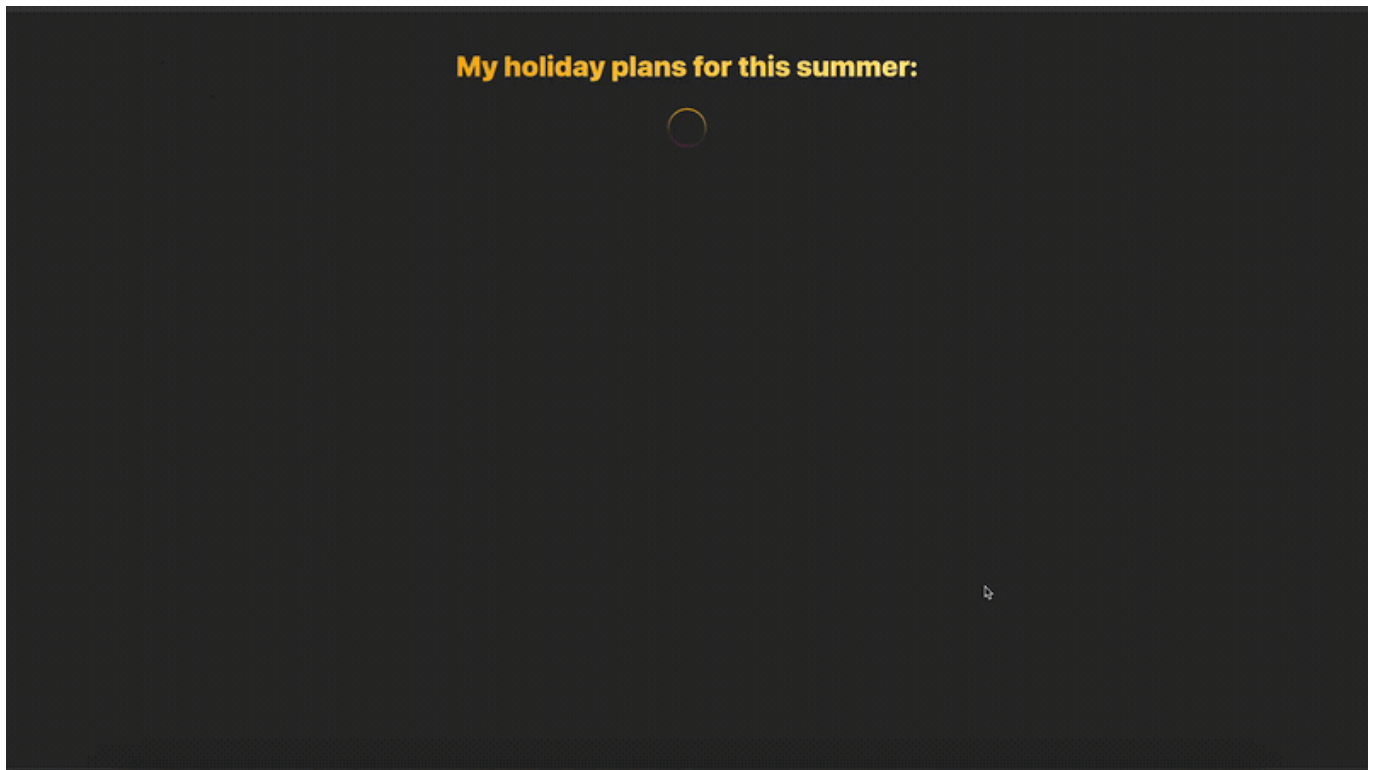


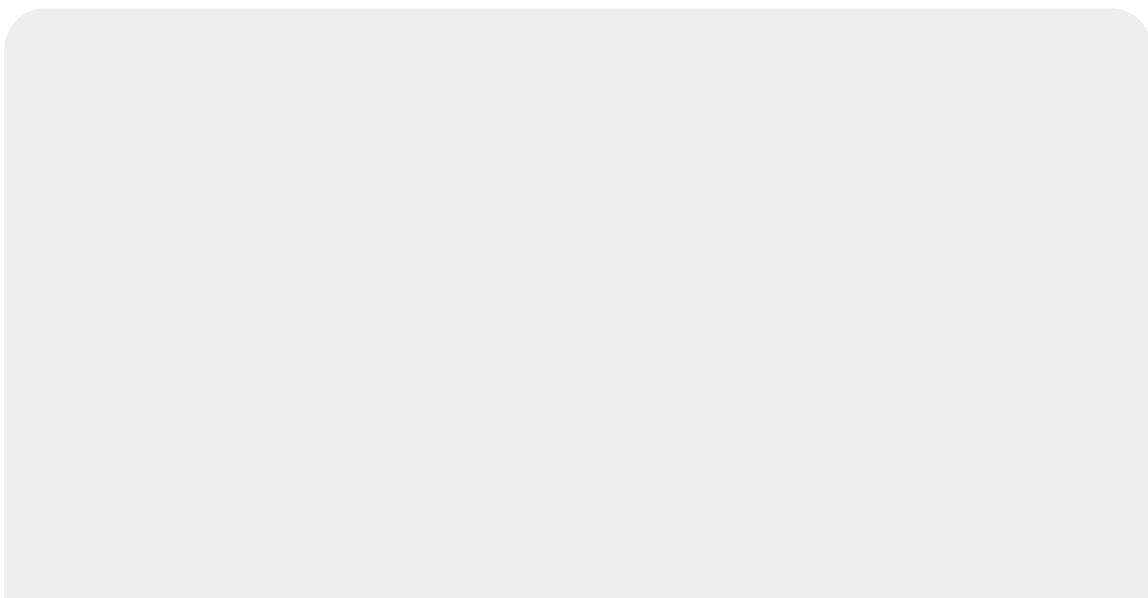
# LU13c - CSS Keyframe-Animationen

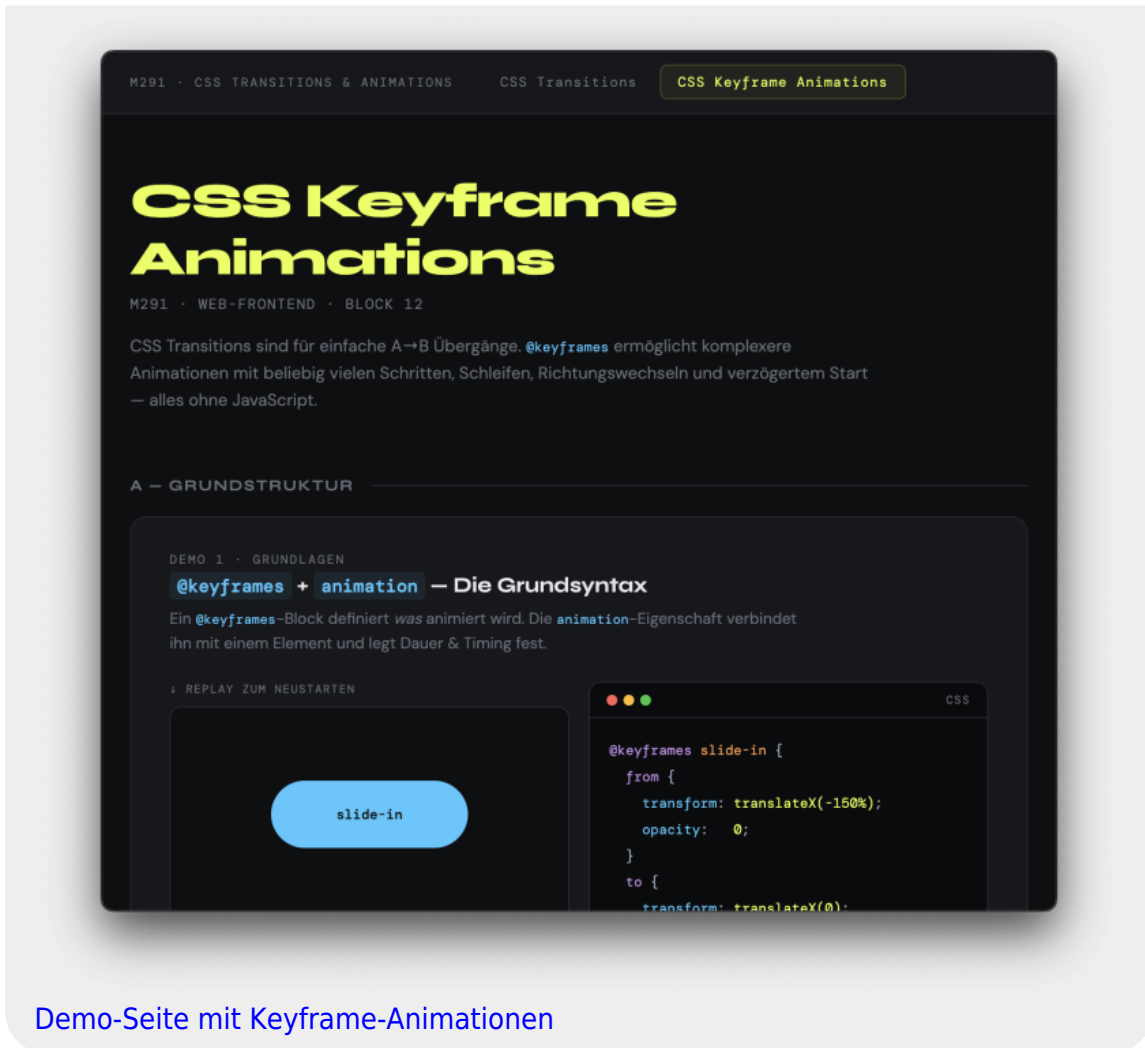


Im Gegensatz zu Transitions laufen Keyframe-Animationen **eigenständig** – sie benötigen keinen Auslöser wie einen Klick oder Hover. Sie eignen sich für Ladeanimationen, Einblend-Effekte beim Seitenaufruf und dekorative Bewegungen.

