

# Uvod u veb i internet tehnologije





# Stilovi i CSS





# Stilski listovi

- Vizuelna prezentacija HTML dokumenata podešava se korišćenjem **stilskih listova** (stylesheets) opisanih u jeziku CSS (Cascading Style Sheets)
- Prva verzija objavljena 1996. godine
- Aktuelna verzija je CSS3





# Opšta sintaksa stilskih listova

- Mogu se zadavati u zaglavlju HTML dokumenata, u okviru elementa `style` ili u posebnim CSS dokumentima
- Stilski list se sastoji od niza pravila

- Primer:

```
p { color: red; }  
h1 { font-family: Arial; margin: 20px; }
```

- Beline nemaju uticaja; stilski list se često nazubljuje radi preglednosti

- Primer:

```
h1 {  
    font-family: Arial;  
    margin: 20px;  
}
```



## Opšta sintaksa stilskih listova (2)

- Stilski list se sastoji od pravila
  - Svako **pravilo** je oblika:  
`selektor opis`
  - Svaki **opis** je oblika:  
`{niz deklaracija međusobno razdvojenih znakom ';' }`
  - Svaka **deklaracija** je oblika:  
`svojstvo: vrednost`
- Pojedinačni opis se pomoću atributa **style** može dodeliti ma kom elementu, bez korišćenja stilskih listova



## Opšta sintaksa stilskih listova (3)

- Selektori mogu biti složeniji nego samo navođenje imena elemenata
- Više selektora može se navesti zajedno (u tom slučaju, selektori se razdvajaju zapetama)

```
h1, h2, h3 { color: blue; }
```

- Komentari se navode između simbola `/*` i `*/`

- Primer:

```
p { /* podesavamo sve pasuse */  
    color: red; /* crvena boja teksta */  
    margin: 10px; /* margina od 10 piksela */  
}
```





# Uključivanje stilskih opisa u HTML dokumente

Tri načina za uključivanje CSS opisa u HTML dokument:

## 1. Opisi na nivou elementa (atribut **style**)

- pojedinačnom elementu može se promeniti stil navođenjem atributa **style**, čija je vrednost niz CSS svojstava i njihovih vrednosti
- opis vizuelne prezentacije je isprepleten sa opisom njene strukture

```
<p style="color:red; margin-left:10px;">Ovo je pasus</p>
```



# Uključivanje stilskih opisa u HTML dokumente (2)

Tri načina za uključivanje CSS opisa u HTML dokument:

## 2. Opisi na nivou dokumenta (atribut **style**)

- CSS opis se može navesti u zaglavlju dokumenta, kao sadržaj elementa **style**

```
<head>
```

```
...
```

```
<style type="text/css">
```

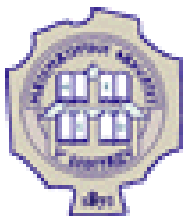
```
  p { color : blue; }
```

```
</style>
```

```
...
```

```
</head>
```





# Uključivanje stilskih opisa u HTML dokumente (3)

Tri načina za uključivanje CSS opisa u HTML dokument:

## 3. Spoljašnji opisi

- Koriste se za stilizaciju većeg broja veb strana na isti način
- Zapisuju se u vidu tekstualne datoteke sa ekstenzijom **.css**
- Pojednostavljuje izmenu vizuelne prezentacije celog veb sajta
- Uključuje se korišćenjem elementa **link** u zaglavlju dokumenta, navođenjem atributa **rel** sa vrednošću **stylesheet**

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

- Jezik CSS dozvoljava uvoz nekog drugog stilskog lista u dati stilski list, korišćenjem direktive **@import**

```
<style type="text/css"> @import url("stil.css"); </style>
```



# Nasleđivanje stilskih listova

- Za neka svojstva važi da, ako elementu pridružimo neku deklaraciju stila, nju automatski nasleđuju svi elementi sadržani u tom elementu

```
body { color : red; }
```

- Nasleđeno svojstvo se može promeniti

```
p { color : blue; }
```

- Neka svojstva se ne nasleđuju

```
body { margin : 20px; }
```



# Kaskada stilskih opisa

- Stilske opise moguće je istovremeno navesti na nekoliko različitih mesta.
- Razlikujemo:
  1. stilske listove autora veb strane (date na neki od prethodna tri načina)
  2. stilske listove korisnika (zadaju se podešavanjima veb pregledača)
  3. podrazumevane stilske listove veb pregledača
- Različiti opisi za neki element se kumulativno obrađuju; ako dođe do konflikta, prednost se daje u opadajućem prioritetu gore navedenih načina zadavanja stilskih listova
- Ako do konflikta dođe na nivou stilskih opisa autora, najveći prioritet imaju pravila navedena na nivou elementa, zatim na nivou dokumenta, a najmanji pravila u spoljašnjim listovima



