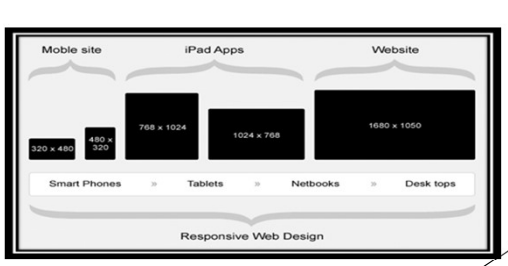
Prilagodljivi dizajn

(engl. *Responsive Web Design)*

Prilagodljivi dizajn

- Pojam:
  - 2010 Ethan Marcotte uvodi pojam: “Responsive Web Design - RWD”.
- Zasniva se na korišćenju tehnika:
  - **Medija upita**
  - **Skalabilnih slika**
  - **Fluidne mreže - Fluid grids**



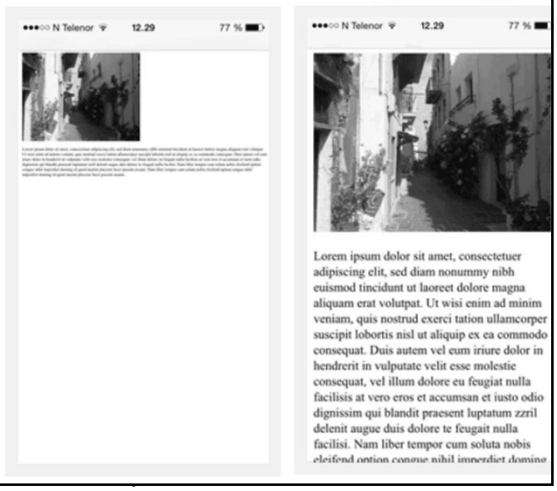
Viewport

- Viewport je vidljiva površina web stranice za korisnika.
- Kontrolom viewport-a kontroliše se prikaz web stranice na mobilnim uređajima. Bez kontrole viewport-a mobilni uređaj prikazuje web stranicu na isti način kao i desktop računari.
- Generalno, viewport je promenljive širine i može biti mali kod mobilnih uređaja i jako širok kod televizijskih ekrana ili bioskopskih platana.
- HTML5 daje mogućnost web dizajnerima da preuzmu kontrolu nad Viewport objektom koristeći <meta> tag.
- Uključivanje viewport objekta u svim web stranicama se izvodi ubacivanjem koda:
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`

`width=` – definiše širinu koju čitač vidi na širinu uređaja. (CSS pikseli)
`initial-scale=` – definiše vrednost početnog skaliranja (zoom-a)
`maximum-scale, minimim-scale` – definiše maksimalnu i minimalnu verednost za skaliranje

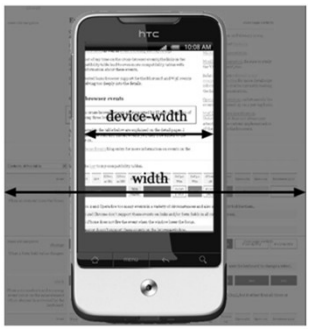
```
<meta name="viewport"
content="width=device-width,
initial-scale=1.0" />
<style>
    img {
        max-width: 100%;
        height: auto;
    }
</style>
```

- width=device-width – podešava širinu stranice čitača na osnovu širine ekrana uređaja tj. širina stranice biće promenljiva.
- initial-scale=1.0 – podešava početni zoom faktor kada se stranica prvi put učita u web čitač.
- Pogledati primere za html kod:
- <img src="slika.jpg" width="460" height="340">
- Ukoliko je neka veličina podešena reletivno na širinu ekrana, kao slika u drugom primeru, onda će se u slučaju postojanja definisanog viewport-a slika prilagoditi različitim ekranima.



Viewport – otvor za prikaz sadržaja na mob. uređaju

- U slučaju mobilnih uređaja ovo je prikazano na slikama



Veličina sadržaja i Viewport

- Skrolovanje:
- Smatra se da su korisnici naučeni da skroluju web stranice vertikalno, ali ne i horizontalno!
- Kaže se „ loše korisničko iskustvo“ ukoliko je web dizajn takav da od korisnika očekuje da skroluje horizontalno ili da vrši promenu zoom-a da bi video neki sadržaj.
- Zato poštujte principe:
- **1. Ne koristite fiksne široke elemente** - Na primer, ukoliko je neka slika prikazana šire nego što je viewport to znači da će korisnik morati da skroluje horizontalno.
- **2. Sadržaj ne sme da se oslanja na određenu širinu viewport-a da bi imali dobar izgled.**
- **3. Koristite CSS medija upite da bi ste primenili drugačiji stil za drugačije veličine ekrana.**

Nove relativne jedinice mere

Jedinica	Opis
vw	Predstavlja 1% od širine viewport-a
vh	Predstavlja 1% od visine viewport-a
vmin	Predstavlja 1% od manje dimenzije viewport-a
vmax	Predstavlja 1% od veće dimenzije viewport-a

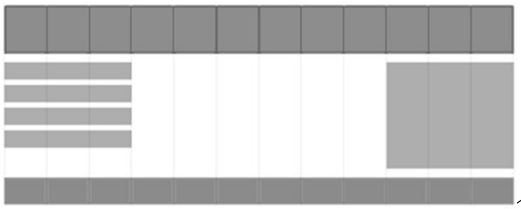
Organizacija sadržaja

- U prilagodljivom dizajnu jedini prihvatljiv način organizacije sadržaja je baziran na primeni relativne mere pre svega u odnosu na širini ekrana.
- Dakle širine kolona sadržaja treba da budu u procentima-% ili em-jedinicama.
- Tipično je da se definiše struktura dokumenta u zavisnosti od širine ekrana.

