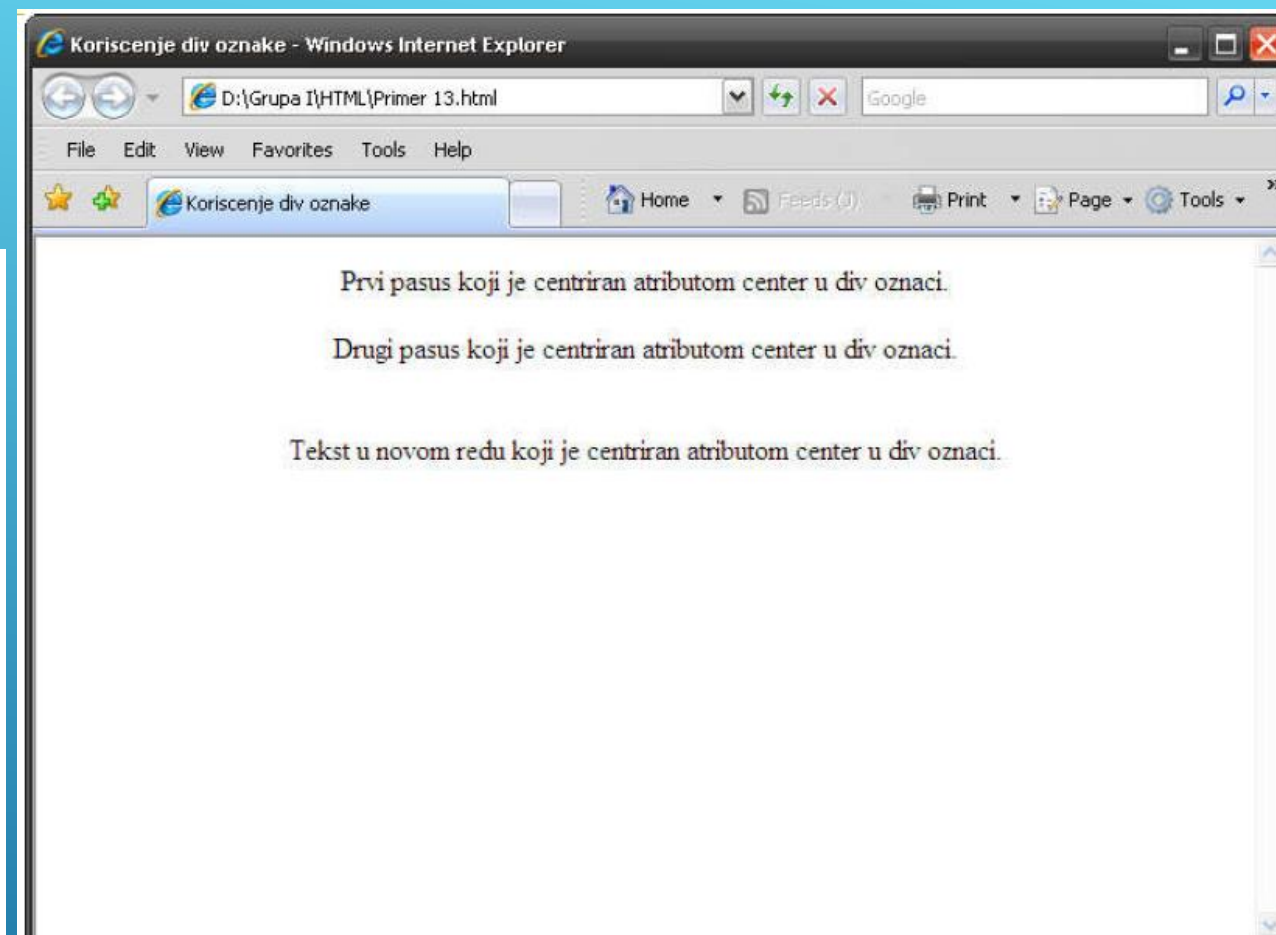
- ▶ Grupa HTML elemenata koji imaju istu grafičku reprezentaciju kao i elementi za stilizovanje ali su pogodne za programera da ukažu na neke detalje koje nisu vidljive korisniku.
- ▶ Elementi koji imaju istu grafičku reprezentaciju kao elementa definisan tagom <i>
 - ▶ naglašavanje dela teksta
`<em>Naglašen tekst</em>`
 - ▶ pisanje definicija i pojmova
`<def>Definicija ili pojam</def>`
 - ▶ navođenje kontakt podataka i adrese
`<address>Podvučen tekst</address>`
 - za citiranje
`<cite> Citat...</cite>`
 - za definisanje promenljivih
`<var>Definicija promenljivih</var>`
- Element koji ima istu grafičku reprezentaciju kao elementa definisan tagom
`<strong>Naglašen tekst</strong>`
- Element koji ima istu grafičku reprezentaciju kao elementa definisan tagom <u>
`<ins>Podvučen tekst</ins>`

