



Cascade Style Sheet 4

dr Suzana Marković, dipl.ing. el.
suzana.markovic@bpa.edu.rs



CSS flexbox

- 4 načina za struktuiranje veb stranice:
 - **Blok** (npr. div), za sekcije na veb stranici
 - **Inline**, za struktuiranje teksta
 - **Table**, dvodimenzionalne tabele podataka
 - **Position**, za definisanje eksplicitne pozicije elementa.
- **Fleksibilan box** raspored omogućava izradu fleksibilne strukture bez korišćenja svojstva **float** i **position**.



CSS flexbox

- Flexbox je jednodimenzionalni način rasporeda postavljanja stavki u redovima ili kolonama.
- Elementi se šire kako bi ispunili dodatni prostor i se smanjuju da bi se uklapali u manje prostore.



Raspoređivanje elemenata flex mehanizmom

- Mehanizam se koristi za raspoređivanje blok elemenata.
- Postoje 2 tipa elemenata:
 - Element roditelj (container)
 - Element dete (item)

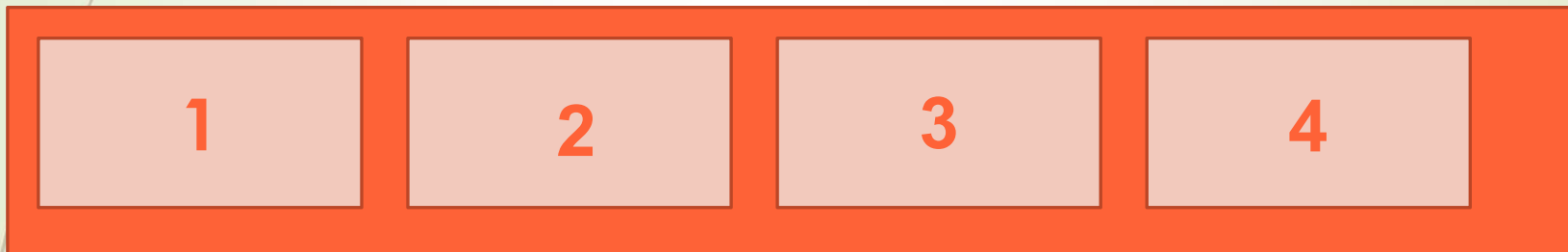


Raspoređivanje elemenata flex mehanizmom

- Element roditelj:
 - može da sadrži više elemenata dece;
 - Određuje način raspoređivanja elemenata dece
- Elementi deca određuju:
 - Koliko im je prostora potrebno
 - Na kojoj poziciji unutar roditelja se prikazuju
 - Na koji način su raspoređeni u odnosu na druge.

Flexbox elementi

➔ Flexbox kontejner (narandžasti postor):



```
<div class="flex-kontejner">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
  <div>3</div>  
</div>
```

CSS

```
.flex-kontejner {  
  display: flex;  
  background-color:#db4423;  
}
```

```
.flex- kontejner > div { background-color:  
#fce5e0; margin: 10px; padding: 20px;  
font-size: 30px;}
```

Primer

```
<!doctype html>
<head>
  <title>Flexbox</title>
  <style>
    .flex-kontejner {
      display: flex;
      background-color: #db4423;
    }
    .dete {
      background-color: #fce5e0;
      margin: 10px;
      padding: 20px;
      font-size: 30px;
      color: #db4423;
    }
  </style>
</head>
<body>
  <div class="flex-kontejner">
    <div class="dete">1</div>
    <div class="dete">2</div>
    <div class="dete">3</div>
    <div class="dete">4</div>
  </div>
</body>
</html>
```

Apps | ma petite cyjuisine... | Online kurs digitalni...



Ako elementu dete damo svojstvo **flex-grow:1** ili samo **flex:1** sva četiri boxa će se proširiti i zauzeti ceo div kontejner.



Primer

```
<head>
  <title>Flexbox</title>
  <style>
    .flex-kontejner {
      display: flex;
      background-color: #db4423;
    }

    .dete {
      background-color: #fce5e0;
      margin: 10px;
      padding: 20px;
      font-size: 30px;
      color: #db4423;
      flex-grow: 1;
    }

    #treći {
      flex-grow: 2;
    }
  </style>
</head>
<body>
  <div class="flex-kontejner">
    <div class="dete" id="prvi">1</div>
    <div class="dete" id="drugi">2</div>
    <div class="dete" id="treći">3</div>
    <div class="dete" id="cetvrti">4</div>
  </div>
</body>
```



$$1+1+2+1=5$$

1. 2. i 4. element zauzimaju po 1/5 prostora dok 3. element zauzima 2/5 prostora.

