

Uvod u Veb i Internet tehnologije

CSS

Filip Marić
Vesna Marinković
Milan Banković
Jelena Graovac

Stilski listovi



- Vizuelna prezentacija HTML dokumenata podešava se korišćenjem **stilskih listova** (*stylesheets*) opisanih u jeziku **CSS** (*Cascading Style Sheets*)
- Prva verzija objavljena 1996. godine (CSS 1)
- Današnje konture dobija sa verzijom CSS 2 (CSS 2.1)
- Aktuelna verzija CSS 3
- U razvoju (od kraja 2017. godine): CSS 4
- Verzija CSS 4 još uvek nije potpuno podržana od strane veb pregledača

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**

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

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

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

- Svako pravilo je oblika:

selektor { niz deklaracija međusobno razdvojenih znakom ';' }

- Svaka deklaracija je oblika:

svojstvo: **vrednost**

Opšta sintaksa stilskih listova

- Selektori mogu biti složeniji nego samo navođenje imena elemenata
- Više selektora može se navesti zajedno (razdvajaju se zarezima)

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

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

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

Uključivanje stilskih listova u HTML dokumente

- Tri načina za uključivanje CSS opisa u HTML dokument
- *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>
```

- *Opisi na nivou dokumenta* (element 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 listova u HTML dokumente

- *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 stilskeg 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:
 - stilske listove autora veb-strane (date na neki od prethodna tri načina)
 - stilske listove korisnika (zadaju se podešavanjima pregledača veba)
 - podrazumevane stilske listove pregledača veba
- Različiti opisi za neki element se sabiraju; ako dođe do konflikta prednost se daje u opadajućem prioritetu gore navedenih načina zadavanja stilskih listova (ovo se može promeniti navodjenjem **!important** specifikatora):

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

- Ako do konflikta dođe na nivou stilskih opisa autora, najveći prioritet imaju pravila navedena na nivou elementa (style atribut), dok se pravila navedena u spoljašnjim listovima i u okviru strane (style element) rangiraju najpre prema specifičnosti, a zatim prema mestu u dokumentu (kasnije navedeni stilovi imaju veći prioritet)

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**

```
<p id="opis">U ovom pasusu biće opisana glavna svojstva...</p>
```

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

isto je što i:

```
#opis { color : blue; }
```

Selektori

- 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**

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

```
...
```

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

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

nije nužno isto što i:

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

- Opštije: `E[atribut="vrednost"]` – svi elementi *E* koji imaju dati atribut postavljen na datu vrednost
- Slično: `E[atribut~="vrednost"]` – svi elementi *E* koji imaju dati atribut sa vrednošću koja sadrži datu reč (`E.klasa` je ekvivalentno sa `E[class~="klasa"]`)

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 (:**hover**), a pseudoelementi pomoću dvostruke dvotačke (**::first-line**)
- Najčešće pseudoklase i pseudoelementi:
 - **:link**, **:active**, **:visited**
 - **:hover**
 - **:checked**
 - **:first-child**
 - **:nth-child($An+B$)**, **:nth-child(even)**, **:nth-child(odd)**
 - **::first-line**
 - **::first-letter**
 - **::before**, **::after**

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

Ugnežđeni elementi

- **Ugnežđeni elementi** se koriste kada je potrebno promeniti stil samo onih elemenata koji se nalaze u nekom širem elementu
- Sintaksa: `selector1 selector2` – ovim se označavaju elementi opisani selektorom `selector2` koji se nalaze u okviru selektora `selector1`
- Sintaksa: `selektor1 > selektor2` – ovim se označavaju elementi opisani selektorom `selector2` koji su neposredni potomci selektora `selector1`
- Sintaksa: `selektor1 + selektor2` – ovim se označavaju elementi opisani selektorom `selector2` čiji je prethodni brat u HTML stablu opisan selektorom `selektor1`
- Sintaksa `selektor1 ~ selektor2` – ovim se označavaju elementi opisani selektorom `selector2` čiji je neki od prethodne braće u HTML stablu opisan selektorom `selektor1`

