

CSS und CSS-Animationen

Inhaltsverzeichnis

HTML

CSS

Zusammenspiel mit JavaScript

Animationen

Arten

Transitions (fließende Übergänge)

KeyFrame-Animationen

Scroll-driven Animationen

Animationen in der Praxis

Fazit

Quellen

Einleitung

HTML



HTML auch Hypertext Markup Language ist eine Auszeichnungssprache zur Darstellung von Inhalten im Internet. HTML wird gegliedert aufgebaut in Überschriften, Absätzen, Textblöcke, Aufzählungen und sonstigen Inhalten wie Bilder, Videos oder Downloads.

Ganz Wichtig: HTML ist keine Programmiersprache da es nur Inhalte und Struktur umfasst, welche dann von Browser gelesen und umgesetzt werden.

Dateien die HTML beinhalten haben meist die Endung: .html

Umso eine Datei dann darzustellen, werde Browser verwendet, die den Inhalt anzeigen, wie zum Beispiel: Chrome, Firefox oder Microsoft Edge. Sie lesen die Daten ein und stellen sie dann dar.



HTML kann man grundsätzlich mit einem einfachen Texteditor schreiben, jedoch verwendet man in der Praxis spezielle Editoren, die die Erstellung unterstützen wie: Visual Studio Codes, Notepad++ oder Sublime Text.

Die Inhalte in HTML werden in so genannte Tags gekapselt, die mit diesen Zeichen `< >` ausgeführt werden, z.B für eine Überschrift mit erster Ordnung: `<h1>` als Start-tag und um den Tag dann wieder zu schließen verwendet man den End-tag: `</h1>`.

Mittels der Tags kann man unterschiedliche Strukturen definieren. Einfache Linien einfügen `<hr>` oder mit dem tag `<p>` einfache Texte schreiben aber auch das erstellen von Tabellen ist möglich, wobei man aber mehrere geschachtelte Tags braucht: `<table>`, `<tr>`, `<th>`, `<td>`.

Das Verlinken auf andere Seiten ist durch das Tags `<a>` möglich, doch man muss aufpassen wenn die Datei zum verlinken in einem anderen Order ist. Zudem sollte man noch ein Attribut hinzufügen: ``. Ursprünglich sind die Links jedoch als Anker innerhalb der Seite gedacht und werden auch heute noch teilweise so verwendet.

CSS

CSS



CSS ist die Abkürzung von Cascading Style Sheet. Damit kann man einen langweiligen Text, Hintergrund usw. gestalten und formatieren.

Ohne CSS wird HTML vom Browser sehr einfach und nur rudimentär formatiert entsprechend der HTML – Struktur dargestellt. Mittels CSS kann man diese Formatierung farbig gestalten, Abstände oder Hintergründe hinzufügen. Für die Übersichtlichkeit und Wiederverwendbarkeit werden die Formatierungen in separate Dateien mit der Endung .css ausgelagert. Formatangaben mittels CSS direkt im Header sind zwar gängig sind aber auch nicht wiederverwendbar.

