

Web - CSS Skript

GREGOR HAASE
MATNR: 1824234
3IB

Inhalt

1	Das Konzept der Cascading Style Sheets	3
1.1	Die Motivation hinter CSS	3
1.2	Grundaufbau der CSS.....	3
2	Integration von CSS in HTML	5
2.1	Zweck	5
2.2	Inline.....	5
2.3	Internal.....	5
2.4	External	5
3	Selektoren.....	7
3.1	Einfache Selektoren.....	7
3.1.1	Typselektor	7
3.1.2	ID Selektor	7
3.1.3	Class Selektor	7
3.1.4	Globalselektor	7
3.1.5	Attribut Selektor	8
3.1.6	Kombinierte Selektoren.....	8
3.2	Selektor Operatoren.....	8
3.2.1	Descendant Selektor	8
3.2.2	Adjacent Selektor	8
3.2.3	Parent Selektor	8
3.3	Pseudoelemente	9
3.3.1	::before, ::after	9
3.3.2	::first-line	9
3.3.3	::selection.....	9
3.4	Pseudoklassen.....	9
3.4.1	:link und :visited	10
3.4.2	:hover.....	10
3.4.3	:nth-child(n).....	10
3.4.4	:focus	10
4	Eigenschaften	11
4.1	Shorthand Notation.....	11
4.2	Eigenschaften	12
4.2.1	border.....	12
4.2.2	margin.....	13
4.2.3	padding	13
4.2.4	height und width	13
4.2.5	max-height, min-height, max-width, min-width	13

4.2.6	outline	13
4.2.7	text	14
4.2.8	display	14
4.2.9	position	14
4.2.10	z-index	15
4.2.11	overflow	15
5	Größenangaben	16
6	Farben	18
6.1	Technik	18
6.2	RGB und RGBA	18
6.3	HEX	18
6.4	HSL und HSLA	18
6.5	Farbnamen	19
7	Responsives Webdesign	21
7.1	Layout	21
7.2	Media Query	21
7.2.1	Bedingungen	22
7.3	Mobile First	23
8	Anhang	25
8.1	Lösungen der Aufgaben	25
8.2	Literaturverzeichnis	30
8.3	Abbildungsverzeichnis	31

1 Das Konzept der Cascading Style Sheets

In diesem Kapitel wird die Idee der Cascading Style Sheets (CSS) behandelt. Zuerst wird die Motivation hinter CSS erklärt, danach wird auf die Grundkonzepte eingegangen.

Nach der Bearbeitung dieses Kapitels sollen sie in der Lage sein,

- *das Grundkonzept von CSS wiederzugeben.*
- *die Vorteile von CSS gegenüber anderen Formatierungsmöglichkeiten zu nennen.*
- *die Syntax einer einfachen CSS Datei zu verstehen.*

1.1 Die Motivation hinter CSS

Hauptmotivation hinter Cascading Style Sheets (CSS) und ähnlichen Stylesheet-Sprachen ist, die Trennung von Inhalt und Darstellung. Vereinfacht gesagt bedeutet dies, dass der Inhalt (z.B. eine Website) ohne CSS in der Darstellung nicht oder nur leicht formatiert ist und erst durch das Anwenden von Gestaltungsregeln in seine endgültige Form gebracht wird. Diese Gestaltungsregeln werden mittels CSS festgelegt. Durch die Trennung und oftmals auch Auslagerung in eine eigene Datei erhöht sich die Wartbarkeit und Wiederverwendbarkeit enorm. Mittlerweile zählt CSS neben HTML zu den wichtigsten Webtechnologien (vgl. [Fuc19 S.13]).

1.2 Grundaufbau der CSS

Eine CSS-Gestaltungsregel setzt sich aus einem Selektor und einem Deklarationsblock zusammen.

Über den Selektor (es sind auch mehrere möglich) werden alle Elemente gewählt, auf die die Anweisungen im Deklarationsblock angewendet werden sollen.

Danach folgt in geschweiften Klammern der Deklarationsblock. Er enthält mindestens eine Eigenschaft und den ihm zugewiesenen Wert. Eigenschaft und Wert werden durch einen Doppelpunkt getrennt. Möchte man mehrere Eigenschaften hinzufügen, werden diese durch ein Semikolon getrennt.

