

Átmenetek (transitions)

A pszeudo-osztály kijelölőknél tanult `:hover` állapothoz (jelentése: rámutatás egérrel) és a `:active` állapothoz (jelentése: aktiválás lenyomva tartott bal egérgombbal) szinte mindenféle formázási lehetőség hozzárendelhető. Például ha az egérkurzorral rápozicionálunk a kijelölt elem fölé vagy lenyomva tartjuk az egérgombot, akkor az elem egy előre megadott új formázási állapotba kerülhet. **A két állapot valamelyikének alkalmazásával az elem formátumában ugrásszerű változás jön létre - ezt tudjuk folyamatos átmenetté alakítani a CSS `transition` tulajdonságával, illetve többszöri átmenetet (azaz ismétlést) az animációval kiegészítve lehet definiálni.**

Az átmenetek (transitions) olyan prezentációs effekteket jelentenek, amelyek alkalmazása során egyes tulajdonságok az idő függvényében folyamatosan mennek át az egyik állapotból (értékből) egy másik állapotba (értékbe). Az átmenet végi állapot természetesen megegyezik a hirtelen átmenetnél kapottal (tehát ha nem használjuk a *transition* tulajdonságot). Az állapot megszűntével az alakzat a folyamatos átmenet tükörképeként visszaalakul a kiinduló állapotba.

Az átmenetek definiálásához **két kötelezően** megadandó (mi változik és mennyi idő alatt) és **két opcionális** paraméter (késleltetés és időzítés) áll a rendelkezésre.

Kötelezően megadandó tulajdonságok

Kötelezően megadandó a **transition-property** (a változáson áteső tulajdonság vagy a tulajdonságok listája) és a **transition-duration** (az átmenet vagy átmenetek időtartama). Ha az időtartamot nem adjuk meg, akkor az átmenetnek nincs hatása, mert az alapértelmezett érték a 0 másodperc.

transition-property: none / all / property / initial / inherit;

Meg kell adni **a változással érintett CSS-tulajdonságot** vagy alkalmazni kell a megfelelő kulcsszó egyikét (**none** – egyiket sem akarjuk módosítani, **all** – mindegyik tulajdonságot módosítani akarjuk, **initial** – alapértelmezett értékek, **inherit** – szülőtől örökölt értékek).

pl. `#div_1 { transition-property: width; }`

transition-duration: time / initial / inherit;

Minden egyes változtatott értékhez meghatározhatjuk **az átmenet időtartamát**, idejét másodpercben megadva.

pl. `#div_1 { transition-duration: 5s; }`

Egyszerre azonban **több tulajdonságot** is megváltoztathatunk és ezek **változtási időtartama is eltérhet** (különbözhet) egymástól.

- A **megváltoztatandó tulajdonságokat** a **transition-property** értékeként, vesszővel és szóközzel felsorolva kell megadni; a sorrend tetszőleges.

pl. `transition-property: width, height, color, margin;`

- Az **egyes változások időtartamát** is tulajdonságoként, a **transition-duration** értékeként, vesszővel és szóközzel elválasztva kell felsorolni, ahol a sorrend kötelezően a **transition-property** értékeinek sorrendjét követi, így fog minden változtatandó tulajdonsághoz egyértelműen egy-egy időtartam tartozni. A különböző tulajdonságok változásaihoz különböző időtartamok tarthatnak.

pl. `transition-duration: 2s, 5s, 3s, 4s;`

Opcionálisan megadható tulajdonságok

A folyamatos átmenethez opcionálisan megadható két további paraméter: a **transition-delay** (az átmenet késleltetése, azaz mikor kezdődjön az átmenet) és a **transition-timing-function** (az átmenetek időzítésének függvénye, az az időbeli lefolyás egyenletessége vagy egyenlőtlensége).

transition-delay: time / initical / inherit;

A **tényleges átmenet kezdetének időbeli eltolódását definiálja**, vagyis azt, hogy mennyi idővel később kezdődjön el a változás a változásra vonatkozó utasítás kiadásához képest.