ELEMENTI DIV I SPAN

- ▶ Elementi div i span nemaju nikakvu posebnu semantiku. Njihova svrha je da budu kontejneri za druge elemente i tekst.
- ▶ **HTML <div>** je elemenat na nivou **bloka** pa se u veb čitaču prikazuje prazna linija pre i posle njega.
- ▶ Uokviruje - ograđuje jedan blok elemenata.
- ▶ Koristan je za poravnavanje teksta i pridruživanje određenih svojstava određenom delu dokumenta, da se ne bi morale koristiti za svaki element bloka posebno.
- ▶ Zamenio je stari način definisanja rasporeda elemenata i organizacije veb stranice pomoću tabela.
- ▶ **HTML ** je **inline** (jednoredni) elemenat koji se koristi kao kontejner za tekst.
- ▶ Jedina razlika između elemenata div i span jeste ta što je div block element, a span inline element.
- ▶ Ovi elementi najviše se koriste u kombinaciji sa CSS-om za smeštanje elemenata koji treba da se stilizuju.

PRIMER

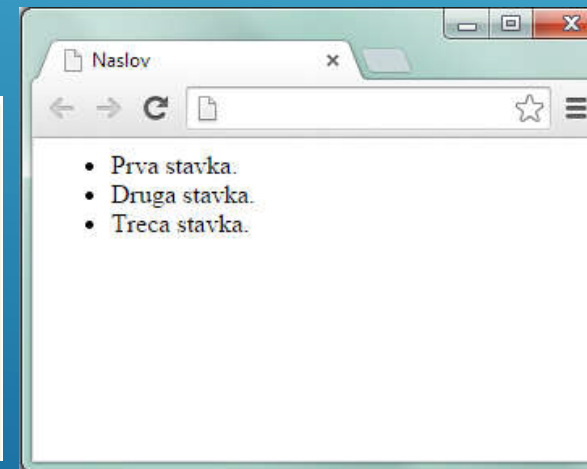
```
<html>
  <head>
    <title>Veličina slova u naslovu</title>
  </head>
  <body>
    <div align="center">
      <p> Prvi pasus koji je centriran atributom center u div oznaci. </p>
      <p> Drugi pasus koji je centriran atributom center u div oznaci. </p>
      <br> Tekst u novom redu koji je centriran atributom center u div oznaci.
    </div>
  </body>
</html>
```



HTML LISTE

- Postoje tri vrste lista koje su definisane u HTML-u: unordered, ordered i description liste.
- Unordered lista se definiše elementom `<ul></ul>` (unordered list).
- Element `<ul></ul>` se sastoji od `<li></li>` (list item) elemenata, pri čemu svaki li element predstavlja jednu stavku liste.

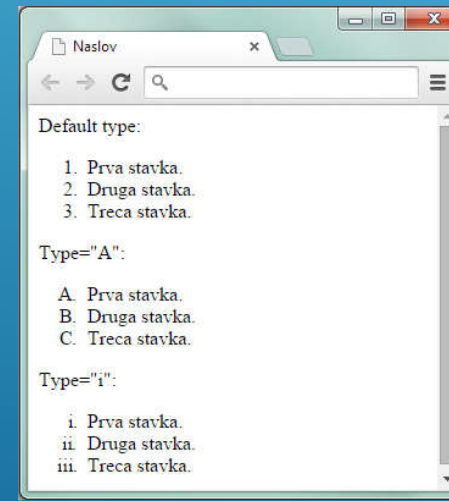
```
<ul>  
  <li>Prva stavka.</li>  
  <li>Druga stavka.</li>  
  <li>Treci stavka.</li>  
</ul>
```



HTML LISTE

- Ordered lista se definiše elementom `<ol>` (ordered list).
- Stavke u listi se definišu na isti način kao i u unordered listama – koriste se li elementi.
- Moguće je definisati tip liste uz pomoć type atributa. Vrednosti koje ovaj atribut može imati su: 1, A, a, I, i..

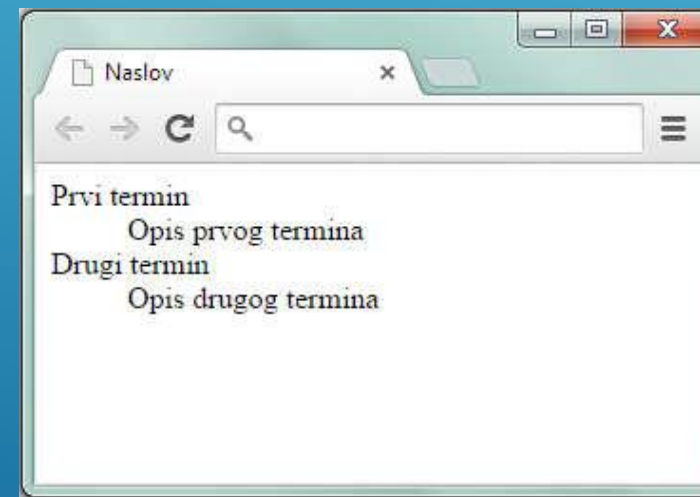
```
<p>Default type:</p>
<ol>
  <li>Prva stavka.</li>
  . . .
<p>Type="A":</p>
<ol type="A">
  <li>Prva stavka.</li>
  . . .
<p>Type="i":</p>
<ol type="i">
  <li>Prva stavka.</li>
  . . .
```