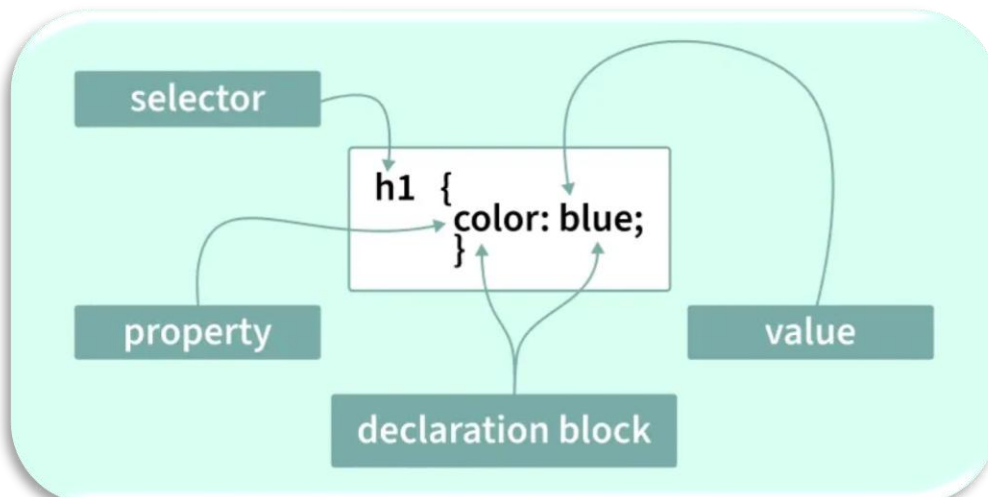
Vorteile und Anwendung

Formatierungen kann direkt an ein HTML-Tag knüpfen, z.B. ein Rahmen um ein Objekt `<div style="border: 1px solid black">`. Die sogenannte Inline-Formatierung ist jedoch nur für einfache Angaben geeignet, wenn mehrere Formatierungen verwendet werden, wird das sehr schnell unübersichtlich. Um die Formatierung übersichtlicher zu gestalten kann man HTML-Tags mittels Objektangabe, ID oder CSS-Klasse formatieren.

Das Stylen fängt man immer mit den Selektoren an, die bestimmen, welcher teil von HTML jetzt gestylt wird, dann wird die Styleangabe und der Wert mit Doppelpunkt getrennt definiert. Die Einzelnen Styleangaben werden mit eine Semikolon abgeschlossen.

```
hr {  
    text-align: center; color: ■ red;  
}
```

Vorteile für die Verwendung von CSS sind die deutlich übersichtlichere und ansprechendere Darstellung der Webseiten. Mittels CSS ist es aber auch möglich, dieselben Inhalte auf unterschiedlichen Geräten von Mobile über Tablets bis hin zu Widescreenmonitoren lesbar und nutzbar darzustellen.





Zusammenspiel mit JavaScript

HTML bildet das Grundgerüst einer Webseite, CSS ist die Formatierung und Darstellung verantwortlich. JavaScript bildet die interaktive Komponente die dem Anwender ermöglicht nicht nur eine Seite zu lesen sondern auch mit dieser zu interagieren.

Eine JavaScript-Datei wird wie eine CSS-Datei gespeichert nur das diese Datei die Endung: .js hat. Beide Dateien werden dann mit der HTML-Datei verknüpft. In der Praxis werden CSS-Dateien im „Header“ verknüpft, JavaScript-Dateien jedoch im Body am Ende des übrigen HTML-Quelltextes. Dadurch ist sicher gestellt, dass der Browser das HTML erst darstellt, wenn die CSS-Datei auch wirklich geladen ist und somit auch ordentlich dargestellt wird. Beim JavaScript hingegen ist dadurch, dass es sich am Ende des HTML-Body-Tags befindet, sicher gestellt dass alle Elemente im Zugriff sind und es zu keiner Verzögerung bei der Anzeige der Seite kommt. CSS wird durch interne Funktionen des Browsers berechnet und ist daher performanter als JavaScript, dass erst über einen Interpreter des Browsers kompiliert werden muss.

Das unterschiedliche Rendering von CSS und JavaScript führt dazu, dass z.B. Animationen mit CSS flüssiger und weniger Prozessorintensiv sind.

Über JavaScript-Aktionen ist es möglich direkt die CSS-Eigenschaften für ein HTML-Objekt zu definieren. Z.B. mit einer Interaktion vom Anwender die Schrift blau machen:
`h1Elem.style.color = „blue“;`

Auch kann man mit gezielten Aktionen CSS-Klassen zu HTML-Objekten hinzufügen oder entfernen. Sobald diese dann zugewiesen sind, werden sie von dem CSS formatiert.

Animationen können durch dieses Zusammenspiel mit CSS-Eigenschaften und Klassen feiner gesteuert werden als mit reinem CSS möglich ist. Auch können in JavaScript Berechnungen durchgeführt werden um z.B. Positionierungen oder Ausmaße eines Objekt zu bestimmen. Das ist in der Form mit CSS nicht möglich.

Formatierungen mit CSS aktiv sobald einem Objekt eine entsprechende Formatierung zugewiesen ist. Bei JavaScript kann ich zwar direkt beim Laden einer Seite Aktionen ausführen, aber in der Regel reagiert JavaScript auf Interaktionen des Anwenders.



Hauptteil

Animationen

Grundsätzlich ist jede CSS-Eigenschaft animierbar, bei der der Browser Zwischenschritte berechnen kann. Ausnahmen bilden Eigenschaften wie Tabelleneigenschaften, Schriftarten oder Listenzeichen.

