CSS tranzicije (s primjerima)

CSS tranzicije su vrsta animacije HTML elemenata. Kod HTML elemenata moguće je mijenjati izgled nakon učitavanja stranice. Umjesto da se dogodi trenutno, ova promjena izgleda može biti postupna, odnosno animirana, što svakako stvara ljepši vizualni dojam.

Kako prisiliti elemente da promijene izgled, kada je stranica već učitana i prikazana?

Možda je najjednostavniji način stvoriti CSS stil koji se mijenja kada korisnik prijeđe mišem preko elementa. To postižemo pomoću dobro poznate pseudo-klase **hover**. Kao što znamo, u pseudoklasi definiramo kako promijeniti izgled elementa u određenim situacijama. Ta promjena može biti postupna (animirana).

Drugi način za promjenu izgleda elementa je s JavaScriptom, korištenjem Web DOM programiranja. Iako se radi o programiranju, promjena izgleda zapravo nije teška - kada "dobijemo" HTML element, možemo promijeniti njegov inline stil pomoću podobjekta style ili cijelu klasu putem svojstva className. Svaka od ovih promjena može se animirati pomoću CSS prijelaza.

Za postizanje efekta prijelaza koristimo sljedeće CSS atribute:

transition -property → Popis CSS atributa koji će biti animirani tijekom tranzicije

transition-duration → Trajanje prijelaza

transition -delay → Duljina pauze prije početka prijelaza

transition -timing-function → Određivanje vremena međufaza u prijelazu

tranzicija → Složeni atribut za definiranje CSS tranzicije

Animirani atributi

Prvo ćemo se pozabaviti CSS atributom **transition-property**, koji predstavlja popis onih CSS atributa koje želimo animirati. Trebamo znati da se samo neki atributi mogu animirati, pa ih jedino ima smisla uključiti u ovaj popis.

Atribut transition-property također ima nekoliko unaprijed definiranih vrijednosti:

all - svi atributi elementa koji se mogu animirati bit će animirani

none - animacija se neće dogoditi ni na jednom atributu, praktički ovo je isključivanje animacije

Primjenjuju se i globalne vrijednosti:

Initial - atribut će se vratiti na početnu vrijednost (u ovom slučaju prazna lista)

inherit - atribut nasljeđuje vrijednost od nadređenog elementa

unset - atribut će se vratiti na početnu vrijednost (budući da nije naslijeđen)

Obično ćemo navesti CSS atribute koje želimo animirati. Alternativno, možemo navesti vrijednost all, što će značiti animirati sve što može.

```
transition-property: [atribut]
transition-property: [atribut], [atribut], ...
transition-property: all
```

Primjer za tranzicije

Promjena oblikovanja CSS-a putem pseudo-klase **hover**

Najjednostavniji način aktiviranja tranzicije je korištenje pseudoklase **hover**.

```
.primjer {
  color: red;
  transition-property: color;
  transition-duration: 1s;
}
.primjer:hover {
  color: blue;
```