- Većina unapred pripremljenih okvira za projektovanje prilagođenih stranica postiže pojednostavljenje u projektovanju prethodnim definisanjem mrežaste strukture - grid.
- Mrežasta struktura se zasniva na unapred definisanim kolonama u koje se smeštaju elementi stranice.

Mrežasta organizacija sadržaja

- Ovo je način prilagodljivog dizajna koji se zasniva na mreži (engl. grid) koja se koristi pri definisanju rasporeda elemenata za različite širine prikaza.
- Za mrežu od 12 istih kolona jedan mogući raspored bi bio kao na slici u slučaju desktop prikaza. Svih 12 kolona zajedno čini 100% širine stranice.



Svojsvo box-sizing

- Obično mreže imaju zaista 12 kolona. Tako veliki broj kolona je neophodan kako bi se dobio što bolji dizajn. Neophodna je i potpuna kontrola cele širine prozora.
- Zato svi HTML elementi treba da imaju podešeno
- ```
  *{
      box-sizing: border-box;
  }
  ```
- Ovim svojstvom se definiše kako se računa širina i visina. Ovde se osigurava da su `pedding` i `border` uključeni u ukupnu širinu i visinu elemenata. (Standardni box model je: `box-sizing: content-box` tj. padding i border se dodaju na širinu/visinu.)
- U konkretnom slučaju sa slike iz prethodnog slajda, iskazano u procentima širina jedne kolone je: 100% / 12 = 8.33%.
- Zatim formiramo posebne klase za kolone u 12 širina, `class="col-" .`
- Napomena: Ovako formirano ime klase omogćava CSS pristup pomoću atributa koji ciljaju imena svojstva (`*= ^= $=`). Na primer:

```
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
```
```

```
[class*="col-"] {
    float: left;
    padding: 15px;
    border: 1px solid red;
}
```

- Svaki red podataka koristi jedan element koji sadrži nekoliko kolona različite širne, ali sume širina 12. Na primer:

```
<div class="col-3">
    <ul>
        <li>NRT</li>
        <li>RT</li>
        <li>EPO</li>
        <li>...</li>
    </ul>
</div>
<div class="col-9">
    <h1>Najbolji studijski programi</h1>
    <p>Na osnovu nepoznatih kriterijuma...</p>
    <p>Ipak NRT je naj.</p>
</div>
```

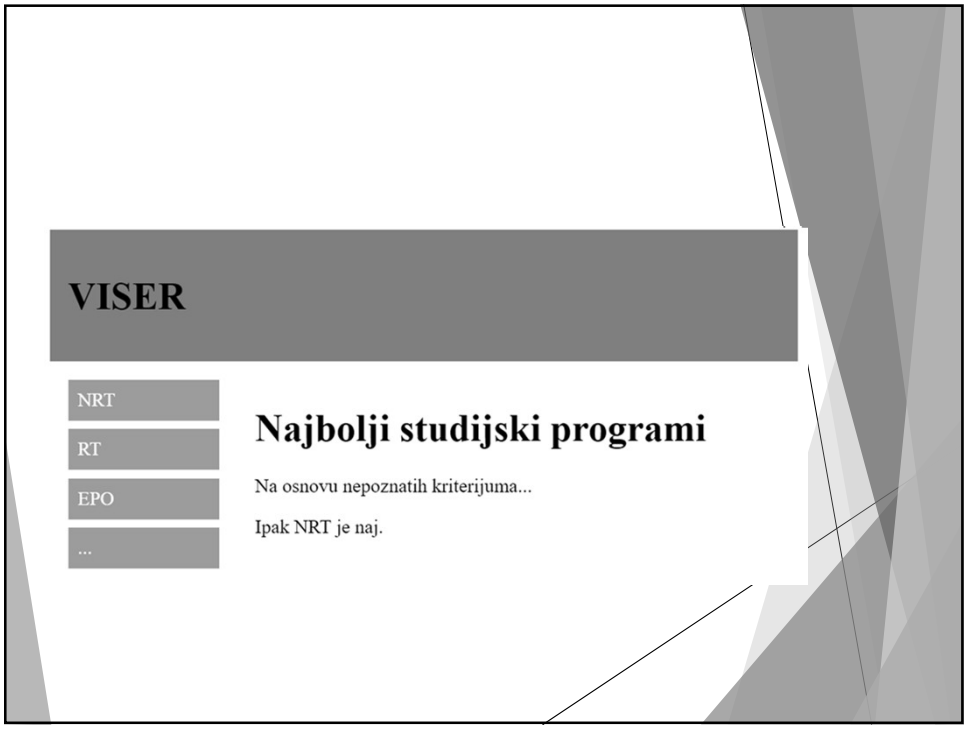
- VAŽNO!
- Sve kolone u redu imaju postavljeno stvojstvo float na vrednost left. Da bi se osigurao prelaz u novi red blokovskih elemenata ukida se plutanje nakon postavljanja reda.