Animationen ermöglichen es die Website noch lebhafter zu gestalten und sie interessanter zu gestalten. In CSS kann man Animationen bereits mit wenigen Zeilen erstellen ohne gleich JavaScript zu benötigen. Im Gegensatz zu JavaScript laufen die CSS Animationen auch bei mäßiger Systemauslastung flüssiger da der Browser die Leistung und Effizienz hier besser steuern kann.

Das Animieren in CSS verwendet man meist mit der Eigenschaft „animation“, womit man unter Anderem Timing, Dauer, Häufigkeit und andere Animationseigenschaften bestimmen kann. Eigenschaften wie „animation-delay“ verzögert z.B. den Start der Animation nach dem Laden der Seite.

Beispiel 1: p {

```
    animation-duration: 3s;  
  
    animation-name: slide-in;  
  
    animation-iteration-count: infinite;  
  
    animation-direction: alternate;  
  
}
```

Beispiel 2: <file:///C:/Users/admin/Projekte/Animation/index.html> (muss noch eine andere website)

Animationen sind ein sehr gutes Mittel, dem Benutzer den Umgang mit den Informationen auf der Seite zu erleichtern und die Interaktionen intuitiv zu gestalten. Dabei ist aber auch zu achten, dass es auch Gruppen von Benutzern gibt, die nur eingeschränkt Webseiten und digitale Produkte nutzen können z.B. Menschen mit starken Gleichgewichtsstörungen. Daher sollte man die Animationen über die Media-Query „prefers-reduced-motion“ steuern, so dass man diesen Benutzern eine reduzierte Version zur Verfügung stellt.

Arten von Animationen

Es gibt mehre Arten von Animationen mit verschiedenen Wirkungen und Effekten, welche man in CSS anwenden kann, ohne JavaScript zu benötigen. Bei einer sogenannten Hover-Animation verändert sich das Aussehen eines Objekts mittels geänderten Eigenschaften wobei hier Transitions (fließende Übergänge) und Key-Frame-Animationen zur Anwendung kommen. Scroll-Driven-Animationen sind der neueste Entwicklungsschritt der CSS-Animationen.

Transitions

Sie bilden das Fundament für zustandsbasierte Interaktionen. Ihr Zweck besteht darin den Wechsel einer CSS-Eigenschaft zu glätten, sobald ein bestimmter Auslöser wie z.B. ein Hover aktiv wird. Der Browser berechnet hierbei die Zwischenschritte zwischen den beiden definierten Zuständen.

Eine Transition wird durch 4 Eigenschaften gesteuert.

„transition-property“ bestimmt welche CSS-Eigenschaft animiert werden soll (z.B. width, background-color oder opacity). Der Standardwert „all“ verändert alle berechenbaren Eigenschaften sobald diese sich ändern.

„transition-duration“ definiert die dauer der Animation in Sekunden (z.b. 0.3s) oder Millisekunden (z.B. 500ms). Dies ist die einzige Eigenschaft die erforderlich ist, um einen Übergang zu berechnen.

„transition-timing-function“ regelt den Verlauf der Animation. „linear“ bedeutet, dass die Animation gleichmäßig berechnet wird. Eigenschaften wie „ease“ oder „ease-in“ oder „ease-out“ lassen die Animation asynchron laufen, also am Anfang, am Ende oder in der Mitte langsamer, fast so als ob die Animation eine Beschleunigung oder ein Abbremsen erfährt.

„transition-delay“ verzögert den Start der Animation um die angegebene Zeitspanne, z.B. 0.25s.

Diese Eigenschaften können auch unter der Eigenschaft „transition“ zusammengefasst werden, hier gilt dann die Reihenfolge der Angaben „property“, „duration“, „timing-function“ und „delay“.

Key-Frame Animation

Während Transition-Animation linear ablaufen, erlauben Key-Frame-Animation mehrstufige, komplexere und zeitgesteuerte Animationen und Bewegungsabläufe zu realisieren.

Das Konzept erinnert an die klassische Zeichentrickanimation in der auch das Prinzip der Schlüsselbilder (Keyframes) angewendet wird. Dabei werden im CSS lediglich markante Zustände zu bestimmten Zeitpunkten definiert, der Browser rendert dann die dazwischenliegenden Abläufe bzw. Zustände.

