



Animacje CSS, rozpoznawanie mowy

Animacje CSS

Animacje w CSS pozwalają na tworzenie dynamicznych efektów wizualnych, które poprawiają wygląd i interaktywność stron internetowych. Dzięki nim elementy mogą się poruszać, zmieniać kolory, powiększać się, znikać czy płynnie przechodzić między różnymi stanami – wszystko bez potrzeby użycia JavaScript.

Właściwość animation

Pozwala na tworzenie dynamicznych efektów wizualnych bez użycia JavaScript. Jest to właściwość, która łączy kilka parametrów kontrolujących animację.

parametry animation:

animation-name – nazwa animacji, którą definiujemy w @keyframes.

animation-duration – czas trwania animacji (np. 2s, 500ms).

animation-timing-function – sposób, w jaki animacja przyspiesza lub zwalnia (ease, linear, ease-in, ease-out, ease-in-out).

animation-delay – opóźnienie animacji przed rozpoczęciem.

animation-iteration-count – ile razy animacja ma się powtórzyć (1, 3, infinite).

animation-direction – kierunek animacji (normal, reverse, alternate, alternate-reverse).

animation-fill-mode – co dzieje się z elementem przed i po animacji (none, forwards, backwards, both).

animation-play-state – pauzowanie (running, paused).

Reguła @keyframes

Reguła @keyframes określa kluczowe etapy animacji, czyli jak zmieniają się właściwości elementu w czasie.

```
@keyframes nazwa-animacji {  
  from {  
    /* początkowy stan animacji */  
  }  
  to {  
    /* końcowy stan animacji */  
  }  
}
```

```
@keyframes moveBox {  
  0% {  
    transform: translateX(0);  
    background-color: blue;  
  }  
  50% {  
    transform: translateX(200px);  
    background-color: red;  
  }  
  100% {  
    transform: translateX(0);  
    background-color: blue;  
  }  
}
```



W `@keyframes` opisujemy, jak element ma się zmieniać w czasie. To tutaj definiujemy kolejne etapy animacji, np. zmianę koloru, przesunięcie czy obrót.

W `animation` (właściwościach CSS) określamy, jak ta animacja ma się zachowywać, m.in.:

- Czas trwania (`animation-duration`)
- Liczbę powtórzeń (`animation-iteration-count`)
- Opóźnienie przed startem (`animation-delay`)
- Kierunek animacji (`animation-direction`)
- Funkcję czasową (tempo animacji) (`animation-timing-function`)

Przykład

```
@keyframes bounce {  
    0% {  
        transform: translateY(0);  
    }  
    50% {  
        transform: translateY(-50px);  
    }  
    100% {  
        transform: translateY(0);  
    }  
}
```

0% → Piłka jest na swojej początkowej pozycji.

50% → Piłka podskakuje o -50px (w góre).

100% → Wraca do początkowej pozycji.

```
.ball {  
    width: 50px;  
    height: 50px;  
    background-color: orange;  
    border-radius: 50%;  
  
    /* Animacja skoku */  
    animation: bounce 2s ease-in-out infinite;  
}
```

bounce → Nazwa animacji (zdefiniowana w @keyframes).

2s → Animacja trwa 2 sekundy.

ease-in-out → Płynne rozpoczęcie i zakończenie animacji.

infinite → Animacja powtarza się w nieskończoność.