# Selektori

- Najjednostavniji selektor je **naziv elementa**  
`p { color : blue; }`
- Kao selektor se može koristiti **jedinstveni identifikator elementa**: elementu pridružujemo identifikator korišćenjem atributa **id**, a zatim ga koristimo kao selektor oblika **#id**
- Primer:  
ako smo definisali pasus:  
`<p id="opis">U ovom pasusu biće opisana glavna svojstva...</p>`  
tada pravilo:  
`p#opis { color : blue; }`  
u ovom kontekstu ima isti efekat kao:  
`#opis { color : blue; }`



## Selektori (2)

- Kao selektor može se koristiti i **naziv klase**: svakom elementu koji želimo na isti način da stilizujemo dodeljuje se klasa korišćenjem atributa **class**, a zatim se ta klasa koristi kao selektor oblika **.class**

Primer:

ako segment HTML dokumenta ima sledeći oblik:

```
<p class="rezime">Jezik HTML je...</p>
```

...

```
<p class="rezime">Jezik CSS je...</p>
```

tada pravilo:

```
p.rezime { color : blue; border: 1px solid black; }
```

ne mora da ima isti efekat kao pravilo:

```
.rezime { color : blue; border: 1px solid black; }
```



# Pseudoklase i pseudoelementi

- Pseudoklase i pseudoelementi služe za finija podešavanja u jeziku CSS
- Unapred su definisani i nipošto ih ne treba dodeljivati elementima u sklopu HTML opisa
- **Pseudoklase** služe za stilizovanje elemenata dok su u posebnom stanju, dok **pseudoelementi** služe za stilizaciju dela nekog elementa
- Pseudoklase se označavaju pomoću dvotačke (npr. **:hover**), a pseudoelementi pomoću dvostruke dvotačke (npr. **::first-line**)



# Pseudoklase i pseudoelementi (2)

- Najčešće pseudoklase i pseudoelementi:

- **:link**

- **:visited**

- **:hover**

- **:first-child**

- **::first-line**

- **::first-letter**

- Primer:

```
:hover { color : green; }
```





# Ugnježdženi elementi

- Ugnježdženi elementi se koriste kada je potrebno promeniti stil samo onih elemenata koji su obuhvaćeni nekim širim elementom
- Sintaksa: **selector1 selector2** - ovim se označavaju samo elementi opisani selektorom selector2 koji se nalaze u okviru selektora selector1
- Primeri:

```
main p { color: blue; }
```

```
div#gallery img.small { width: 50px; }
```



# Fontovi

- Font predstavlja oblik znakova
- Naziv (familija) fonta se zadaje svojstvom font-family; može se navesti:
  - tačan naziv fonta: imena fontova od više reči navode se pod navodnicima (npr. "Times New Roman")
  - ime familije fontova (npr. Times)
  - ime vrste fontova (npr. serif)
- Razlikujemo:
  - serifne fontove (serif) koji na ivicama znakova imaju neke detalje
  - neserifne fontove (sans-serif) koji to nemaju
  - neproporcionalne fontove (monospace) kod kojih su sva slova iste širine



## Fontovi (2)

- Moguće je navesti više opisa u opadajućem prioritetu

```
p { font-family: "New Century Schoolbook", Times, serif }
```
- Veličina fonta ( $1\text{px}=1/96$  deo inča,  $1\text{pt}=1/72$  deo inča) se zadaje svojstvom font-size

```
p { font-size: 12pt; }  
p { font-size: 120%; }  
p { font-size: 1.2em; }
```

- Varijante fonta
  - font-style određuje iskošenost znakova; moguće vrednosti: normal, italic, oblique
  - font-weight određuje debljinu slova; najčešće vrednosti: normal, bold, lighter
- Dozvoljeno je više karakteristika fonta dati jednom deklaracijom

```
p { font: italic bold 12pt Times, serif; }
```



