CSS-Animationen

Lernziele

- [ ] Die `transition`-Eigenschaft verstehen
- [ ] Easing-Funktionen verstehen
- [ ] Verstehen, welche Eigenschaften animiert werden können
- [ ] Die `animation`-Eigenschaft verstehen

Die `transition`-Eigenschaft

Die `transition`-Eigenschaft ist eine Kurzform für die Eigenschaften `transition-property`, `transition-duration`, `transition-timing-function` und `transition-delay`. (Man kann diese Eigenschaften auch einzeln verwenden, aber es ist am üblichsten, die Kurzform zu nutzen.) Sie ermöglicht es dir, den Übergang zwischen zwei Zuständen eines Elements zu definieren. Du kannst mehrere Übergänge definieren, indem du sie mit einem Komma trennst.

```
/* Übergang für die Deckkraft */
transition: opacity 500ms;
/* Übergang für Deckkraft und Hintergrundfarbe */
transition: opacity 500ms, background-color 500ms;
/* Ease-in-out-Übergang für Deckkraft und Hintergrundfarbe */
transition: opacity 500ms ease-in-out, background-color 500ms ease-in-out;
/* Ease-in-out-Übergang für Deckkraft und Hintergrundfarbe mit einer
Verzögerung von 1s */
transition: opacity 500ms ease-in-out 1s, background-color 500ms ease-in-
out 1s;
```

Hier sind alle Eigenschaften, die du mit transition verwenden kannst:

- `transition-property`: Die CSS-Eigenschaft, die du animieren möchtest.
- `transition-duration`: Die Dauer des Übergangs.
- `transition-timing-function`: Die Timing-Funktion des Übergangs.
- `transition-delay`: Die Verzögerung, bevor der Übergang beginnt.

> 💡 `duration` und `delay` verwenden eine Zeiteinheit (`ms` oder `s`).

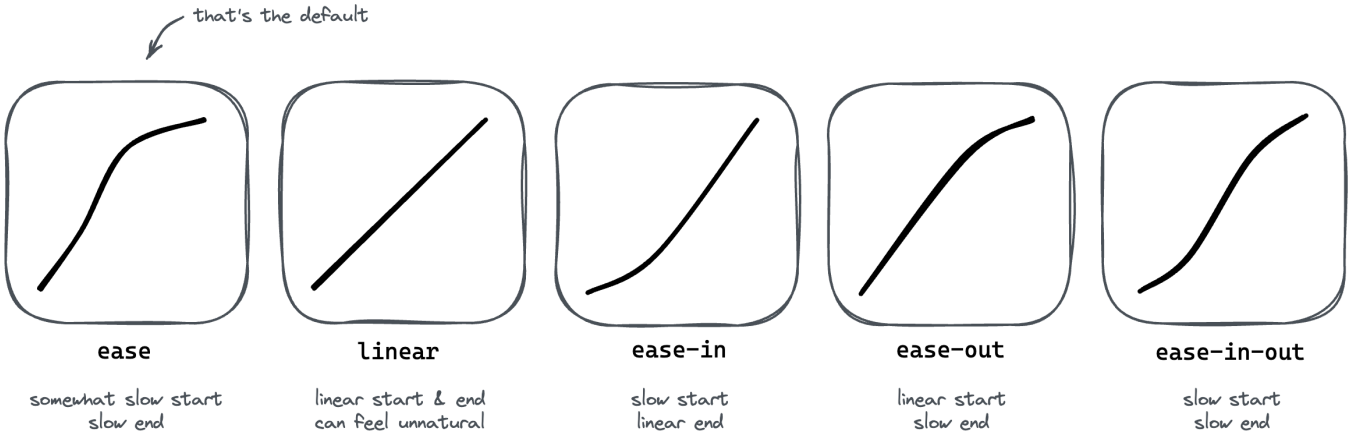
> 💡 Du kannst das Schlüsselwort `all` verwenden, um den Übergang auf alle Eigenschaften anzuwenden:

```
transition: all 500ms ease-in-out;
```

> Sei jedoch vorsichtig damit, da dies unbeabsichtigte Folgen haben kann, wie z.B. das Animieren der `height` oder `width` eines Elements bei Layout-Änderungen.