In der folgenden Abbildung sieht man links noch einmal den Grundaufbau einer Gestaltungsregel. Auf der rechten Seite folgt eine Gestaltungsregel, welche alle Absätze (p Element) eines HTML Dokuments ausgewählt, und mit einer Linie umrandet.

```
Selektor {  
Attribut: Wert;  
}
```

```
p {  
border-style: solid;  
}
```

Abb. 1.1 Allgemeiner Selektoraufbau & konkretes Beispiel

Zusammenfassung

In diesem Kapitel wurde zunächst die Motivation hinter CSS beleuchtet, um ein Verständnis für die Nutzbarkeit von CSS zu geben. Danach wurde auf den Grundaufbau und die grundsätzliche Syntax der CSS eingegangen.

Übungsaufgaben

Übung 1.1

Was ist der Hauptgrund für die Verwendung von CSS?

Übung 1.2

Aus welchen zwei Grundelementen besteht eine normale Darstellungsregel?

2 Integration von CSS in HTML

In diesem Kapitel wird auf die drei Möglichkeiten eingegangen, wie CSS in HTML eingebunden wird.

Nach der Bearbeitung dieses Kapitels sollen sie in der Lage sein,

- *CSS Regeln direkt in die HTML Datei zu schreiben.*
- *CSS Anweisungen aus externen Quellen einzubinden.*

2.1 Zweck

Ohne die Einbindung in ein HTML-Dokument hat CSS keinen nutzen (vgl. [Fuc19 S.28]), da es ohne HTML keine Elemente gibt, auf welche sich die Gestaltungsregeln anwenden lassen würden.

2.2 Inline

Bei der Verwendung der Inline Methode wird die CSS Anweisung direkt im HTML-Code in das style Attributfeld eines beliebigen Elements geschrieben. Das hat den Vorteil, dass die Regel schnell implementiert ist, allerdings auf Kosten der Wartbarkeit.

```
<p style='background-color:blue'> Testtext <p>
```

Abb. 2.1 CSS mit der Inline Methode

2.3 Internal

Wenn die Anweisungen in der HTML Datei innerhalb eines Style-Elements stehen, heißt die Integrationsart Internal. Dies bietet sich an, wenn man die CSS-Regeln nicht noch einmal und nur in dieser Datei verwenden möchte.

2.4 External

Wird CSS aus einer anderen Datei bezogen, spricht man von External. Dabei wird eine CSS-Datei über das HTML Link Element eingebunden. Hierbei kann die Adresse der Datei über einen Dateipfad oder eine http Link angegeben werden. Somit können die Gestaltungsregeln auch von einer externen Website bezogen werden, der diese verwaltet.

```
<link rel="stylesheet" href="stylesheet.css">
```

Abb. 2.2 CSS mit der External Methode

Zusammenfassung

In diesem Kapitel wurde auf die verschiedenen Möglichkeiten eingegangen, wie sich CSS in ein HTML Dokument integrieren lässt. Die einzelnen Möglichkeiten wurden unterschieden und jeweilige Vorteile genannt.

Übungsaufgaben

Übung 2.1

Suchen sie sich ein beliebiges HTML Element aus und formatieren sie es mit einer Eigenschaft ihrer Wahl unter Verwendung der Inline Methode.

Übung 2.2

Nennen Sie die Vorteile der External Methode gegenüber der Inline Methode.

3 Selektoren

In diesem Kapitel wird auf die grundlegenden Selektoren von CSS eingegangen.

Dazu gehören die verschiedenen Arten von Selektoren sowie die verschiedenen Arten von Kombinationen dieser. Zusätzlich werden noch Pseudoelemente und Pseudoklassen behandelt.

Nach der Bearbeitung dieses Kapitels sollen sie in der Lage sein,

- *die grundlegenden CSS-Selektoren kennen.*
- *die grundlegenden CSS-Selektoren zu kombinieren um ihre Auswahl einzuschränken.*
- *Pseudoklassen und Pseudoelemente zu verstehen.*

3.1 Einfache Selektoren

Über Selektoren lässt sich auswählen, welche HTML Elemente von den Darstellungsregeln beeinflusst werden. Im Folgenden wird auf die verbreitetsten Basisselektoren eingegangen.

