

Animation mit CSS

Unterscheidung: Animation und Transition

In CSS lassen sich zwei Arten von Animation definieren: nicht Interaktive (CSS-Animation) und Interaktive (CSS-Transitions, siehe separates Handout).

CSS-Animationen starten automatisch, CSS-Transitions sind abhängig vom Zustand eines Elements, z.B. normal und `:hover`.

Faustregel: Was lässt sich animieren?

Einfach zu animieren sind folgende Eigenschaften:

- Skalierung (wachsen/schrumpfen)
- Bewegen von A nach B
- Rotation
- Verzerrung
- Ein-/Ausblenden

Problematisch ist alles, was eine *Verwandlung* beinhaltet, zum Beispiel das Überführen von Text in eine geometrische Form, von einem Film in ein Vektorbild etc. Alles was nach «Trickfilm» riecht, sollte sicherheitshalber mit dem Prädikat «technisch nicht machbar» versehen werden.

Animation am Bildschirm bedingt *Interpolation*: Solange es Eigenschaften mit Zahlenwerten gibt, die manipulierbar sind, dann ist eine Animation sehr wahrscheinlich möglich.

Easing

Easing bezeichnet die Beschleunigung einer Interpolation.

Wenn die Schritte einer Interpolation immer gleich gross sind, so ist es eine «lineare» Interpolation. Im obigen Beispiel wird das Schriftgewicht bei jedem Einzelbild der Animation um den gleichen Wert erhöht.

Manchmal erfolgt eine Interpolation aber nicht in gleichmässigen Schritten. Zum Beispiel kann es sein, dass der Ausgangswert am Anfang nur in kleinen Schritten gesteigert wird, gegen Ende aber in immer Grösseren. Das ergibt eine Animation, die zu Beginn langsam ist und gegen Ende schneller wird. Ein Beispiel dafür wäre eine startende Rakete.

Umgekehrt kann eine Animation am Anfang schnell sein, gegen Ende aber langsamer werden. Ein Beispiel dafür wäre ein bremsendes Auto.

- ease-in: langsam zu Beginn, schnell am Schluss
- ease-out: Verlangsamung gegen Ende.
- ease-in-out: Verlangsamung zu Beginn und gegen Ende

Je nach Anwendungsfall kann es sein, dass eine Animation natürlicher wirkt, wenn sie nicht linear abläuft. Oder anders gesagt: In der Natur sind lineare Bewegungen extrem selten.

Das Tempo einer Interpolation lässt sich als Pfad in einem Rechteck visualisieren, in welchem die X-Achse für die Dauer und die Y-Achse für den interpolierten Wert steht. Ist es eine lineare Interpolation, so verläuft der Pfad gerade. Ist es nicht linear, so gibt es eine Kurve.

CSS-Eigenschaften zum Einstellen von Animationen

Zur Animation von HTML-Elementen gibt es folgende CSS-Eigenschaften.

animation-name	/* Referenz für @keyframes	*/
animation-duration	/* Dauer in (Milli-)Sekunden	*/
animation-timing-function	/* Beschleunigung	*/
animation-delay	/* Verzögerung	*/
animation-iteration-count	/* Wiederholungen	*/
animation-direction	/* Abspielrichtung	*/
animation-fill-mode	/* Wie bleibt es stehen?	*/
animation-play-state	/* abspielen/pausieren	*/

Am wichtigsten ist `animation-name`, weil es den Bezug zur `@keyframes` Regel herstellt. Ebenfalls wichtig ist `animation-duration`. Alles weitere ist optional.

@keyframes

In der CSS-Regel `@keyframes`, gefolgt vom Namen einer Animation, werden die Werte definiert, zwischen denen interpoliert wird: Start, Schluss und bei Bedarf Zwischen-Etappen. Erlaubt sind Angaben in Prozenten von 0–100, oder `from` und `to`, wenn es keine Zwischenschritte gibt.

```
@keyframes grow {  
  from { transform: scale(1); }  
  to   { transform: scale(2); }  
}
```

Mit Prozent-Angaben kannst du den Ablauf in mehrere Teile gliedern:

```
@keyframes grow {  
  0%   { transform: scale(1); }  
  50%  { transform: scale(2); }  
  100% { transform: scale(1.5); }  
}
```

In diesem Beispiel wird bis zur Hälfte der Animationsdauer die Grösse verdoppelt, dann wieder auf das Anderthalbfache gesenkt.

Einzelnen und übersichtlich oder kurz und bündig

Du kannst die Animations-Eigenschaften separat schreiben:

```
.foo {  
  animation-name: grow;  
  animation-duration: 2s;  
  animation-timing-function: ease;  
  animation-delay: 0.5s;  
  animation-iteration-count: infinite
```

```
animation-direction: alternate;
}
```

Oder du kannst sie in der Eigenschaft `animation` zusammenfassen:

```
.foo { animation: grow 2s ease 0.5s infinite alternate; }
```

Anmerkungen

Reihenfolge der Werte in 'animation'

Die Reihenfolge der Werte ist nur teilweise geregelt. Als erstes muss der Name aufgeführt werden. Falls Dauer und Verzögerung definiert werden, muss die Dauer zuerst angegeben werden.

Ankerpunkt von Text-Elementen

Achtung beim Animieren von Text: Inline-Elemente wie `<p>` erstrecken sich über die ganze Spalte. Bei Texten, die kürzer als die Spalte sind, ist der Ankerpunkt darum ebenfalls in der Mitte der Spalte, nicht in der Mitte des Textes.

Tipp: Wenn du die Eigenschaft `display` auf `inline-block` setzt, passt sich die Grösse des Elements an seinen Inhalt an.

Um zu sehen, wie gross ein Element ist, kann es helfen, einen `border: solid 1px black;` darum zu zeichnen oder den `background: hotpink;` zu färben.

```
.kurzer-text { display: inline-block; background: hotpink; }
```

Links

- [MDN, Using CSS Animation](#)
- [Google Developer Web Fundamentals – The Basics of Easing](#)
- [easings.net](#)
- [cubic-bezier.com](#)