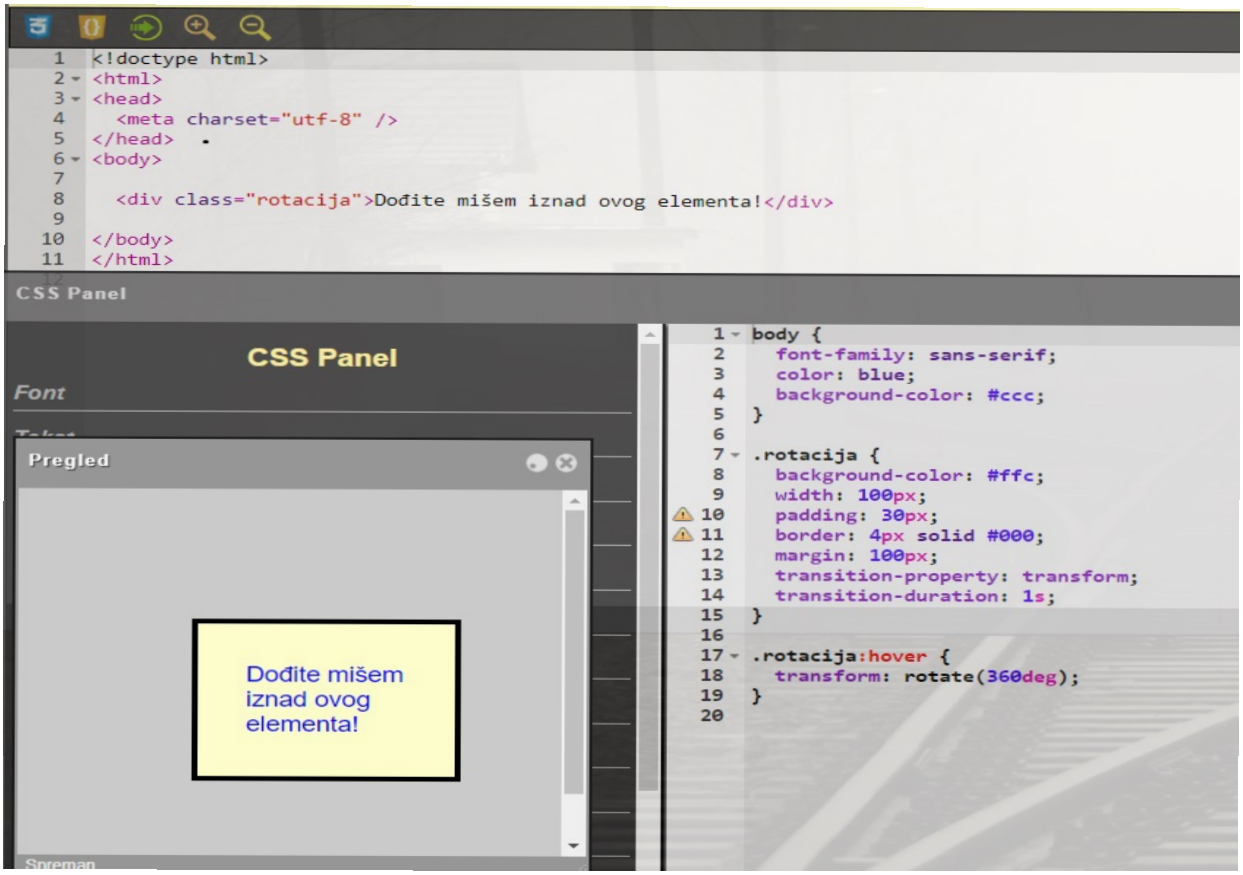
U ovom primjeru vidimo da je za element klase primjera definirano da ako se mišem pomakne preko njega, tekst se mijenja iz crvene u plavu. Međutim, klasa je također definirana za prijelaz atributom boje za 1 sekundu.

Kad god se iz nekog razloga promijeni boja teksta elementa, ta će promjena biti animirana. Slučajno se to događa kada se miš postavi preko elementa, ali i kada se makne.

Pogledajte primjer gdje smo koristili mogućnost animiranja transformacije (rotacije).

U ovom primjeru vidimo da je za element klase primjera definirano da ako se mišem pomakne preko njega, tekst se mijenja iz crvene u plavu. Međutim, klasa je također definirana za prijelaz atributom boje za 1 sekundu.

Kad god se iz nekog razloga promijeni boja teksta elementa, ta će promjena biti animirana. Slučajno se to događa kada se miš postavi preko elementa, ali i kada se makne.



Trajanje animacije

Trajanje animacije je još jedna važna stvar koju želimo prilagoditi kada radimo s transition. Od svih atributa koji služe za kontrolu trajanja, najviše ćemo koristiti transition-duration, čija vrijednost predstavlja trajanje prijelaza u obliku metrike. Vremensko trajanje dano je s dvije moguće mjerne jedinice:

s - sekunde (npr. 5 s, 1,5 s)

ms - milisekunde - milisekunda je tisućinka sekunde (npr. 1000ms, 500ms)

Vidjeli smo da se vrijednost atributa **transition-property m**ože specificirati kao lista. Isto vrijedi i za ovaj atribut. Pri tome svako trajanje s popisa treba odgovarati jednom atributu navedenom na popisu. To znači da animacija po različitim atributima može imati različito trajanje.

```
transition-duration: [trajanje]
transition-duration: [trajanje], [trajanje], ...
```

Kako riješiti situaciju kada broj atributa iz transition-property ne odgovara broju trajanja navedenih u trajanju prijelaza?

