




CSS Animation Methods



GitHub 'dan beni takip etmeyi unutma  [@burakkrt](#)

Not : Aşağıda tanımlanan rotate adındaki animasyon tüm animasyon metotlarını anlatırken kullanılacaktır. rotate kavramı aşağıdaki animasyon 'dan gelmekte olacak.

```
@keyframes rotate {  
  from{  
    transform: rotate(0);  
  }  
  to {  
    transform: rotate(360deg);  
  }  
}
```

animation-name()

Oluşturulan animasyonu belirli bir elemente ekleme.

Önemli: Animasyonu tanımlarken süresini de tanımlamak zorundayız aksi halde animasyon görüntüleyemeyeceğimiz hızda çalışır ve bunu fark etmeyiz.

```
.box{  
  animation-name: rotate; /* rotate adındaki animasyonu box class 'ına tanımladık */  
  animation-duration: 0.7;  
}
```

animation-duration()

Animasyonu toplam ne kadar sürede tamamlanacağını belirtir.

```
.box{  
  animation-name: rotate;  
  animation-duration: 1s; /* toplam 1 saniyede animasyonu oynat */  
}
```

animation-direction()

Animasyonun ileriye mi yoksa geriye doğru mu oynatılacağını belirtir.

```
animation-duration: normal; /* animasyonu ileriye doğru oynat (default) */
animation-duration: reverse; /* animasyonu geriye doğru oynat*/
/* animasyonu ilk ileriye doğru oynatıp bitirir ve ardından geriye doğru oynatır */
animation-duration: alternate;
/* animasyonu ilk geriye doğru oynatıp bitirir ve ardından ileriye doğru oynatır */
animation-duration: alternate-reverse;
```

animation-iteration-count()

Animasyonun kaç kere oynatılacağını belirtir.

```
animation-iteration-count: 4; /* animasyonu 4 kere oynatıp bitirir */
animation-iteration-count: 1.5; /* animasyonu 1 kere tam 1 kere yarım oynatıp bitirir */
animation-iteration-count: infinite; /* animasyonu sonsuz kere oynatır.*/*
```

animation-timing-function()

Animasyonun her döngüde nasıl bir hızda ilerleyeceğini belirler.

```
/* başlangıçtan sona kadar eşit hızla ilerler */
animation-timing-function: linear;

/* başlangıçta yavaş sonunda yavaş */
animation-timing-function: ease;

/* başlangıç yavaş sonlara doğru hızlı */
animation-timing-function: ease-in;

/* başlangıç hızlı sonlara doğru yavaş*/
animation-timing-function: ease-out;

/* başlangıç yavaş ortası hızlı sonlar yavaş*/
animation-timing-function: ease-in-out;

/* cubic-bezier ile özelleştirilmiş animasyon hızı oluşturulabilir*/
animation-timing-function: cubic-bezier(0.42,0,0.8,1);

animation-timing-function: steps(adımSayısı, atlamaAnahtarKelime);
/* Animasyonu 5 adımda tamamla (%20, %20, %20, %20, %20) */
animation-timing-function: steps(5, start)
```

animation-delay()

Animasyona tanımlar ve başlamadan önce ne kadar süre bekleyeceğini belirtir.

```
animation-delay: 1s; /* 1saniye sonra animasyonu oynatmaya başlar */
animation-delay: -3s; /* animasyonu 3. saniyesinden itibaren başlatır */
```

animation-fill-mode()

Belirli bir süre içinde nasıl davranacağını tanımlayan bir özelliktir.

```
@keyframes rotate {
  from{
    transform: rotate(45deg);
  }
  to {
    transform: rotate(180deg);
  }
}

/* Animasyona başlangıç değeri rotate 45deg verdik ve bitişi 180deg olacağını söyledik.
.box class 'ında rotate ile ilgili bir değer olmadığına dikkat edelim. */

/* none methodu animasyon bittikten sonra animasyonda tanımlanan özelliklerin
hiçbirinin .box class 'ına aktarılmayacağını belirtir. Yani animasyon 45 dereceden
180 dereceye dönmeye başlar ve animasyon bitince animasyonda tanımlanan tüm
methodlar .box class 'ından kaldırılır. */

.box {
  animation-name: rotate;
  animation-duration: 3s;
  animation-fill-mode: none;
}

/* forwards 'da animasyonda tanımlanan son (yani bitiş) methodlarının değerleri .box
class 'ına eklenir. Yani burada son adım olan rotate(180deg) animasyon bitiminde
.box class 'ına aktarılır ve animasyon bitse bile .box class 'ı 180 derece döndürülmüş
şekilde kalmaya devam eder */

.box {
  animation-name: rotate;
  animation-duration: 3s;
  animation-fill-mode: forwards;
}

/* ek olarak backwards ve both özellikleride verilebilir ama kullanılam gerek yoktur. */
```

animation-play-state()

İlgili animasyonun durdurulması veya oynatılması durumunu kontrol etmemizi sağlar.

```
animation-play-state: running; /* animasyon etkin, devam ediyor */
animation-play-state: paused; /* animasyon durduruldu, devam etmiyor */
```

```
/* Javascript ile animasyonu durdurup, devam ettirebiliriz
running ve paused adında iki class oluşturdum ve yukarıdaki methodları tanımladım.
Bu class 'ları butona her tıklandığında animasyon bulunan elemente ekleyerek animasyonu
durdurup veya yeniden başlatabiliriz.
*/

.paused{
  animation-play-state: paused;
}

.running{
  animation-play-state: running;
}

document.querySelector('#button').addEventListener("click", () => {
  let tt = document.querySelector('#animasyonDiv');
  if(tt.classList.contains("running")){
    tt.classList.remove("running");
    tt.classList.add("paused")
  } else {
    tt.classList.remove("paused");
    tt.classList.add("running")
  }
})
```