HTML LISTE

- Description lista se definiše elementom `<dl>`.
- Unutar elementa `<dl>` nalaze se dva tipa podelemenata:
 - Element `<dt>` – unutar njega se navodi termin.
 - Element `<dd>` – unutar njega se navodi opis ili definicija termina

```
<dl>  
  <dt>Prvi termin</dt>  
  <dd>Opis prvog termina</dd>  
  <dt>Drugi termin</dt>  
  <dd>Opis drugog termina</dd>  
</dl>
```



HTML FORME

- Forme se koriste za prikupljanje podataka od korisnika.
- Forma se definiše elementom `<form>`.
- HTML forma može da sadrži ulazne (input) elemente, koji su ujedno i najvažniji deo formi.
- Input elementi imaju razne varijacije, koje su definisane vrednošću atributa `type`.
- Najvažniji tipovi ulazanih elemenata forme definisanih sa vrednošću atributa `type` su:
 - `text` – definiše regularni tekstualni input.
 - `radio` – definiše radio button-e.
 - `submit` – definiše submit dugme kojim se podaci iz forme šalju na obradu.
- U okviru forme mogu biti i elementi: padajuće liste, `textarea`, `fieldset`, `label`.

HTML FORME – INPUT ELEMENTI

- **Tekstualni input element** podrazumeva prazno polje za jednu liniju teksta.
- Definiše se na sledeći način:
`<input type="text" name="firstname">`
- Širina text input-a se definiše u broju karaktera i to unutar atributa size.
- Podrazumevana dužina tekstualnog polja je 20 karaktera.
- **Polje za lozinku** definiše se na sledeći način:
`<input type="password" name="pwd">`

HTML FORME – INPUT ELEMENTI

- **Radio input (radio button) element**– omogućava korisniku da izabere tačno jednu od više ponuđenih opcija koje pripadaju jednoj grupi.
- Radio button se definiše navođenjem input elementa sa **atributom type= „radio“**.
- Unutar radio button-a potrebno je definisati i **value atribut**. Vrednost ovog atributa slaće se zajedno sa ostalim podacima forme ukoliko je baš taj radio button selektovan.
- Atribut checked, koji ne mora imati vrednost, se navodi u slučaju da određeni radio button pri inicijalizaciji stranice treba biti selektovan.

```
<input type=„radio“ name=„pol“ value=„zenski“ checked> Ženski  
<input type=„radio“ name=„pol“ value=„muski“> Muški
```


HTML FORME – INPUT ELEMENTI

- **Polje za potvrdu (checkbox)** – omogućava korisniku da izabere više od jedne ponuđene opcije koje pripadaju jednoj grupi.
- Definiše se navođenjem input elementa sa atributom **type=„checkbox“**.

```
<form>  
<input type="checkbox" name="vehicle"  
value="Bike">I have a bike<br>  
<input type="checkbox" name="vehicle"  
value="Car">I have a car  
</form>
```