Eine Key-Frameanimation im CSS hat einen zweistufigen Aufbau. Dieser Prozess führt zu einer strikten Trennung zwischen Definition der Bewegungsabläufe und Implementierung zu den jeweiligen CSS-Elementen. So können Key-Frame-Definitionen mehrfach in unterschiedlichen Elementen verwendet werden.

Die Definition der eigentlichen Key-Frames wird über das Tag „@keyframes“ gefolgt von einem frei wählbaren, aber eindeutigen Namen zur Identifizierung der Animation. Innerhalb der Deklaration kann entweder über die Schlüsselwörter „from 0%“ und „to 100%“ die zeitlichen abschnitte deklariert werden oder aber die Prozentwerte 0 bis 100 können individuell auch Zwischenschritte definiert werden. Für jeden Schritt kann man beliebige animierbare CSS-Eigenschaften deklarieren und kombinieren, z.B. Transformationen, Positionen, Transparenz oder Farbe. Prozentuale Zwischenschritte erlauben komplexe Animationen in einem einzigen Animationszyklus zu realisieren.

Die Zuweisung einer Animation zu einem CSS-Element erfolgt über das Tag „animation-name“. Hier wird als Eigenschaft der Name der Key-Frame-Deklaration eingetragen. Die Eigenschaft „animation-duration“ legt die exakte Dauer der Animation in Sekunden oder Millisekunden fest. Die Eigenschaft „animation-timing-function“ steuert wie den Bewegungsablauf entweder linear standard oder aber über dieselben Mechanismen wie die Transitionanimation einen nicht linearen Verlauf mit positiven und negativen Beschleunigungen z.B: „ease-in-out“. Dadurch wirken die Bewegungen etwas lebendiger.

Über die Eigenschaft „animation-iteration-count“ kann festgelegt werden, wie oft eine Animation ablaufen soll, von 1 bis unendlich „infinite“.

Damit die Animation nach dem Ablauf nicht gleich abrupt in den Ursprungszustand zurück springt, kann man mit der Eigenschaft „animation-fill-mode“ mit der Deklaration „forward“

dafür sorgen, dass der Zustand des letzten Key-Frames beibehalten wird, während mit „backward“ der Zustand des 1. Key-Frames angenommen wird.

Mit der Eigenschaft „animation-delay“ kann man definieren, wie lange die Verzögerung dauern soll, bis die Animation startet.

Scroll-Driven Animationen

Scroll-Driver Animationen sind die neueste Entwicklung der CSS – Animation und lösen hier einen weiteren JavaScript Teil ab. Dadurch werden die Animation flüssiger und benötigen weniger Prozesslast.

Scroll-Driven- Animationen unterteilen sich in zwei Kategorien, abhängig von der „animation-timeline“.

- Progress-Timeline (scroll()) : hier orientiert sich der Animationsfortschritt am absoluten Scrollfortschritt des Containerelements. Diese Animation eignet sich für Fortschrittsbalken oder sogenannte Parallaxeffekte.
- View-Timeline (view()) : hier orientiert sich die Animation an der Sichtbarkeit eines Elements im Scrollbereich. Entscheidend ist der Eintritt und der Austritt des Elements aus dem Sichtbereich. Diese Animation wird angewendet um Elemente beim Scrollen einzublenden, wenn diese in den Sichtbereich kommen oder auszublenden, wenn diese den Sichtbereich wieder verlassen.

Neben der „animation-timeline“ bestimmt die „animation-range“ den Start und Endpunkt der Animation innerhalb des Scrollwegs. Also 0% für den Einstiegs- und 100% für den Ausstiegspunkt.

Über die Eigenschaft „scroll-timeline-name“ ist es möglich einem Container einen Namen zu geben und dann gezielt die Animation in dem Container anzusteuern.

Die Scrollrichtung auf die Animation anspricht, wird über die Eigenschaft „scroll-timeline-axis“ definiert. Die vertikale Scrollrichtung wird über den Wert „block“ gesetzt, während „inline“ für eine horizontale Scrollrichtung verwendet wird.

Wie bei den Key-Frame-Animationen können auch hier alle animierbaren CSS-Eigenschaften für die Animation verwendet werden, der Browser rendert dann automatisch die Zustände während des Scrollvorgangs.

