**CSS transformacije i tranzicije**

**00:05:40**

Jedinica: 18 od 30

**+Rezime**

Na početku svoj razvoja, jezik CSS bio je jezik jednostavnih mogućnosti, s osnovnom svrhom da formatira sadržaj definisan CSS oznakama. Vremenom, moć CSS-a se povećavala, pa je tako danas moguće CSS koristiti i za kompleksnije zahvate nad HTML elementima. Neke od kompleksnijih operacija, koje se korišćenjem CSS-a mogu sprovesti nad elementima su transformacije i tranzicije.

**Šta su transformacije?**

Kao što i samo ime kaže, *transformacije*omogućavaju transformaciju HTML elementa uticanjem na njihove vizuelne karakteristike. Na taj način elementi prelaze iz jednog u drugo stanje, a sam proces prelaska iz stanja u stanje naziva se transformacija.

CSS podržava dvodimenzionalne i trodimenzionalne transformacije.

U 2D CSS transformacije ubrajaju se:

* rotiranje
* pomeranje
* iskošavanje
* promena veličine

3D CSS transformacije omogućavaju vrlo sličan skup transformacija nad elementima, ali u koordinatom sistemu sa tri ose. U nastavku lekcije prvo će biti prikazane 2D transformacije, a zatim i najznačajnije 3D transformacije.

|  |
| --- |
| **Napomena**    *2D transformacije primenjuju se korišćenjem CSS svojstva transform. Određene verzije browsera zahtevaju specijalnu sintaksu, tako da se može naići i na CSS svojstva -ms-transform, za Internet Explorer, -webkit-transform za Chrome, Safari i Opera  i -moz- transform za Firefox browsere.* |

**2D Pomeranje (translate)**

Kako bi se element pomerio u odnosu na svoj normalan položaj, moguće je koristiti translate transformaciju. Transformacije se primenjuju korišćenjem različitih metoda, koje se navode kao vrednost transform svojstva. Tako je za transformaciju pomeranja elementa dovoljno iskoristiti metodu translate(). Sintaksa ove metode je sledeća:

|  |  |
| --- | --- |
| 1 | translate(x-axis, y-axis) |

Parametri metode translate() odnose se na vrednosti za koje je element potrebno pomeriti po x i y osama. Sledeći primer ilustruje način na koji je moguće iskoristiti CSS transformaciju translate:

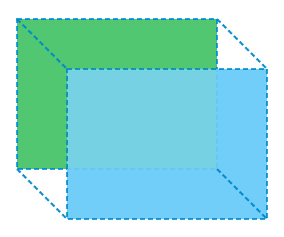
|  |  |  |
| --- | --- | --- |
| 1  2  3 | | #transformed{   transform: translate(50px, 50px);  } |
| **Napomena**    *Prikazana transformacija se u produkcionim uslovima često navodi korišćenjem dodatnih svojstava koja su specifična za određene browsere i neke njihove verzije. Tako se prethodni primer može navesti i ovako:*    #transformed{            transform: translate(50px, 50px);          -ms-transform: translate(50px, 50px);           -webkit-transform: translate(50px, 50px); }    *U nastavku lekcije će zbog preglednosti biti korišćeno isključivo svojstvo transform.* | |

Translate transformacija proizvodi efekat kao na slici 18.1.



*Slika 18.1 – Translate transformacija*

Pravougaonik ispod ilustruje normalni položaj div elementa, dok je pravougaonik iznad div element sa primenjenom transformacijom, tako da je on izmešten sa svog prirodnog položaja. Sve ovo se može još ilustrativnije prikazati na slici 18.2.



*Slika 18.2 – Translate transformacija(2)*

Slika 18.2. ilustruje efekat koji translate transformacija ima na primenjeni div element.

|  |
| --- |
| **Napomena**    *Prikazani primer, kao i svi primeri koji će se pojaviti u ovoj lekciji, nalaze se u prilogu.* |

**2D rotiranje (rotate)**

CSS metoda rotate svojstva transform omogućava rotiranje elementa. Njena sintaksa je sledeća:

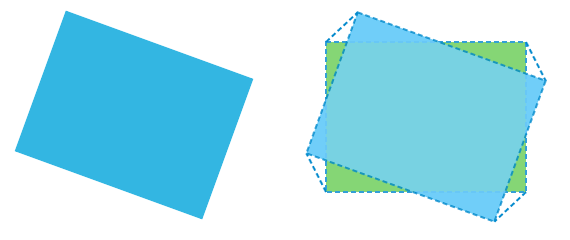
|  |  |
| --- | --- |
| 1 | rotate(degrees) |

Parametar degrees se odnosi na ugao rotiranja i izražava se korišćenjem jedinice deg.