3.1.1 Typselektor

Oftmals möchte man dem gleichen Element auf einer Website das gleiche Aussehen verleihen, damit deren Erscheinung einheitlich und übersichtlich ist. Über den Typselektor lässt sich genau das realisieren. Er wählt alle Elemente eines bestimmten Typs. Im folgenden Beispiel wird für alle Absätze die gleiche Schriftfarbe festgelegt.

```
p{  
color:blue;  
}
```

Abb. 0.1 Typselektor

3.1.2 ID Selektor

Möchte man nur ein einzelnes Element unabhängig von seiner Klasse oder seinem Typ auswählen, lässt sich das über eine ID realisieren. Jedem Element kann eine eindeutige ID zugeordnet werden, sodass mittels ID Selektor auf dieses eine Element zugegriffen werden kann. Das ist nützlich, wenn gezielt einzelne Aspekte einer Website gestaltet werden sollen. Eine ID wird in CSS immer mit einer voranstehenden Raute notiert.

3.1.3 Class Selektor

Jedes Element lässt sich über das HTML Class Attribut einer Klasse zuweisen. Über den Class Selektor werden alle Elemente ausgewählt, die zu einer Klasse gehören. Jeder Class Selektor beginnt mit einem Punkt vor dem Namen der Klasse.

3.1.4 Globalselektor

Über diesen werden alle Elemente des Dokuments angesprochen. Er wird jedoch hauptsächlich in Kombination mit anderen Selektoren verwendet, da man nur selten jedem einzelnen Element eines Dokuments eine Eigenschaft hinzufügen möchte. Der Globalselektor ist ein einzelner Stern.

3.1.5 Attribut Selektor

Möchte man das Vorhandensein eines Attributes als Auswahlkriterium benutzen, lässt sich das über einen Attribut Selektor realisieren. Das Attribut wird in ein eckiges Klammernpaar eingeschlossen. Fügt man dem Attribut mittels =wert noch einen Wert hinzu, muss der Wert im Attribut stehen. In der folgenden Abbildung werden alle Checkboxes markiert, in deren name Attributfeld „sel“ steht.

3.1.6 Kombinierte Selektoren

Der Typselektor und der Klasselektor lassen sich kombinieren. Schreibt man den Typselektor direkt gefolgt von einem Class Selektor werden nur Elemente des Typs ausgewählt, die ebenfalls der Klasse angehören. Ebenfalls kann der Attributselektor an jeden der oben genannten Typen angehängt werden. Das bewirkt, dass das Element zusätzlich zur Attributs Bedingung, noch von dem anderen Selektor erfasst werden muss.

3.2 Selektor Operatoren

Nur die Basisselektoren sind zur Eingrenzung von Elementmengen leider nicht immer ausreichend, weshalb man die einzelnen Selektoren auch miteinander mittels Operators verbinden kann. Im Folgenden werden die verbreitetsten Operatoren erläutert.

3.2.1 Descendant Selektor

Oft kommt es vor, dass man nicht alle Elemente eines Typs verändern möchte, sondern nur jede, welche sich innerhalb eines anderen Elements befinden. Der Descendant Selektor (deutsch Nachkommen Selektor) bietet diese Möglichkeit. In Kombination mit dem Descendant Selektor bietet sich auch der Globalselektor an, da man mit diesem alle Elemente innerhalb eines anderen Elements auswählen kann.

Durch die Darstellungsregel in der Abbildung sollen alle Listeneinträge einer Ungeordneten Liste unterstrichen werden, die einer geordneten Liste allerdings nicht.