Smer ređanja elemenata dece unutar flexboxa

- ➔ Smer ređanja se određuje preko **flex-direction** svojstva kontejnera sa vrednostima:
 - ➔ row – podrazumevana vrednost
 - ➔ row-reverse
 - ➔ column
 - ➔ column-reverse

```
<title>Flexbox</title>
style>
  .flex-kontejner {
    display: flex;
    background-color: #db4423;
    flex-direction: row-reverse;
  }
```



Flex-direction: column

```
<head>
  <title>Flexbox</title>
  <style>
    .flex-kontejner {
      display: flex;
      background-color: #db4423;
      flex-direction: column;
      height: 300px;
    }

    .dete {
      background-color: #fce5e0;
      margin: 5px;
      padding: 2px;
      font-size: 30px;
      color: #db4423;
      flex-grow: 1;
    }

    #treci {
      flex-grow: 2;
    }
  </style>
</head>
```



Ako je visina elemenata dece veća od 300px oni će biti jednaki, a ako je manja oni će zauzeti prostor 1/5, 1/5, 2/5, 1/5.



Poravnanje elemenata po horizontali

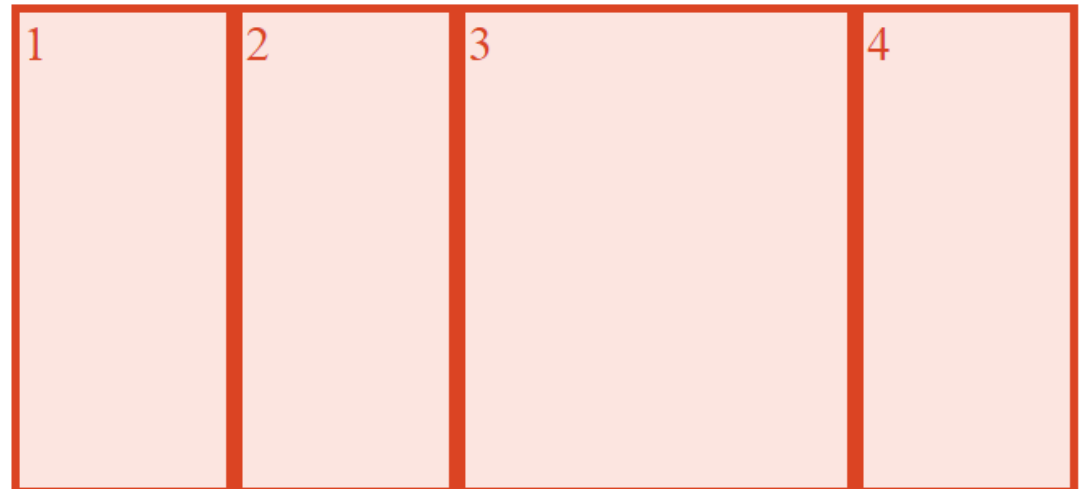
- Koristi se svojstvo **align-items** sa vrednostima:
 - flex-start
 - flex-end
 - center
 - baseline
 - stretch – podrazumevana vrednost.

Primer

```
head>  
<title>Flexbox</title>  
<style>  
  .flex-kontejner {  
    display: flex;  
    background-color: #db4423;  
    flex-direction: row;  
    height: 300px;  
    align-items: flex-start;  
  }  
  .dete {  
    background-color: #fce5e0;  
    margin: 5px;  
    padding: 2px;  
    font-size: 30px;  
    color: #db4423;  
    flex-grow: 1;}  
  #trece {  
    flex-grow: 2;
```



```
.flex-kontejner {  
  display: flex;  
  background-color: #db4423;  
  flex-direction: row;  
  height: 300px;  
  align-items: stretch;  
}  
.dete {  
  background-color: #fce5e0;  
  margin: 5px;  
  padding: 2px;  
  font-size: 30px;  
  color: #db4423;  
  flex-grow: 1;}  
#trece {  
  flex-grow: 2;
```



