

## 2D CSS transformacije

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

### Šta su transformacije?

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

CSS podržava dvodimenzionalne i trodimenzionalne transformacije.

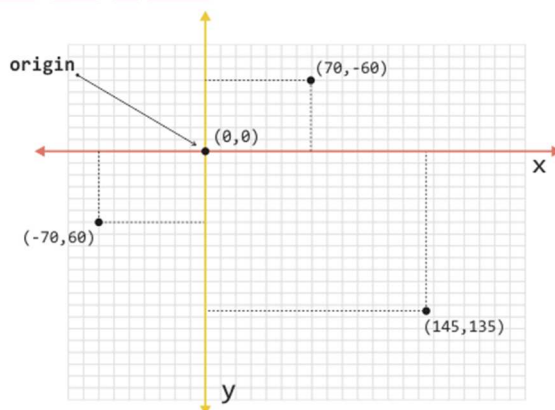
U 2D CSS transformacije ubrajaju se:

- pomeranje (*translate*);
- rotiranje (*rotate*);
- promena veličine (*scale*);
- iskošavanje (*skew*).

### 2D koordinatni sistem CSS-a

Definisanje položaja nekog predmeta u prostoru postiže se korišćenjem koordinata koordinatnog sistema. Takvih koordinata je dve, ukoliko se utvrđuje položaj nekog predmeta u prostoru sa dve ose.

Kada je reč o koordinatnim sistemima koji se koriste na webu, neko ustaljeno pravilo nalaže da se horizontalna osa naziva x-osa, a vertikalna y-osa (slika 5.1).



Slika 5.1. 2D koordinatni sistem na webu

Unutar koordinatnog sistema na slici 5.1. predstavljene su četiri tačke koje su obeležene odgovarajućim koordinatama, prikazanim u zagradi. Prva vrednost u zagradi se odnosi na vrednost  $x$ , a druga na vrednost  $y$  koordinate. Početak koordinatnog sistema definisan je tačkom sa koordinatama (0,0), a takva tačka se drugačije naziva **origin**. Origin tačka zapravo je referentna pozicija unutar koordinatnog sistema, u odnosu na koju se iskazuje položaj ostalih tačaka u sistemu.

Na slici 5.1. potrebno je zapaziti i to da sve tačke iznad referentne imaju negativne vrednosti  $y$  koordinata. Isto tako, sve tačke levo od referentne imaju negativne vrednosti  $x$  koordinata.

#### Napomena

*Potrebno je obratiti pažnju na to da se koordinatni sistem na webu u izvesnoj meri razlikuje od koordinatnih sistema u matematici ili mehanici. Pozitivne vrednosti po  $x$  i  $y$  osama se na webu nalaze u četvrtom kvadrantu (dole, desno), dok se u matematici one nalaze u prvom kvadrantu (gore, desno).*

## Kako se definišu CSS transformacije?

Transformisanje HTML elemenata korišćenjem CSS-a postiže se upotrebom svojstva **transform**. Sve do standardizacije ovog svojstva, u upotrebi su bili i oblici sa prefiksima specifičnim za različite browsere:

- -webkit-transform
- -moz-transform
- -ms-transform
- -o-transform

Kao i sa većinom svojstava sa prefiksima karakterističnim za različite browsere, za definisanje transformacija danas ih nije potrebno koristiti ukoliko se web sajt ne optimizuje za veoma stare web pregledače.