```
main p { color: blue; }  
div#gallery > img.small { width: 50px; }
```

Font

- 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
- Moguće je navesti više opisa u opadajućem prioritetu

```
p { font-family: "New Century Schoolbook", Times, serif }
```

Font

- Veličina fonta se zadaje svojstvom **font-size**, na dva načina:
 - apsolutno: izraženo u pikselima (px) ili u „tačkama” (pt). Veličina piksela zavisi od rezolucije. 1pt=1/72 inča
 - relativno: izraženo u procentima ili em-ovima (1em=100%), u odnosu na veličinu fonta nasledjenu od roditeljskog elementa

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

- Varijante fonta
 - **font-style** određuje iskošenost karaktera; moguće vrednosti: normal, italic, oblique
 - **font-weight** određuje debljinu slova; najčešće vrednosti: normal, bold, lighter
 - Dozvoljeno je više karakteristika fonta zadati jednom deklaracijom
- ```
p { font: italic bold 12pt Times, serif; }
```

## Stilizovanje teksta – ukrašavanje teksta

- U CSS-u se (za razliku od procesora teksta) podešavanje teksta razlikuje od podešavanja fonta
- U podešavanje teksta spadaju: podvlačenje teksta, precrtavanje, uvlačenje prve linije, poravnanje teksta, podešavanje razmaka između reči i slova,...
- U CSS-u se svojstvima fonta određuje izbor glifa za ispis karaktera, a svojstvima teksta kako se glifovi raspoređuju i da li je potrebno još nešto dočrtati

```
text-decoration: none
text-decoration: overline
text-decoration: underline
text-decoration: line-through
```

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

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

## Stilizovanje teksta – razmak između karaktera

- U CSS-u moguće je fino podešavanje horizontalnog razmaka između karaktera
- Za podešavanje razmaka između karaktera u jednoj reči koristi se svojstvo `letter-spacing`, a za podešavanje razmaka između susednih reči svojstvo `word-spacing`; zadaje se 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: 1.5em; }
```

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



# Stilizovanje teksta – uvlačenje prve linije teksta, 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`

| <code>text-align: left</code> | <code>text-align: center</code> | <code>text-align: right</code> | <code>text-align: justify</code> |
|-------------------------------|---------------------------------|--------------------------------|----------------------------------|
| *** ** ***** **               | *** ** ***** **                 | *** ** ***** **                | *** ** ***** **                  |
| ***** ** *****                | ***** ** *****                  | ***** ** *****                 | ***** ** *****                   |
| ** ***** **                   | ** ***** **                     | ** ***** **                    | ** ***** **                      |
| *** *****                     | *** *****                       | *** *****                      | *** *****                        |
| ***** ** *****                | ***** ** *****                  | ***** ** *****                 | ***** ** *****                   |
| ***** ** *****                | *****                           | ***** ** *****                 | **                               |

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

# Boje

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

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

```
p { color: #00ff00; }
```

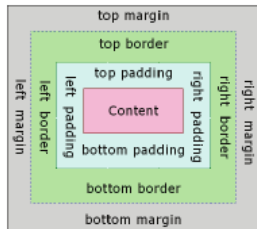
|        |        |        |        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| FFFFFF | 000000 | 333333 | 666666 | 999999 | cccccc | CCCC99 | 9999CC | 666699 |
| 660066 | 663399 | 996633 | 003300 | 003333 | 003399 | 000066 | 330066 | 660066 |
| 990000 | 993300 | CC9900 | 006600 | 336666 | 0033FF | 000099 | 660099 | 990066 |
| CC0000 | CC3366 | FFCC00 | 009900 | 006666 | 0066FF | 0000CC | 663399 | CC0099 |
| FF0000 | FF3366 | FFFF00 | 00CC00 | 009999 | 0099FF | 0000FF | 9900CC | FF0099 |
| CC3333 | FF6600 | FFFF33 | 00FF00 | 00CCCC | 00CCFF | 3366FF | 9933FF | FF00FF |
| FF6666 | FF6633 | FFFF66 | 66FF00 | 66CCCC | 00FFFF | 3399FF | 9966FF | FF66FF |
| FF9999 | FF9966 | FFFF99 | 99FF99 | 66FFCC | 99FFFF | 66CCFF | 9999FF | FF99FF |
| FFCCCC | FFCC99 | FFFFCC | CCFFCC | 99FFCC | CCFFFF | 99CCFF | CCCCFF | FFCCFF |