```
.row::after {
    content: "";
    clear: both;
    display: block;
}
```

- Meni
- Nakon definisanja svojstava kolone i reda mogu se dodati posebne klase za sređivanje izgleda, menija na primer:

```
.menu ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}
.menu li {
    padding: 8px;
    margin-bottom: 7px;
    background-color :#33b5e5;
    color: #ffffff;
}
.menu li:hover {
    background-color: #0099cc;
}
```

```
<div class="col-3 menu">
    <ul>
        <li>NRT</li>
        <li>RT</li>
        <li>EPO</li>
        <li>...</li>
    </ul>
</div>
```



Medija upiti

(engl. Media queries – MQ)

Uvod

- Medijski upiiti (engl. Media queries – MQ) - CSS tehnika uvedena u CSS3
- Bazira se na korišćenju `@media` pravila. Sintaksa je:
- `@media = mediji and/or uslovZaOsobinuMedija`
-
```
<style>
   @media screen and (max-width: 500px) {
        body {
           background-color: yellow;
        }
   }
</style>
```

Tipovi

- Neki drugi tipovi medija su:
- `all` – svi mediji
- `print` – stranični medij u režimu prikaza štampača
- `speech` – snimanje glasa
- `tv` – televizija,...
- Tip medija može se koristiti pri definisanju stila ili pri povezivanju sa spoljnim fajlovima, na primer:

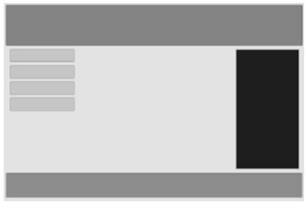
1. `<link href="stilovi.css" media="screen">`
2. `<style media="print">`

Osobine medija

- Postoji 13 različitih osobina koje se mogu testirati. Neke od njih su:
- `aspect-ratio` – odnos širine i visine uređaja
- `color` – broj bitova po boji komponente boje
- `resolution` – gustina piksela
- `orientation` – portrait ili landscape režim
- `height` – visina oblasti prikaza (po standardu treba izbegavati device-height – visina prikazane površi)
- `widht` – širina oblasti prikaza (po standardu treba izbegavati device-width – širina prikazane površi)
- `. . .`
- Svim osobinama može da se doda prefiks min- ili max- . Na primer:
- `@media all and (min-width: 350px) { . . . }`
  `@media all and (min-width: 350px) and (max-width: 750px) { . . . }`

Tačke preloma

- Skaliranje kolona sadržaja je dobro do određene veličine, ali u slučaju mobilnih telefona i u *portrait* prikazu svakako nije. Za takve prikaze bilo bi bolje da imamo drugačiju organizaciju sadržaja, u jednoj koloni.
- Medija upiti rešavaju ovaj problem postavljanjem graničnih vrednosti tj. tačaka preloma za koje se menja dizajn.



Tačka preloma u CSSu

```
/* Desktop: */
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}

@media only screen and
(max-width: 768px) {
    /* Mobile */
    [class*="col-"] {
        width: 100%;
    }
}
```

Tačka preloma

Novi pojam: *Mobile First*

- *Mobile First* – označava tehniku dizajniranja za mobilne uređaje pre dizajniranja za desktop ili neki drugi uređaj.
- To podrazumeva pristup i izmene u CSSu.

```
/* First mobile */
[class*="col-"] {
    width: 100%;
}
@media only screen and (min-width:
768px) {
    /* Then Desktop: */
    .col-1 {width: 8.33%;}
    .col-2 {width: 16.66%;}
    .col-3 {width: 25%;}
    .col-4 {width: 33.33%;}
    .col-5 {width: 41.66%;}
    .col-6 {width: 50%;}
    .col-7 {width: 58.33%;}
    .col-8 {width: 66.66%;}
    .col-9 {width: 75%;}
    .col-10 {width: 83.33%;}
    .col-11 {width: 91.66%;}
    .col-12 {width: 100%;}
}
```

Više tačaka preloma

- Može se dodati više tačaka preloma po potrebi.
- Razlog? Na primer između desktop dizajna i dizajna za mobilne telefone može postojati tablet dizajn.

```
/*Mobile: */
[class*="col-"] {
    width: 100%;
}
@media only screen and (min-width: 600px) {
    /* Tablets: */
    .col-m-1 {width: 8.33%;}
    .col-m-2 {width: 16.66%;}
    .col-m-3 {width: 25%;}
    .col-m-4 {width: 33.33%;}
    .col-m-5 {width: 41.66%;}
    .col-m-6 {width: 50%;}
    .col-m-7 {width: 58.33%;}
    .col-m-8 {width: 66.66%;}
    ****************
}
```

```
@media only screen and (min-width: 768px) {
    /* Desktop: */
    .col-1 {width: 8.33%;}
    .col-2 {width: 16.66%;}
    .col-3 {width: 25%;}
    .col-4 {width: 33.33%;}
    .col-5 {width: 41.66%;}
    .col-6 {width: 50%;}
    .col-7 {width: 58.33%;}
    .col-8 {width: 66.66%;}
    ******************
}
```

Odvojene kolekcije stilova

- Dizajneri često posežu za rešenjima u kojima posebno definišu stilove za posebne prikaze tj. tipove uređaja. Mogući nedostatak je ponavljanje istih stilova što se može donekle prevazići izdvajanjem zajedničkih stilova u posebnim zajedničkim dokumentima.
- Na primer, u fajlovima *small.css* i *large.css* su stilovi za određene ekrane.

