

Síkbeli (2D-s) transzformációk

Síkbeli transzformációk esetén a vízszintes (X) és/vagy függőleges (Y) tengely mentén egy elemen eltolás, forgatás, nagyítás/kicsinyítés vagy döntés (ferdítés) hajtható végre.

A `transform` (átalakít) tulajdonság értékei, azaz a transzformációs függvények definiálják, hogy milyen művelet hajtandó végre és a zárójelben megadott szám(ok) pedig az átalakítás mértékét határozza(k) meg. Értékei nem öröklődnek!

1. **Áthelyezés / eltolás:** `translate`

= egy elem középpontját a megadott X és/vagy Y irányba tolja el

kódolás	leírás	példa
<code>translateX(ertek)</code>	<i>eltolás vízszintesen</i> az X tengely mentén, ahol a pozitív szám a jobbra, a negatív szám a balra tolást jelenti (px-ben megadva vagy %-érték esetén az adott elem szélességi értékének arányában)	<pre>#div_tX { transform: translateX(50px); }</pre>
<code>translateY(ertek)</code>	<i>eltolás függőlegesen</i> az Y tengely mentén, ahol a pozitív szám a lefelé, a negatív szám a felfelé tolást jelenti (px-ben megadva vagy %-érték esetén az adott elem magassági értékének arányában)	<pre>#div_tY { transform: translateY(-20%); }</pre>
<code>translate(ertekX, ertekY)</code>	<i>eltolás egyidőben a vízszintes és a függőleges irányba is</i> , ahol az első adat a vízszintes, a második adat a függőleges eltolás mértékét jelenti (vessző választja el egymástól a két értéket, px-ben és %-ban is megadható az értékük)	<pre>#ketirany_t { transform: translate(15px,25px); }</pre>

2. **Átméretezés:** scale

= egy elem középpontjából kiindulva a megadott X és/vagy Y irányba megváltoztatja az elem méretét, azaz az adott irányba nagyítja (1-nél nagyobb szám esetén) vagy kicsinyíti (1-nél kisebb szám esetén)

kódolás	leírás	példa
scaleX(ertek)	átméretezés (nagyítás / kicsinyítés) az X tengely mentén (azaz vízszintes irányú széthúzás vagy összenyomás)	<pre>#div_kicsinyitX { transform: scaleX(0.69); }</pre>
scaleY(ertek)	átméretezés (nagyítás / kicsinyítés) az Y tengely mentén (azaz függőleges irányú széthúzás vagy összenyomás)	<pre>#div_nagyitY { transform: scaleY(1.8); }</pre>
scale (ertekX, ertekY)	átméretezés egyidőben a vízszintes és a függőleges irányba is , ahol az első adat a vízszintes, a második adat a függőleges átméretezés mértékét határozza meg (vessző választja el egymástól a két értéket)	<pre>#aranytalanul_nagyit { transform: scale(1.2,1.8); }</pre>

3. **Elforgatás:** rotate

= egy elem középpontja körül adott szöggel (deg) való elforgatást hajt végre

kódolás	leírás	példa
rotate(szogertek)	adott szöggel való elforgatás az elem origója körül úgy, hogy a pozitív szögérték az óramutató járásával megegyező irányt, a negatív szögérték pedig az óramutató járásával ellenkező irányt jelenti	<pre>#balra_forgat { transform: rotate(-30deg); } #jobbra_forgat { transform: rotate(45deg); }</pre>

4. **Döntés / ferdítés: skew**

= a megadott X és/vagy Y irányban ferdén megdönti az elemet a megadott szögértékkel (deg)

kódolás	leírás	példa
skewX(szogertek)	döntés a vízszintes tengely mentén (a pozitív érték esetén az elem balra dől, negatív érték esetén jobbra dől)	<pre>#div_dont_balra { transform: skewX(-30deg); }</pre>
skewY(szogertek)	döntés a függőleges tengely mentén (a pozitív érték esetén az elem lefelé dől, negatív érték esetén felfelé dől)	<pre>#div_dont_le { transform: skewY(60deg); }</pre>
skew (szogertekX,szogertekY)	döntés egyidőben a vízszintes és a függőleges irányba is , ahol az első adat a vízszintes, a második adat a függőleges döntés mértékét határozza meg (vessző választja el egymástól a két értéket)	<pre>#ketiranyba_dont { transform: skew(25deg,45deg); }</pre>

5. Több transzformációs művelet egyidejű alkalmazása

Egy elemen egyidejűleg természetesen több transzformációs művelet is végrehajtható. Ezekben az esetekben kétféle lehetőségünk van:

- a) **összevont alakban, vessző nélkül, a végrehajtandó műveletek sorrendjében felsorolva kell megadni a transzformációs függvényeket és az azokhoz tartozó értékeket**, például:

```
#tobbfele_transzformacio {
  transform: translate(30%,20px) scaleX(0.75) rotate(-60deg);
}
```

- b) **a `matrix(...)` függvény alkalmazásával hat paraméter adott sorrendű (`scaleX`, `skewY`, `skewX`, `scaleY`, `translateX`, `translateY`) megadásával** lehet elérni a kívánt hatást, például:

```
#tobbfele_transzformacio2 {
  transform: matrix(1, 0, 0.5, 1, 150, 0);
}
```

Egyéb beállítások és lehetőségek

Amennyiben a transzformáció origója nem esik egybe a transzformálandó elem középpontjával, akkor ezen a **transform-origin** tulajdonsággal változtathatunk. (A korábbi példák alapértelmezetten mind az elem közepére vonatkoztak.)

kódolás	leírás	példa
transform-origin: Xirany Yirany;	<p><i>a transzformáció origójának helyzete az elem bal felső sarkához képest az X és az Y tengely irányában</i> úgy, hogy az első érték mindig a vízszintes, a második érték a függőleges irányú elmozdítás adatát jelenti és a két érték között szóköz található</p> <ul style="list-style-type: none"> • megadható értékek: %, hosszúsági érték, szavakkal: X-tengely esetén a <i>left</i>, <i>center</i> és <i>right</i>, az Y-tengelynél a <i>top</i>, <i>center</i> és <i>bottom</i> használható • ha csak egyetlen paramétert adunk meg, akkor a viszonyítási pont csak vízszintesen mozdul el, mert a függőleges paraméter automatikusan a center érték lesz • pozitív érték esetén balra/fel, negatív értéknél jobbra/le mozdul el a viszonyítási pont 	<pre>#div_jobb_also_sarok_forgat { transform: rotate(45deg); transform-origin: right bottom; } #div_baloldal_felsoel_marad { transform: scale(1.5,1.5); transform-origin: left top; }</pre>

A **transform-origin** tulajdonság segítségével akár a transzformálandó elemen kívüli viszonyítási pontot is meghatározhatunk, így pl. az elemtől távolabb eső tengelyre tudunk tükrözni elemeket:

(demó: https://www.w3schools.com/cssref/trycss3_transform-origin_inuse.htm)

```
div { width: 200px; height: 200px; background-color: green; }

div:hover {
  transform: scaleX(-1);
  transform-origin: 300px;
}
```

Mivel az egyes alakzatokba tartalom (pl. szöveg vagy kép) is helyezhető, és a tartalom követi a CSS-ben a (külső) alakzatra kódolt utasításokat, ezért ezeket a transzformációs utasításokat arra is használhatjuk, hogy egyes elemeket kiemeljünk a környezetük-ből. Tipikus példa: rámutatás után a választott kép kinagyítódik, az elem megdől vagy az egyik irányba elfordul.