- heksadekadnim kôdom oblika `#rgb` (`#27c` = `#2277cc`)
- dekadnom specifikacijom oblika `rgb(r,g,b)` – tri dekadna broja

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

# Model kutije

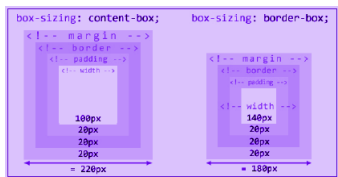
- Svi HTML elementi mogu da se posmatraju kao pravougaone površine – **kutije** (*box*)
- Svaka kutija ima **sadržaj** (*content*) i može da ima **okvir** (*border*)
- Okvir je razdvojen od sadržaja **unutrašnjom marginom**, tj. **punjenjem** (*padding*), a od okolnih elemenata **spoljašnjom marginom**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



Модел кутије

# Širina i visina

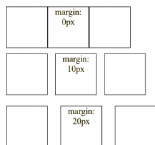
- Širina i visina elementa zadaju se svojstvima `width` i `height`; vrednost se zadaje u px
  - Podrazumevano se punjenje, okvir i margine ne računaju u širinu i visinu
  - Svojstvom `box-sizing` se podešava šta se računa u širinu i visinu elementa: podrazumevana vrednost je `content-box` i tada se računa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



- 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`, `min-height`

# 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 gornju i donju marginu, a druga na levu i desnu
  - ako se navedu tri vrednosti, prva se odnosi na gornju, druga na levu i desnu, a treća na donju marginu
  - ako se navedu četiri vrednosti, one se odnose na gornju, desnu, donju, levu marginu redom
- Postoje i pojedinačna svojstva: `margin-top`, `margin-right`, `margin-bottom`, `margin-left`
- ako leva i desna margina imaju vrednost `auto`, margine će se automatski podesiti da budu jednake
- Gornje i donje margine susednih elemenata se ne sabiraju, računa se veća od vrednosti; leve i desne margine susednih elemenata se sabiraju



# Unutrašnje margine

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

padding: 0px

padding: 10px

# 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`,...
- Boja okvira podešava se svojstvom `border-color`
- Sva tri svojstva moguće je zadati korišćenjem svojstva `border`  

```
p { border: 1px solid black; }
```
- Postoje i svojstva `border-top`, `border-right`, `border-bottom`, `border-left`
- Zaobljenost okvira se može zadati svojstvom `border-radius`: vrednost odgovara poluprečniku krugova upisanih u svako teme pravouganka

border: 1px  
solid black

border: 2px  
dotted blue

border: 3px  
dashed red

border-radius:  
10px

# Pozadina elemenata

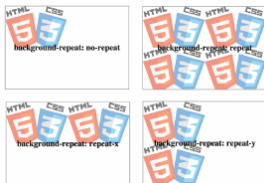
- Svakom elementu moguće je podesiti pozadinu: boju ili sliku
- Boja pozadine elementa može se podesiti svojstvom **background-color**: vrednost je boja ili transparent

```
background-color: yellow
```

```
background-color: #ecccff
```