```
<link rel="stylesheet" media="screen
                              and (min-width:200px)
                              and (max-width: 500px)" href="small.css">

<link rel="stylesheet" media="screen
                              and (min-width:501px)
                              and (max-width: 1100px)" href="large.css">
```

Napomena: Zbog nekih starijih čitača po nekada se koristi i *only screen*, ali i kao naglašavanje da se ne koristi neki drugi medijum osim *screen*.

Orijentacija ekrana

- Medija upiti mogu da prepoznaju tekuću orijentaciju ekrana. Moguće orijentacije su:
  - `Portrait`
  - `Landscape`
- Tako, u zavisnosti od orijentacije ekrana mogu da se dodele posebna svojstva.
- ```
  @media only screen and (orientation: landscape)
  {
      body {
          background-color: yellow;
      }
  }
  ```

Slike u prilagodljivom dizajnu

- Da bi slika u prilagodljivom dizajnu bila prilagođena širini viewporta neophodno je koristiti relativno podešavanje širine
- ```
  img {
      width: 100%;
  }
  ```
- Međutim, realan slučaj je da se koristi slika ne preterano velike rezolucije, recimo širine 480px što bi u slučaju desktop varijante prikaza dovolo do širenja slike do dimenzija prikaza odnosno istovremeno i do narušavanja kvaliteta prikaza. Zato se umesto toga koristi
- ```
  img {
      max-width: 100%;
  }
  ```
  ```

- Ako je slika dimenzije 1500px onda bi na mobilnim uređajima trebalo da bude skaliran dok na desktop uređajima da ne bude šira od 1500px. Dakle:

```
img {
     width: 100%;
     height: auto;
     max-width: 1500px;
}
```

Prilagođavanje pozadinske slike

- Pozadinske slike mogu takođe biti prilagodljive i skalirajuće.
- Postoji nekoliko različitih metoda za dobijanje efekta prilagodljivosti:
- 1. Ako je svojstvo background-size postavljeno na vrednost "contain", pozadinska slika će biti skalirana i prilagodiće se površini koja je sadrži. Slika će sačuvati *„aspect ratio“* (proporcionalnost između širine i visine slike):

- 2. Ako se svojstvo background-size postavi na vrednosti "100% 100%", pozadinska slika će biti razvučena preko cele klijentske oblasti:

- 3. Ako se svojstvo background-size postavi na vrednost "cover", pozadinska slika će biti skalirana preko cele površi kontejnera. Zapazite da vrednost "cover" čuva proporcionalnost širine i visine slike a da se zbog toga neki deo slike mora iseći:

Različite slike za različite uređaje

- Kao što je velika slika dobra za veliki ekran, mala je za mali. Da bi se izbeglo nepotrebno prenošenje velike slike za male ekrane može se definisati potpuno druga ili već obrađena manja slika.

```
body {
    background-image: url('pejzaz1_mobile.jpg');
}
@media only screen and (min-width: 400px) {
   body {
     background-image: url('pejzaz1_desktop.jpg');
   }
}
```

- Napomena:
- Može se koristiti medija upit min-device-width, umesto min-width, tj. umesto da se proverava širina web čitača da se proverava širina ekrana. U ovom slučaju se slika neće menjati sa širinom čitača:

```
@media only screen and (min-device-width: 400px) {
    body {
        background-image: url('pejzaz1_desktop.jpg');
    }
}
```

Video

- Video element ima svojstvo koje može da se prilagođava podešavanjem na 100% čime se prilagođava širini ekrana ili čitača, sa eventualnim ograničenjem koristeći max-width svojstvo.

```
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <style>
video {
    max-width: 100%;
}
    </style>
</head>
<body>
    <video width="600" controls>
        <source src="mo1.mp4" type="video/mp4">
    </video>
</body>
</html>
```

Uvod u fleksibilni model okvira

Flexbox

- Predstavlja još jedan novi način RWD uveden uz CSS3.
- Upotreba flexbox koncepta garantuje predvidivo prikazivanje elemenata u slučaju različitih uređaja za prikaz. Za neke primene ovo je napredniji model prikaza koji daje bolje karakteristike i veće mogućnosti u odnosu na blok model u kome se ne koriste plutajući elementi niti se za flex elemente margine sažimaju sa marginama kontejnera.
- Flexbox se sastoji od flex kontejnera i stavki.
- Kontejner se deklariše postavljanjme svojstva diplay na vrednost flex (za blok element) odnosno inline-flex (za inline elemente).
- Unutar jednog kontejnera smešta se jedna ili više stavki.
- Sve van flex kontejnera, ali takođe i unutar flex stavki se prikazuje na uobičajen način. Flexbox definiše kako se prikazuju flex stavke u kontejneru.
- Stavke se pozicioniraju duž jedne flex linije. Podražumevano u jednom kontejneru postoji samo jedna flex linija. Podrazumevano pozicioniranje ide od leve ka desnoj strani.

Pravac prostiranja stavki

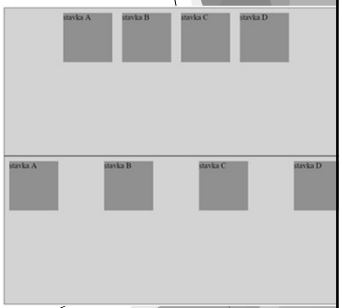
- Još uvek ovo svojstvo nije potpuno identično za sve čitače pa treba koristiti prefikse –moz, -webkit
- Definisanje flex linije se vrši postvljanjem svojstva directon , na primer:
- ```
  .flex-container {
      ...
      display:flex;
      flex-direction:row;
  }
  ```
- Podrazumevana vrednost je s leva na desno i odozgo na dole - **row**.
- Ostale moguće vrednosti su:
- •row-reverse – suprotno od podrazumevanog tj s desna na levo
- •**column** – stavke se prikazuju po kolonama tj vertikalno
- •column-reverse – po kolonama ali u suprotnom redosledu