Easing-Kurven

Easing-Kurven werden verwendet, um die Geschwindigkeit und Beschleunigung des Übergangs zu definieren. Es gibt fünf eingebaute Easing-Kurven:



Du kannst die `cubic-bezier`-Funktion verwenden, um deine eigene Easing-Kurve zu definieren. Am einfachsten ist es, ein Tool wie cubic-bezier.com oder die Dev Tools deines Browsers zu verwenden, um deine eigenen Easing-Kurven zu erstellen.

> 💡 Hier ist das technische Detail: Die Funktion nimmt vier Parameter entgegen: `cubic-bezier(x1, y1, x2, y2)`. Die Parameter definieren die beiden Kontrollpunkte der Kurve. Die Kontrollpunkte sind die Punkte, zu denen sich die Kurve hinbiegt. Der Startpunkt ist immer `(0, 0)` und der Endpunkt ist immer `(1, 1)`.

Animierbare Eigenschaften

Nicht alle Eigenschaften können animiert werden. Einige Eigenschaften wie `display` oder `position` haben diskrete Werte und werden nicht glatt übergehen.

Einige Eigenschaften können animiert werden, haben aber negative Auswirkungen auf die Leistung. Zum Beispiel führt das Animieren der `height` oder `width` eines Elements dazu, dass der Browser das Layout der Seite 60 Mal pro Sekunde neu berechnen muss. Das bedeutet, dass der Browser, da sich die Abmessungen des animierten Objekts ändern, die neue Position aller anderen Elemente auf der Seite herausfinden muss.

Es ist am besten, Eigenschaften zu bevorzugen, die günstig zu animieren sind (da sie das Layout nicht beeinflussen können) wie:

- `color`
- `background-color`
- `border-color`
- `opacity`
- `transform`
- `box-shadow`
- `filter`

Eine vollständige Liste der animierbaren Eigenschaften findest du in der mdn Dokumentation.

Komplexe Animationen mit @keyframes und animation

Transitions sind großartig für einfache Animationen, aber sie sind begrenzt in dem, was sie tun können. Man kann sie nur verwenden, um zwischen zwei Zuständen zu animieren. Wenn man zwischen mehreren Zuständen animieren oder wiederkehrende Animationen haben möchte, kann man die `@keyframes`-Regel und die `animation`-Eigenschaft verwenden.

`@keyframes`

Die `@keyframes`-Regel definiert die Animation. Sie nimmt einen Namen und eine Liste von Keyframes an. Ein Keyframe ist ein Prozentsatz und eine Liste von Eigenschaften. Der Prozentsatz definiert, zu welchem Zeitpunkt der Animation die Eigenschaften angewendet werden sollen, wobei 0% der Start und 100% das Ende der Animation ist.

`@keyframes` wird außerhalb eines Selektors verwendet und wird nicht direkt auf ein Element angewendet.

```
@keyframes my-animation {
  0% {
    opacity: 0;
    transform: scale(1);
  }
  30% {
    opacity: 1;
    transform: scale(1.2);
  }
  50% {
    opacity: 1;
    transform: scale(1);
  }
  100% {
    opacity: 0;
  }
}
```

> 💡 Du kannst auch die Schlüsselwörter `from` und `to` anstelle von Prozentsätzen verwenden. `from` entspricht `0%` und `to` entspricht `100%`. Du kannst auch Prozentangaben zwischen `from` und `to` definieren, wie zum Beispiel:

```
@keyframes animationName {
  from {
    opacity: 0;
  }
  50% {
    opacity: 1;
  }
  to {
    opacity: 0;
```

```
  }
}
```

