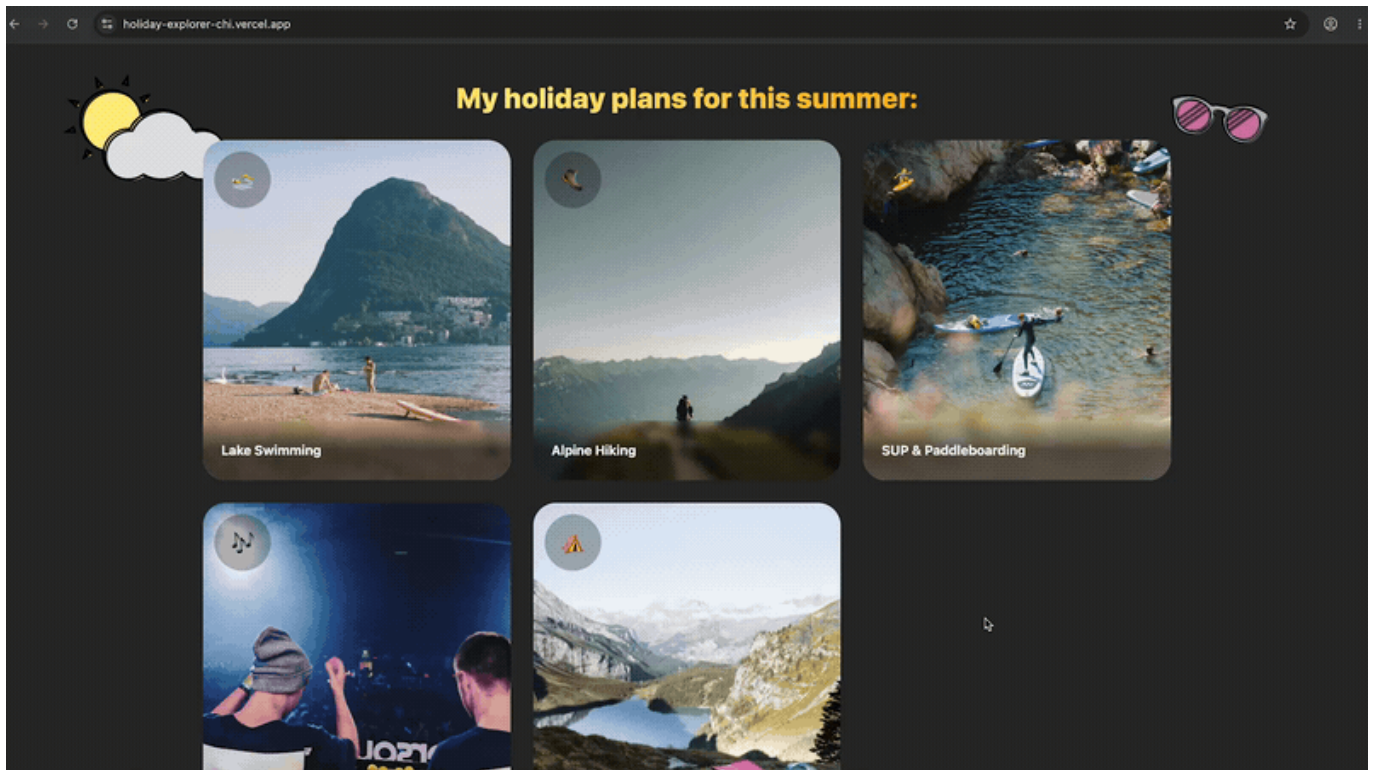


# LU13a - CSS Transitions (Wiederholung)



CSS Transitions kennen Sie bereits aus früheren Lerneinheiten. Diese Seite fasst das Wichtigste zusammen.