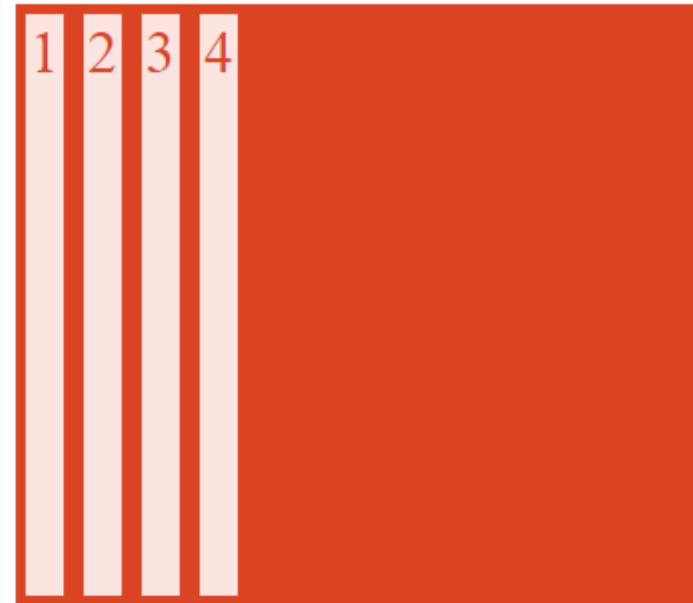


Raspoređivanje prostora između elemenata

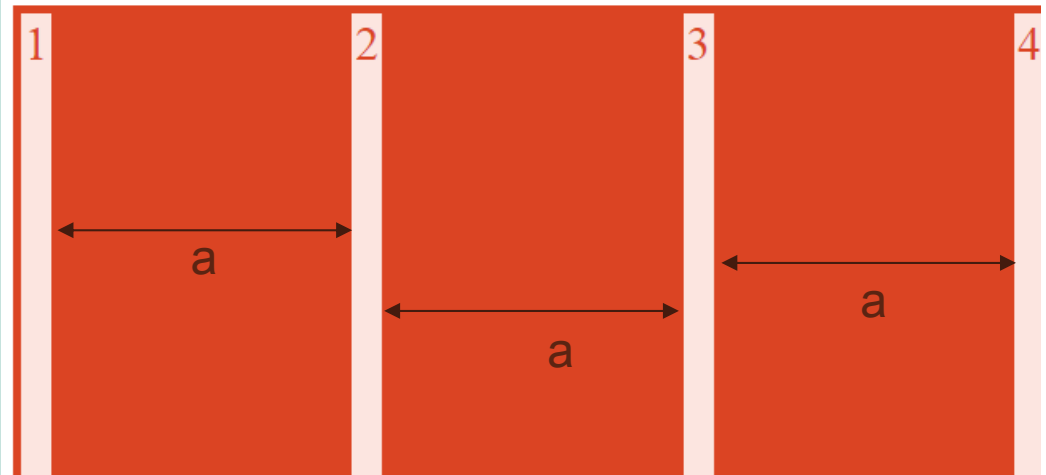
- Kada se ne postavi flex-grow elementima deci, njihovo raspoređivanje može da se izvrši pomoću **justify-content** sa vrednostima:
 - flex-start – podrazumevana vrednost
 - flex-end
 - center
 - space-between
 - space-around
 - space-evenly

Primer

```
.flex-kontejner {  
    display: flex;  
    background-color: #db4423;  
    flex-direction: row;  
    height: 300px;  
    align-items: stretch;  
}  
  
.dete {  
    background-color: #fce5e0;  
    margin: 5px;  
    padding: 2px;  
    font-size: 30px;  
    color: #db4423;  
    <!-- flex-grow:1; -->  
  
<!-- #trece { -->  
    <!-- flex-grow:2; -->  
}
```



```
.flex-kontejner {  
    display: flex;  
    background-color: #db4423;  
    flex-direction: row;  
    height: 300px;  
    align-items: stretch;  
    justify-content: space-between;  
}  
  
.dete {  
    background-color: #fce5e0;  
    margin: 5px;  
    padding: 2px;  
    font-size: 30px;  
    color: #db4423;  
    <!-- flex-grow:1; -->  
  
<!-- #trece { -->
```