`animation`

Die `animation`-Eigenschaft ist eine Kurzform für die Eigenschaften `animation-name`, `animation-duration`, `animation-timing-function`, `animation-delay` (plus einige weitere - siehe unten). Sie wird verwendet, um die Animation auf ein Element anzuwenden. Du kannst mehrere Animationen definieren, indem du sie mit einem Komma trennst.

```
/* wende die Animation my-animation auf das Element an */
animation: my-animation 2s;
/* wende die Animation my-animation auf das Element mit einer Verzögerung
von 1s und einer ease-in-out-Easing-Kurve an */
animation: my-animation 2s ease-in-out 1s;
```

Dies sind die Eigenschaften, die du mit `animation` verwenden kannst (du kannst sie alle in der Kurzform kombinieren, aber es wird schwer lesbar):

- `animation-name`: Der Name der anzuwendenden Animation.
- `animation-duration`: Die Dauer der Animation.
- `animation-timing-function`: Die Easing-Kurve der Animation.
- `animation-delay`: Die Verzögerung, bevor die Animation beginnt.
- `animation-iteration-count`: Die Anzahl der Wiederholungen der Animation. Du kannst infinite verwenden, um die Animation unendlich oft zu wiederholen.
- `animation-direction`: Die Richtung der Animation. Du kannst `alternate` verwenden, um die Animation bei jeder zweiten Wiederholung umzukehren.
- `animation-fill-mode`: Das Fill-Mode der Animation. Du kannst `forwards` verwenden, um den Endzustand der Animation nach ihrem Ende beizubehalten.
- `animation-play-state`: Der Abspielzustand der Animation. Du kannst `paused` verwenden, um die Animation zu pausieren.

Du kannst eine beliebige Mischung aus Kurz- und Langform-Eigenschaften verwenden:

```
.element {
  /* wende die Animation my-animation auf das Element mit einer ease-in-
out-Easing-Kurve, unendlich abwechselnd an */
  animation: my-animation 2s ease-in-out;
  animation-iteration-count: infinite;
  animation-direction: alternate;
}

.element:hover {
  /* pausiere die Animation, wenn das Element gehovt wird */
  animation-play-state: paused;
}
```

> 💡 Eine sehr coole Ressource für CSS-Animationen ist animate.style. Es enthält viele Animationen, die du in deinen Projekten verwenden kannst. Du kannst den CSS-Code für die gewünschte Animation aus dem animate.css GitHub Repository kopieren und in dein Projekt einfügen.

Animationen mit `prefers-reduced-motion` verwenden

Animationen können für manche Menschen ablenkend sein. Menschen mit vestibulären Störungen oder Menschen mit Aufmerksamkeitsdefizit-Hyperaktivitätsstörung (ADHS) können leicht von Animationen beeinflusst werden. Menschen mit Epilepsie können durch blinkende Lichter Anfälle bekommen.

Um deine Website barrierefreier zu machen, solltest du eine Möglichkeit bieten, Animationen zu deaktivieren. Du kannst dies tun, indem du die Media Query `prefers-reduced-motion` verwendest. Sie kann verwendet werden, um zu erkennen, ob der Benutzer die Bewegung auf seinem Gerät reduzieren möchte.

```
/* füge nur dann Animationen oder Übergänge hinzu, wenn der Benutzer keine
Präferenz für prefers-reduced-motion hat */
@media (prefers-reduced-motion: no-preference) {
  .element {
    animation: my-animation;
    transition: color 2s;
  }
}

/* oder umgekehrt, entferne Übergänge und Animationen, wenn der Benutzer
eine reduzierte Bewegung bevorzugt */
@media (prefers-reduced-motion: reduce) {
  .element {
    animation: none;
    transition: none;
  }
}
```

> 💡 Du kannst die Media Query `prefers-reduced-motion` in den Einstellungen deines Betriebssystems festlegen. Auf macOS findest du es unter `Systemeinstellungen > Bedienungshilfen > Anzeige > Bewegung reduzieren`.

Resources

- transition on mdn
- animation on mdn
- easing-function on mdn
- cubic-bezier.com
- animate.style
- Responsive Design for motion with Ally Examples