# Stilizovanje teksta

- U CSS-u se (za razliku od procesora teksta) podešavanje teksta razlikuje od podešavanja fonta
- U podešavanje (stilizovanje) teksta spadaju: podvlačenje teksta, precrtavanje, uvlačenje prve linije, poravnanje teksta, podešavanje razmaka između reči i slova, itd.
- U CSS-u se svojstvima fonta određuje izbor slike (glyph) za ispis znaka, a svojstvima teksta se definiše kako se oni raspoređuju i da li je potrebno još nešto do crtati



# Stilizovanje teksta – dekoracija

- Dodatno ukrašavanje teksta zadaje se svojstvom `text-decoration`; moguće vrednosti su `none`, `underline`, `overline`, `line-through`
- Primer:

```
a { text-decoration: none; }  
a:hover { text-decoration: underline; }
```

text-decoration: none

text-decoration: overline

text-decoration: underline

~~text-decoration: line-through~~



# Stilizovanje teksta – razmaci

- U CSS-u moguće je fino podešavanje horizontalnog razmaka između znakova
- Za podešavanje razmaka između znakova u jednoj reči koristi se svojstvo **letter-spacing**, a za podešavanje razmaka između susednih reči svojstvo **word-spacing**; vrednosti za ova svojstva se zadaju u pt, px ili em

```
letter-spacing: 5px
word-spacing: 25px
```

- Vertikalni razmak nazivamo prored i on se može podešavati korišćenjem svojstva **line-height**; zadaje se kao decimalni broj ili kao procenat

```
p { line-height: 1.5; }
```

U ovom primeru se 'line-height: 200%' koristi da bi se povećao prored u pasusu.



# Stilizovanje teksta – uvlačenje i poravnanje

- Uvlačenje prve linije teksta zadaje se svojstvom **text-indent**; vrednost se zadaje u px, pt ili u procentima, odnosno u em

```
p { text-indent: 3em; }
```

U ovom primeru je postavljeno 'text-indent: 3em' tako da je prva linija pasusa uvučena.

- Poravnanje teksta u okviru elementa se zadaje svojstvom **text-align**; moguće vrednosti su **left**, **right**, **center**, **justify**

text-align: left	text-align: center	text-align: right	text-align: justify
*** ** ***** **	*** ** ***** **	*** ** ***** **	*** ** ***** **
***** ** *****	***** ** *****	***** ** *****	***** ** *****
** ** ***** **	** ***** ** *****	** ** ***** **	** ***** ** *****
*** *****	***** *****	*** *****	***** ***** **
***** ** *****	** ** * ***** *	***** ** *****	** * ***** * *****
***** * *****	*** **	***** * *****	**

- Poravnanje jednog elementa u odnosu na drugi element koji ga obuhvata zadaje se na drugi način





# Boja

- Boja teksta zadaje se svojstvom **color**
- Boje se mogu zadati na više načina:
  - korišćenjem imena  

```
p { color: red; }
```
  - heksadekadnim kodom oblika **#rrggbb** - tri dvocifrena heksadekadna broja  

```
p { color: #00ff00; }
```
  - dekadnom specifikacijom oblika **rgb(r,g,b)** - tri dekadna broja  