Primer1

Raspodela praznog prostora

Linjska raspodela

- Važno je upravljati viškom belog prostora u fleksibilnom rasporedu kada stavke ne zauzmu ukupan unutrašnji prostor. Za to se koristi svojstvo **justify-content**. Sledeće vrednosti se koriste:
- flex-start – Podrazumevano. Stavke su pozicionirane od početka kontejnera.
- flex-end – Stavke su pozicionirne na kraju kontejnera
- center – Stavke su pozicionirane oko centra kontejnera
- space-between – Stavke su pozicionirane sa prostorom između linija
- space-around – Stavke su pozicionirane sa prostorom ispred, između i posle linija.


  ```

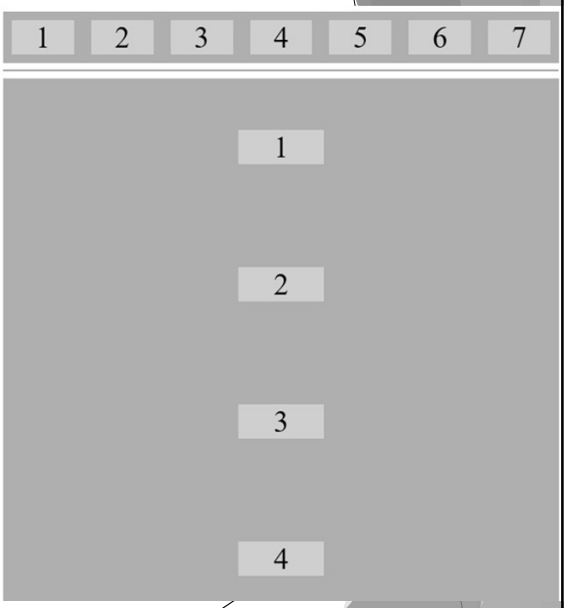
Raspodela praznog prostora po drugoj osi

- Raspodela praznog prostora po drugoj osi, normalna u odnosu na linijsku, definiše se pomoću svojstva align-items u slučaju kada stavke nisu iskoristile ukupan raspoloživ prostor. Odgovarajuće vrednosti su:
- stretch – Podrazumevana vrednost. Stavke se razvlače i tako popunjavaju ukupan kontejner.
- flex-start – Pozicionirane od vrha.
- flex-end – Pozicionirane na dno kontejnera.
- center – Stavke su pozicionirane na sredini.
- baseline – Stavke su pozicionirana na baznoj liniji kontejnera.

```
<head>
  <style>
    .flex-container {
            display: flex;
            flex-wrap: nowrap;
            background-color: darkseagreen;
            justify-content: space-around;

        }
        .flex-containerVer {
            flex-direction: column;
            align-items: center;
            height: 40em;
        }

        .flex-container > div {
            background-color: lightblue;
            width: 100px;
            margin: 10px;
            text-align: center;
            font-size: 35px;
        }
  </style>
</head>
<body>
  <content class="flex-container">
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
    <div>5</div>
    <div>6</div>
    <div>7</div>
  </content>
  <hr />
  <content class="flex-container flex-containerVer">
    <div>1</div>
    <div>2</div>
```



Prelom sadržaja

- Svojstvo flex-wrap definiše da li se ili ne stavke u flex kontejneru prelamaju u slučaju da nema dovoljno praznog prostora u jednoj liniji. Moguće vrednosti su sledeće:
- nowrap – Podrazumevano.
- wrap – Prelom postoji ako sadržaj ne može da stane
- wrap-reverse – Prelom ali u obrnutom redosledu

Raspodela prostora po stavkama

- Svojstvom flex specificira dužinu flex stavke, relativno u odnosu na ostatatak preostalih stavki u istom kontejneru.
- U narednom primeru prvi div dobija 3 od ukupno 6 vrednosti (50%), drugi, treći i četvrti dobija 1 od 6.
- Ova vrednost definiše proporciju sa kojom se stavke povećavaju.


```
#prvi1
{
    flex:3;
}
#prvi2
{
    flex:1;
}
#prvi3
{
    flex:1;
}
#prvi4
{
    flex:1;
}
```

Fleksibilna širina stavki i redosled

- Kao što se definiše širina stavki u fleksibilnom okviru, odnosno način promene, isto tako se može definisati i način tj srazmernost pri proširenju/skupljanju sadržaja. Ta vrednost se može zadati kao prva/druga vrednost, a kao treća se zadaje početna veličina u odnosu na koju se vrši računanje. Na primer: flex: 1 3 150px; (Podrazumevano je: 0 1 auto)

- Redosled se može posebno definisati svojstvom order. Niža vrednost ovog svojstva znači prvenstvo u prikazu. Ako dve stavke imaju istu vrednost onda se primenjuje podrazumevani redeosled.