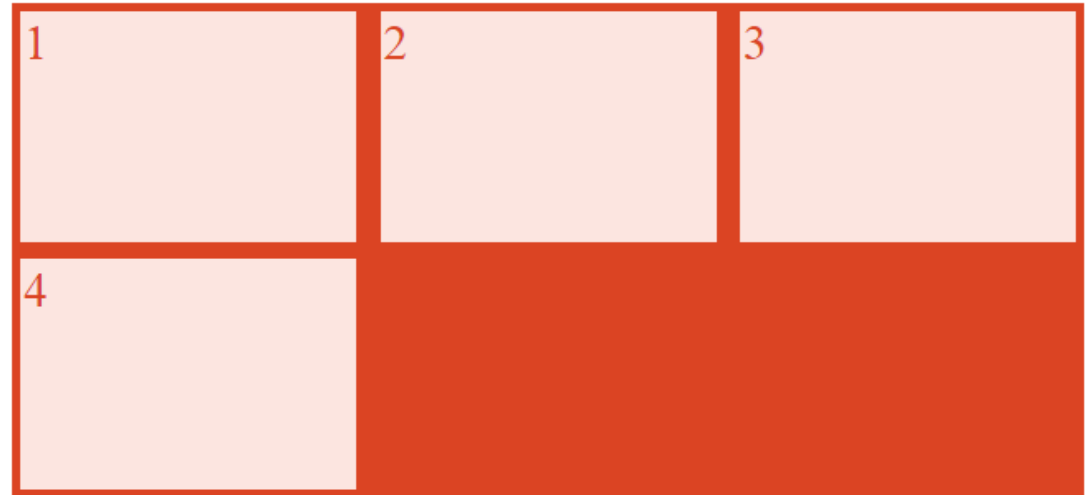


Raspoređivanje u više redova

- Koristi se kada se daju dimenzije elementima deci, pa ceo sadržaj ne može da se smesti unutar kontejnera.
- Koristi se opcija **flex-wrap** sa vrednostima:
 - nowrap – podrazumevana vrednost
 - wrap
 - wrap – reverse

Primer

```
.flex-kontejner {  
  display: flex;  
  background-color: #db4423;  
  flex-direction: row;  
  height: 300px;  
  align-items: stretch;  
  justify-content: space-between;  
  flex-wrap: wrap;  
}  
  
.dete {  
  background-color: #fce5e0;  
  margin: 5px;  
  padding: 2px;  
  font-size: 30px;  
  color: #db4423;  
  width: 200px;  
}
```





Raspoređivanje elemenata po vertikali

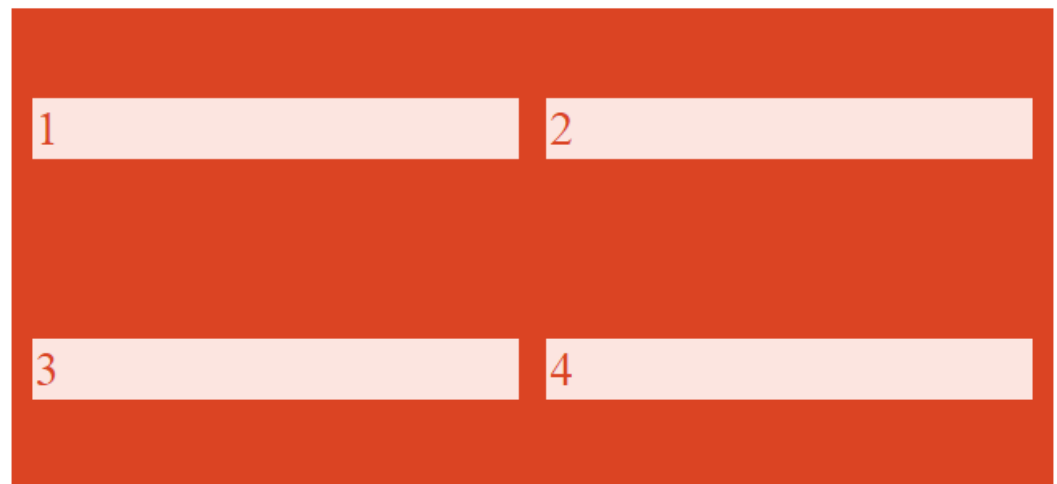
- Koristi se opcija **align-content** sa vrednostima:
 - flex-start
 - flex-end
 - center
 - space-between
 - space-around
 - space-evenly
 - stretch – podrazumevana vrednost