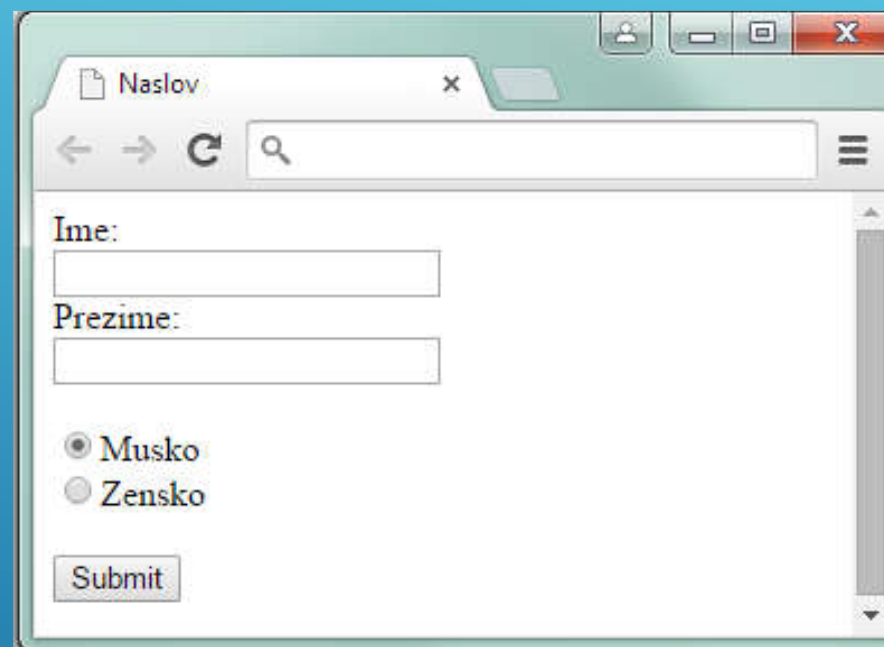
HTML FORME – SUBMIT

- **<input type=„submit“ value=„Posalji“>** definiše dugme čijim se klikom šalju svi podaci iz forme na obradu.
- Pri definiciji ove vrste input-a, atribut type treba da ima vrednost submit.
- Atributom value se definiše tekst koji će se ispisati na samom dugmetu.
- **Podaci se šalju stranici** koja je navedena u **action** atributu forme.

```
<form name="input" action="html_form_action.php"
method="get">
Username: <input type="text" name="user">
<input type="submit" value="Submit">
</form>
```

HTML FORME – PRIMER

```
<form action="action_page.php">  
Ime:<br>  
<input type="text" name="ime">  
<br>  
Prezime:<br>  
<input type="text" name="prezime">  
<br><br>  
<input type="radio" name="pol"  
value="musko" checked>Musko  
<br>  
<input type="radio" name="pol"  
value="zensko">Zensko  
<input type="submit" value="Submit">  
</form>
```



The screenshot shows a web browser window with a single tab titled 'Naslov'. The browser's address bar is empty. The rendered form contains two text input fields labeled 'Ime:' and 'Prezime:'. Below these fields are two radio buttons; the first is labeled 'Musko' and is selected (indicated by a filled circle), while the second is labeled 'Zensko' and is not selected (indicated by an empty circle). At the bottom of the form is a 'Submit' button.

HTML TAGOVI ZA FORME

Tag	Description
<form>	Definiše HTML formu
<input>	Definiše ulazni (input) element
<textarea>	Definiše višelinijski input element (text area)
<label>	Definiše labelu za <input> element
<fieldset>	Grupiše elemente u formi
<legend>	Definiše naslov za <fieldset> element
<select>	Definiše padajuću (drop-down) listu
<optgroup>	Definiše grupu opcija u padajućoj listi
<option>	Definiše opciju (stavku) u padajućoj listi

Tag	Opis
<button>	Definiše dugme (koje reaguje na akciju)
<datalist> (HTML5)	Specifies a list of pre-defined options for input controls
<keygen> (HTML5)	Defines a key-pair generator field (for forms)
<output> (HTML5)	Defines the result of a calculation

HTML LINKOVI

- Linkovi omogućavaju povezivanje sadržaja na unutar web sajtu kao i van njega.
- **Tag <a> definiše hiperlink.**
- Hiperlink (link) predstavlja reč, grupu reči ili neki drugi objekat koji omogućava prelazak na drugi dokument.
- **Najvažniji atribut <a> taga je href** u koji se upisuje odredište na koji link vodi.
- Podrazumevano, linkovi se u veb čitačima pojavljuju kao:
 - Neposećeni linkovi
 - Posećeni linkovi
 - Aktivni linkovi

SINTAKSA HTML LINK ELEMENTA

- `<a href=„Navodi se URL“>Objekat, tekst koji se linkuje</a>`
- Atribut **target** se navodi u okviru početnog `<a>` taga za specificiranje gde i kako će biti otvoren linkovani dokumenat.
- `<a href=„http://www.example.com“ target=„_blank“>Otvara se u novom listu odnosno tabu ili novom prozoru veb čitača</a>`

APSOLUTNA PUTANJA

- Kada se kreira veza ka veb stranici koju nije autor napravio, specificira se aposlutna adresa - putanja URL (engl. *Uniform Resource Locator*) te stranice.
 - **protokol://domen/putanja do datoteke**
- Protokoli: http, https, ftp..
- Domen: www.viser.edu.rs
- Putanja uključuje direktorijum i informacije o traženoj datoteci
 - npr. https://www.viser.edu.rs/uploads/2019/10/RT5p2019.pdf

APSOLUTNA PUTANJA

- Pored osnovne namene, **URL putanja se može koristiti za prenos podataka od strane korisnika ka web serveru.**
- Početni deo URL- a ostaje isti ali se proširenje realizuje iza imena stranice kojoj se pristupa.
- Za odvojanje imena stranice i podataka koristi se simbol ? Iza koga se definiše standardizovani paket informacija koje se prenose u formatu:
ime_podatak1=sadrzaj_podatak1
- Ako postoji **više podataka povezuju se znakom &**

