

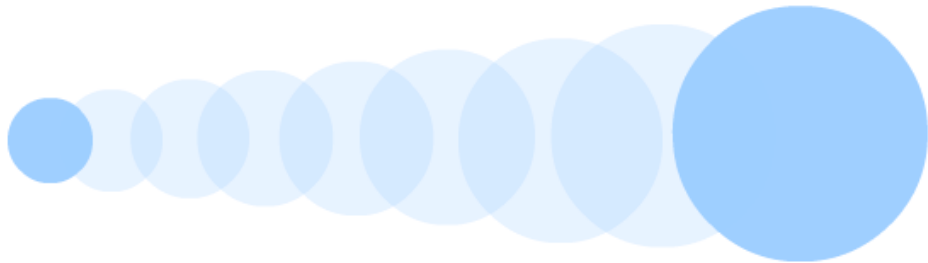


Animacja

dimon.work/kurs.html

1. Animacja z wykorzystaniem CSS

CSS i animacja



Animacja na stronie internetowej to zmiana w czasie tego czy innego stylu właściwości elementu.

*Przed **CSS 3** animacja była tworzona wyłącznie przy użyciu **JavaScript**, ale nic nie stoi w miejscu.*

Dwa podejścia do animacji w CSS

Animacja przejścia płynne przejście pomiędzy różnymi wartościami właściwości stylu, w oparciu o właściwość **przejścia** .

Animacja skryptowa to predefiniowana sekwencja akcji odtwarzanych podczas animacji, w oparciu o właściwość **animacji** .

2. Animacja

„ według scenariusza ”

Animacja skryptowa

Animacja skryptowa składa się z dwóch etapów . **Pierwszy krok** : utwórz „skrypt” za pomocą **dyrektywy @keyframes** określone są „ramki odniesienia”, tj. stany, przez które powinna przejść nasza animacja. Klatki numerowane są **procentowo** (dla animacji pomiędzy dwoma stanami można zastosować **from - to**) , dzięki czemu określana jest nie tylko kolejność klatek, ale także obliczany jest czas potrzebny na odtworzenie przejścia pomiędzy klatkami . Skrypt musi mieć nazwę.

```
16 @keyframes my-script{
17     0%{
18         background: yellow;
19         transform: rotate(0deg) ;
20     }
21
22     70% {
23         background: blue;
24         transform: rotate(45deg) ;
25     }
26
27     90%{
28         background: orange;
29         transform: rotate(-15deg) ;
30     }
31
32     100%{
33         background: yellow;
34         transform: rotate(0deg) ;
35     }
36 }
```

Animacja skryptowa

Animacja skryptowa składa się z dwóch etapów . **Drugi krok** polega na odtworzeniu animacji dla tagu. W tym celu należy użyć odpowiednich właściwości i określić: nazwę skryptu (**animation-name**) , czas trwania animacji (**animation-duration**) , funkcję rozkładu czasu ((**animation-timing-function**) , opóźnienie rozpoczęcia animacji (**animation-delay**), liczbę powtórzeń animacji (**animation-iteration-count** – od **1** do nieskończoności) , kierunek animacji (**animation-direction**) i jakie style zastosować po zakończeniu animacji (**animation-fill-mode**). Możesz też użyć ogólnej właściwości **animation** umożliwiając ustawienie wszystkich parametrów animacji w jednej linii .

```
3  div{
4      animation-name: my-script;
5      animation-duration: 3s;
6      animation-timing-function: linear;
7      animation-delay: 1s;
8      animation-iteration-count: infinite;
9      animation-direction: alternate;
10     animation-fill-mode: forwards;
11
12     /* Или же */
13
14     animation: my-script 3s linear 1s infinite alternate forwards;
15 }
```

3. Trochę praktyki

Animacja według „scenariusza”



Składa się z dwóch kroków:
Krok pierwszy : utwórz „skrypt” przy użyciu **dyrektywy @keyframes** określone są „ramki odniesienia”, tj. stany, przez które powinna przejść nasza animacja.
Drugi krok : użycie właściwości **animacji** „skrypt” jest dołączony do dowolnego elementu.

4. Animacja w **Font Awesome**

Animacja w czcionce Awesome



```
<div class="fa-3x">  
  <i class="fas fa-spinner fa-spin"></i>  
  <i class="fas fa-circle-notch fa-spin"></i>  
  <i class="fas fa-sync fa-spin"></i>  
  <i class="fas fa-cog fa-spin"></i>  
  <i class="fas fa-spinner fa-pulse"></i>  
  <i class="fas fa-stroopwafel fa-spin"></i>  
</div>
```

Biblioteka **Font Awesome** zawiera szereg efektów, które mogą być przydatne, przede wszystkim przy animacjach dla programów ładujących.

<https://fontawesome.com/how-to-use/on-the-web/styling/animating-icons>

Praca domowa

Praca domowa



Masz obraz **drogi** i **samochodu** . Utwórz kreskówkę CSS przedstawiającą **samochód jadący drogą** .
P.S. możesz dodać więcej samochodów...

P.S. Ustaw rozmiar obrazka na 1000x1000 px i nie zwracaj sobie głowy kwestią skalowania.