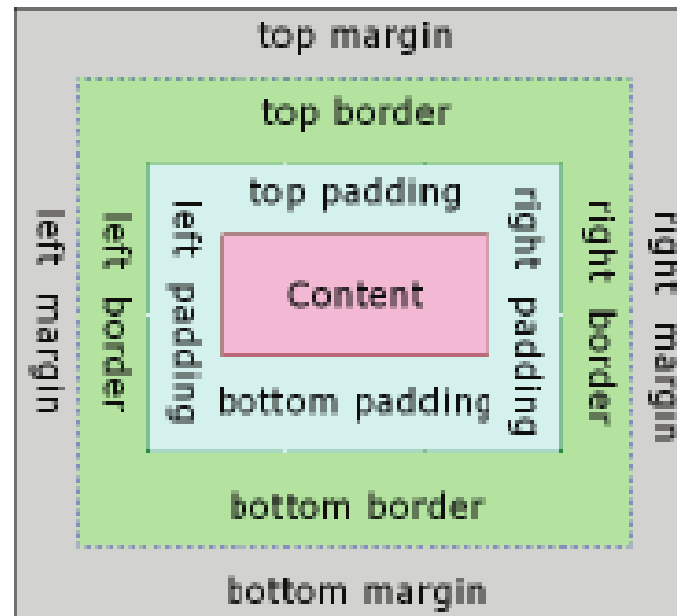
```
p { color: rgb(0,0,255); }
```

FFFFFF	000000	333333	666666	999999	CCCCCC	CCCC99	9999CC	666699
660000	663300	996633	003300	003333	003399	000066	330066	660066
990000	993300	CC9900	006600	336666	0033FF	000099	660099	990066
CC0000	CC3300	FFCC00	009900	006666	0066FF	0000CC	663399	CC0099
FF0000	FF3300	FFFF00	00CC00	009999	0099FF	0000FF	9900CC	FF0099
CC3333	FF6600	FFFF33	00FF00	00CCCC	00CCFF	3366FF	9933FF	FF00FF
FF6666	FF6633	FFFF66	66FF66	66CCCC	00FFFF	3399FF	9966FF	FF66FF
FF9999	FF9966	FFFF99	99FF99	66FFCC	99FFFF	66CCFF	9999FF	FF99FF
FFCCCC	FFCC99	FFFFCC	CCFFCC	99FFCC	CCFFFF	99CCFF	CCCCFF	FFCCFF



# Model kutije

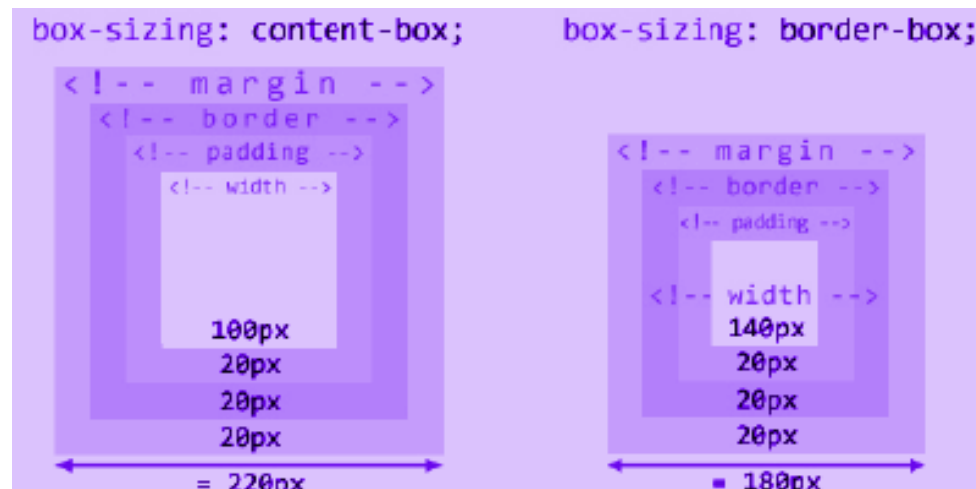
- HTML elementi mogu da se posmatraju kao pravougaone površine - **kutije** (box)
- Svaka kutija ima **sadržaj** (content) i može da ima **okvir** (border)
- Okvir je razdvojen od sadržaja unutrašnjom marginom, tj. **punjenjem** (padding), a od okolnih elemenata **spoljašnjom marginom** (margin)





# Model kutije - širina i visina

- Širina i visina elementa zadaju se svojstvima **width** i **height**; vrednost se zadaje u px
  - Podrazumevano je da se punjenje, okvir i margine ne računaju u širinu i visinu
  - Svojstvom **box-sizing** se podešava šta se računa u širinu i visinu elementa: podrazumevana vrednost je **content-box** i tada se računa samo sadržaj, ako se navede **border-box** onda se računa visina i širina sadržaja, punjenja i okvira (bez spoljašnjih margina)





## Model kutije - širina i visina (2)

- Nekada je zgodno ne fiksirati širinu i visinu, već samo zadati najmanje ili najveće dopuštene vrednosti; to se postiže svojstvima `max-width`, `min-width`, `max-height` i `min-height`