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

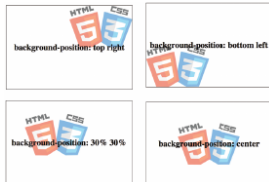
- Kao pozadina elementa može se postaviti slika svojstvom **background-image**: vrednost se zadaje u obliku `url(...)`
- Svojstvom **background-repeat** kontroliše se da li da se slika ponavlja dok ne ispuni širinu/visinu elementa ili ne; moguće vrednosti su `repeat`, `repeat-x`, `repeat-y`, `no-repeat`





# Pozadina elemenata

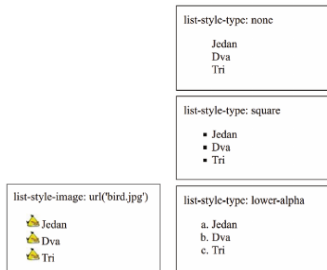
- Pozicija slike u pozadini elementa može se podesiti svojstvom **background-position**: moguće je navesti dve vrednosti (horizontalna, vertikalna) ili samo jednu (horizontalna, podrazumevano center)
- Vrednosti mogu biti date kao:
  - procenat –  $x\%$  znači da poravnava tačku koja se nalazi na  $x\%$  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`, ...
  - 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

- Stilizacija tabela obuhvata postavljanje dimenzija, okvira, boje pozadine, unutrašnjih i spoljašnjih margina i sl.
- Podešavanja se vrše na nivou tabele, na nivou pojedinačnih redova i pojedinačnih ćelija
- Opšta svojstva border, background-color, padding, margin

```
table {
 margin-left: auto;
 margin-right: auto;
}
td {
 background-color: #eeeeee;
 padding: 5px;
 border: 1px solid red;
}
```

| Ime     | Prezime    | Poeni |
|---------|------------|-------|
| Petar   | Petrović   | 100   |
| Ana     | Anić       | 95    |
| Mihailo | Mihailović | 82    |
| Mara    | Marić      | 73    |

# Stilizovanje tabela

- Postoje i svojstva koja su karakteristična samo za tabele
- Svojevremeno `border-collapse` sa vrednošću `collapse` postavlja se da se susedne ćelije slepe i 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`: vrednosti `top`, `middle`, `bottom`

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

# 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`, `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`

Nekoliko korisnih linkova: [w3c](#), [w3schools](#)

Puno sreće u učenju veb-tehnologija.

Однос између блок и линијских  
елемената

# Prikaz

- Svojtvo **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, ali ne i visina i širina (osim kod eksternih objekata, poput slika, kod kojih može, iako su inline elementi); ima smisla podešavati samo levu i desnu marginu jer samo one pomeraju okolni sadržaj

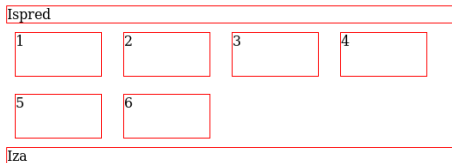
Ovo je jedna rečenica na početku teksta. U ovom tekstu se nalazi jedan linijski element. Ovo je jedna rečenica na kraju teksta.

Маргине линијског елемента

# Prikaz

- **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

```
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>
```



# Pozicioniranje

- Elementi se pozicioniraju postavljanjem svojstva **position** na:
  - **static** – podrazumevano pozicioniranje je **statičko**, elementi se uklapaju u tzv. normalni tok (blok elementi jedan ispod drugog, linijski elementi jedan uz drugog)
  - **relative** – elementi se mogu pomeriti u odnosu na svoju statičku poziciju korišćenjem **relativnog pozicioniranja**; koordinate se zadaju svojstvima **top**, **left**, **bottom**, **right**
  - Promenom pozicioniranja elementi se mogu preklopiti; svojstvo **z-index** određuje koji će element biti iznad/ispod (veću vrednost ima element iznad)