Primer

```
.flex-kontejner {  
  display: flex;  
  background-color: #db4423;  
  flex-direction: row;  
  height: 300px;  
  align-items: stretch;  
  justify-content: space-around;  
  flex-wrap: wrap;  
  align-content: flex-start;  
}  
  
.dete {  
  background-color: #fce5e0;  
  margin: 5px;  
  padding: 2px;  
  font-size: 30px;  
  color: #db4423;  
  width: 300px;  
}
```



```
.flex-kontejner {  
  display: flex;  
  background-color: #db4423;  
  flex-direction: row;  
  height: 300px;  
  align-items: stretch;  
  justify-content: space-around;  
  flex-wrap: wrap;  
  align-content: space-around;  
}  
  
.dete {  
  background-color: #fce5e0;  
  margin: 5px;  
  padding: 2px;  
  font-size: 30px;  
  color: #db4423;  
  width: 300px;  
}
```



Promena redosleda dece bez promene HTML koda

- ➔ Redosled se setuje pomoću vrednosti **order**.
- ➔ Podrazumevano je ta vrednost jednaka 0.

```
.flex-kontejner {  
  display: flex;  
  background-color: #db4423;  
  flex-direction: row;  
  height: 300px;  
  align-items: stretch;  
  justify-content: space-around;  
  flex-wrap: wrap;  
  align-content: space-around;  
}  
  
.dete {  
  background-color: #fce5e0;  
  margin: 5px;  
  padding: 2px;  
  font-size: 30px;  
  color: #db4423;  
  width: 300px;  
  
  <!-- flex-grow:1; -->  
  
#treci {  
  order: -1;  
}
```





Animacije


- Dva glavna svojstva koja omogućavaju animaciju iz CSS-a:
- **Transition** – koristi se da *ublaži* naglu promenu vrednosti izabranog CSS svojstva u određenom vremenskom periodu, tako što *postepeno* menja vrednosti CSS svojstva.
- **Animation** - zasniva se na promeni svojstva nekog elementa u toku vremena, ali *dozvoljava malo bolju kontrolu* i ima dodatne druge specifičnosti.

Koja svojstva animirati?

- Spisak svojstava koji se mogu animirati nalazi se na linku: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_animated_properties
- Današnji browser-i obezbeđuju hardversko ubrzanje određenih osobina za bolje performanse pri renderingu.
- Hardversko ubrzanje (eng. *Hardware acceleration*) znači da će " *Graphics Processing Unit*" (GPU) pomoći u renderovanju stranice tako što će umesto procesora obavljati neke od težih zadataka.

Svojstva za animaciju

- Svojstva "*transform*" i "*opacity*" su u prednosti kada je u pitanju animacija, jer im GPU pomaže pri izvršavanju, pa ih treba koristiti kad god je to moguće:
 - *opacity*
 - *transform: translate()*
 - *transform: rotate()*
 - *transform: scale()*
- Svojstva koja menjaju geometriju stranice (layout), jesu skupa svojstva jer promenom jednog elementa, često je potrebno da browser ponovo preračuna geometriju svih drugih elemenata.



Razlike između "transition" i "animation"

- U svojstvu "*transition*" se animacija izvršava samo na osnovu početnih i krajnjih vrednosti nekog CSS svojstva, dok kod svojstva "*animation*" imamo mogućnost da kroz **@keyframe** definišemo neograničen broj međuvrednosti.
- Svojstvo "*transition*" izvodi animaciju samo kao reakciju na promenu CSS svojstva koje se prati (npr. hover element). Animacije sa svojstvom "*animation*" ne zahtevaju eksplicitno aktiviranje (iako je i to moguće), pa mogu automatski započeti reprodukciju odmah po učitavanju.

Razlike između "transition" i "animation"

- **Looping** - svojstvo "*transition*" može da se pokrene samo jednom (za iste vrednosti), a svojstvo "*animation*" može da definiše koliko puta želimo da se izvrši ista animacija (uključujući i infinite).
- **Odloženo pokretanje** - Svojstvo "*animation*" za razliku od "*transition*" može da odloži animacije.
- **Transition** – za jednostavne animacije

Transform svojstvo

- **Ne vrši nikakvu animaciju**, ali se često koristi uz prethodno pomenuta CSS svojstva koja su zadužena za animaciju (transition/animation).
 - Transliranje duž x-ose: `transform: translate(12px, 50%);`
 - Rotiranje: `transform: rotate(10deg);`
 - Skaliranje po veličini: `transform: scale(2, 0.5);`
 - Košenje: `transform: skew(30deg, 20deg);`