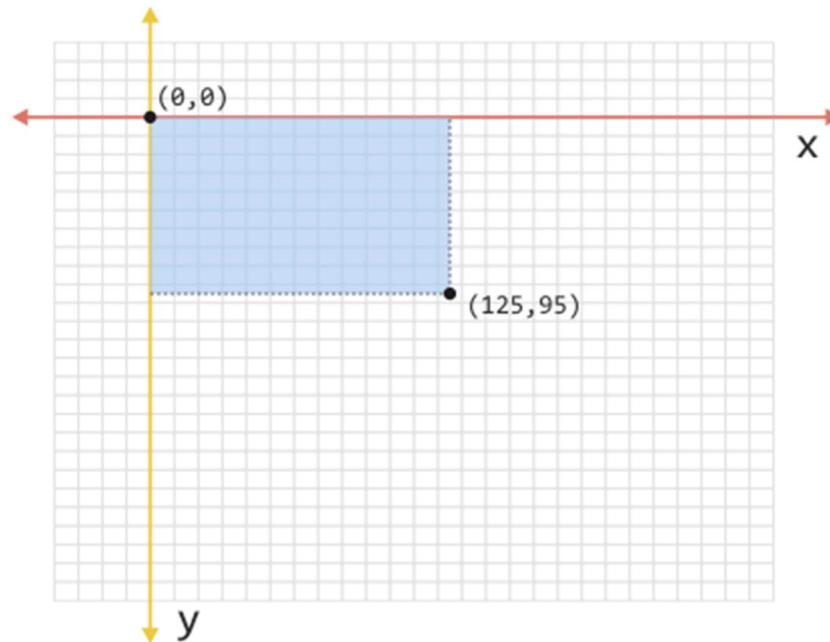
CSS svojstvo `transform` može da prihvati različite vrednosti koje se iskazuju CSS funkcijama. Zapravo, svaka od transformacija poseduje svoj skup funkcija sa parametrima različite namene, čime se definišu osobine transformacije.

## Pomeranje

Kako bi se element pomerio u odnosu na svoj normalni položaj, moguće je koristiti `translate` transformaciju. Tako je za transformaciju pomeranja elementa dovoljno iskoristiti funkciju **translate()**. Sintaksa ove funkcije je sledeća:

```
translate(x-axis)
translate(x-axis, y-axis)
```

Funkcija `translate()` može da prihvati jedan ili dva parametra. Kada prihvata jedan parametar, odnosi se na vrednost pomeranja po  $x$  osi. Druga vrednost definiše pomeranje po  $y$  osi. S obzirom na to da je reč od 2D transformaciji, primenjuje se nešto ranije prikazani koordinatni sistem sa dve ose, kod koga se referentna tačka (*origin*) poklapa sa gornjom levom tačkom elementa koji se transformiše (slika 5.2).

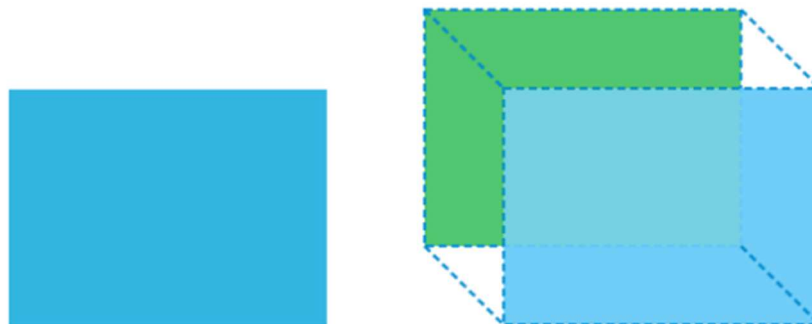


Slika 5.2. HTML element unutar 2D koordinatnog sistema

Sledeći primer ilustruje način na koji je moguće iskoristiti CSS transformaciju `translate`:

```
#transformed {  
  transform: translate(50px, 50px);  
}
```

Transformacija pomeranja proizvodi efekat kao na slici 5.3.



Slika 5.3. Translate transformacija

Na levoj polovini slike 5.3 ilustrovan je HTML element (`div` obojen u plavo) nad kojim je primenjena transformacija pomeranja. Ipak, pravi efekat ove transformacije se može videti na desnoj polovini slike 5.3, gde je u pozadini zelenom bojom predstavljena inicijalna pozicija `div` elementa, a plavom bojom njegova pozicija nakon transformacije pomeranja.

Pored funkcije `translate()`, za transformaciju pomeranja moguće je koristiti i specifične funkcije:

- `translateX(x-axis)` – za horizontalno pomeranje (po `x` osi); identičan efekat ima funkcija `translate(x-axis, 0)`;
- `translateY(y-axis)` – za vertikalno pomeranje (po `y` osi); identičan efekat ima funkcija `translate(0, y-axis)`.

