

JavaScript CSS animacije

Čas 26



BY



COMTRADE

CSS animacije su deo stilizacije HTML elemenata koji nam omogućava da se kroz određeno vreme element transformiše.

CSS animacije

Transformacije tokom vremena mogu biti bilo koja svojstva brojevni svojstava. Npr. rotacije (stepeni), pomeranje (pikseli), transparentnost, boje (procenti...) ...

CSS animacije

Da bismo napravili animaciju, prvo moramo da definišemo njen početak i kraj (ili neki drugi trenutak). To radimo pomoću svojstva `@keyframes`. Ovo svojstvo obično stavljamo na kraj CSS dokumenta.

Unutar njega, definišemo tačke u vremenu. To možemo učiniti sa procentualnom vrednošću.

Takođe možemo koristiti i reči `from` i `to` (od i do)

```
@keyframes bojenje {  
  from{  
    background-color:  rgb(0,0,0);  
  }  
  to{  
    background-color:  rgb(255, 255, 255);  
  }  
}
```

```
@keyframes rotiranje {  
  0%{  
    transform: rotate(0deg)  
  }  
  100%{  
    transform: rotate(360deg)  
  }  
}
```

CSS animacije

Možemo definisati koliko god tačaka želimo. Bitno je napomenuti da svaka vrednost kreće da se menja od nulte tačke ako je definisano svojstvo negde u budućnosti. Te, ako hoćemo da menjamo neko svojstvo od polovine, moramo tačku pre nje reći odakle kreće. Primer desno pokazuje da nam je boja crna sve do pola animacije, a tek onda kreće da se menja u belu, dok se rotacija vrši od početka do kraja.

```
div{  
  width: 400px;  
  height: 400px;  
  background-color: black;
```

```
@keyframes bojenjeRotiranje {  
  0%{  
    transform: rotate(0deg);  
    background-color: black;   
  }  
  50%{  
    background-color: black;  
  }  
  100%{  
    transform: rotate(360deg);  
    background-color: white;  
  }  
}
```

CSS animacije

Da bismo zadali tu animaciju na element, moramo koristiti svojstvo animation i definisati njeno trajanje. Ako ne stavimo trajanje, biće 0 sekundi te se neće ni izvršiti.

Dakle naziv koji stavimo u keyframes iskoristimo u animation.

100% u ovom primeru znači na drugoj sekundi. Da smo stavili 50% između, to bi se desilo na prvoj (pola od dve).

```
.kvadrat{  
  animation: rotiranje 2s;  
}
```

```
@keyframes rotiranje {  
  0%{  
    transform: rotate(0deg)  
  }  
  100%{  
    transform: rotate(360deg)  
  }  
}
```

CSS animacije

Osim naziva i trajanja, animation je kombinacija ovih stvari (možemo dopisati u nastavku):


- name - naziv keyframes svojstva
- duration - trajanje
- timing-function - ubrzanje, usporenje
- delay - sačekati pa onda krenuti sa animacijom
- iteration-count - koliko puta će se izvršiti (kao u petlji u js) - infinite za beskonačno
- direction - napred ili nazad ili kombo
- fill-mode - kako će element izgledati nakon animacije

```
animation: rotiranje 2s linear 1s 23 normal forwards;
```

```
@keyframes rotiranje {  
  0%{  
    transform: rotate(0deg)  
  }  
  100%{  
    transform: rotate(360deg)  
  }  
}
```

CSS animacije – timing-function

Standardna animacija kreće polako i završava polako, a između održava ravnu brzinu. Poput automobila koji se isparkirava, odveze pa uparkira.

- ease - standardno, polako krene pa ubrza i polako završi a između ide normalno brzo
- ease-in - polako krene ali ravno ide i završi
- ease-out - polako završi
- ease-in-out - poput ease ali manje ubrzanje
- linear - bez ikakvih ubrzanja ili usporenja, uvek ravna brzina
- cubic-bezier(vrednosti) - za precizno naglašavanje ubrzanja i usporenja, obično se koristi neki generator [cubic-bezier\(.25,.1,.25,1\)](https://cubic-bezier.com)  cubic-bezier.com

```
animation: rotiranje 2s linear 1s 23 normal forwards;
```

```
@keyframes rotiranje {  
  0%{  
    transform: rotate(0deg)  
  }  
  100%{  
    transform: rotate(360deg)  
  }  
}
```


CSS animacije – direction

Možemo birati da li će animacija ići od 0 do 100 ili obrnuto ili svaki put različito po iteraciji

- normal - od 0 do 100
- reverse - od 100 do 0
- alternate - prvo normalno pa u sledećoj iteraciji obrnuto i tako u krug
- alternate-reverse - prvo obrnuto, pa normalno i tako u krug

```
animation: rotiranje 2s linear 1s 23 normal forwards;
```

```
@keyframes rotiranje {  
  0%{  
    transform: rotate(0deg)  
  }  
  100%{  
    transform: rotate(360deg)  
  }  
}
```

CSS animacije – fill-mode

Kako će se element ponašati nakon kraja animacije

- forwards - zadržiće css koji je dobio nakon poslednje iteracije animacije
- backwards - zadržiće css koji se dobije na prvoj tački animacije
- both - kombinacija oba

```
animation: rotiranje 2s linear 1s 23 normal forwards;
```

```
@keyframes rotiranje {  
  0%{  
    transform: rotate(0deg)  
  }  
  100%{  
    transform: rotate(360deg)  
  }  
}
```

CSS animacije

Ako želimo da menjamo samo neku vrednost od ovih svojstava, možemo to uraditi tako što ćemo dodati animation-svojstvo, npr:

animation-timing-function: ease-out;
animation-duration: 23s;
animation-fill-mode: forwards

Osim toga, vredno je napomenuti i animation-play-state koji može biti paused ili running (pauzirano ili traje).

```
animation: rotiranje 2s linear 1s 23 normal forwards;
```

```
@keyframes rotiranje {  
  0%{  
    transform: rotate(0deg)  
  }  
  100%{  
    transform: rotate(360deg)  
  }  
}
```

CSS animacije

Primer loader animacije:

[W3Schools Tryit Editor](#)

Domaći:

Napravi animaciju koja pomera kvadrat od gornjeg levog ćoška, do donjeg desnog, do donjeg levog, do gornjeg desnog, do gornjeg levog. Između svakog ćoška se okrene za 90 stepeni i promeni boja. Kad odradi ceo put, animacija se završi i kvadrat ostaje tu. Početna i krajnja boja treba da budu različite.

Hvala na pažnji



BY



COMTRADE