

CSS

**Četvrti dio**

# PREGLED

- Display
- Max-width
- Position, z-index
- Overflow
- Float i clear
- Align
- Opacity
- Forme

# OBNAMLJANJE

- Box model, od čega se sastoji visina i širina box-a
- Zašto nekad definišemo više vrijednosti za font-family
- Outline, da li spada u width i height box modela
- Tabele, šta je colgroup element, šta caption element
- Kako definišete odstupanje teksta od okvira

# DISPLAY

- Najbitniji CSS property za kontrolu layout-a
  - `display` definiše da li će i kako element biti prikazan
  - Svaki html element ima default vrijednost za `display`, za većinu elemenata je to `block` ili `inline`
- **Block elementi** – uvijek počinju na novoj liniji i zauzimaju full moguću širinu (rastežu se lijevo i desno koliko je moguće)
  - `div`, `h1-h6`, `p`, `form`, `header`, `footer`, `section`
- **Inline elementi** – ne počinju na novoj liniji i zauzimaju onoliko prostora koliko je potrebno
  - `span`, `a`, `img`
- Sa `display: none` skrivate element. Razlika između `visibility: hidden` ?

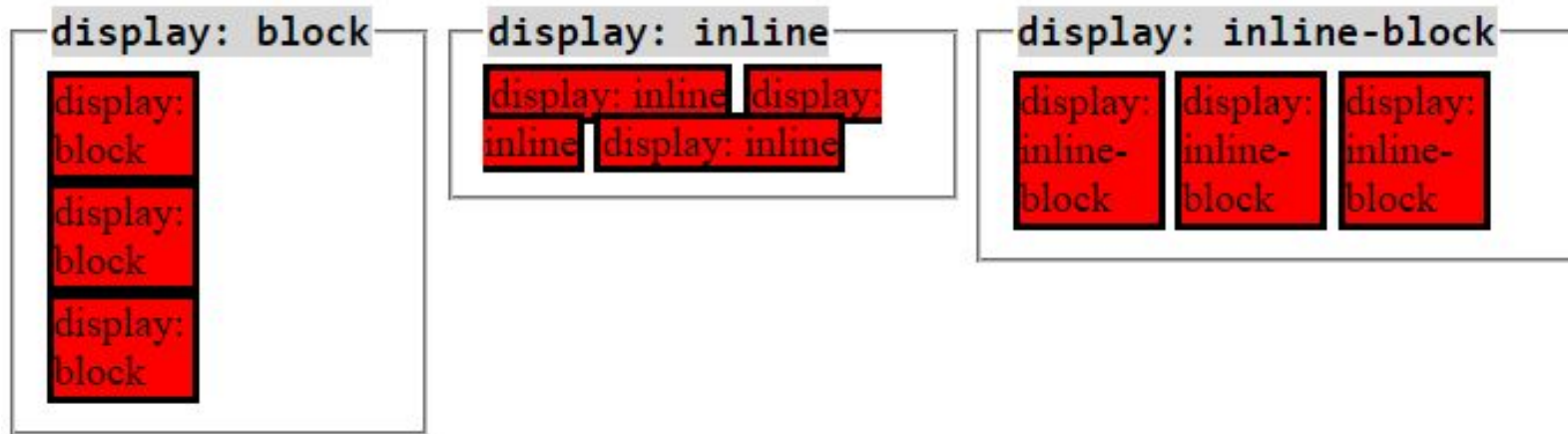
# DISPLAY, INLINE

- Neke od osobina elemenata za `display: inline`
  - Poštuju lijevu i desnu za marginu i padding, ali ne top i bottom
  - Visina i širina ovih elemenata ne može da se mijenja
    - Za sliku ovo nije slučaj
  - Ostali elementi mogu da budu lijevo i desno od ovih elemenata
  - `span`, a i `img` imaju default vrijednost `display: inline`
  - Za više informacija pročitajte i [članak](#)

# DISPLAY, BLOCK I INLINE-BLOCK

- Neke od osobina elemenata za **display: block**
  - Poštuju sve, lijevu, desnu, gornju i donju marginu i padding
  - Generišu line-break nakon blok elementa (dva bloka ne mogu da stoje jedan pored drugog)
  - Zauzima se cijela širina elementa ako nije definisana
- Neke od osobina elemenata za **display: inline-block**
  - Omogućava da drugi elementi mogu da budu jedni pored drugih
  - Poštuju sve stilove, lijeva, desna, gornja i donja margina/padding
  - Poštuju visinu i širinu

