

Interface Software

CSS Animations

Kristof Michiels

Waarom webpagina's animeren?

- Beweging is een interessant design mechanisme
- Ons brein heeft een natuurlijke reflex om aandacht te schenken aan beweging
- CSS animatie is een krachtige manier om aandacht van de gebruiker naar belangrijke zaken te richten
- Indien goed gebruikt brengt het je site tot leven en maakt het je gebruikersinterface doeltreffender

Daarom: CSS Animaties

- Er zijn twee belangrijke bouwstenen om animaties te maken:
 - @keyframes: definiëren de fases en de stijlen van de animatie. Bevat vaak ook een transform
 - Animatie-eigenschappen: binden de keyframes aan een CSS-element en bepalen de aard van de animatie
- Deze bouwstenen worden aan het CSS-bestand toegevoegd

@keyframes

@keyframes definiëren hoe de animatie zal verlopen

```
@keyframes mijnanimatie {  
  0% {  
    transform: scale(0.1);  
    opacity: 0;}  
  60% {  
    transform: scale(1.2);  
    opacity: 1;}  
  100% {  
    transform: scale(1);}  
}
```

@keyframes

- Elk @keyframes-directief is samengesteld uit:
 - De naam van de animatie: een naam die de animatie beschrijft, gevolgd door accolades
 - De tussenstappen van de animatie: weergegeven als percentage. 0% = begin en 100% het eindpunt. Je mag zoveel tussenstappen toevoegen als je wil. De browser animeert zelf de ruimte tussen de stadia
 - De CSS eigenschappen: worden gedefinieerd voor elk stadium van de animatietijdslijn
 - Hier vind je een lijst van de "animatable" CSS properties: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_animated_properties

De animatie-eigenschappen

- Wijzen de @keyframes toe aan het element dat je wil animeren
- Bepalen hoe het element geanimeerd wordt
- Er zijn 2 eigenschappen die steeds moeten aanwezig zijn:
 - animation-name: link met de @keyframes via de naam. Je kan dus @keyframes-directieven herbruiken en koppelen aan verschillende elementen
 - animation-duration: de looptijd van de animatie, in seconden

```
div {  
  animation-name: mijnanimatie;  
  animation-duration: 2s;  
}
```

Optionele eigenschappen

- animation-timing-function
- animation-delay
- animation-iteration-count
- animation-direction
- animation-fill-mode
- animation-play-state

animation-timing-function

- Bepaalt - zoals bij transitions - de snelheidscurve of het tempo van de animatie
- toegelaten waarden: ease (=standaard), linear, easein, ease-out, ease-in-out, initial, inherit

```
div {  
  animation-name: mijnanimatie;  
  animation-duration: 5s;  
  animation-timing-function: linear;  
}
```


animation-delay

- Standaard begint een animatie onmiddellijk
- animation-delay laat toe een aantal seconden vertraging te zetten op de animatie
- Een positieve waarde (bvb 5s) zal de animatie pas na 5 seconden starten
- Een negatieve waarde (bvb -4s) zal de animatie onmiddellijk doen starten, maar op het punt dat de animatie zoals beschreven in de @keyframes al 4 seconden bezig is.

```
div {  
  animation-name: mijnanimatie;  
  animation-duration: 5s;  
  animation-delay: 3s;  
}
```

animation-iteration-count

- Standaard wordt een animatie slechts 1 keer uitgevoerd
- Met animation-iteration-count bepaal je het aantal keer dat de animatie zal worden uitgevoerd
 - Bvb. voor waarde 5 wordt de animatie 5x na elkaar uitgevoerd om dan stil te vallen
 - Waarde infinite zorgt ervoor dat de animatie in een loop wordt uitgevoerd en dus nooit stil zal vallen

```
.mijndiv, .anderediv {  
  animation-name: mijnanimatie;  
  animation-duration: 10s;  
  animation-iteration-count: 5; // animatie wordt 5x uitgevoerd  
}  
  
.anderediv {  
  animation-iteration-count: infinite; // zorgt voor een loop  
}
```

animation-direction

- Standaard verloopt een animatie chronologisch van begin (0%) naar eind (100%)
- Met animation-direction kan je dit wijzigen. Toegelaten waarden:
 - normal (standaard): voorwaartse volgorde. Na elke cyclus start de animatie op 0% en loopt naar 100%
 - reverse: achterwaartse volgorde. Na elke cyclus start de animatie op 100% en gaat ze terug naar 0%
 - alternate: afwisselend voorwaarts & achterwaarts
 - alternate-reverse: afwisselend achterwaarts & voorwaarts

```
div {  
  animation-name: mijnanimatie;  
  animation-duration: 10s;  
  animation-direction: alternate // afwisselend voorwaarts & achterwaarts  
}
```

animation-fill-mode

- Geeft aan of stijlen uit @keyframes toegepast worden/blijven voor of na de animatie. Toegelaten waarden:
 - backwards: voor animatie begint worden stijlen beschreven in eerste keyframe (0%) reeds toegepast
 - forwards: nadat animatie is voltooid worden stijlen beschreven in het laatste keyframe (100%) behouden
 - both: de animatie volgt zowel de regels voor forwards en backwards
 - normal (standaard): de animatie gaat geen stijlen toekennen voor of na de animatie

```
div {  
  animation-name: mijnanimatie;  
  animation-duration: 10s;  
  animation-fill-mode: forwards; // eindstijl animatie wordt behouden  
}
```

animation-play-state

- Met animation-play-state kan je de animatie pauzeren
- Toegelaten waarden:
 - running: de animatie loopt momenteel
 - paused: de animatie is gepauzeerd momenteel
- Bij een verandering van paused naar running: de animatie gaat verder vanaf het punt waar eerder werd gestopt

```
div {  
  animation-name: mijnanimatie;  
  animation-duration: 10s;  
  animation-play-state: paused; // de animatie is gepauzeerd  
}
```

Meerdere animaties aan één element koppelen

Doe je door meerdere waarden mee te geven, gescheiden door een komma

```
div {  
  animation: binnenGlijden 2s, draaien 1.75s;  
}
```

IS - CSS Animations

kristof.michiels01@ap.be