Ako je broj zadanih trajanja veći od broja atributa, onda nema problema - "višak" se jednostavno ignorira. S druge strane, ako je navedeno manje trajanja od broja atributa, tada se popis trajanja "kotrlja" od početka dok svi atributi ne budu "zadovoljeni". Ovo je prilično zgodno, jer ako odredimo samo jedno trajanje, ono će se primijeniti na sve atribute.

Primjer za prijelazno trajanje

Sve napisano možda neće imati puno smisla dok ne pogledamo nekoliko primjera...

```
.primjer {
  transition-property: width, height;
  transition-duration: 1s, 500ms;
}
```

Ovo je prilično jednostavna situacija. Atributi širine i visine animirani su kroz tranziciju. Bit će potrebna 1 sekunda za promjenu širine i samo pola sekunde (500 milisekundi) za promjenu visine.

```
.primjer {
  transition-property: left, top, color, background-color, font-size;
  transition-duration: 1s, 0.5s, 0.8s;
}
```

U ovom primjeru imamo situaciju s 5 atributa i samo 3 trajanja. Kako će se to riješiti? Animacije lijevog, gornjeg i atributa boja trajat će, kako je navedeno na popisu, 1 s, 0,5 s, 0,8 s. Za sljedeća dva atributa popis "počinje" ispočetka, tako da animacije atributa boje pozadine i veličine fonta traju 1 s i 0,5 s.

```
.primjer {
  transition-property: transform, border;
  transition-duration: 500ms;
}
```

U većini slučajeva, željet ćemo da animacija svih atributa traje jednako, pa će to biti najčešća situacija - specificiramo više atributa, ali samo jedno trajanje koje se odnosi na sve atribute. To znači da se promjena atributa transformacije i granice animira u pola sekunde.

Redoslijed navođenja svojstava u atributu transition-property nije značajan. Redoslijed kojim postavljamo atribute ne znači da će se prvi animirati prvi, zatim drugi, pa treći itd. Kada dođe do promjene, svi atributi počinju animirati istovremeno, a ovisno o njihovom trajanju mogu završiti u različito vrijeme.

Međutim, ako želimo, možemo utjecati na početak animacije po atributu, koristeći transition-delay prijelaza. Ovaj atribut je vrlo sličan atributu prijelaznog trajanja, budući da kao vrijednost može imati i popis trajanja izraženih u vremenskim jedinicama.

transition-delay: [trajanje]

transition-delay: [trajanje], [trajanje], ...

Trajanje navedeno u ovom atributu označava odgodu početka animacije određenim atributom.

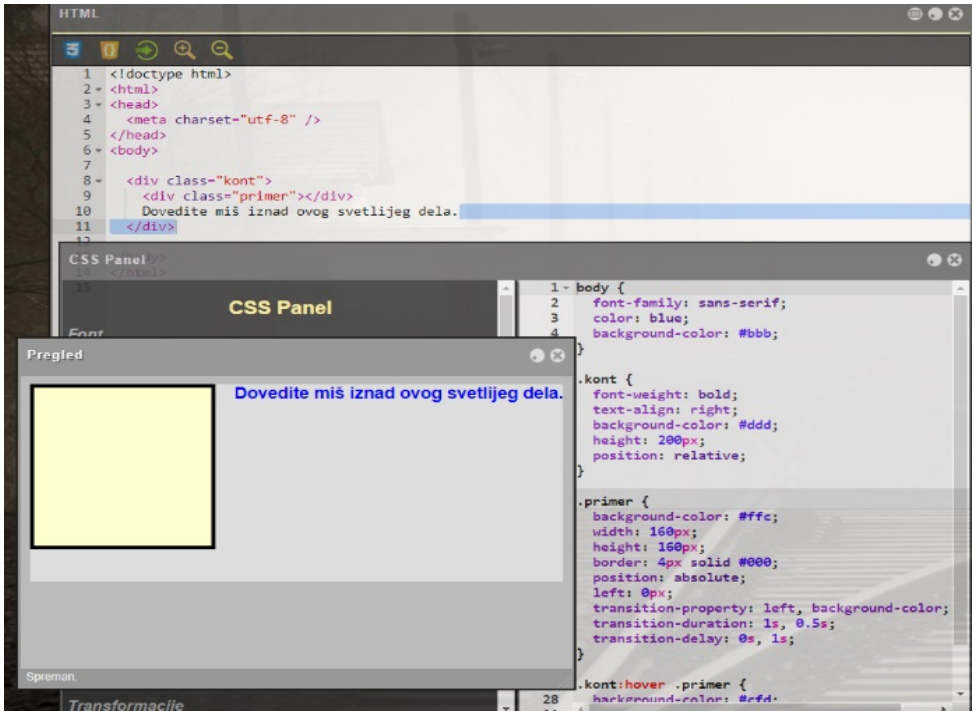
Ako je naveden manji broj trajanja od broja animiranih atributa navedenih u svojstvu tranzicije, atributi kojima nedostaju trajanja neće imati odgodu. Ovo se ponašanje razlikuje od atributa trajanja tranzicije.

