



Cascade Style Sheet 4

dr Suzana Marković, dipl.ing. el.
suzana.markovic@vps.ns.ac.rs



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: rotateX(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: green;}  
  50% {background-color: violet;}  
  100% {background-color:blue;}  
}
```

```
<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: yellow; left: 0px; top: 0px;}  
    25% {background-color: green; left: 200px;  
    top: 0px;}  
    50% {background-color: violet; left: 200px;  
    top: 200px;}  
    75% {background-color: blue; left: 0px;  
    top: 200px;}  
    100% {background-color: pink; left: 0px; top: 0px;}}
```

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



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**.

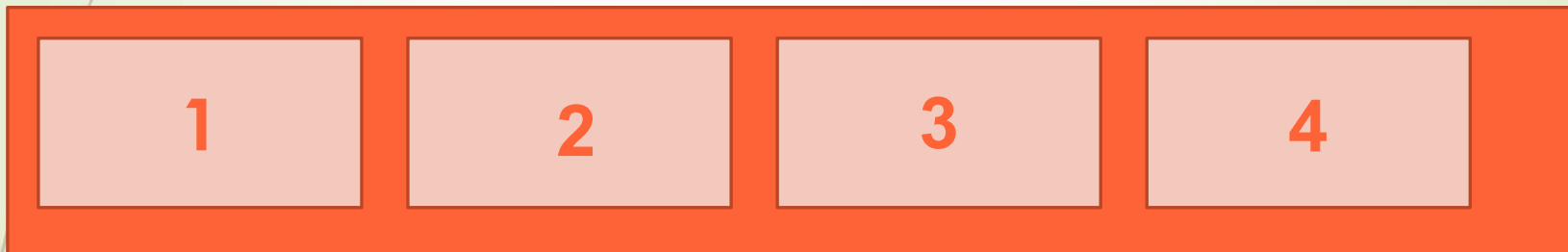


CSS flexbox

- 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.

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



Proporcije

- ➔ Svojstvo **flex:1;** - svi elementi će zauzeti jednaku količinu slobodnog prostora nakon što se dodaju uvlake i margine.
- ➔ Svojstvo **flex:2;** element sa ovim svojstvom će biti veći od ostalih, npr. za 3 elementa: dva će zauzeti po $\frac{1}{4}$ prostora, a treći $\frac{1}{2}$.

Svojstvo clip-path

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



Primeri



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



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