APSOLUTNA PUTANJA

- URL адреса za prenos podataka o imenu i prezimenu korisnika formata:
 - <http://primer.com/index.php?ime=mika&prezime=mikic>
- Na ovaj način se podaci popunjeni od strane korisnika na klijentskoj strani prenose do udaljene stranice na veb serveru koja je definisana URL adresom i gde se podaci obrađuju.
- Alternativni način prenosa podataka putem URL adrese podrazumeva drugačiju sintaksu korišćenjem MVC (engl. *Model-view-controller*) arhitekture.
 - <http://primer.com/obrada/mika/mikic>

RELATIVNA PUTANJA

- Relativna putanja se koristi za referenciranje datoteka u okviru veb sajta.
- **Link ka datoteci koja se nalazi u istom direktorijumu** kao i stranica sa koje se linkuje:
`<a href=„footbar.html“> Footbar stranica </a>`
- **Link ka datoteci koja se nalazi u poddirektorijumu** u odnosu na stranicu sa koje se linkuje.
Stranica footbar.html se nalazi u direktorijumu footbar koji je poddirektorijum www direktorijuma
`<a href=„footbar/footbar.html“> Footbar stranica </a>`
- **Link ka stranici koja se nalazi u naddirektorijumu** u odnosu na stranicu sa koje se linkuje – skok u taj direktorijum:
`<a href=„../index.html“> Početna</a>`

HTML BOJE

- Boje koje se prikazuju na HTML stranici zasnovane su na RGB modelu (Red, Green, Blue)
- Najmanja moguća vrednost je 0 (hex 00) a najveća je 255 (hex FF)
- Sa RGB modelom dobija se 16 miliona različitih boja.
- U HTML i CSS specifikacija definisano 140 boja (17 standardnih)
- Standardnih 17 boja:
aqua, black, blue, fuchsia, gray,
green, lime, maroon, navy,
olive, orange, purple, red, silver,
teal, white, yellow

Boja	BojaHEX	Boja RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#CoCoCo	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

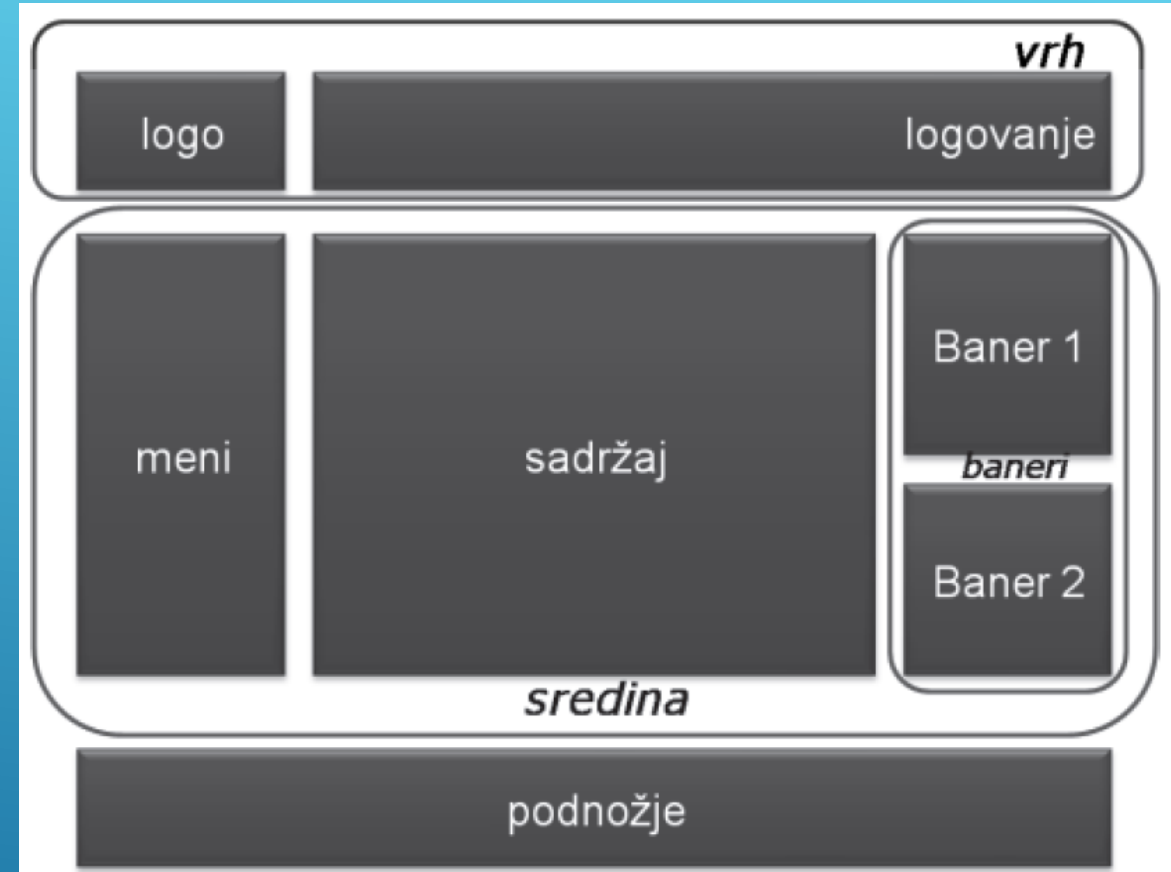
- **Slike se definišu neuparenim tagom .**
- Ključni atribut ovog taga je **src** u okviru koga se definiše putanja i naziv slike koja se prikazuje.
- Putanja do slike može biti relativna ili apsolutna.
- **Veličina** slike definisana **atributima height i width** može biti apsolutna (u piskelima) ili releativna (u procentima u odnosu na originalnu veličinu slike).

```
<img src=„pitanja do slike“ height=„150px“ width=„200px“  
alt=„alterantivni tekst“ align=„center“>
```