```
#prvi1
{
    flex: 1 3 150px;
}
#prvi2
{
    flex: 1 1 150px;
}
#prvi3
{
    flex: 1 1 150px;
}
#prvi4
{
    flex: 1 1 150px;
}
```

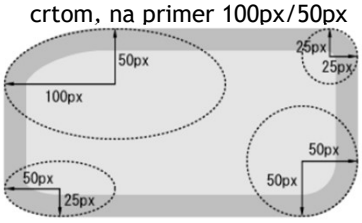
Nova vizuelna svojstva u CSS3

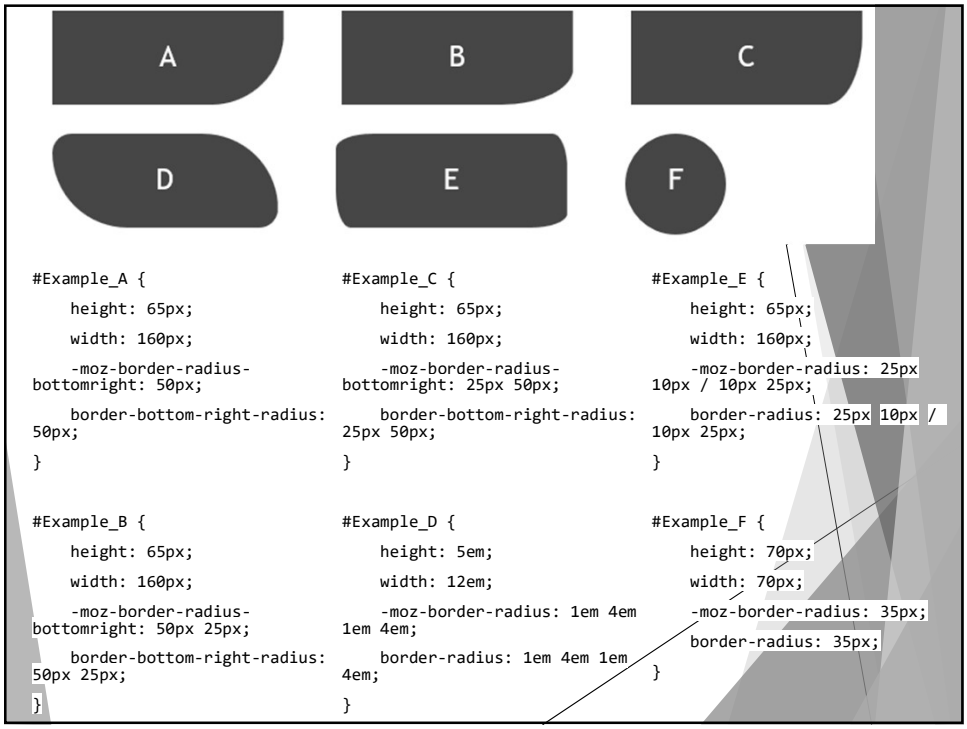
Vendor Prefiksi

- Neka nova CSS3 svojstva još uvek nisu primenljiva sa istom sintaksom u svim web čitačima. Zato se, za takva svojstva, koriste specifični prefiksi:
  - Mozilla Browsers (Firefox)
    -moz
  - Webkit Browsers (Safari, Chrome)
    -webkit
  - Opera
    -o
  - Internet Explorer
    -ms
    <!--[if IE]> <![endif]-->
- Neki sajtovi koji mogu pomoći u razvoju u zavisnosti od primenjenog web čitača:
  http://www.caniuse.com/ - provera funkcionalnosti za pojedine web čitače
  http://www.css3.info/selectors-test/ - testiranje selektora za tekući web čitač

border-radius

- Svojstvo *border-radius* je skraćeno svojstvo za 4 pojedinčna svojstva ivice *border-*-radius*.
  (* - top-left, top-right, bottom-right, bottom-left). Ove pojedinačne vrednosti su malo drugačije za Mozilla pretraživač.
- border-radius: *1-4 dužine*|*%* / *1-4 dužine*|*%*|initial|inherit;
- Ako se vrednost zadaje sa dve vredsnosti razdvojene kosom crtom, na primer 100px/50px onda je prva vrednost horizontalni poluprečnika a druga vrednost vertikalni poluprečnik.






```
#Example_A {
    height: 65px;
    width: 160px;
    -moz-border-radius-bottomright: 50px;
    border-bottom-right-radius: 50px;
}

#Example_B {
    height: 65px;
    width: 160px;
    -moz-border-radius-bottomright: 50px 25px;
    border-bottom-right-radius: 50px 25px;
}
```

```
#Example_C {
    height: 65px;
    width: 160px;
    -moz-border-radius-bottomright: 25px 50px;
    border-bottom-right-radius: 25px 50px;
}

#Example_D {
    height: 5em;
    width: 12em;
    -moz-border-radius: 1em 4em 1em 4em;
    border-radius: 1em 4em 1em 4em;
}
```

```
#Example_E {
    height: 65px;
    width: 160px;
    -moz-border-radius: 25px 10px / 10px 25px;
    border-radius: 25px 10px / 10px 25px;
}

#Example_F {
    height: 70px;
    width: 70px;
    -moz-border-radius: 35px;
    border-radius: 35px;
}
```

Box-shadow

- box-shadow: #rrggbb xof yof rza
- #rrggbb ili rgb(r,g,b) - boja
- xof, yof – pomeraj senke u odnosu na objekat
- rza – rastojanje zamagljenja

```
width: 200px;
height: 100px;
background-color: lightblue;
box-shadow: 10px 10px 5px #999;
```

```
width: 200px;
height: 100px;
background-color: lightblue;
box-shadow: 5px -5px 5px
```

```
width: 200px;
height: 100px;
background-color: lightblue;
box-shadow: inset rgb(0,50,100) -5px -5px 15px,
inset white 5px 5px 15px
```

Text-shadow

- Potpuno analogno sa box-shadow samo sa funkcijom da se obezbedi senčenje teksta a ne okvira.