Schluss

CSS-Animationen in der Praxis

CSS-Animationen sind schon seit langem Standard in der modernen Webentwicklung und haben die schwerfälligen und rechenintensiven JavaScript und Flash-Animationen abgelöst. Sie lassen Benutzeroberflächen intuitiver, lebendiger und dynamischer erscheinen.

Css-Animationen sind heute im Web und auch modernen Oberflächen fast überall zu finden, wenn diese auch nicht auf den 1. Blick sichtbar sind. So nutzen kommerzielle Webseiten subtile Effekte wie fließende Farbwechsel bei Buttons, oder dezent animierte Dropdown-Menüs.

Aufwändigere Animationen finden sich eher auf Seiten, die eine sehr markante Wirkung erzielen wollen, um z.B. eine Marke gezielt zu bewerben oder bekannt zu machen.

Der Grad und die Komplexität der Animation hängt stark vom Typ und dem Zweck der Webseite ab.

Landingpages und Produktseiten haben einen starken Fokus auf Scrollanimationen bei denen eine visuelle Wirkung mit gezieltem Einblenden von Informationen erzielt werden soll (<https://www.apple.com/ipad-pro/>).

E-Commerce und Onlineshops nutzen eher kleinere unauffälliger Animation wie Ladebalken oder Fokusbereiche, sanfte Animation für sich einblendende Objekte wie Filter, Menüs oder Warenkorb, wenn hier ein Produkt hinzugefügt wird.

Dashboards oder Backendanwendungen nutzen Animation um Daten, Charts oder Prozessfortschritte flüssiger darzustellen und zu visualisieren.

Digitale Visitenkarten nutzen Animationen um sich von der Masse abzuheben und eine gezielte Wirkung die Aufmerksamkeit der Benutzers zu gewinnen (<https://qr.to/DrJanHerzog>).

Fazit

Gut gestaltete und durchdachte Animationen informieren und erwecken Emotionen. Sie lenken die Aufmerksamkeit gezielt auf Elemente einer Webseite, da wir instinktiv auf Bewegung reagieren.

Trotzdem sollten Animationen mit Bedacht und wohl dosiert eingesetzt werden. Denn zu viele Animation und Bewegungen auf eine Webseite können schnell verwirrend sein und dann den Benutzer eher stören.

Auch sollten Animationen stets mit dem Media-Query „prefers-reduced-motion“ eingesetzt werden, damit Menschen mit Einschränkungen diese abschalten können.

Quellen

Bilder

[https://upload.wikimedia.org/wikipedia/commons/6/61/HTML5 logo and wordmark.svg](https://upload.wikimedia.org/wikipedia/commons/6/61/HTML5_logo_and_wordmark.svg)

[https://upload.wikimedia.org/wikipedia/commons/7/7e/Microsoft Edge logo %282019%29.png](https://upload.wikimedia.org/wikipedia/commons/7/7e/Microsoft_Edge_logo_%282019%29.png)

[https://upload.wikimedia.org/wikipedia/commons/e/e1/Google Chrome icon %28Februar y 2022%29.svg](https://upload.wikimedia.org/wikipedia/commons/e/e1/Google_Chrome_icon_%28Februar_y_2022%29.svg)

[https://upload.wikimedia.org/wikipedia/commons/a/a0/Firefox logo%2C 2019.svg](https://upload.wikimedia.org/wikipedia/commons/a/a0/Firefox_logo%2C_2019.svg)

[https://upload.wikimedia.org/wikipedia/commons/d/d5/CSS3 logo and wordmark.svg](https://upload.wikimedia.org/wikipedia/commons/d/d5/CSS3_logo_and_wordmark.svg)

<https://www.rocket-backlinks.com/optimierung-ladegeschwindigkeit/>

Informationen

<https://www.youtube.com/watch?v=BGb2dggTxoY>

https://www.youtube.com/watch?v=l84aQhbJl_Y

<https://www.youtube.com/watch?v=Q3MlitoSQkE>

https://page.mi.fu-berlin.de/mbudde/css_kurs/vorteile.html

<https://www.youtube.com/watch?v=yIzwhjWqDpM>

<https://www.youtube.com/watch?v=dXr-cr4qUtA>

Mit Unterstützung von meinem Vater/Programmierer