CSS

- **Osnovne prednosti CSS-a:**
 - Razdvajanje glavnog sadržaja stranice od koda namenjenog za dizajn.
 - CSS kod se može iskoristiti nad više HTML stranica, čime se izbegava ponovno kucanje istog koda.
- Od HTML 4 - omogućena organizacija sadržaja veb dokumenta primenom CSS mehanizma.
- Neophodno prvo kreirati odgovarajuće HTML elemente kojim se realizuje struktura dokumenta a potom izvršiti stilizovanje (formatiranje) primenom CSS metoda.
- **<div> i ** upareni tagovi se koriste za organizaciju i raspored sadržaja na veb stranici .

- Kada je kreirana struktura stranice potrebno je napisati odgovarajući HTML kod za realizaciju.
- Tokom pisanja HTML koda potrebno obezbediti kasnije napisanom CSS mogućnost primene i implementacije na odgovarajuće HTML elemente.
- Ovo se realizuje preko atributa id i class tagova div i span.



GDE PISATI CSS KOD?

- Postoje tri načina na koja se CSS ubacuje u HTML stranicu.
- **CSS kod se može definisati:**
 - Unutar style atributa određenog HTML elementa.
 - Unutar style elementa.
 - U eksternom fajlu koji se onda referencira iz HTML dokumenta.
- Ponekad se može dogoditi da se na navedena tri mesta iste stvari definišu na različit način.
- Zbog toga je definisan prioritet nad ova tri načina, koji odgovara redosledu u kojem su načini iznad navedeni.
- Primer: ukoliko je boja slova u style atributu setovana na plavo, a u style elementu na crveno, browser će navedeni tekst obojiti plavom bojom zbog gore navedenog prioriteta.

INLINE CSS

- CSS kod koji se nalazi unutar style atributa.
- Ovo je najlošiji pristup pisanju CSS koda, s obzirom da se na ovaj način sve navedene pogodnosti CSS-a gube.
- Ovaj pristup koristiti samo kada ne postoji drugo rešenje.

```
<p style="color:blue;margin-left:30px;">Paragraf.</p>
```


INTERNAL CSS

- CSS kod se nalazi u okviru style elementa.
- Element style se navodi unutar head elementa HTML dokumenta.
- CSS kod koji je tu definisan se primenjuje na ceo HTML dokument.
- Ovaj pristup je nešto bolji od prethodnog (inline CSS-a), budući da je CSS kod grupisan na jednom mestu – u zaglavlju dokumenta.
- Ovim pristupom preglednost koda je bolja, ali višestruko korišćenje istog koda i dalje nije moguće.

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body {
      background-color: linen;
    }
    h1 {
      color: maroon;
      margin-left: 40px;
    }
  </style>
</head>
<body>
  . . .
</body>
</html>
```

EXTERNAL CSS

- CSS kod se definiše u eksternom fajlu, a potom se taj fajl referencira iz HTML-a.
- Ovaj pristup je najbolji i treba ga koristiti kad god je to moguće.
- Na ovaj način se CSS u potpunosti izdvaja iz HTML-a, čime se dobija pregledan kod.
- Tako izdvojen kod se može referencirati od strane više HTML dokumenata, te se time izbegava dupliranje istog CSS koda.

EXTERNAL CSS

- Referenciranje eksternog CSS fajla se vrši korišćenjem link HTML elementa koji u generalnom slučaju služi za linkovanje eksternih resursa.
- Link se navodi unutar head elementa.
- Unutar link elementa potrebno je navesti nekoliko **atributa: rel, type i href.** (Postoji još atributa koji se mogu navesti unutar link elementa, a ovde su navedeni najbitniji.)

EXTERNAL CSS

- Atributom **rel** se definiše relacija između HTML dokumenta i linkovanog resursa, tj. definiše se šta linkovani resurs HTML dokumentu predstavlja. U slučaju CSS-a vrednost ovog atributa je **stylesheet**.
- Atributom **type** se definiše tip linkovanog resursa. U slučaju CSS-a navodi se vrednost **text/css**.
- Atributom **href** se definiše **url do linkovanog resursa**. (Uglavnom je to relativna putanja u odnosu na HTML dokument.)