Ein einzelnes Bild aus einer Animation heisst ein **Frame**. Dabei werden Schlüsselframes (Keyframes) vorgegeben – vor allem Start und Ende – und alle Frames dazwischen werden vom Browser automatisch berechnet.

## Demo-Seite





Demo-Seite mit Keyframe-Animationen

## Teil 1: Die Animation definieren

Eine Keyframe-Animation besteht aus **zwei getrennten Teilen**: der Definition und der Zuweisung. Ohne Zuweisung an ein Element bewegt sich nichts.

### @keyframes mit from / to

```
@keyframes einblenden {
  from {
    opacity: 0;
    transform: translateY(20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}
```

```
}

```

1. @keyframes leitet die Animationsdefinition ein.
2. Danach folgt ein frei wählbarer **Name** - er wird beim Zuweisen benötigt.
3. from (= 0%) ist der erste Frame, to (= 100%) der letzte.
4. Im Keyframe stehen CSS-Eigenschaften, deren Wert sich **interpolieren** lässt: Grösse, Position, Farbe, opacity, transform und viele mehr. Eigenschaften wie background-image können nicht interpoliert werden - sie wechseln sofort.

## @keyframes mit Prozent-Schritten

Für mehr als zwei Zwischenschritte können beliebig viele Prozent-Schritte angegeben werden:

```
@keyframes farbwechsel {
  from { background-color: yellow; }
  33% { background-color: red; }
  66% { background-color: blue; }
  to { background-color: yellow; }
}
```

## Teil 2: Die Animation zuweisen

Die Animation wird einem Element über den animation-Befehl zugewiesen:

```
.card {
  animation-name: farbwechsel;
  animation-duration: 0.6s;
  animation-timing-function: ease;
  animation-delay: 0s;
  animation-iteration-count: 1;
  animation-direction: normal;
  animation-fill-mode: both;
}
```

Eigenschaft	Beispielwert	Bedeutung
animation-name	farbwechsel	Name des @keyframes-Blocks
animation-duration	0.6s	Dauer der Animation
animation-timing-function	ease	Beschleunigungskurve
animation-delay	0.2s	Verzögerung vor dem Start

Eigenschaft	Beispielwert	Bedeutung
animation-iteration-count	1, 3, infinite	Wie oft die Animation wiederholt wird
animation-direction	normal, alternate, reverse	Richtung der Wiederholung - alternate geht hin und zurück
animation-fill-mode	both	Zustand vor / nach der Animation (→ siehe unten)
animation-play-state	running, paused	Animation pausieren und fortsetzen

## Kurzschreibweise

Alle Eigenschaften lassen sich in einer einzigen Zeile zusammenfassen:

```
.card {  
  animation: farbwechsel 0.6s ease 0.2s 1 normal both;  
  /*      Name      Dauer  Easing Delay Anzahl  
  Richtung fill-mode */  
}
```

<note important> **Reihenfolge bei zwei Zeitangaben:** Wenn Dauer und Delay beide angegeben werden, gilt: **Dauer kommt zuerst**, Delay an zweiter Stelle. animation: name 0.6s 0.2s → 0.6 s Dauer, 0.2 s Delay. </note>

## Zeitangaben

```
animation-duration: 1s;      /* 1 Sekunde */  
animation-duration: 300ms;   /* 300 Millisekunden = 0.3  
Sekunden */
```

## animation-fill-mode

Ohne fill-mode springt ein Element nach der Animation sofort in seinen ursprünglichen Zustand zurück. Ausserdem ist es vor dem Start in seinem normalen Zustand sichtbar - auch wenn die Animation mit opacity: 0 beginnen würde.

Wert	Bedeutung
none	Standardverhalten: kein spezieller Zustand vor oder nach der Animation
forwards	Das Element bleibt nach dem Ende im to-Zustand
backwards	Das Element startet im from-Zustand (auch während eines Delays)
both	Kombination aus forwards und backwards: empfohlener Wert für Einblend-Animationen

```
@keyframes hereinschieben {
  from { opacity: 0; transform: translateX(-30px); }
  to   { opacity: 1; transform: translateX(0); }
}

.element {
  opacity: 0; /* Ohne fill-mode wäre das Element vor dem
  Start kurz sichtbar */
  animation: slideIn 0.5s ease 0.3s both;
}
```

Mit `fill-mode: both` übernimmt das Element den `from`-Zustand (`opacity: 0`) bereits während des Delays – kein Aufflackern.

## Weiterführende Ressourcen

- [Kevin Powell – Simplifying CSS animations](#) (5 Min., Englisch)
- [MDN – @keyframes](#)
- [MDN – animation \(Shorthand\)](#)
- [Josh W. Comeau – Keyframe Animations](#) (sehr empfehlenswert)

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
[https://wiki.bzz.ch/de/modul/m291/learningunits/lu13/theorie/c\\_keyframes](https://wiki.bzz.ch/de/modul/m291/learningunits/lu13/theorie/c_keyframes)

Last update: **2026/06/01 00:22**