# Pozicioniranje

- **absolute** – kod **apsolutnog pozicioniranja** element se može pozicionirati u okviru šireg elementa zadavanjem njegovih koordinata u odnosu na širi element
  - pozicioniranje se vrši u odnosu na najbliži element koji ga sadrži, a pozicioniran je nestatički (ako takav element ne postoji, pozicionira se u odnosu na element body)
  - apsolutno pozicioniranje izbacuje element iz normalnog toka, a ostali elementi se pozicioniraju kao da on ne postoji
  - koordinate se zadaju svojstvima `top`, `left`, `bottom`, `right`

# Primer apsolutnog pozicioniranja

```
<div>...</div>
<div id="container">
 <div id="content">...</div>
</div>
<div>...</div>
```

```
div {
 border: 1px solid black;
}
#container {
 position: relative;
 width: 500px;
 height: 200px;
 top: 50; left: 70;
}
#content {
 position: absolute;
 top: 50px; left: 50px;
}
```

position: static;

position: relative;

position: absolute;

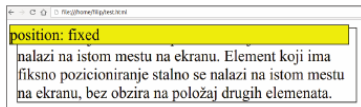
position: static;

# Pozicioniranje

- **fixed** – slično apsolutnom pozicioniranju; element se pozicionira u odnosu na ekran pregledača veba i ne pomera se prilikom skrolovanja strane
  - koristi se za naslove i potpise
  - fiksirano pozicioniranje izbacuje element iz normalnog toka

```
.fixed{
 position: fixed;
 top: 0;
 left: 0;
 width: 200px;
}
```

```
<div class="fixed">
... nalazi na istom mestu na ekranu. Element koji ima fiksno pozicioniranje...
</div>
```



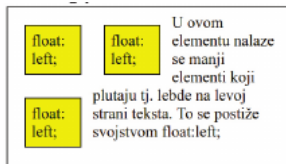
## Plutajući elementi

- Neki elementi se mogu podesiti da budu u levom ili desnom kraju elementa koji ih okružuje, a da se ostali sadržaj obmotava oko njih
- Element se postavlja plutajućim svojstvom `float` sa mogućim vrednostima `left` i `right`
- Ako dva elementa imaju istu vrednost svojstva `float` oni se postavljaju jedan pored drugog
- Za postavljanje plutajućih elemenata jedan ispod drugog potrebno je navesti i svojstvo `clear` sa vrednostima `left`, `right` ili `both`

```

<div>
 <div style="float: left;">
 <div style="float: left;">
 <div style="float: left; clear: left;">
 U ovom elementu nalaze se manji elementi
 koji plutaju...
</div>

```



- **NAPOMENA:** plutajući elementi se u odnosu na svoj sadržaj ponašaju kao blok elementi, bez obzira na vrednost `display` svojstva

# Fleksibilno prikazivanje

- Novina u CSS3 je **fleksibilno prikazivanje**
- **display: flex;** – element postaje **flex kontejner** za svoj sadržaj
- Flex kontejner se ponaša kao blok element za svoje okruženje, a unutar sebe kreira **fleksibilni kontekst formatiranja**.
- Neposredna deca flex kontejnera nazivaju se **flex stavke** (flex items)
- Unutar flex kontejnera flex stavke se redjaju duž **glavne ose** koja može da se podešava svojstvom **flex-direction**. Moguće vrednosti su:
  - **row**: sa leva na desno
  - **row-reverse**: sa desna na levo
  - **column**: odozgo na dole
  - **column-reverse**: odozdo na gore
- Osa normalna na glavnu osu je **poprečna osa**
- Flex stavke za svoj sadržaj uspostavljaju standardni blok kontekst formatiranja
- Fleksibilno prikazivanje uvodi **ravnopravnost osa**, jer se sada isti obrasci formatiranja mogu primenjivati i po vertikali i po horizontali
- Fleksibilno prikazivanje omogućava jednostavno pozicioniranje duž glavne i sporedne ose, kao i podesivo širenje i skupljanje elemenata, što omogućava bolje prilagodjavanje dimenzijama ekrana

