

Animációk (animations)

Az animáció az egyszeri, folyamatos átmenet (a transition) kiterjesztése.

A `transition`-nél egy vagy több tulajdonság megadott kezdő értékei és végső értékei között a közbenső értékeket a böngésző határozza meg és csak az időbeli lefolyás kódolható az időfüggvénnyel (`transition-timing-function`), az átmenet időtartamával (`transition-duration`) és az időzítéssel (`transition-delay`).

Az animációk (animations) esetében a kezdő és a végső értékek közötti átmenet finomabban / részletesebben meghatározható, az átmenetek többször is végrehajthatók (ismételhető, akár végtelen számban, tehát időkorlát nélkül, akár folyamatosan is), még a lejátszás iránya (mindig előlről kezd vagy oda-vissza játssza le) is megadható.

Az átmenet és az animáció kódolásnak összefüggései

A fentieknek megfelelően az animáció kódolása részben a folyamatos átmenettel analóg módon történik, részben azonban annál bővebb és részletesebb meghatározási lehetőségeket biztosít. Az alábbi táblázatban ez látható:

típus	folyamatos átmenet (transition)	animáció (animation)
tulajdonság/név	transition-property	animation-name
időtartam	transition-duration	animation-duration
időzítés függvénye	transition-timing-function	animation-timing-function
időzítés	transition-delay	animation-delay
ismétlésszám	----	animation-iteration-count
irány	----	animation-direction
kitöltési mód	----	animation-fill-mode

Az egyes tulajdonságok részletezése

- A *három analóg tulajdonság* (`duration`, `timing-function`, `delay`) lehetséges értékei és azok értelmezése az átmenetnél és az animációnál egyeznek.
- Az **animation-iteration-count** mutatja az *ismétlések számát*, lehetséges értéke mindig *pozitív szám* vagy a *végtelen* (*infinite*) érték lehet.

- Az **animation-direction** értékei az *irány* megadására: **normal** | **reverse** | **alternate** | **alternate-reverse** | **initial** | **inherit**
 - **normal**: a ciklus végén a megváltoztatott tulajdonságok értéke visszaugrik a kezdő állapotba és a következő ciklus ismét onnan indul
 - **alternate**: a páratlan ciklusok az elejéről, a párosak pedig a végétől (tehát visszafelé) hajtódnak végre
 - **alternate-reverse**: a páratlan ciklusok a végétől visszafelé, a párosak pedig az elejéről hajtódnak végre (vagyis ez az előző fordítottja)
- Az **animation-fill-mode** tulajdonságnak az egyszeri átmenetnél van jelentősége, amikor az átmenet előtti és utáni állapotra lehet megkötést tenni a **forward**, a **backward** és a **both** értékkel (alapértelmezett: **none**)
 - **forward**: az elem megtartja az utolsó kulcskép által beállított stílusértékeket
 - **backward**: az elem megkapja az első kulcskép által beállított stílusértékeket és azt megtartja az animáció késleltetési ideje alatt is
 - **both**: az animáció előre és hátra egyaránt követi a szabályokat, kiterjesztve az animáció tulajdonságait mindkét irányban
 - **none**: az animáció nem alkalmaz semmilyen stílust az elemre a végrehajtás előtt vagy után

Közbenső értékek definiálása

Az animáció során a folyamatos átmenetet azzal haladhatjuk meg, hogy **közbenső értékeket** definálunk a **@keyframes** segítségével. Ha ezt meghatározzuk, akkor az ebben megadott névvel (**animation-name**) kapcsolhatjuk hozzá azt az animációs tulajdonságokhoz.

A @keyframes (kulcs képkockák) szintaxisa:

```
@keyframes animation-name { keyframes-selector {CSS-styles;} }
```

Példák a @keyframes-re:

- **from ... to ...** módon (a **from** a kezdő-, a **to** pedig a zárópontot jelöli):


```
@keyframes pelda-1 {
  from { kezdotulajdonsag: ertekek-1; }
  to { zarotulajdonsag: ertekek-2; }
}
```
- **százalékokkal**: egy teljes animációs ciklusnak a kezdőpontja a 0%, a végpontja a 100%, a közbenső pontokat pedig a két érték közötti %-ok jelölik (csak a kezdő- és a végpont megadása kötelező, a belső pontoké opcionális)

```
@keyframes pelda-2 {  
  0% { kezdotulajdonsag: ertek-1; }  
  ..% { belsotulajdonsag: ertek-2;}  
  ..% { belsotulajdonsag: ertek-3;}  
  100% { zarotulajdonsag: ertek-z;}  
}
```

Példa

A "minta" nevű animáció egy piros kis téglalapot átalakít egy zöld lekerekített sarkú nagyobb négyzetté, végül egy óriási kék körre. Az átalakítást 10 másodperc alatt (lineárisan) hajtja végre, majd ezt követően végtelenszer megismétli, oda-vissza elvégezve azt.

Elsőként a kulcsképet kódoljuk, vagyis a kezdő-, a közbenső és a végállapotokat:

```
@keyframes minta {  
  0% { background-color: red; width: 10px; height: 30px;  
      border-radius: 0%; margin: 10px 0 0 25px; }  
  50% { background-color: green; width: 150px;  
      height: 150px; border-radius: 50px;  
      margin: 10px 0 0 100px; }  
  100% { background-color: blue; width: 500px;  
      height: 500px; border-radius: 250px;  
      margin: 10px 0 0 200px; }  
}
```

Ezt követően hozzákapcsoljuk a kezdő elemhez az animációt és megadjuk az animáció tulajdonságait:

```
div {  
  animation-name: minta;  
  animation-timing-function: linear;  
  animation-iteration-count: infinite;  
  animation-duration: 10s;  
  animation-direction: alternate;  
}
```