- az értéket másodpercben kell meghatározni
- 0 értéknél azonnal megkezdődik a változás (ez az alapértelmezett érték)
- egyébként a megadott pozitív érték nagyságú késleltetéssel indul el
- ha véletlenül negatív értéket definiálunk, akkor nem indul el előbb a változás, mint az azt kiváltó utasítás, de a tulajdonság kezdeti értéke nem az egyébként induló érték, hanem egy olyan közbenső állapot lesz, mintha a negatív késleltetés ideje alatt már bekövetkezett változásról indulna a folyamat
- az egyes változtatott tulajdonságokra egyesével külön is előírható ez az érték (vesszővel és szóközzel kell elválasztani egymástól az értékeket) - azt a sorrendet kell követni, mint amelyet a **transition-property**-nél megadtunk, így egyértelműen meghatározhatjuk minden egyes változás kezdetéhez egy-egy késleltetést

pl. `transition-delay: 1s, 2s, 0, 3s;`

transition-timing-function:

linear / ease / ease-in / ease-out / ease-in-out / step-start / step-end / steps(int, start|end) / cubic-bezier(n,n,n,n) / initial / inherit;

Az **átmenetek időzítésének függvényét** ennek értékével határozhatjuk meg. Ez az érték azt jelenti, hogy az átmenet kezdeti és végpontja közötti közbenő időben milyen legyen a változó érték sebességváltozásának módja - pl. lineáris, lassuló, gyorsuló.

- az átmenet lefolyása négyzetes [Bezier-görbével](#) írható le: amely négy jellegzetes pontjával definálható - a függvény megadásakor ezt kell megadnunk (a részleteket lásd a hivatkozásban...) - pl. cubic-bezier(x1,y1,x2,x2)
- *névvel* is megadhatók az átmenetek:
 - linear: egyenletes sebességű
 - ease: gyors kezdésből lassul a végére (ez az alapértelmezett)
 - ease-in: egyre gyorsul az elejétől kezdve
 - ease-out: hasonló az ease-hez, de kevésbé gyors az elején
 - ease-in-out: gyorsul az elején, lassul a végén
- a rövid átmeneteknél gyakorlatilag mindegy, hogy melyiket használjuk
- ebben az esetben is használható az egyes tulajdonságokra eltérő függvény (vesszővel és szóközzel kell itt is elválasztani)

pl. `transition-timing-function: linear,ease,linear,ease-in-out;`

Összevont alak

A legegyszerűbben az összevont **transition** tulajdonság is használható úgy, hogy elsőként a módosítandó értéket, majd szóközzel attól elválasztva a végrehajtás időtartamát adjuk meg (ez a kettő kötelező). Ezt követheti az időzítés függvénye, majd az időzítés értéke. (Egyelőre a transition összevont alakot egyes böngészőverziók nem tudják értelmezni és hirtelen átmenetet hoznak létre, ezért javasolt a paraméterek egyesével történő kódolása.)

Az átmenet leírásának összevont alakú szintaxisa:

transition: property duration timing-function delay;

Példa-1: csak a szélességet változtatja meg 3 mp alatt:

```
#doboz1 { width: 100px; height: 50px; background-color: red;
          color: white; transition: width 3s; }
```

Példa-2: a magasságot és a betűszínt is megváltoztatja (eltérő időtartammal):

```
#doboz2 { width: 100px; height: 50px; background-color: red;
          color: white; transition: width 3s, color 5s; }
```

Hol használható célszerűen a folyamatos átmenet?

Példák:

- képre történő rámutatáskor lassan és nagyobb méretben jelenik meg a kép
- az elemre egérrel történő rápozicionálást követően láthatóbbá (kevésbé átlátszóvá) válik az elem
- az adott elemre történő rámutatáskor egy információs szöveg jelenik meg az elem helyén vagy rajta (pl. egy képen információs szöveg)
- rámutatáskor mozduló képek
- menüpontok kiélesítése rámutatáskor