# DISPLAY, SLIKA I PRIMJERI



- Primjeri
  - Primjer 1
  - Primjer 2
  - Primjer 3

# WIDTH I MAX-WIDTH

- Kao što smo vidjeli, block element uvijek zauzme cijelu širinu. Međutim, ako postavite width block elementu, onemogućavate širenje tog elemente do ivica njegovog parent elementa. Onda možete postaviti margine na auto za horizontalno centriranje unutar parenta
- Max-width se koristi za manje prozore i korisno je za manje uređaje

```
div.ex1 {  
    width: 500px;  
    margin: auto;  
    border: 3px solid #73AD21;  
}
```

```
div.ex2 {  
    max-width: 500px;  
    margin: auto;  
    border: 3px solid #73AD21;  
}
```



# POSITION

- Služi za pozicioniranje elementa na stranici. Vrste pozicioniranja:
  - `static`
  - `relative`
  - `fixed`
  - `absolute`
  - `sticky` (sami pogledajte)
- Nakon što odaberete metod pozicioniranja, koristite `top`, `left`, `right`, i `bottom` za pozicioniranje.

# POSITION, STATIC

- Ovo je default vrijednost za position.
- Element pozicioniran na ovaj način ne može biti pomjeren sa top, bottom, left i right. Ovi elementi su pozicionirani na osnovu slaganja html elemenata na strani

```
div.static {  
    position: static;  
    border: 3px solid #73AD21;  
}
```

# POSITION, RELATIVE

- Elementi pozicionirani na ovaj način su relativno pozicionirani u odnosu na normalnu poziciju

## HTML

```
<h2>position: relative;</h2>
```

```
<p>An element with position: relative; is  
positioned relative to its normal  
position:</p>
```

```
<div class="relative">
```

```
This div element has position: relative;
```

```
</div>
```

## CSS

```
div.relative {  
    position: relative;  
    left: 10px; /*kako ovo moze jos da se zapise*/  
    top: 10px; /*kako ovo moze jos da se zapise*/  
    border: 3px solid #73AD21;  
}  
body {  
    position: relative;  
    left: 40px;  
}
```

# POSITION, FIXED

- Relativno pozicioniranje u odnosu na viewport, što znači da će element ostati uvijek na istom mjesto i ako se stranica scrolla.

## HTML

```
<h2>position: fixed;</h2>
```

```
<p>An element with position: fixed; is positioned  
relative to the viewport, which means it always  
stays in the same place even if the page is  
scrolled:</p>
```

```
<div class="fixed">
```

```
This div element has position: fixed;
```

```
</div>
```

## CSS

```
div.fixed {  
    position: fixed;  
    bottom: 0;  
    right: 0;  
    width: 300px;  
    border: 3px solid #73AD21;  
}
```

# POSITION, ABSOLUTE

- Relativno pozicioniran u odnosu na najbliži pozicioniran parent element
  - Element je **pozicioniran** ako position **nije static**
  - Čest je slučaj da se kombinuje parent element čija je pozicija relative sa child elementom čija je pozicija absolute
- Primjer 1

# PREKLAPANJE ELEMENATA, Z-INDEX

- Kad su elementi pozicionirani oni mogu da se preklapaju
- Property **z-index** služi za definisanje redosleda slaganja elemenata (koji element će da bude ispod, koji iznad, itd.). Element čiji je z-index veći, ima i veći prioritet, a ako dva elementa imaju isti z-index onda se prikazuje onaj zadnji dodat u

```
<h1>This is a heading</h1>  
  
<p>Because the image has a z-index of -1, it  
will be placed behind the text.</p>
```

```
img {  
    position: absolute;  
    left: 0px;  
    top: 0px;  
    z-index: -1;  
}
```

# OVERFLOW

- Definiše **overflow** content elementa
  - **visible** - sadržaj koji ne može da stane unutar elementa vidljiv
  - **hidden** - sadržaj skriven
  - **scroll** - sadržaj skriven ali na scroll vidljiv
  - **auto** - ako sadržaj ne može da stane, dodaje se scrollbar
- Overflow radi samo za blok elemente sa specifičnom visinom
- Imamo još i **overflow-x** i **overflow-y**

<h2>CSS Overflow</h2>

<p>By default, the overflow is visible, meaning that it is not clipped and it renders outside the element's box:</p>

<div>You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.</div>

```
div {  
  background-color: #eee;  
  width: 200px;  
  height: 50px;  
  border: 1px dotted black;  
  overflow: visible; /*isprobati ostale*/  
}
```

# FLOAT

- Property `float` se koristi za pozicioniranje i layout stranice, a vrijednosti su:
  - `left` - elementi se pomjeraju lijevo u parent elementu
  - `right` - elementi se pomjeraju desno u parent elementu
  - `none` - default
  - `inherit` - nasljeđuje float u odnosu na parent element
- Primjer 2 (isporbati razne varijacije, šta ako je slika child od body, itd.)