Primer koji ilustruje rotiranje jednog div elementa je sledeći:

|  |  |
| --- | --- |
| 1  2  3 | #transformed{   transform: rotate(20deg);  } |

Efekat koji definisana transformacija ima prikazan je na slici 18.3.



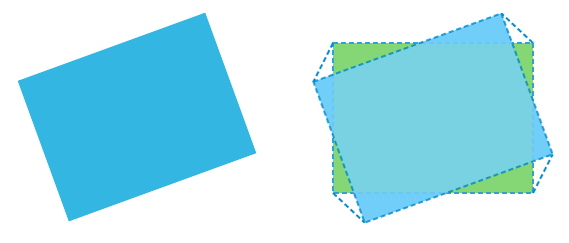
*Slika 18.3 – Rotate transformacija*

Na slici 18.3 prikazan je div element na kome je primenjena transformacija rotiranja. Na desnoj polovini slike prikazan je div element u odnosu na svoj prirodni položaj.

Rotiranje HTML elementa je moguće obaviti i korišćenjem negativnih vrednosti uglova rotiranja:

|  |  |
| --- | --- |
| 1  2  3 | #transformed{   transform: rotate(-20deg);  } |

Efekat prikazanog koda je kao na slici 18.4.



*Slika 18.4 – Rotiranje korišćenjem negativnog ugla*

**2D povećavanje i smanjivanje (scale)**

Za povećavanje i smanjivanje HTML elemenata moguće je koristiti metodu scale() svojstva transform. Njena sintaksa je sledeća:

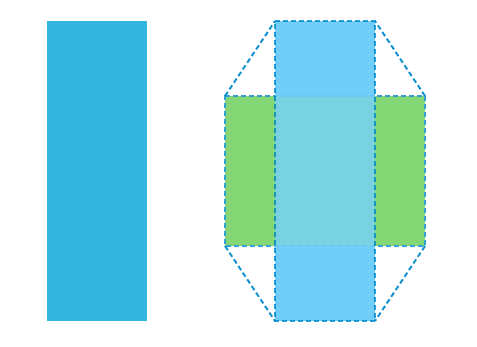
|  |  |
| --- | --- |
| 1 | scale(width\_ratio, height\_ration) |

Metoda scale() prihvata dva parametra koja se koriste kako bi se proračunale nove dimenzije elementa. Prvi parametar se odnosi na širinu, a drugi na visinu. Vrednost 1 označava da neće doći ni do kakve promene veličine. Na sličan način, vrednost 2 označava da će se veličina dva puta povećati, a vrednost 0.5 da će se veličina dva puta smanjiti.

Sledeći primer ilustruje korišćenje metode scale za transformaciju:

|  |  |
| --- | --- |
| 1  2  3 | #transformed{   transform: scale(0.5, 2);  } |

U prikazanom primeru definisano je da će se širina elementa smanjiti 2 puta, a visina povećati 2 puta. Tako primer proizvodi efekat kao na slici 18.5.



*Slika 18.5 – Scale transformacija*

**2D zakošavanje (skew)**

CSS poseduje tri različite metode, kojima je moguće obaviti zakošenje elementa:

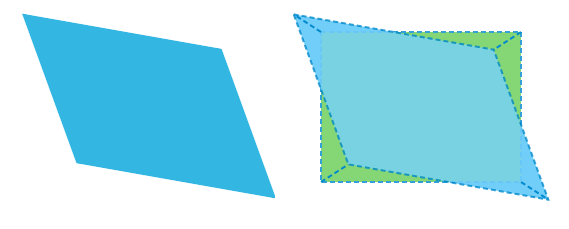
* skewX(angleX)
* skewY(angleY)
* skew(angleX, angleY)

Metoda skewX() vrši zakošavanje elementa po x-osi. Metoda skewY() to isto čini, ali po y-osi, dok metoda skew() može da zakosi element po obe ose.

Primer korišćenja ove transformacije je sledeći:

|  |  |
| --- | --- |
| 1  2  3 | #transformed{   transform: skew(20deg, 10deg);  } |

Efekat je kao na slici 18.6.



*Slika 18.6 – Skew transformacija*

Metoda skew može da zameni sve metode za zakošavanje, tako što ne mora da prihvati dva parametra. Ukoliko se drugi parametar ne navede, kao njegova vrednost se uzima 0.