## Rotiranje

Rotiranje HTML elemenata može se postići korišćenjem CSS funkcije **`rotate()`**:

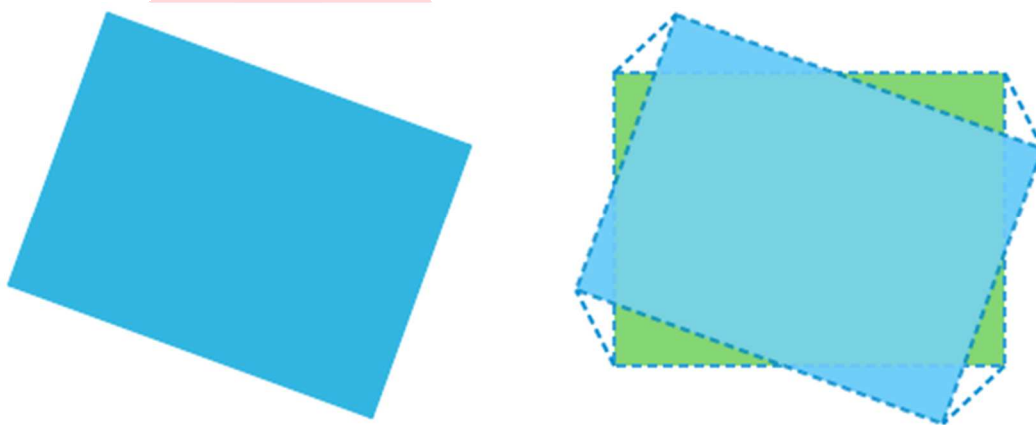
```
rotate(degrees)
```

Parametar `degrees` se odnosi na ugao rotiranja i izražava se korišćenjem jedinice `deg` (stepeni).

Primer koji ilustruje rotiranje jednog `div` elementa je sledeći:

```
#transformed {  
  transform: rotate(20deg);  
}
```

Efekat koji definisana transformacija ima prikazan je slikom 5.4.



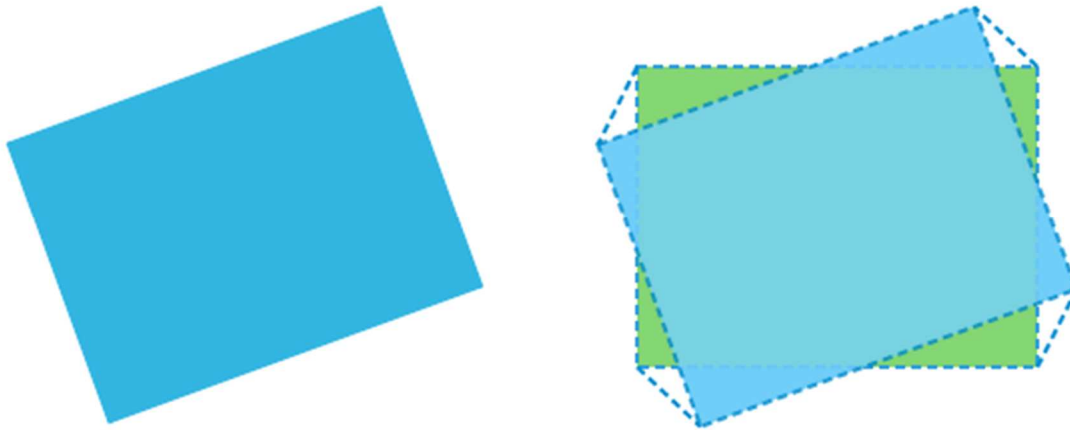
*Slika 5.4. Rotate transformacija*

Na slici 5.4. prikazan je `div` element na kome je primenjena transformacija rotiranja. Na desnoj polovini slike 5.4. prikazan je `div` element u odnosu na svoj početni položaj.

Rotiranje HTML elemenata korišćenjem funkcije `rotate()` podrazumevano se obavlja u smeru kazaljke na satu. Ukoliko je potrebno postići rotiranje u suprotnom smeru, dovoljno je definisati negativnu vrednost ugla:

```
#transformed {  
  transform: rotate(-20deg);  
}
```

Efekat prikazanog koda je kao na slici 5.5.