Tranzicije i transformacija 1

```
.box{
```

```
width:150px;  
padding:15px;  
margin:20px auto;  
text-align:center;
```

```
.box:hover{
```

```
transform: scale(2);
```

```
.box1 {
```

```
background-color: aqua;
```

```
.box2 {
```

```
background-color: lime;  
transition: all 5s;
```

```
<body>
```

```
<div class="box box1">
```

```
<h3>Bez tranzicije</h3>
```

```
</div>
```

```
<div class="box box2">
```

```
<h3>Sa tranzicijom</h3>
```

```
</div>
```

```
</body>
```

Trajanje tranzicije

- Svojstvo **transition-duration** određuje vremenski period tranzicije.
- Može biti u sekundama ili milisekundama.
- Svojstvo **transition-timing-function** omogućava definisanje brzine tranzicije tokom njenog trajanja.
 - Podrazumevano je **ease** (jednostavna), koja počinje sporo, zatim se ubrzava i usporava na kraju.
- Svojstvo **transition-delay** omogućava određivanje početka transformacije. Podrazumevano je na klik (hover) miša, ali to može da se odloži ovim svojstvom.

Tranzicije i transformacija 2

```
.box{
    border-radius: 50%;
    height: 40px;
    margin: 50px auto;
    width: 40px;    }

.box:hover{
transform: skew(30deg, 30deg);
}

.box1 {
    background: #60D4C8;
    transition: all 300ms; }

.box2 {
    background: #46BAAF;
    transition: all 1s; }

.box3 {
    background: #3e9990;
    transition: all 3s; }
```

```
<body>
<div class="box box1">
  </div>
  <div class="box box2">
    </div>
  <div class="box box3">
    </div>
</body>
```

Tranzicije i transformacija 3

```
.box{  
    border-radius: 50%;  
    height: 40px;  
    margin: 50px auto;  
    width: 40px; }  
  
.box:hover{  
    transform: translateX(200px);  
}
```

```
.box1 {  
    background: salmon;  
    transition: all 1.5s ease; }
```

```
.box2 {  
    background: mediumturquoise;  
    transition: all 1.5s ease-in-out; }
```

```
.box3 {  
    background: thistle;  
    transition: all 3s ease-in-out; }
```

```
<body>  
<div class="box box1">  
  </div>  
  <div class="box box2">  
  </div>  
  <div class="box box3">  
  </div>  
</body>
```

Animacija 1

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: yellow;  
  animation-name: promena_boje;  
  animation-duration: 10s;  
}  
  
  @keyframes promena_boje {  
    from {background-color:yellow;}  
    to {background-color: green;}  
  }
```

```
<body>  
<div></div>  
</body>
```

Animacija 2

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: yellow;  
  animation-name: promena_boje;  
  animation-duration: 10s;  
}  
  
@keyframes promena_boje {  
  0% {background-color:yellow;}  
  25% {background-color: lime;}  
  50% {background-color: lightgreen;}  
  100% {background-color:green;}}
```

```
<body>  
<div></div>  
</body>
```

Animacija 3

```
div {
  width: 100px;
  height: 100px;
  background-color: yellow;
  position: relative;
  animation-name: promena_boje;
  animation-duration: 10s;
  /* animation-delay: 2s; animation-iteration-count: 3; */
  @keyframes promena_boje {
    0% {background-color: red; left: 0px; top: 0px;}
    25% {background-color: orange; left: 200px; top: 0px;}
    50% {background-color: yellow; left: 200px; top: 200px;}
    75% {background-color: lightgreen; left: 0px; top: 200px;}
    100% {background-color: lightblue; left: 0px; top: 0px;}}
```

```
<body>
<div></div>
</body>
```


Svojstvo clip-path

- ➔ Npr. clip-path: polygon(x1 y1, x2 y2, x3 y3, x4 y4)



Slide 33

SM4

Vise o svojstvu na linku: <https://bennettfeely.com/clippy/>

Suzana Marković; 21.11.2018.

Primeri



clip-path: polygon(50% 0%, 100% 38%,
82% 100%, 18% 100%, 0% 38%);



clip-path: circle(50% at 50% 50%);