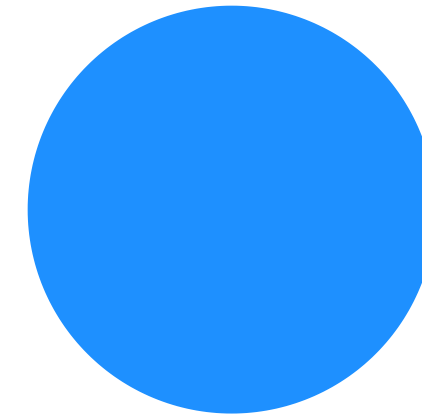


Animacje

Przypisanie klatek kluczowych do selektora

Właściwości klatek kluczowych

Po stworzeniu opisu animacji za pomocą `@keyframes` czas podłączyć tą deklarację do jakiś elementów. Pojedynczą deklarację możemy podpinać do wielu elementów na raz. Służy do tego właściwość `animation`.



```
<div class="loading"></div>
```

```
@keyframes pulse {  
  from {  
    transform: scale(1);  
  }  
  
  to {  
    transform: scale(2);  
    background-color: dodgerblue;  
  }  
}  
  
.loading {  
  animation: pulse 2s infinite alternate;  
}
```

Nazwa i czas trwania animacji

Nazwa animacji

Właściwość `animation-name` służy do wskazywania `@keyframes`, które chcemy podpiąć pod dany element.

W naszym przykładzie **pulse**.

```
.loading {  
  animation-name: pulse;  
}
```

Czas trwania animacji

Właściwość `animation-duration` określa czas trwania pojedynczego przebiegu animacji. Domyślnie właściwość ta wynosi 0s. Możemy go określać za pomocą sekund (s) lub milisekund (ms). Ważne by zawsze tutaj podawać jednostkę - nawet jeżeli podajemy 0.

```
.loading {  
  animation-name: pulse;  
  animation-duration: 2s;  
  
  /* lub */  
  animation: pulse 2s;  
}
```

Opóźnienie animacji

Właściwość `animation-delay` określa pierwsze opóźnienie startu animacji.

Opóźnienie startu animacji. Opóźnienie animacji **nastąpi tylko przy jej pierwszym uruchomieniu.**

Jeżeli podamy **ujemną wartość**, to animacja uruchomi się tak od razu rozpoczynając odtwarzanie od danej sekundy animacji. Przykładowo podanie -2s sprawi, że animacja odegra się od razu, ale rozpocznie się w 2s @keyframes.

Przykład

```
.loading {  
  animation-name: pulse;  
  animation-duration: 2s;  
  animation-delay: 1s;  
  
  /* lub */  
  animation: pulse 2s 1s;  
}
```

Tempo animacji

Właściwość `animation-timing-function` określa rodzaj płynności trwania animacji. Tak samo jak w przypadku transition, możemy tutaj użyć kilku zapisów:

- `linear` – stałe tempo animacji,
- `ease` – domyślne tempo animacji,
- `ease-in` - wolniejsze tempo animacji na początku,
- `ease-out` - wolniejsze tempo animacji na końcu,
- `ease-in-out` - wolniejsze tempo efektu przejścia (na początku oraz na końcu),
- `steps` - liczba klatek animacji efektu przejścia określona za pomocą funkcji.



Przykład

```
.loading {  
  animation-name: pulse;  
  animation-duration: 2s;  
  animation-delay: 1s;  
  animation-timing-function: ease;  
  
  /* lub */  
  animation: pulse 2s 1s ease;  
}
```

Więcej <http://matthewlein.com/ceaser>

Liczba powtórzeń

Za pomocą właściwości `animation-iteration-count` określamy liczbę powtórzeń naszej animacji. Domyślnie właściwość ta wynosi 1.

Jeżeli chcemy, by animacja była zapętłona w nieskończoność, używamy słowa kluczowego `infinite`.