# Fleksibilno prikazivanje

- **flex-wrap** svojstvo (vrednosti **nowrap**, **wrap**, **wrap-reverse**) omogućava da se flex stavke koje ne mogu da se smeste duž glavne ose u jednoj liniji prebace u sledeću liniju
- **justify-content** svojstvo određuje poravnanje stavki duž glavne ose (vrednosti **center**, **space-between**, **space-around**, **flex-start**, **flex-end**)
- **align-items** svojstvo određuje poravnanje stavki iz iste linije po poprečnoj osi (vrednosti **center**, **flex-start**, **flex-end**, **stretch**, **baseline**)
- **align-content** svojstvo određuje poravnanje linija duž poprečne ose (vrednosti **space-between**, **space-around**, **center**, **stretch**, **flex-start**, **flex-end**)
- NAPOMENA: vrednosti **flex-start** i **flex-end** se računaju u odnosu na **flex-direction** svojstvo (npr. za **flex-direction: row**, **flex-start** je levo, a **flex-end** desno)

# Fleksibilno prikazivanje

- **order** svojstvo određuje relativni poredak flex stavki

```
<div style="display: flex; flex-direction: row;">
<div style="order: 3">1</div>
<div style="order: 2">2</div>
<div style="order: 2">3</div>
<div style="order: 1">4</div>
</div>
```

- Stavke sa manjom vrednošću **order** svojstva se redjaju prve. Stavke sa istim vrednostima se redjaju u poretku u kom su navedene
- **flex-grow** svojstvo određuje u kojoj meri će se stavka širiti duž glavne ose prilikom promene dimenzija prozora (gledano relativno u odnosu na druge stavke). Podrazumevana vrednost je 1
- **flex-shrink** svojstvo određuje u kojoj meri će se stavka skupljati duž glavne ose prilikom promene dimenzija prozora (gledano relativno u odnosu na druge stavke). Podrazumevana vrednost je 0
- **flex-basis** svojstvo određuje polaznu veličinu stavke (duž glavne ose)
- **flex** svojstvo je skraćeni zapis za prethodna tri svojstva  
`flex: 1 0 25%; /* grow je 1, shrink je 0, basis je 25% */`

# Prilagodjavanje izgleda dimenzijama ekrana

- Tradicionalno, za prikaz veb strana koristili smo monitore desktop računara
- Danas se sve više koriste uredjaji manjih dimenzija, poput tableta i mobilnih telefona
- Javlja se potreba da izgled veb strane bude prilagodljiv dimenzijama ekrana na kome se prikazuje
- Da bi se ovo postiglo, potrebno je najpre naložiti pregledaču da dimenzije veb strane (body elementa) prilagodi dimenzijama ekrana, kao i da podesi inicijalno uveličanje (zoom):

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- Nakon toga je u CSS opisima potrebno postaviti uslove vezane za ekran uredjaja pod kojim odredjena svojstva važe:

```
<link rel="stylesheet" href="style.css" media="screen and (min-width: 780px)" />
...
<style>
@media screen and (max-width: 779px) {
 body {
 background-color: rgb(200,220,220);
 }
}
</style>
```



# Prilagodjavanje izgleda dimenzijama ekrana

- Direktiva `@media` prepoznaje više različitih tipova medijuma:
  - `screen`
  - `print`
  - `projection`
  - `tty`
  - `tv`
  - `braille`
  - ...
- Najčešće korišćeni tip je `screen`
- Moguće je navesti i `all`, ako želimo da stilovi važe za sve medijume (medjutim, `all` se može i izostaviti)
- Neki od dodatnih uslova koji se mogu navoditi su:
  - `max-width`, `min-width`, `max-height`, `min-height`
  - `orientation` (`landscape` ili `portrait`)
  - `color` (da li je ekran u boji ili ne)
  - `min-color`, `max-color` (minimalni/maksimalni broj bitova po boji)
  - `min-resolution`, `max-resolution` (izraženo u dpi)
  - ...