*Slika 5.5. Rotiranje korišćenjem negativnog ugla*

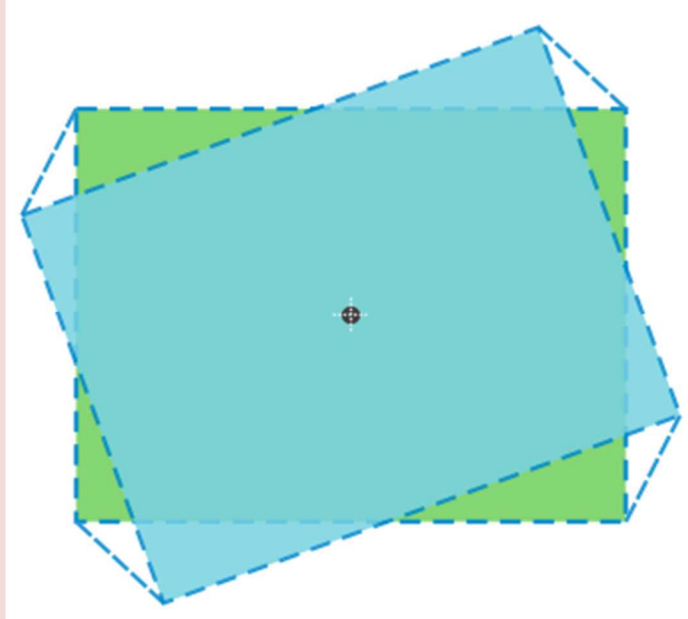
Ono što se može primetiti na slikama 5.4. i 5.5. jeste to da se transformacija rotiranja obavlja u odnosu na centar elementa koji se transformiše. Na osobine takve referentne tačke u odnosu na koju se obavljaju transformacije može se uticati korišćenjem CSS svojstva `transform-origin`.

### **Svojstvo `transform-origin`**

Svojstvo `transform-origin` može se koristiti za postavljanje referentne tačke za obavljanje transformacija. Podrazumevana vrednost ovog svojstva je:

```
transform-origin: 50% 50% 0;
```

Podrazumevana vrednost definiše referentnu tačku transformacije u samom centru elementa, korišćenjem vrednosti od 50% za pomeraj u odnosu na koordinate referentne tačke (koja se nalazi u gornjem levom uglu). Poslednja vrednost definiše pomeraj po z-osi, u slučaju da se sprovode trodimenzionalne transformacije, o čemu će više reči biti u narednoj lekciji. Osobine podrazumevane vrednosti mogle su se videti i na prethodnom primeru rotacije, s obzirom na to da se rotiranje obavljalo oko centra elementa (slika 5.6).



*Slika 5.6. Rotiranje elementa sa podrazumevanom vrednošću transform-origin svojstva*

Upravo ste mogli da vidite da svojstvo `transform-origin` može da prihvati ukupno tri vrednosti:

```
transform-origin: x-offset y-offset z-offset;
```

Pored definisanja sve tri vrednosti, `transform-origin` se može navesti i u obliku sa dve ili jednom vrednošću:

```
transform-origin: x-offset y-offset;  
transform-origin: x-offset;
```