# CLEAR

- Property `clear` definiše koji elementi mogu da “plutaju” (`float`) do `cleared` elementa i na kojoj strani
  - `none`, by default
  - `left` - nema floating elemenata na lijevoj strani
  - `right` - nema floating elemenata na desnoj strani
  - `both` - nema floating elemenata
  - `inherit` - element nasleđuje `clear` vrijednost od parent elementa
- Najčešće se koristi nakon što se iskoristi `float` property na prethodni (ne parent) element
- Kad se čisti `float`, `clear` i `float` trebaju da se mećuju
  - Ako element pluta lijevo, onda se `clear` postavi na `left`. Element koji pluta će i dalje da pluta, ali `cleared` element će se pojaviti ispod na stranici

# CLEARFIX HACK

- Šta ako je element koji pluta veći od parent elemente
  - `overflow: auto`
  - Primjer 4
- Ovaj način funkcionije sve dok uspijevate da iskontrolirate margine i padding, inače biste vidjeli scrollbar. Moderni način da se riješi ovaj problem
  - Koristiti `clear` i `::after` pseudo element
  - Primjer 5

# PRIMJERI, FLOAT I CLEAR

- Grid jednakih sekcija
  - Primjer 6
- Grid slika
  - Primjer 7
- Vidjeli smo kako da napravimo sekcije jednakih širina, međutim za visinu je to malo drugačije
  - Primjer 8
    - Dobro rješenje ako je sadržaj box-a uvijek isti
  - Primjer 9
    - Flex, detaljno ćemo govoriti o njemu
    - Rješenje za prethodni primjer

# JOŠ PRIMJERA

- Navigation primjer
  - [Link](#)
- Layout primjer
  - [Link](#)

# ALIGN (PORAVNANJE), HORIZONTAL

- Jedna od bitnijih stvari o kojoj se treba voditi računa pri razvoju stranica
  - Za horizontalno centriranje najčešće se postavi širina parent div-a i `margin: auto`
    - Napomena: ako se ne postavi width, margin:auto ne funkcioniše
  - Za centriranje teksta koristi se `text-align: center`
  - Za centriranje slike, postavite `display: block` i `margin-left: auto`, `margin-right: auto`, postavite obavezno i width
  - Kako bi odradili right align pomoću position: absolute
    - Pogledajte
  - Takođe, već smo radili poravnanje sa float

# ALIGN, VERTICAL

- Postoji više načina za vertikalno poravnanje
  - Jedan od načina je da koristite istu vrijednost za padding top i bottom, `padding: 70px 0;`
    - Ako dodate `text-align: center` ujedno se odradili vertikalno i horizontalno poravnanje
    - Primjer
  - Još jedan zanimljiv način poravnanja sa `line-height`
    - Primjer
      - Šta će da se desi ako maknem height za div element, šta ako ima više paragrafa, šta ako je paragraf duži (staje u više linija)
  - Još jedan od načina za centriranje koji se često koristi
    - Pogledati

# OPACITY

- Ovaj property služi za definisanje vidljivosti elementa
  - Vrijednost ovog atributa (property-a) kreće se od 0 do 1
  - Što je niža vrijednost to je slabija vidljivost elementa koji ima ovaj property
  - Pogledajmo sledeća dva primjera
    - Primjer 1
    - Primjer 2
      - Isporbajte malo šta se dešava kad mijenjate marginu child elementa (p)

# FORME

- Sa CSS-om drastično možete da unaprijedite izgled forme
  - Npr. width, border, padding, boje
  - Takođe, možete da definišete stilove u zavisnosti od stanja input polja
  - Možete da dodate ikonicu
  - Animaciju
  - Možete da mijenjate na sličan način i textarea element
    - **resize: none** (da onemogućite resize-ovanje)
  - Select element
  - Button element



Pitanja