## Was ist eine Transition?

Eine Transition animiert den Wechsel einer CSS-Eigenschaft von einem Zustand in einen anderen. Sie benötigt immer einen **Auslöser** – zum Beispiel einen Hover-Effekt oder das Hinzufügen einer CSS-Klasse per JavaScript.

```
.panel {
  height: 0;
  opacity: 0;
  overflow: hidden;
  transition: all 300ms ease-in;
}

.panel.open {
  height: auto;
  opacity: 1;
}
```

Sobald die Klasse open per JavaScript hinzugefügt oder entfernt wird, animiert der Browser den Übergang automatisch.

## Die transition-Eigenschaft

Eigenschaft	Beispielwert	Bedeutung
transition-property	all, opacity, transform	Welche Eigenschaft animiert wird
transition-duration	300ms, 0.5s	Wie lange die Animation dauert
transition-timing-function	ease, ease-in, ease-out, linear	Beschleunigungskurve
transition-delay	0s, 0.1s	Verzögerung vor dem Start

Kurzschreibweise:

```
/* transition: Eigenschaft Dauer Easing Delay */  
transition: opacity 300ms ease-in 0s;  
  
/* Mehrere Eigenschaften mit Komma trennen */  
transition: opacity 300ms ease, transform 300ms ease;  
  
/* all animiert alle veränderten Eigenschaften gleichzeitig */  
transition: all 300ms ease-in;
```

## height: auto animieren

CSS kann normalerweise **nicht** von einer fixen Höhe zu height: auto animieren, weil auto kein numerischer Wert ist. Mit einer zusätzlichen CSS-Zeile auf :root können wir heute korrekt animieren:

```
/* In main.css, einmalig für das gesamte Projekt */  
:root {  
  interpolate-size: allow-keywords;  
}
```

Danach funktioniert die Transition auf height: auto in modernen Browsern direkt:

```
.description {  
  height: 0;  
  overflow: hidden;  
  transition: height 300ms ease;  
}  
  
.description.sichtbar {  
  height: auto; /* jetzt animierbar! */  
}
```

## Transition vs. @keyframes

	CSS Transition	CSS @keyframes Animation
<b>Auslöser</b>	Benötigt einen Zustandswechsel	Läuft eigenständig, kein Auslöser nötig
<b>Zustände</b>	Von einem Zustand zum nächsten	Beliebig viele Zwischenschritte ( % bis 100 %)
<b>Wiederholungen</b>	Nicht vorgesehen	infinite, alternate, beliebige Wiederholungen
<b>Geeignet für</b>	Hover-Effekte, Auf-/Zuklappen	Ladeanimationen, Einblend-Effekte, Dekorationen

## Demo-Seite



M291 · CSS TRANSITIONS & ANIMATIONS    CSS Transitions    CSS Keyframe Animations

M291 · WEB-FRONTEND · BLOCK 12

# CSS Animationen

Willkommen, angehende Mediamatiker:innen! Entdecke die verschiedenen Möglichkeiten, wie wir mit CSS Leben in Webseiten bringen können.

## Transitions vs. @keyframes – der Überblick

CSS bietet zwei Wege, um Animationen zu erstellen. Der Unterschied liegt im **Auslöser** und in der **Komplexität**

MERKMAL	TRANSITION	@KEYFRAMES + ANIMATION
Auslöser	Zustandsänderung (z. B. <code>:hover</code> , <code>:focus</code> )	Läuft automatisch oder beim Hinzufügen einer Klasse
Zwischenschritte	Nur Start → Ende (zwei Zustände)	Beliebig viele Zwischenschritte ( <code>0%</code> , <code>50%</code> , <code>100%</code> ...)
Wiederholung	Nicht möglich	<code>infinite</code> oder beliebig oft

Demo-Seite mit Keyframe-Animationen und Transitions

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

Permanent link: [https://wiki.bzz.ch/de/modul/m291/learningunits/lu13/theorie/a\\_transitions\\_recap](https://wiki.bzz.ch/de/modul/m291/learningunits/lu13/theorie/a_transitions_recap)

Last update: **2026/06/01 00:23**