**Kako se u CSS-u naziva transformacija pomeranja elementa:**

 translate

 skew

 rotate

 pitch

**2D matrix**

Za izvršenje 2D transformacija, CSS poseduje i metodu matrix, koja objedinjuje sve CSS transformacije. Ova metoda prihvata 6 parametara i na taj način omogućava da se definišu vrednosti za sve transformacije odjednom. Sintaksa metode matrix je:

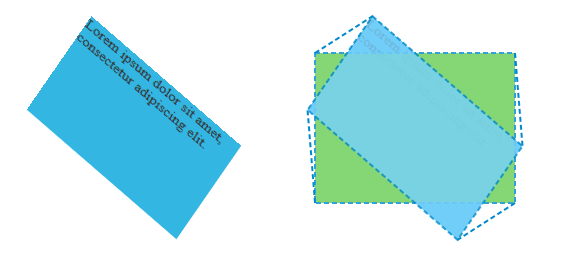
|  |  |
| --- | --- |
| 1 | matrix(scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY()) |

**3D transformacije**

3D transformacije funkcionišu na sličan način kao prikazane 2D transformacije, s tim što omogućavaju transformisanje elemenata korišćenjem koordinatnog sistema sa tri ose. Tako se, na primer, prilikom rotiranja mogu navesti uglovi za svaku od osa pojedinačno. To ilustruje sledeći primer:

|  |  |
| --- | --- |
| 1  2  3 | #transformed{   transform: rotateX(30deg) rotateY(30deg) rotateZ(30deg);  } |

Prikazani primer proizvodi efekat kao na slici 18.7.



*Slika 18.7 – 3D rotiranje*

|  |
| --- |
| **Napomena**    *Pravi efekat 3D transformacija biće vidljiv u nastavke lekcije, kada budu obrađene tranzicije.* |

3D transformacije poseduju i specijalne metode koje objedinjuju pojedinačne metode za transformaciju po jednoj osi. Tako je 3D rotiranje, identično onom iz prethodnog primera, moguće postići:

|  |  |
| --- | --- |
| 1 | transform: rotate3d(1,1,1, 30deg); |

Ovoga puta je iskorišćena metoda rotate3d. Prva tri parametra se odnose na ose oko kojih će se obaviti rotiranje. Broj 1 označava da će se rotiranje obaviti, a 0 da neće. Inače, prva tri parametra se odnose na x, y i z ose, respektivno. Poslednji parametar definiše ugao rotacije.

I ostale 2D transformacije poseduju sve 3D pandane. Najznačajnije metode za obavljanje 3D transformacija prikazane su u tabeli.

|  |  |
| --- | --- |
| **Metoda** | **Opis** |
| translateX(x) | 3D pomeranje, ali korišćenjem samo X-ose |
| translateY(y) | 3D pomeranje, ali korišćenjem samo Y-ose |
| translateZ(z) | 3D pomeranje, ali korišćenjem samo Z-ose |
| scaleX(x) | 3D promena veličine, ali samo po X-osi |
| scaleY(y) | 3D promena veličine, ali samo po Y-osi |
| scaleZ(z) | 3D promena veličine, ali samo po Z-osi |
| rotateX(angle) | 3D rotiranje, ali samo po X-osi |
| rotateY(angle) | 3D rotiranje, ali samo po Y-osi |
| rotateZ(angle) | 3D rotiranje, ali samo po Z-osi |
| translate3d(x,y,z) | 3D pomeranje |
| scale3d(x,y,z) | 3D promena veličine |
| rotate3d(x,y,z,angle) | 3D rotiranje |
| matrix3d  (n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n) | sve 3D transformacije objedinjene u jednu metodu |

*Tabela 18.1 – Metode 3D transformacija*

**CSS tranzicije**

U dosadašnjem toku lekcije bilo je reči o transformacijama. Rečeno je da transformacija predstavlja prelazak elementa iz jednog oblika u drugi. Ukoliko se takva promena obavi glatko i postepeno, tokom određenog vremenskog perioda, tada se govori o tranziciji. Nastavak lekcije biće posvećen tranzicijama.

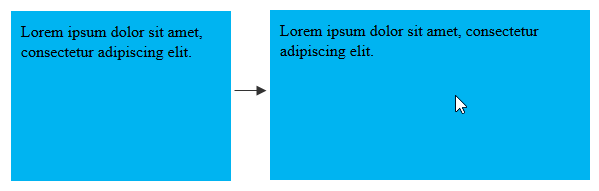
Tranzicije nisu ograničene na rad sa transformacijama, već omogućavaju glatku promenu bilo kojeg CSS svojstva.