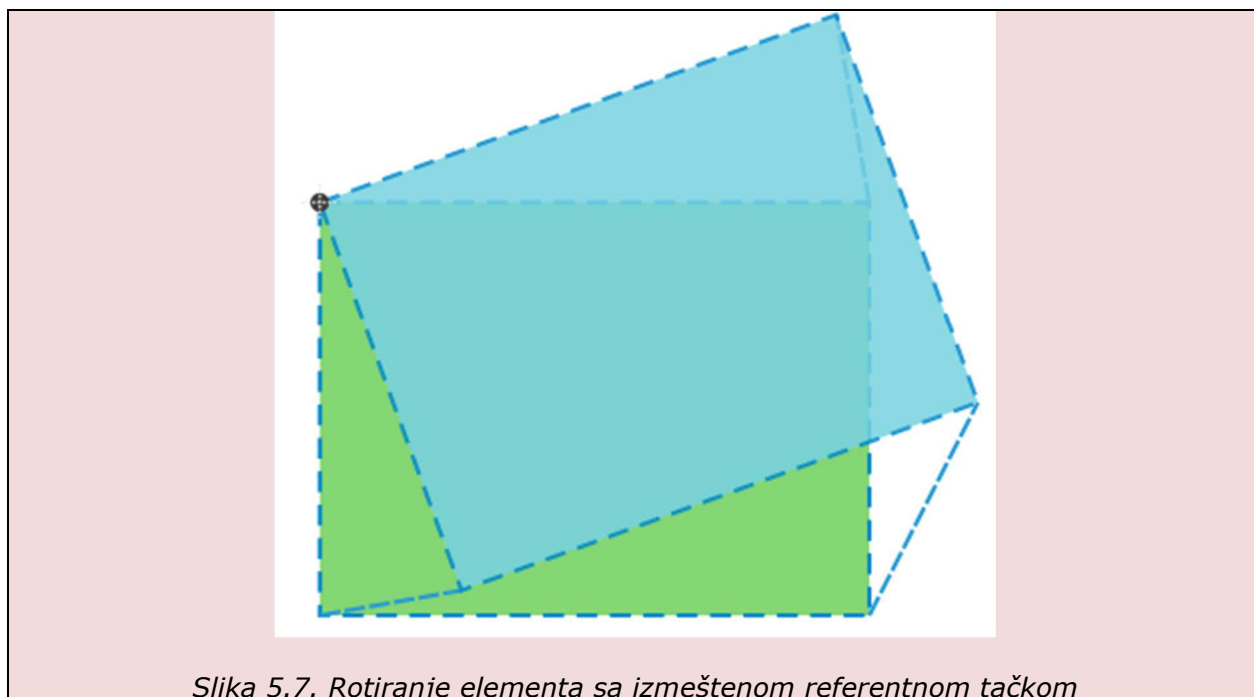
Sintaksa nalaže sledeća pravila:

- kada se navedu tri vrednosti, one se odnose na pomeraje po *x*, *y* i *z* osi, respektivno;
- kada se navedu dve vrednosti, one se odnose na pomeraje po *x* i *y* osi, respektivno, dok se za vrednost pomeraja po *z* osi uzima 0;
- kada se navede jedna vrednost, ona se odnosi na pomeraj po *x* osi, dok se za pomeraj po *y*-osi uzima vrednost 50%, a za pomeraj po *z*-osi vrednost 0.

Kako bi se referentna tačka za obavljanje transformacija poklopila sa početnom tačkom koordinatnog sistema elementa, može se napisati ovako nešto:

```
transform-origin: 0px 0px;
```

U ovakvom slučaju, rotacija će biti obavljena kao na slici 5.7.



Slika 5.7. Rotiranje elementa sa izmeštenom referentnom tačkom

## Povećavanje i smanjivanje

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

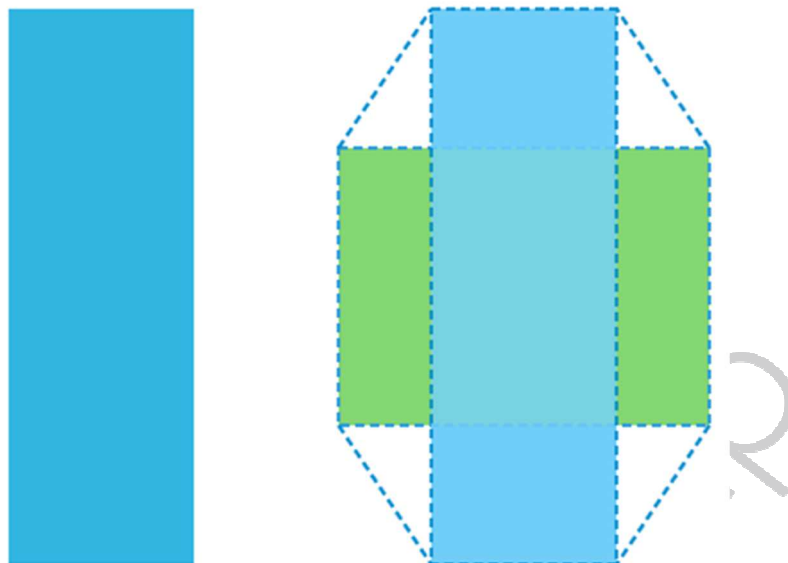
```
scale(width_ratio, height_ratio)
```

Funkcija `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 funkcije `scale()`:

```
#transformed {  
  transform: scale(0.5, 2);  
}
```

U prikazanom primeru, definisano je da će se širina elementa smanjiti dva puta, a visina povećati dva puta. Tako primer proizvodi efekat kao na slici 5.8.