Przykład

```
.pulse {  
  animation-name: pulse;  
  animation-duration: 2s;  
  animation-delay: 1s;  
  animation-timing-function: ease;  
  animation-iteration-count: infinite;  
  
  /* lub */  
  animation: pulse 2s 1s infinite;  
}
```

Kierunek

animation-direction

Właściwość ta określa kierunek powtarzania się oraz rozpoczęcia naszej animacji.

- **normal** – animacja będzie rozpoczynała się od początku i będzie powtarzana w domyślnym kierunku,
- **alternate** – animacja będzie powtarzać się w odwrotnym kierunku,
- **alternate-reverse** – animacja będzie rozpoczynała się od końca oraz będzie powtarzana w odwrotnym kierunku,
- **reverse** – animacja będzie rozpoczynała się od końca.

Przykład

```
.loading {  
  animation-name: pulse;  
  animation-duration: 2s;  
  animation-delay: 1s;  
  animation-timing-function: ease;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}
```

Styl elementu po animacji

animation-fill-mode

Określa, jaki styl będzie zastosowany dla elementu, kiedy animacja się skończy.

- **none** – brak dodatkowego stylu wyglądu elementu w czasie opóźnienia animacji lub po jej wykonaniu.
- **both** – w czasie opóźnienia animacji element odziedziczy wartości właściwości CSS takie jak w pierwszym selektorze czasu animacji. Natomiast po zakończeniu animacji element odziedziczy wartości właściwości CSS takie jak w ostatnim selektorze czasu animacji.

- **forwards** – po zakończeniu animacji element odziedziczy wartości właściwości CSS takie jak w ostatnim selektorze czasu animacji.
- **backwards** – w czasie opóźnienia animacji element odziedziczy wartości właściwości CSS, takie jak w pierwszym selektorze czasu animacji.

Przykład

```
.loading {  
  ...  
  animation-fill-mode: forwards;  
}
```


Zatrzymanie i wznowienie animacji

animation-play-state

Jest to zatrzymanie lub wznowienie aktualnej animacji.

- **paused** – animacja zostanie wstrzymana,
- **running** – animacja zostanie uruchomiona.

Przykład

```
.loading {  
  animation-name: pulse;  
  animation-duration: 2s;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
  animation-fill-mode: forwards;  
}  
  
.loading: hover {  
  animation-play-state: paused;  
}
```

Zapis animacji dla selektora

```
.loading {  
  animation-name: pulse;  
  animation-duration: 2s;  
  animation-delay: 1s;  
  animation-iteration-count: infinite;  
  animation-timing-function: ease;  
  animation-direction: alternate;  
}
```

```
.loading {  
  animation: pulse 2s 1s infinite ease alternate;  
}
```

Zapis animacji dla selektora

```
.loading {  
  animation-name: pulse;  
  animation-duration: 2s;  
  animation-delay: 1s;  
  animation-iteration-count: infinite;  
  animation-timing-function: ease;  
  animation-direction: alternate;  
}
```

Przykład animacji.

```
.loading {  
  animation: pulse 2s 1s infinite ease alternate;  
}
```

Zapis animacji dla selektora

```
.loading {  
  animation-name: pulse;  
  animation-duration: 2s;  
  animation-delay: 1s;  
  animation-iteration-count: infinite;  
  animation-timing-function: ease;  
  animation-direction: alternate;  
}
```

```
.loading {  
  animation: pulse 2s 1s infinite ease alternate;  
}
```

Ten sam przykład, wersja skrócona.

Wiele animacji

Wiele animacji

Możemy definiować kilka animacji dla jednego selektora, wystarczy rozdzielić je przecinkiem.

```
.loading {  
  animation-name: pulse, fade;  
  animation-duration: 4s, 2s;  
  animation-delay: 4s, -3s;  
  animation-iteration-count: infinite;  
  animation-timing-function: linear, ease-in;  
}
```

```
.loading {  
  animation: pulse 4s 4s infinite linear,  
             fade 2s -3s infinite ease-in;  
}
```