```
<head>  
<link rel="stylesheet" type="text/css" href="style.css">  
</head>
```

SINTAKSA CSS

- CSS kod se sastoji od **CSS pravila**.
- Svako pravilo sačinjeno je od **selektora** i **deklaracionog bloka**.
- **Selektorom** se definiše koje **HTML elemente** treba stilizovati.
- **Deklaracionim blokom** se definiše kako **HTML elemente** treba stilizovati.

body	<pre>{ background-color: #d0e4fe; }</pre>	
.orangetext	<pre>{ color: orange; text-align: center; }</pre>	SELEKTORI
#title	<pre>{ font-family: "Times New Roman"; font-size: 20px; }</pre>	DEKLARACIONI BLOKOVI

SELEKTORI

- Svrha im je da **selektuju HTML elemente koji treba da se stilizuju.**
- Postoji više načina selektovanja elemenata:
 - **Korišćenjem CSS klasa**
 - **Korišćenjem ID-a elemenata**
 - **Navođenjem imena elementa (npr. p, body, ...)**
 - **Korišćenjem pseudo klasa**
 - **Korišćenjem selektora * koji selektuje sve elemente**

SELEKCIJA POMOĆU CSS KLASA

- Class selektor **selektuje** sve elemente koji za vrednost **class atributa** imaju baš onu klasu koja je navedena unutar selektora.
- Selektor se navodi tako što se ispred imena klase stavlja tačka.

```
.bluetext {  
    text-align: center;  
    color: blue;  
}
```

```
<p class="bluetext">Tekst na  
koji će stil iz klase "bluetext" biti  
primenjen.</p>
```

SELEKCIJA POMOĆU ID ELEMENTA

- ID selektor **selektuje** element čiji je **ID jednak vrednosti koja je navedena u selektoru**.
- ID elementa se navodi u istoimenom atributu.
- ID vrednost treba da bude jedinstvena unutar HTML dokumenta.
- ID selektor se može koristiti samo za stilizovanje jednog HTML elementa po dokumentu.

```
#text {  
    text-align: center;  
    color: blue;  
}
```

```
<p id="text">Tekst na koji će stil  
biti primenjen.</p>
```


SELEKCIJA POMOĆU IMENA ELEMENTA

- Elementi se mogu selektovati i navodeći **ime elementa**.
- U tom slučaju se selektuju svi elementi tog imena koji se nalaze na datoj HTML stranici.

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

```
<p>Tekst na koji će stil biti  
primenjen.</p>
```

SELEKCIJA SVIH ELEMENATA

- Navođenjem zvezde (*) na mesto selektora stilizuju se svi HTML elementi.