```
<style>
h1 {
    text-shadow: #FF0000 2px 2px 8px;
}
</style>
```

Efekat senčenja sa zamagljivanjem

Gradijent

- Gradijent ili tonski prelaz je način da se bojom koja prelazi iz jedne u drugu popuni neki objekat. Korišćenjem ovog svojstva, može se napraviti gradijent od najmanje dve boje. Inače, broj boja definiše korisnik.
- Nekoliko online generatora tonskih prelaza je dato na sledećim linkovima:
  - http://www.colorzilla.com/gradient-editor/,
  - http://css3gen.com/gradient-generator/ .
- Postoje dve vrste gradijenta:
  - linearni i
  - radijalni.

- Sintaksa za definisanje gradijenta je:
- `background:linear-gradient(pravac, clr1, clr2, ...);`

- Primeri:
- `background: linear-gradient(to bottom, green, blue);`
- Predstavlja promenu od zelene ka plavoj počev od vrha prema dnu.
- Ako želimo da obezbedimo istu promenu, ali pod uglom od 70 stepeni, CSS kod bi bio:
- `background: linear-gradient(70deg, green, blue);`
- Ukoliko želimo da dobijemo prelaz po svim pravcima počev od sredine elementa koristi se radijalni gradijent.
- `radial-gradient(green, blue, rgba(200,200,200,0.2));`

Više kolona teksta

- CSS3 nudi jednostavan prikaz sadržaja u više kolona, tipično u slučaju prikaza na širioj površi.
- Na ovaj način se dobija načitljivosti.
- Za chrome, opera i firefox potrebno je koristiti prefikse.
- Svojstva:
- `column-count` – broj kolona
- `column-gap` – prostor između kolona
- Definisanje stila između kolona:
  - `column-rule-style` – stilovi između kolona
  - `column-rule-width` – širina korišćenog stila
  - `column-rule-color` – boja korišćenog stila

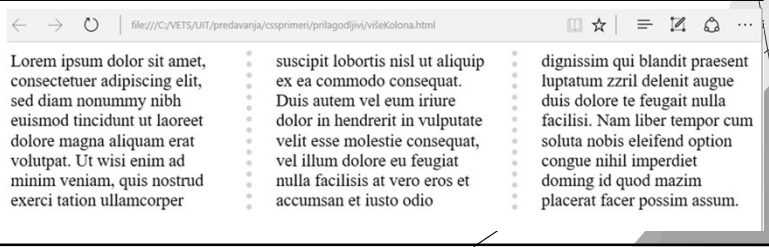
```
.triKolone {
-webkit-column-count: 3; /* Chr, Saf, Op */
-moz-column-count: 3; /* Firefox */
column-count: 3;

-webkit-column-gap: 50px; /* Chr, Saf, Op */
-moz-column-gap: 50px; /* Firefox */
column-gap: 50px;

-webkit-column-rule-style: dotted; /* Chr, Saf, Op */
-moz-column-rule-style: dotted; /* Firefox */
column-rule-style: dotted;

-webkit-column-rule-width: 5px; /* Chr, Saf, Op */
-moz-column-rule-width: 5px; /* Firefox */
column-rule-width: 5px;

-webkit-column-rule-color: lightgreen; /* Chr, Saf */
-moz-column-rule-color: lightgreen; /* Firefox */
column-rule-color: lightgreen;
}
```



Transformacije

Transform

- Svojstvo transform izvršava 2D ili 3D transformacije na nekom elementu. Ovim svojsvom se može izvršiti rotacija, skaliranje, pomeranje, ukošavanje.
- Sintaksa je:
- transform: none|*transform-functions*|initial|inherit;
- Na primer
- ```
  #myDIV {
      border: solid 1px;
      width: 200px;
      height: 200px;
      background: red;
      transform:skewX(-30deg);
  }
  ```
- Vrši transformaciju elementa tako što izvodi ukošenje za -30 stepeni oko X ose.

Funkcije transformacije

- `matrix` (x,x,x,x,x,x), matrix3d(x,x,x,x,x,x,x,x,x,x)
- `translate` (*x*,*y*), translate3d(*x*,*y*,*z*), translateX(*x*), translateY(*y*), translateZ(*z*)
- `scale` (*x*,*y*), scale3d(*x*,*y*,*z*), scaleX(*x*), scaleY(*y*), scaleZ(*z*)
- `rotate` (*ugao*), rotate3d(*x*,*y*,*z*, *ugao*), rotateX(*ugao*), rotateY(*ugao*), rotateZ(*ugao*)
- `skew` (*x- ugao*,*y- ugao*), skewX(*ugao*), skewX(*ugao*)
- `perspective` (*n*) – Određuje pogled za 3D transformisani element
  ```

CSS

```
div {
    width: 100px;
    height: 100px;
    position:absolute;
    left:50px;
}
.normal{
    border: solid 2px blue;
    background:white;
}
.transform{
    border: solid 2px red;
    background:lightblue;
}
```

```
.red1{
    top:0px;
}
 .red2{
    top:110px;
 }
 .red3{
    top:220px;
 }
 .translate{
    transform:translate(20px);
 }
.skew {
    transform:skewX(-30deg);
 }
.rotate {
    transform:rotate(45deg);
 }
```

CSS

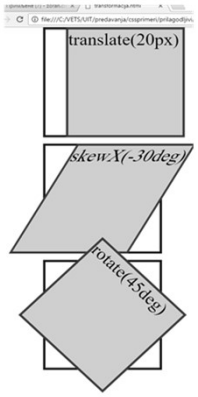
```
div {
    width: 100px;
    height: 100px;
    position:absolute;
    left:50px;
}
.normal{
    border: solid 2px blue;
    background:white;
}
.transform{
    border: solid 2px red;
    background:lightblue;
}
```

```
.red1{
    top:0px;
}
 .red2{
    top:110px;
 }
 .red3{
    top:220px;
 }
 .translate{

transform:translate(20px);
 }
.skew {
    transform:skewX(-30deg);
 }
.rotate {
    transform:rotate(45deg);
 }
```

HTML

```
<div class="normal red1">
</div>
<div class="transform red1
translate">
    translate(20px)
</div>
<div class="normal red2">
</div>
<div class="transform red2 skew">
    skewX(-30deg)
</div>
<div class="normal red3">
</div>
<div class="transform red3 rotate">
    rotate(45deg)
</div>
```