Zanimljivo je da se kašnjenje može postaviti i kao **negativna vrijednost**. To bi značilo da je animacija atributa trebala započeti čak i prije nego što se dogodila radnja koja je uzrokovala animaciju. To je naravno nemoguće, no web preglednik će se "snaći" tako da pokrene animaciju u isto vrijeme kada i ostale animacije, ali će izgledati kao da kreće "od sredine".

Primjer za transition-delay Evo kako bismo zadali **odlaganje početka animacije**.

```
.primjer {

  transition-property: left, background-color;

  transition-duration: 1s, 0.5s;

  transition-delay: 0s, 1s;

}
```

Dakle, promjena položaja na lijevoj strani i boja pozadine su animirani. Vidimo da će tranzicija left trajati 1 sekundu, a promjena boje pola sekunde. Međutim, promjena boje završit će nakon pomicanja. Pomicanje počinje odmah (kašnjenje od 0 sekundi), a promjena boje počinje s odgodom od 1 sekunde. To znači da će promjena boje započeti kada se element završi s pomicanjem i tada će trajati pola sekunde.



Posljednja stvar povezana s kontrolom "vremena" je mogućnost utjecaja na vrijeme međufaza animacije.

Što to znači?

Zamislite da se blok treba pomaknuti za 100px na 5 sekundi. Koliko će se pomaknuti svake sekunde? Logičan odgovor bi bio da se svake sekunde blok pomakne za 20px. Ovo je najjednostavnija animacija, ali u isto vrijeme, kada se gleda, djeluje mehanički. Bilo bi puno bolje da blok npr. polako ubrzava na početku i blago usporava na kraju animacije.

Drugim riječima, idealno bi bilo kada bismo mogli definirati ubrzanje i usporavanje animacije. Upravo te stvari kontroliramo atributom transition-timing-function, gdje kao vrijednost postavljamo "funkciju" kojom se određuje tajming animacije. Ovaj atribut ima nekoliko unaprijed definiranih vrijednosti:

linear - animacija konstantnom brzinom, (već spomenuto "mehaničko kretanje")

ease-in - sporo ubrzanje, nagli prekid (kao kod linearnog vremena)

ease-out - nagli početak (poput linearnog), usporavanje na kraju

ease-in-out - blago ubrzanje i blago usporavanje na kraju (simetrično ubrzanje i usporavanje)

lakoća - ista kao i prethodna, samo što na početku oštrije ubrzava, a na kraju sporije (ovo je zadano vrijeme)

step-start - odmah skače na kraj animacije i ne mijenja se tijekom cijelog trajanja (cijelo vrijeme prikazuje završnu fazu)

step-end - tijekom cijelog trajanja ne mijenja se (prikazuje početnu fazu), tek na kraju "skoči" na kraj animacije