Slika 5.8. Scale transformacija

Na levom strani slike 5.8 prikazan je finalni izgled transformisanog elementa. Na desnoj strani slike može se videti odnos između početnog (zelena boja) i finalnog izgleda elementa (plava boja).

## Zakošavanje

CSS poseduje tri različite funkcije kojima je moguće obaviti zakošenje elemenata:

- `skewX(angleX)`
- `skewY(angleY)`
- `skew(angleX, angleY)`

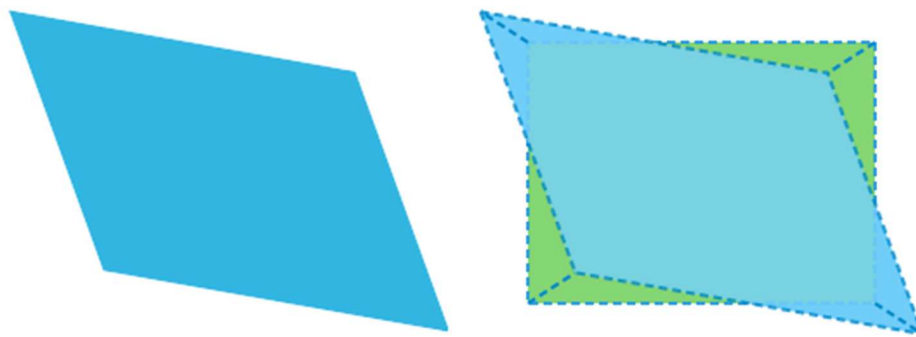
Funkcija `skewX()` vrši zakošavanje elementa po x-osi. Funkcija `skewY()` čini to isto, ali po y-osi, dok funkcija `skew()` može da zakosi element po obe ose.

Primer korišćenja ove transformacije je sledeći:

```
#transformed{
    transform: skew(20deg, 10deg);
}
```

Efekat je kao na slici 5.9.





Slika 5.9. Skew transformacija

### Pitanje

Kako se u CSS-u naziva transformacija pomeranja elementa?

- **translate**
- skew
- rotate
- pitch

### Objašnjenje:

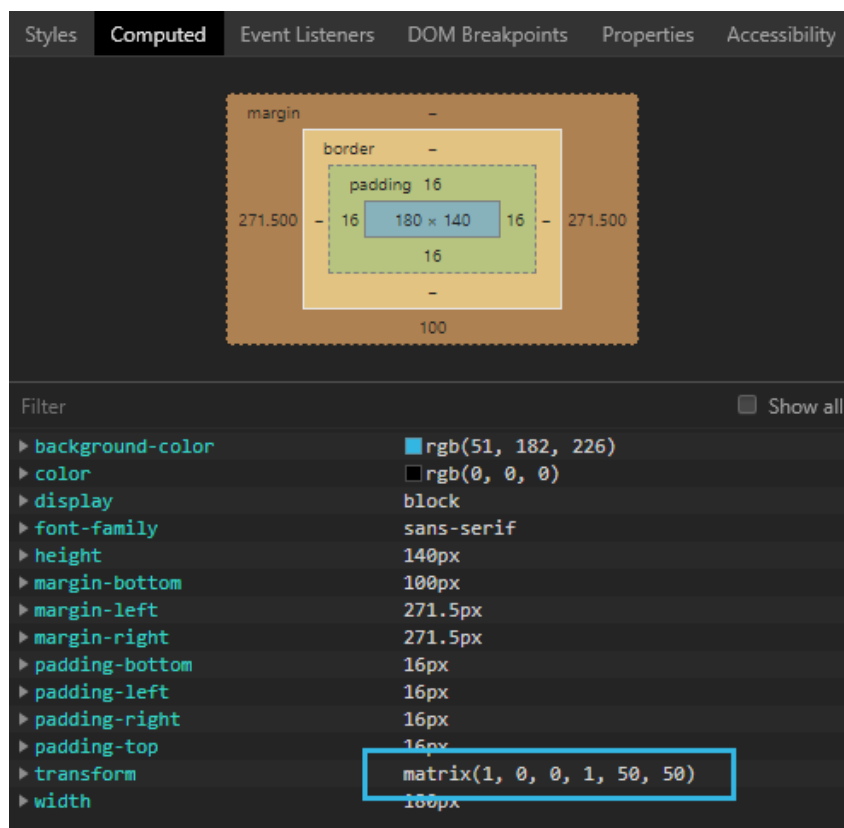
*Kako bi se element pomerio u odnosu na svoj normalni položaj, moguće je koristiti translate transformaciju.*