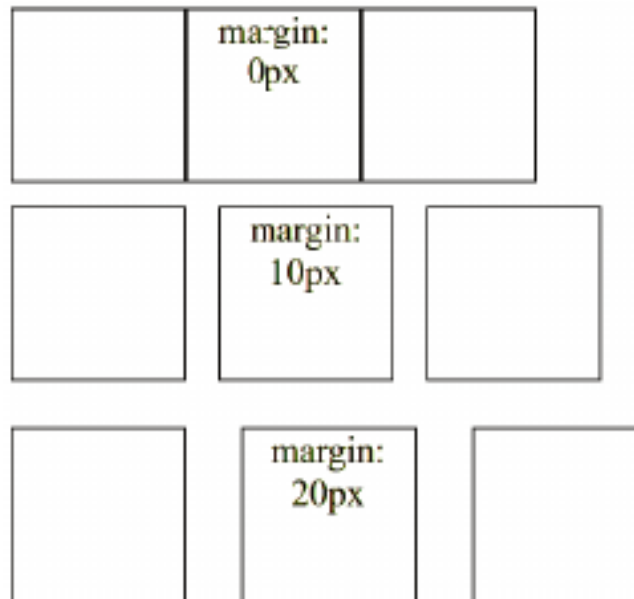
# Model kutije - spoljašnje margine

- Elementima je moguće podesiti spoljašnje i unutrašnje margine
- Spoljašnju marginu je moguće podesiti svojstvom **margin**:
  - ako se navede samo jedna vrednost, ona se odnosi na sve margine
  - ako se navedu dve vrednosti, prva se odnosi na levu i desnu, a druga na gornju i donju marginu
  - ako se navedu četiri vrednosti, one se odnose na levu, gornju, desnu, donju marginu redom
  - ako se navede auto, margine će se automatski rasporediti podjednako



## Model kutije - spoljašnje margine (2)

- Margine susednih elemenata se ne sabiraju, računa se veća od vrednosti





# Model kutije - unutrašnje margine

- Unutrašnju marginu je moguće podesiti svojstvom `padding`
  - Koristi se analogno svojstvu `margin`
  - Postoje i pojedinačna svojstva: `padding-top`, `padding-right`, `padding-bottom` i `padding-left`

`padding: 0px`

`padding: 10px`





# Model kutije - okviri

- Oko svakog elementa moguće je prikazati okvir
  - Debljina okvira podešava se svojstvom `border-width`
  - Tip linije okvira podešava se svojstvom `border-style`: moguće vrednosti su `solid`, `dashed`, `dotted`, itd.
  - Boja okvira podešava se svojstvom `border-color`
  - Sva tri svojstva moguće je zadati odjednom, korišćenjem svojstva `border`

```
p { border: 1px solid black; }
```
  - Postoje i svojstva `border-top`, `border-right`, `border-bottom` i `border-left`



## Model kutije – okviri (2)

- Zaobljenost okvira se može zadati svojstvom **border-radius**: vrednost odgovara poluprečniku krugova pomoću kojih se postiže zaobljenost okvira

border: 1px  
solid black

border: 2px  
dotted blue

border: 3px  
dashed red

border-radius:  
10px



# Pozadina elementa

- Svakom elementu moguće je podesiti pozadinu: boju ili sliku
- Boja pozadine elementa može se podesiti svojstvom **background-color**: vrednost tog svojstva je boja ili **transparent**

`background-color: yellow`

`background-color: #ccccff`

`background-color: rgb(255, 200, 200)`

- Kao pozadina elementa može se postaviti slika svojstvom **background-image**: vrednost se zadaje u obliku `url(...)`



## Pozadina elementa (2)

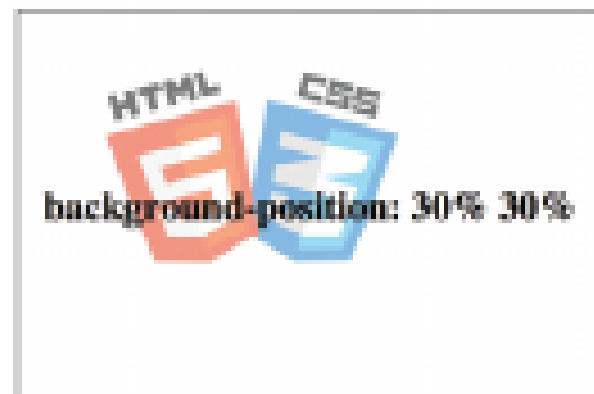
- Svojstvom `background-repeat` kontroliše se da li da se slika ponavlja dok ne ispuni širinu/visinu elementa ili ne; moguće vrednosti su: `repeat`, `repeat-x`, `repeat-y` i `no-repeat`