Prilikom definisanja tranzicije, potrebno je znati naziv svojstva čiju vrednost je postepeno potrebno menjati, kao i dužinu trajanja promene. Pritom će tranzicija započeti onoga trenutka kada dođe do promene vrednosti svojstva koje je prilikom definisanja tranzicije navedeno.

Sledeći primer ilustruje definisanje tranzicije za širinu div elementa:

|  |  |
| --- | --- |
| 1  2  3 | div{   transition: width 2s;  } |

Kada se širina div elementa promeni, takva promena će trajati dve sekunde. Promena širine div elementa se može obaviti prilikom hovera, i u takvoj situaciji će se dobiti efekat kao na slici 18.8.



*Slika 18.8 – Tranzicija širine div elementa na hoveru*

Naravno, pravi efekat tranzicija nije moguće prikazati slikom, zato je najbolje funkcionisanje prikazanog primera isprobati unutar nekog browsera.

Korišćenjem CSS tranzicija, moguće je u istom trenutku definisati efekat tranzicije za više od jednog CSS svojstva:

|  |  |
| --- | --- |
| 1 | div {    transition: width 2s, height 4s; } |

Ovoga puta, tranzicija je definisana za width i height svojstva.

|  |
| --- |
| **Napomena**    *Tranziciju je moguće aktivirati i za sva svojstva nekog HTML elementa. U takvim situacijama se navodi vrednost all.* |

U prikazanim primerima tranzicije, promene vrednosti svojstava su obavljane linearno tokom nekog vremena. Ovakvo ponašanje je moguće promeniti, pa tranzicija ne mora biti linearna. To se postiže svojstvom transition-timing-function. Ovo svojstvo može imati vrednosti prikazane tabelom 18.2.

|  |  |
| --- | --- |
| **Vrednost** | **Opis** |
| ease | spor početak, zatim ubrzanje, spor kraj |
| linear | jednaka brzina tokom celog trajanja |
| ease-in | spor početak |
| ease-out | spor kraj |
| ease-in-out | spor početak i spor kraj |
| cubic-bezier(n,n,n,n) | proizvoljno definisanje dinamike izvršavanja navođenjem vrednost parametara za kreiranje Bezijerove krive |

*Tabela 18.2 – Vrednosti transition-timing-function svojstva*

Korišćenje svojstva transition-timing-function ilustrovano je primerom:

|  |  |
| --- | --- |
| 1  2  3  4 | div{   transition: width 2s;   transition-timing-function: ease;  } |

Efekat tranzicije je moguće odložiti korišćenjem svojstva transition-delay. Na taj način je moguće definisati vremenski interval koji je potrebno da protekne od aktiviranja tranzicije do njenog početka. Na primer, umesto da tranzicija započne odmah kada dođe do promene vrednost svojstva za koje je tranzicija definisana, tranzicija će biti odložena za vreme definisano svojstvom transition-delay.

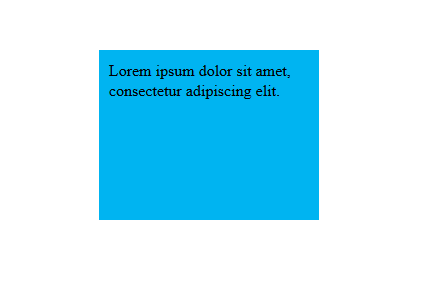
|  |  |
| --- | --- |
| 1  2  3  4  5 | div{   transition: width 2s;   transition-timing-function: ease;   transition-delay: 1s;  } |

Nešto ranije je rečeno da se tranzicije često koriste u kombinaciji sa transformacijama. Jedan takav primer prikazan je u nastavku:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | body{  perspective: 700px;  }  div{  transition: transform 2s;  transform-style: preserve-3d;  }    div:hover {  transform: rotateY(180deg);    } |

U primeru je definisana kombinacija transformacije i tranzicije. Pri tome je reč o 3D transformaciji rotiranja. Kada se strelica miša nađe iznad div elementa (hover), aktiviraće se transformacija 3D rotiranja, a prelazak iz jednog stanja u drugo će trajati 2 sekunde. Na taj način će biti postignut takozvani flip efekat.

Efekat prikazanog primera ilustrovan je slikom:



*Slika 18.9 – 3D rotacija*

|  |
| --- |
| **Napomena**    *Svojstvo transform-style sa vrednošću preserve-3d označava da je element potrebno pozicionirati u 3D koordinatnom sistemu. Svojstvo perspective definiše udaljenost elementa od gledaoca u 3D koordinatnom sistemu.* |