## Matrix

Postizanje 2D transformacija moguće je i korišćenjem funkcije **matrix()**, koja objedinjuje sve CSS transformacije. Ono što je posebno zanimljivo jeste to da browser sve do sada opisane 2D transformacije u pozadini prevodi u `matrix()` funkciju. Potvrda toga može se pronaći i unutar browsera, uvidom u Computed panel. Na primer, uzmimo da je na nekom elementu definisana sledeća transformacija:

```
transform: translate(50px, 50px);
```

Uvidom u *Computed* panel jasno se može videti da browser prikazanu transformaciju prevodi u oblik koji podrazumeva upotrebu `matrix()` funkcije (slika 5.10).



Slika 5.10. Prikaz proračunatih (computed) vrednosti za selektovani element unutar Google Chrome Developer Tools

Sa slike 5.10. može se videti da je transformaciju pomeranja browser u pozadini pretvorio u `matrix()` funkciju sa određenim vrednostima parametara (na slici 5.10. označeno plavim okvirom).

Funkcija `matrix()` prihvata šest parametara:

```
matrix(scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY());
```

Ono što je zanimljivo zapaziti jeste to da funkcija `matrix()` prihvata vrednosti za obavljanje skaliranja, zakošavanja i pomeranja, ali ne i rotiranja. Naime, funkcija `matrix()` rotiranje obavlja kombinovanjem skaliranja i zakošavanja:

```
transform: rotate(20deg);
```

Upravo prikazanu liniju browser pretvara u:

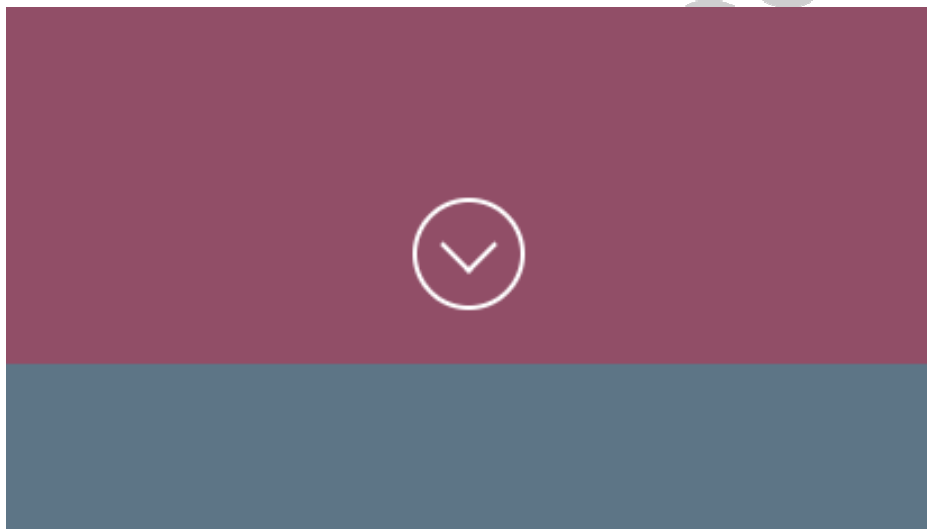
```
transform: matrix(0.939693, 0.34202, -0.34202, 0.939693, 0, 0);
```

## Primer – Kreiranje scroll down strelice

Primer koji će ilustrirati upotrebu pristupa obrađenih u ovoj lekciji biće kreiranje scroll down strelice. Reč je o funkcionalnosti koju poseduju gotovo svi moderni web sajтови, pogotovu ako je reč o web sajtovima one page dizajna. Scroll down strelica obično se nalazi na dnu početne sekcije web stranice i ima dve osnovne namene:

- korisniku stavlja do znanja da stranica poseduje još sadržaja koji će skrolovanjem postati vidljiv;
- omogućava korisniku lak način za prelazak na sledeću sekciju sadržaja.