## Pozadina elementa (3)

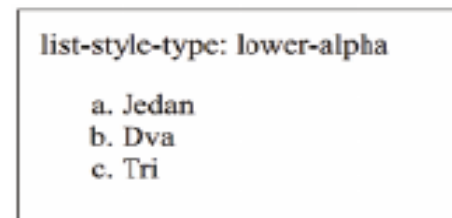
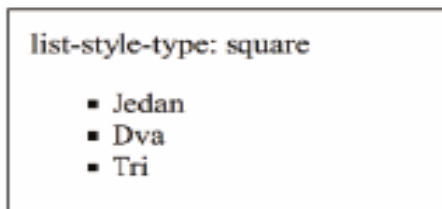
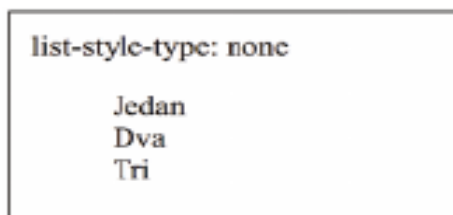
- Pozicija slike u pozadini elementa može se podesiti svojstvom **background-position**
  - moguće je navesti dve vrednosti (horizontalna, vertikalna) ili samo jednu (horizontalna, podrazumevano center)
  - Vrednosti mogu biti date kao:
    - procenat - x% znači da poravnava tačku koja se nalazi na x% širine (dužine) slike sa tačkom koja se nalazi na x% širine (dužine) elementa
    - dužina - gornje levo teme slike postavlja se na tačku pomerenu za ovu vrednost u odnosu na gornje levo teme elementa
    - top, bottom - 0% (100%) za vertikalnu poziciju
    - left, right - 0% (100%) za horizontalnu poziciju
    - center - 50% za horizontalnu/vertikalnu poziciju





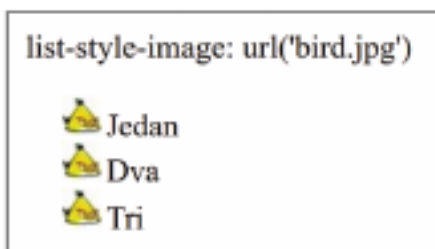
# Stilizovanje lista

- Najčešće se stilizuju oznake stavke liste
  - oblik znaka ispred stavki nabiranja u nenumerisanoj listi ili broja u numerisanoj listi se podešava svojstvom `list-style-type`: moguće vrednosti su: `disc`, `circle`, `square`, `none`, `decimal`, `lower-alpha`, `lower-roman`, `upper-alpha`, `upper-roman`, itd.



- kod nenumerisanih slika umesto znaka za nabiranje može se postaviti slika korišćenjem svojstva `list-style-image`  

```
ul { list-style-image: url("slika.png"); }
```





# Stilizovanje tabela

- Postoje i svojstva koja su karakteristična samo za tabele
  - Svojstvom **border-collapse** sa vrednošću collapse postavlja se da se susedne ćelije „slepe“ tj. da imaju jedinstveni okvir

Devojčice	Dečaci
120	125
118	123

Devojčice	Dečaci
120	125
118	123

- Za poravnanje sadržaja ćelija tabele koriste se svojstva **text-align** i **vertical-align** (ovaj drugi samo za ćelije tabele): vrednosti su **top**, **middle** i **bottom**

text-align: left	text-align: center	text-align: right
vertical-align: top	vertical-align: middle	vertical-align: bottom





# Stilizovanje elemenata `div` i `span`

- Generički elementi `div` i `span` nemaju unapred definisanu semantiku; zadatak im je da grupišu neki sadržaj
- Obično se koriste u kombinaciji sa globalnim atributima `id` i `class`
- Element `div` služi za grupisanje većih celina; prikazuje se kao **blok element** (block-level element)
- Element `span` služi za grupisanje manjih celina; prikazuje se kao **linijski element** (inline element)



# Element div

- U ranijim verzijima HTML-a bio je jedan od najkorišćenijih elemenata
- U verziji HTML5 treba ga koristiti kada nijedan od semantičkih elemenata nije odgovarajući

```
<main>  
  ...  
  <div id="gallery">  
    <!-- tri slicice na dnu centralnog dela strane -->  
  </div>  
</main>
```



# Element span

- Koristi se kada je potrebno delu teksta dodeliti značenje koje nije definisano HTML standardom
- Bez dodatnih podešavanja prikazuju se isto na veb-strani; međžtim na ovaj način moguće je izdvojiti sve označene delove teksta sa veb strane ili korišćenjem CSS-a podesiti prikaz svih označenih elemenata