Animacije

Animacije - 1

- Animacija predstavlja postepeno menjanje stila od jednog ka drugom.
- Može se menjati veći broj CSS svojstava i proizvoljan broj puta. Stilovi elemenata koji će se pojavljivati u animaciji u određenim trenucima nazivaju se ključnim – *engl. „Keyframes“*.
- Da bi se koristile CSS3 animacije potrebno je prvo definisati ključne frejmove za animaciju.
- Sintaksa je:
- `@keyframes naziv {vremenskaOznaka {cssStil;}}`
- `naziv` – naziv animacije,
- `vremenskaOznaka` – pozicija frejma u vremenu animacije iskazana u procentima 0-100% ili ključnim rečima from (0%) odnosno to (100%),
- `cssStil` – CSS stil u definisanom trenutku.

Animacije - definisanje

- Dodatni opis animacije izvodi se svojstvima:
- `animation-delay` – vrednost ovog svojstva definiše odlaganje starta animacije u sekundama,
- `animation-iteration-count` – definiše broj ponavljanja animacije, ako je vrednost infinite onda se animacija neprekidno ponavlja,
- `animation-direction` –ukoliko ima vrednost reverse definiše suprotan tok animacije. Podrazumevana vrednost je normal. Moguća je naizmenična promena ove dve vrednosti pomoću alternate odnosno alternate-reverse. Ako je vrednost alternate onda se ponavlja *normal* pa *reverse*.
- `animation-timing-function` - vrednost ovog svojstva definiše brzinu i krivu promena po vremenu. Moguće vrednosti su:
  - `ease` – lagani start, sredina je brza i lagani kraj, ovo je podrazumevano svojstvo,
  - `linear` – ista brzina od početka do kraja,
  - `ease-in` – spori start,
  - `ease-out` – spori kraj,
  - `ease-in-out` – spori start i kraj,
  - `cubic-bezier(n,n,n,n)` – definisanje sopstvene krive animacije u vidu *cubic-bezier* funkcije.

Primer 1

```
div {
    width: 500px;
    height: 50px;
    background-color: red;
    animation-name: srb;
    animation-duration: 5s;
}
@keyframes srb {
    from {background-color: white;}
    50% {background-color: blue;}
    to {background-color: red;}
}
```

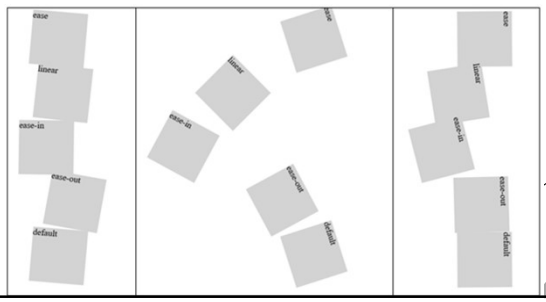
Primer 1

CSS:

```
div {
    width: 100px;
    height: 100px;
    background-color: lightgreen;
    position: relative;
    animation: anim 5s infinite;
}
#div1 {animation-timing-function: ease;}
#div2 {animation-timing-function: linear;}
#div3 {animation-timing-function: ease-in;}
#div4 {animation-timing-function: ease-out;}
@keyframes anim {
    from {left: 0px;}
    to {left: 500px;
    transform:rotate(90deg);
}
```

HTML

```
<div id="div1">ease</div>
<div id="div2">linear</div>
<div id="div3">ease-in</div>
<div id="div4">ease-out</div>
<div id="div5">default</div>
```



***Tranzicije

Tranzicije - 1

- Omogućava jednostavnu promenu jednog ili više svojstava CSSa od jedne vrednosti do druge – slično animacijama, ali bez ključnih frejmova i pri promeni svojstva.
- Da bi se kreirao neki efekat u vidu tranzicije potrebno je definisati:

1. CSS svojstvo kojim se želi postići efekat tranzicije.
2. Trajanje efekta tranzicije.

- Ukoliko trajanje nije definisano, podrazumevana vrednost je 0.

Tranzicije - 2

- transition – Skraćeno svojstvo
  - transition-property - naziv svojstva za tranziciju
  - transition-duration - vreme tranzicije u sekundama ili milisekundama
  - transition-timing-function - naziv funkcije promene brzine tranzicije
  - transition-delay - vreme za koje kreće tranzicija
- Primer:

  http://css3.bradshawenterprises.com/transitions/

- Na primer:
- ```
  div {
      border: solid 1px;
      width: 600px;
      height: 100px;
      background: red;
      -webkit-transition: width 1s;
      transition: width 1s;
  }
  ```
- definiše promenu svojstva width koja se dešava pošto se dogodi zahtev da se vrednost promeni. Na početku je 100px, a ako se zahteva promena za pozicioniranje miša iznad tada se događa tranzicija.
- ```
  div:hover {
      width: 300px;
  }
  ```

Vremenska funkcija animacije

- Svojstvo transition-timing-function definiše brzinu i način promene svojstva u tranziciji. Moguće vrednosti su:
- ease – spori start, brzo i spori kraj
- linear – ista brzina sve vreme tranzicije
- ease-in – spori start
- ease-out – spori kraj
- ease-in-out – spori start i kraj
- cubic-bezier – definisanje sopstvene funkcije

```
div {
  width:200px;
  height: 100px;
  background:transparent;
  transition:background 2s,
             width 3s,
             transform 2s;
}
```

```
div:hover {
  background: green;
  width:100px;
 transform:rotate(90deg);
}
```