```
* {  
    text-align: center;  
    color: blue;  
}
```

SELEKCIJA POMOĆU PSEUDO KLASA

- **Pseudo klase** su predefinisane CSS klase koje se u okviru selektora navode tako što njihovom imenu prethodi dvotačka (**:imeklase**).
- One definišu posebno stanje određenog elementa. Primer: klasa **hover** selektuje dati element samo kada se strelica miša nalazi iznad njega.
- Primeri pseudo klasa: **link** (neposećen link), **visited** (posećen link), **hover** (miš je iznad elementa), **active** (selektovan link), **first-child** (element je prvo dete nekog roditeljskog elementa), itd.

KOMBINOVANJE RAZLIČITIH TIPOVA SELEKTORA

- Ukoliko se različiti tipovi selektora navedu sukcesivno **bez razmaka** unutar istog selektora, tada je rezultat selekcije jednak **preseku rezultata** svakog navedenog tipa selektora pojedinačno.
- Primer: **p.imeklase#idelementa:hover** – u ovom slučaju će se selektovati **p** element čiji **class** atribut ima vrednost “**imeklase**”, koji ima **id** “**idelementa**” i preko kojeg trenutno stoji strelica miša.

KOMBINOVANJE RAZLIČITIH TIPOVA SELEKTORA

- Ukoliko se različiti tipovi selektora navedu sukcesivno sa razmakom unutar istog selektora, tada se selekcija vrši postupno.
- Prvo se izvrši selekcija sa prvim navedenim selektorom na nivou celog dokumenta.
- Potom se vrši selekcija sa drugim selektorom, ali ne na nivou celog dokumenta već samo nad child elementima elemenata koji su selektovani u prethodnoj fazi. Ovaj postupak se ponavlja dok se svi navedeni selektori ne iskoriste.
- Primer: **p .imeklase #idelementa** - u ovom slučaju će se selektovati element koji ima id "idelementa", a koji se nalazi unutar elementa klase "imeklase", koji se opet nalazi unutar nekog p elementa.

KOMBINOVANJE RAZLIČITIH TIPOVA SELEKTORA

- Ukoliko se različiti selektori navedu tako da su **razdvojeni zarezom**, tada je rezultat selekcije jednak **uniji rezultata** svakog navedenog selektora pojedinačno.
- Primer: **p, .imeklase, #idelementa** – u ovom slučaju biće selektovani svi **p elementi**, svi elementi koji imaju klasu “imeklase”, kao i element koji ima id “idelementa”.

CSS MERNE JEDINICE

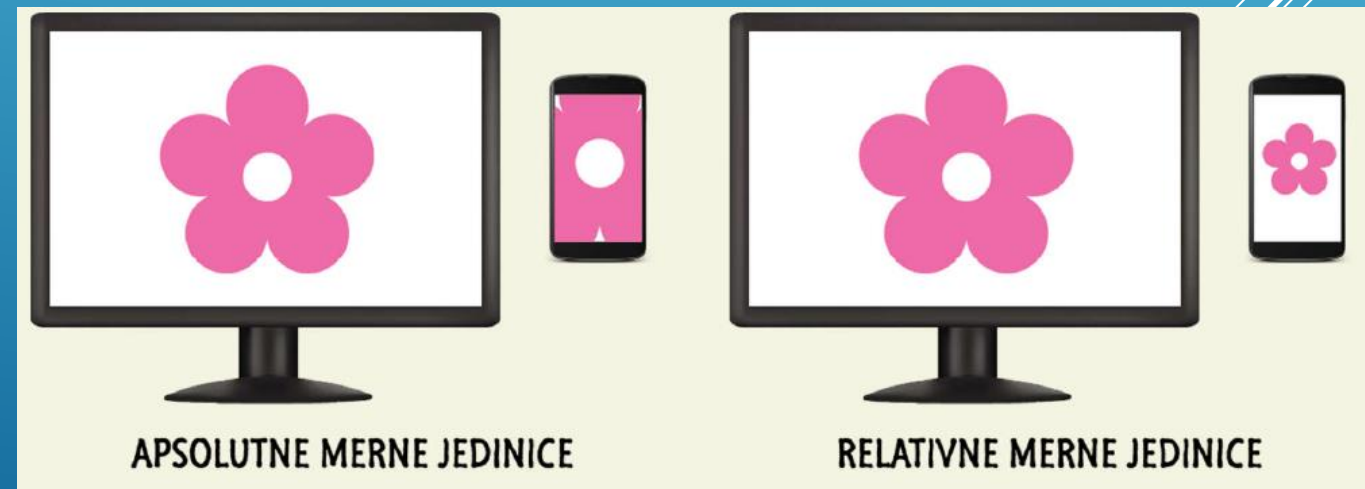
- Pri formatiranju web stranice često je potrebno definisati veličinu elementa, udaljenost između elemenata, veličinu margine, itd.
- CSS-om je definisano nekoliko mernih jedinica koje se koriste pri definisanju gore navedenih stvari:

em	Mera relativna u odnosu na veličinu fonta datog elementa. (2 em = 2 puta veće od veličine fonta)
ex	Mera relativna u odnosu na visinu malog slova "x" datog elementa.
ch	Mera relativna u odnosu na širinu karaktera "0" (nula).
rem	Mera relativna u odnosu na veličinu fonta root elementa HTML stranice.
vw	Mera relativna u odnosu na 1% širine viewport-a.
vh	Mera relativna u odnosu na 1% visine viewport-a.
vmin	Mera relativna u odnosu na 1% manje stranice viewport-a.
vmax	Mera relativna u odnosu na 1% veće stranice viewport-a.
%	Značenje procenta zavisi od konteksta u kom je naveden.
px	Pikseli.

CSS MERNE JEDINICE

- Postoje i apsolutne merne jedinice koje ne zavise od veličine ekrana – na svakom ekranu će njihova fizička veličina biti ista.
- Ove merne jedinice treba koristiti samo u slučaju kada se unapred znaju dimenzije medijuma na kom će biti prikazan dokument.
- Primer: apsolutne merne jedinice se mogu koristiti ukoliko je HTML dokument namenjen za štampu. Tada se dimenzije papira unapred znaju, te je pogodno koristiti apsolutne jedinice.

cm	Centimetar
mm	Milimetar
in	Inč (1 in = 2,54 cm)
pt	Point (1 pt = 1/72 in)
pc	Picas (1 pc = 12 pt)



BOX MODEL

- Pri raspoređivanju HTML elemenata i određivanju njihovih veličina, neophodno je biti upoznat sa **CSS box modelom**.
- Box model definiše “omotač” oko svakog HTML elementa koji se sastoji iz sledećih stvari: **paddinga, bordera i margine**.

BOX MODEL

- **Content** predstavlja sadržaj samog HTML elementa.
- **Paddingom** se definiše razmak između contenta i bordera. Padding nije vidljiv – predstavlja običan prazan prostor.
- **Borderom** se definiše linija koja ide oko contenta i paddinga.
- **Margina** služi za definisanje razmaka između elementa i njegove okoline. Kao i padding, ni margina nije vidljiva.