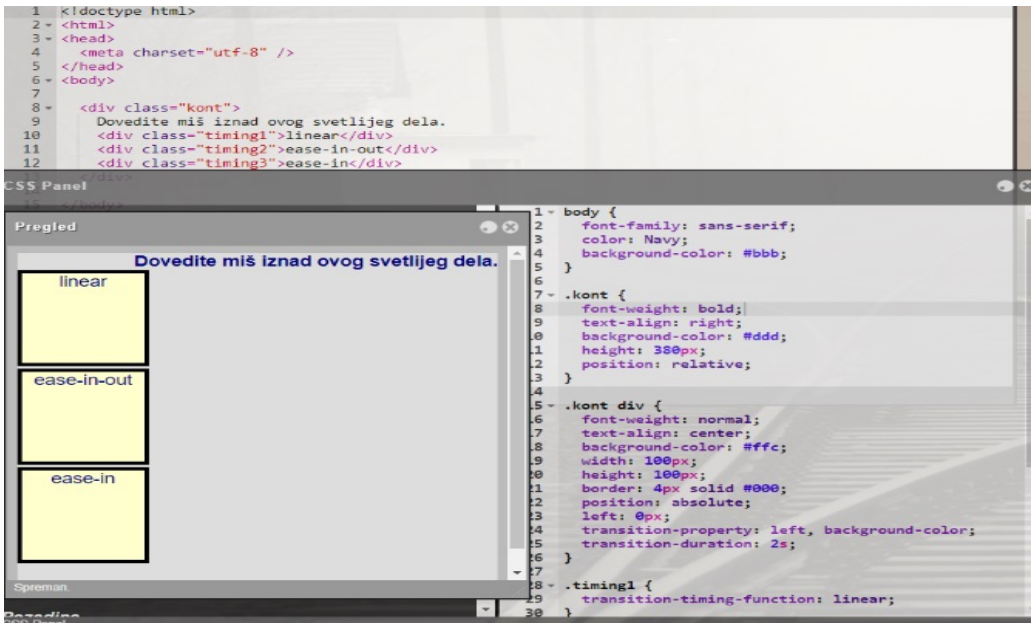
```
transition-timing-function: [funkcija]
transition-timing-function: [funkcija], [funkcija], ...
```

Ako zadamo manje funkcija u odnosu na broj animiranih atributa, oni atributi za koje "fale" funkcije će imati podrazumevanu vrijednost (ease).

Primjer za transition-timing-function

Primjer zadavanja **funkcije tajminga animacije**.

```
.primjer {

  transition-property: left, background-color;

  transition-duration: 2s;

  transition-timing-property: ease-in-out;

}
```



Složen atribut prijelaza

Sve što smo do sada naučili možemo napraviti u jednom potezu, s atributom prijelaza. Ova metoda je vrlo učinkovita ako imamo samo nekoliko atributa kojima želimo postaviti sve parametre animacije, ali izvan toga postaje pomalo zbunjujuće.

Kao vrijednost je zapravo data kombinacija vrijednosti koje definiraju animaciju jednim atributom. Imajte na umu da budući da su navedene dvije vremenske vrijednosti, prva uvijek označava trajanje, a druga odgodu. Naravno, nisu sve vrijednosti obavezne, ali ova kombinacija uvijek treba sadržavati barem atribut i trajanje. Besmisleno je specificirati samo atribute bez trajanja, budući da se tada računa zadano trajanje od 0 sekundi - tako da praktički nema animacije. Ako funkcija mjerenja vremena i kašnjenje nisu navedeni, zadane vrijednosti su lakoća i 0 s.

transition: [atribut] [trajanje]

transition: [atribut] [trajanje] [odlaganje]

transition: [atribut] [trajanje] [tajming-funkcija] [odlaganje]

Naravno, i ovdje je moguće zadati čitavu listu kombinacija.

transition: [kombinacija], [kombinacija]...

Primjer za transition

Pogledajte kako se nekoliko atributa tranzicije mogu zamijeniti jednim **složenim atributom**.

```
.primjer1 {
  transition-property: left, background-color;
  transition-duration: 2s, 1s;
  transition-timing-property: ease-in, linear;
}

.primer2 {
  transition: left 2s ease-in, background-color 1s linear;
}
```

Mozilla Developer Network, Using CSS Transitions