Iako se scroll down strelica može kreirati na bezbroj načina, među kojima prednjači upotreba png slika ili vektora za predstavljanje strelice, u nastavku će scroll down strelica biti kreirana isključivo korišćenjem običnih HTML elemenata. Efekat koji će na kraju biti dobijen ilustrovan je slikom 5.11.



*Slika 5.11. Scroll down strelica*

Za realizaciju ovog primera, pored različitih tehnika opisanih u ovoj lekciji, biće iskorišćeni i pseudoelementi, opisani u prethodnom modulu ovog kursa.

HTML struktura kojom će biti predstavljena scroll down strelica je sledeća:

```
<a id="scroll-down-arrow" href="#next-section"></a>
```

Kao što možete videti, scroll down strelica je realizovana korišćenjem jednog `a` elementa. Primenom posebne stilizacije, učinjeno je da takav element dobije izgled kao na slici 5.11. (krug unutar koga se nalazi strelica koja pokazuje nadole):

```

#scroll-down-arrow {
    position: absolute;
    bottom: 24px;
    width: 46px;
    height: 46px;

    margin: 0 auto;
    left: 0;
    right: 0;

    border: 2px solid #fff;
    border-radius: 100%;
}

#scroll-down-arrow::before {
    position: absolute;
    content: '';
    width: 16px;
    height: 16px;
    left: 0;
    right: 0;
    top: 10px;
    margin: 0 auto;
    border-left: 2px solid #fff;
    border-bottom: 2px solid #fff;
    transform: rotate(-45deg);
}

```

Analizom prikazane stilizacije može se zaključiti sledeće:

- prikaz kruga dobijen je postavljanjem okvira sa zaobljenjem od 100% na samom a elementu;
- prikaz strelice koja pokazuje nadole dobijen je posebnim stilizovanjem `::before` pseudoelementa;

Ovo je idealan primer koji, pored transformacija, ilustruje pravu moć pseudoelementa. Oni se veoma često koriste za obavljanje ovakvih i sličnih zahvata prilikom stilizovanja. Pri tome se misli na kreiranje određenog vizualnog dodatka koji se ni na koji način ne može dobiti stilizovanjem samog elementa. Pseudoelement u primeru ne poseduje nikakav tekstualni sadržaj, ali da bi uopšte bio prikazan na stranici, neophodno je definisati vrednosti `content` atributa. Upravo zbog toga, ovaj atribut u primeru poseduje takozvanu praznu vrednost, koja se dobija navođenjem apostrofa između kojih nema ničega (").

Strelica koja se može videti unutar kruga scroll down elementa zapravo je dobijena postavljanjem levog i donjeg okvira pseudoelementa. Zatim je takav pseudoelement, sa dva vidljiva okvira, zarotiran za -45 stepeni, kako bi se strelica usmerila nadole.

## Rezime

- Transformacije omogućavaju transformaciju HTML elementa uticanjem na njihove vizualne karakteristike.
- Transformisanje HTML elemenata postiže se upotrebom CSS svojstva `transform`.
- Različite CSS transformacije definišu se različitim CSS funkcijama koje se navode kao vrednost svojstva `transform`.
- Da bi se element pomerio u odnosu na svoj normalni položaj, koristi se CSS funkcija `translate()`.
- CSS metoda `rotate()` omogućava rotiranje elementa.
- Za povećavanje i smanjivanje HTML elemenata moguće je koristiti CSS funkciju `scale()`.
- CSS funkcija `skewX()` vrši zakošavanje elementa po x-osi, dok funkcija `skewY()` čini to isto, ali po y-osi; funkcija `skew()` može da zakosi element po obe ose.
- Funkcija `matrix()` prihvata šest parametara i na taj način omogućava da se definišu vrednosti za sve transformacije odjednom.
- Web pregledač sve transformacije u pozadini prevodi u oblik iskazan `matrix()` funkcijom.