- Najčešće se koristi u kombinaciji sa atributom **class**

```
Osnovni delovi racunarskog sistema su <span class="termin">
procesor</span>,
<span class="termin">memorija</span> i
<span class="termin">ulazno-izlazni uredaji</span>.
<pre>
<code>
<span class="keyword">program</span> <span class="id">Hello</span>;
<span class="keyword">begin</span>
<span class="id">WriteLn</span>(<span class="str">'Zdravo, svete'</span>)
<span class="keyword">end</span>.
</code>
</pre>
```



# Prikaz

- Razlikujemo dva načina prikaza elemenata:
  - blok elementi
    - **display: block;**
    - mogu da sadrže tekst, linijske elemente i druge blok elemente
    - prostiru se celom širinom bloka, slažu se jedan ispod drugog
    - **div, section, article, header, footer, main, aside, form, p, ul, ol i li**
  - linijski elementi
    - **display: inline;**
    - mogu da sadrže tekst i linijske elemente
    - zauzimaju koliko i sadržaj, slažu se jedan pored drugog
    - **span, a, img, em, strong, small, i, b, u, sub, sup**



## Prikaz (2)

- Svojstvo **display** može imati različite vrednosti:
  - **none** - element se u potpunosti izostavlja iz prikaza (ne zauzima nikakav prostor na strani)
  - **block** - element se prikazuje kao blok element; mogu mu se postavljati širina, visina, okvir i margine
  - **inline** - element se prikazuje kao linijski element; može se podešavati okvir, margine, visina, ali ne i širina;
    - visina se postavlja svojstvom **line-height**; ima smisla podešavati samo levu i desnu marginu jer samo one pomeraju okolni sadržaj

Ovo je jedna rečenica na početku teksta. U ovom tekstu se nalazi jedan linijski element . Ovo je jedna rečenica na kraju teksta.

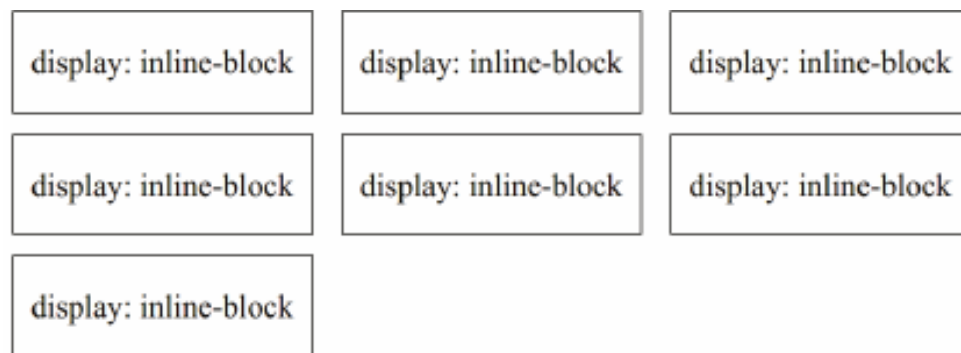


## Prikaz (3)

- Svojstvo **display** može imati različite vrednosti:
  - **inline-block** - element se prikazuje kao linijski blok element; ne prostire se celom širinom, ali mu se mogu podešavati i širina i visina i margine

- Primer:

```
div { border: 1px solid red; }  
div.inblock {  
  display: inline-block;  
  width: 100px; height: 50px;  
  margin: 10px;  
}  
<div>Ispred</div>  
<div class="inblock">1</div>  
<div class="inblock">2</div>  
<div class="inblock">3</div>  
<div class="inblock">4</div>  
<div class="inblock">5</div>  
<div class="inblock">6</div>  
<div>Iza</div>
```





# Zahvalnica

Delovi materijala ove prezentacije su preuzeti iz:

- Skripte iz predmeta Uvod u veb i internet tehnologije, na Matematičkom fakultetu Univeziteta u Beogradu, autor prof. dr Filip Marić
- Prezentacija iz predmeta Uvod u veb i internet tehnologije, na Matematičkom fakultetu Univeziteta u Beogradu, autor dr Vesna Marinković
- Skripte iz predmeta Informatika na Univerzitetu Milano Bicocca, autor dr Mirko Cesarini