

Animacja umożliwia stopniowe przechodzenie elementu z jednego stylu do drugiego.

Możesz zmienić dowolną liczbę właściwości CSS, ile razy chcesz.

Aby użyć animacji CSS, musisz najpierw określić kilka klatek kluczowych dla animacji.

Klatki kluczowe zawierają style, które element będzie miał w określonych momentach.

Reguła @keyframes

Gdy określisz style CSS w regule @keyframes, animacja będzie stopniowo zmieniać się od bieżącego stylu do nowego stylu w określonych momentach.

Aby animacja działała, musisz powiązać animację z elementem.

Poniższy przykład wiąże animację „zmiana” z elementem <div>. Animacja będzie trwała 6 sekund i będzie stopniowo zmieniać kolor tła elementu <div> z „szarego” na „niebieski”

```
<style>
div {
  width: 100px;
  height: 100px;
  background-color: pink;
  animation-name: zmiana;
  animation-duration: 6s;
}

@keyframes zmiana {
  from {background-color: grey;}
  to {background-color: blue;}
}
</style>
```

Właściwość **animation-duration** określa, jak długo powinna trwać animacja. Jeśli właściwość czas trwania animacji nie zostanie określona, animacja nie wystąpi, ponieważ wartość domyślna to 0s.

W powyższym przykładzie określiłam, kiedy styl ulegnie zmianie, używając słów kluczowych „od” i „do” (co oznacza 0% (początek) i 100% (zakończenie)).

Możliwe jest również użycie procentów. Używając procentów, możesz dodać dowolną liczbę zmian stylu.

Poniższy przykład zmieni kolor tła elementu <div>, gdy animacja jest ukończona w 25%, w 50% i ponownie, gdy animacja jest ukończona w 100%:

```
<style>
div {
  width: 100px;
  height: 100px;
  background-color: pink;
  animation-name: zmiana;
  animation-duration: 7s;
}

@keyframes zmiana {
  0% {background-color: grey;}
  25% {background-color: blue;}
  50% {background-color: orange;}
  100% {background-color: black;}
}
</style>
```

Poniższy przykład zmieni zarówno kolor tła, jak i położenie elementu <div>, gdy animacja jest ukończona w 25%, w 50% i ponownie, gdy animacja jest ukończona w 100%:

```
<style>
div {
  width: 100px;
  height: 100px;
  background-color: pink;
  position: relative;
  animation-name: zmiana;
  animation-duration: 7s;
}

@keyframes zmiana {
  0% {background-color:grey; left:0px; top:0px;}
  25% {background-color:black; left:500px; top:0px;}
  50% {background-color:blue; left:200px; top:400px;}
  75% {background-color:green; left:0px; top:100px;}
  100% {background-color:orange; left:0px; top:0px;}
}
</style>
```

Opóźnij animację

Właściwość **animation-delay** określa opóźnienie początku animacji.

Poniższy przykład ma 4-sekundowe opóźnienie przed rozpoczęciem animacji:

```
<style>
div {
  width: 100px;
  height: 100px;
  background-color: pink;
  position: relative;
  animation-name: zmiana;
  animation-duration: 7s;
  animation-delay: 4s;
}

@keyframes zmiana {
  0% {background-color:grey; left:0px; top:0px;}
  25% {background-color:black; left:400px; top:0px;}
  50% {background-color:blue; left:400px; top:400px;}
  75% {background-color:green; left:0px; top:400px;}
  100% {background-color:orange; left:0px; top:60px;}
}
</style>
```

Dozwolone są również wartości ujemne. Jeśli używasz wartości ujemnych, animacja rozpocznie się tak, jakby była odtwarzana już przez N sekund.

W poniższym przykładzie animacja rozpocznie się tak, jakby była już odtwarzana przez 1 sekundę:

```
<style>
div {
  width: 100px;
  height: 100px;
  background-color: pink;
  position: relative;
  animation-name: zmiana;
  animation-duration: 7s;
  animation-delay: -1s;
}

@keyframes zmiana {
  0% {background-color:grey; left:0px; top:0px;}
  25% {background-color:black; left:400px; top:0px;}
  50% {background-color:blue; left:400px; top:400px;}
  75% {background-color:green; left:0px; top:400px;}
  100% {background-color:orange; left:0px; top:60px;}
}
</style>
```

Ustaw, ile razy ma być uruchomiona animacja

Właściwość **animation-iteration-count** określa, ile razy powinna zostać uruchomiona animacja.

Poniższy przykład uruchomi animację 3 razy, zanim się zatrzyma:

```
<style>
div {
  width: 100px;
  height: 100px;
  background-color: pink;
  position: relative;
  animation-name: zmiana;
  animation-duration: 7s;
  animation-delay: -1s;
  animation-iteration-count: 3;
}

@keyframes zmiana {
  0% {background-color: grey; left: 0px; top: 0px;}
  25% {background-color: black; left: 400px; top: 0px;}
  50% {background-color: blue; left: 400px; top: 400px;}
  75% {background-color: green; left: 0px; top: 400px;}
  100% {background-color: orange; left: 0px; top: 600px;}
}
</style>
```

wartości „**infinite**”, aby animacja była kontynuowana na zawsze

```
animation-iteration-count: infinite;
```

Właściwość **animation-direction** określa, czy animacja powinna być odtwarzana do przodu, do tyłu czy w naprzemiennych cyklach.

Właściwość może mieć następujące wartości:

normal - animacja jest odtwarzana normalnie (do przodu). To jest domyślne

reverse - animacja jest odtwarzana w odwrotnym kierunku (do tyłu)

alternate - animacja jest odtwarzana najpierw do przodu, a następnie do tyłu

alternate-reverse - animacja jest odtwarzana najpierw do tyłu, a następnie do przodu

```
animation-direction: reverse;
```

Właściwość **animation-timing-function** określa krzywą szybkości animacji.

Właściwość funkcji czasu animacji może mieć następujące wartości:

ease - określa animację z powolnym początkiem, a następnie szybkim, a następnie wolnym zakończeniem (jest to ustawienie domyślne)

linear - określa animację z taką samą prędkością od początku do końca

ease-in - określa animację z powolnym startem

ease-out - określa animację z powolnym zakończeniem

ease-in-out - określa animację z powolnym początkiem i końcem

cubic-bezier(n, n, n, n) - pozwala zdefiniować własne wartości w funkcji sześcienniej-beziera

```
<style>
div {
  width: 100px;
  height: 50px;
  background-color: pink;
  font-weight: bold;
  position: relative;
  animation: ruch 7s infinite;
}

#div1 {animation-timing-function: linear;}
#div2 {animation-timing-function: ease;}
#div3 {animation-timing-function: ease-in;}
#div4 {animation-timing-function: ease-out;}
#div5 {animation-timing-function: ease-in-out;}

@keyframes ruch {
  from {left: 0px;}
  to {left: 500px;}
}
</style>
```

Zgrupowanie dla właściwości **animation** KOLEJNOŚĆ

```
div {
  animation-name: example;
  animation-duration: 5s;
  animation-timing-function: linear;
  animation-delay: 2s;
  animation-iteration-count: infinite;
  animation-direction: alternate;
}

div {
  animation: example 5s linear 2s infinite alternate;
}
```