3.2.2 Adjacent Selektor

Mittels Adjacent Selector (Angrenzender Selektor) lässt sich ein Element selektieren, welches direkt nach einem anderen folgt. Im folgenden Beispiel wird ein Zitat nur das erste Zitat in einem Absatz umrandet.

```
p + q {  
border: solid;  
}
```

Abb. 0.2 Adjacent Selektor

3.2.3 Parent Selektor

Über den Parent Selektor werden nur direkte Nachkommen eines Elementes selektiert. Befindet sich ein Element auf einer anderen als der nächsten Verschachtelungsebene, wird es nicht ausgewählt. Der ähnliche Descendant Selektor wählt hingegen alle Nachkommen aus.

3.3 Pseudoelemente

Über Pseudoelemente lassen sich, im Gegensatz zu den Basisselektoren, über welche sich immer nur ein ganzes Element und somit auch nur dessen gesamter Inhalt auswählen lässt, auch einzelne Teile eines Elements und somit auch einzelne Teile dessen Inhalt auswählen. Pseudoelemente beginnen immer mit zwei Doppelpunkten. Möchte man auf den gleichen Selektor mehrere Pseudoelemente anwenden, muss man für jedes weitere Pseudoelement den gleichen Selektor mit dem neuen Pseudoelement und einem neuen Deklarationsblock noch einmal notieren.

Im Folgenden werden einige gebräuchliche Pseudoelemente erläutert.

3.3.1 ::before, ::after

Möchte man vor dem eigentlichen Inhalt eines Elements noch eigenen Inhalt einfügen, lässt sich dies mittels ::before bewerkstelligen. Das Pseudoelement ::after verhält sich genau gleich, nur wird der Inhalt nach dem eigentlichen Inhalt des Elementes angefügt. Im Beispiel wird dem Inhalt eines jeden Elements des Typs p der Text „Hallo Welt“ vorangestellt.

```
p::before{
  content:"Hallo Welt";
}
```

Abb. 0.3 Text dem Inhalt eines Elements voranstellen

3.3.2 ::first-line

Über first-line wird nur die erste Zeile eines Elements ausgewählt. Es werden aber nicht nur selbst eingefügte Zeilenumbrüche beachtet, sondern auch durch CSS bzw. den Browser selbst erzeugte.

3.3.3 ::selection

Das ::selection Pseudoelement kommt ohne Selektor aus, denn dieser ist bereits festgelegt. Der Inhalt seines Deklarationsblocks wird auf den Text angewendet, den Benutzer markiert bzw. selektiert hat. Es lassen sich allerdings nur die Eigenschaften color, background, cursor und outline verwenden. Diese verändern in dieser Reihenfolge die Farbe des Textes, die Hintergrundfarbe der Auswahl, das Aussehen des Mauszeigers und die Umrandung der Auswahl.

3.4 Pseudoklassen

Über Pseudoklassen lässt sich abfragen, ob ein Element eine anderen als seinen Normalzustand hat. Spezielle Zustände eines Elements sind unter anderem, ob der Benutzer mit dem Mauszeiger über einem Element steht, oder ein Textfeld ausgewählt wurde. Ähnlich zu Pseudoelementen wird eine Pseudoklasse an einen Selektor angehängt, allerdings beginnen diese nur mit einem Doppelpunkt.

Auch hier wieder eine Erläuterung zu den gängigsten Pseudoklassen.

3.4.1 :link und :visited

Die Pseudoklassen :link und :visited hängen eng zusammen, denn sie beziehen sich auf den gleichen Elementtyp. Über :link werden alle Links ausgewählt, die noch nicht angeklickt und vom Browser als solche gespeichert wurden, :visited wählt genau jede aus, auf die das zutrifft.

3.4.2 :hover

Über :hover wird das Element selektiert, über dem die Maus gerade steht.

3.4.3 :nth-child(n)

Verwendet wird :nth-child(n) um das nte Kind Element eines anderen Elements auszuwählen. Über even oder odd lassen sich auch Eigenschaften der Position angeben. Das kann z.B. dazu eingesetzt werden, um jede zweite Zeile (odd) in einer Tabelle in einer anderen Farbe dazustellen.

3.4.4 :focus

Gilt es das aktuell ausgewählte input Element zu selektieren, bietet sich :focus an. So lässt sich beispielsweise die Farbe eines Textfeldes ändern, sobald der Benutzer es anklickt.

Zusammenfassung

In diesem Kapitel wurden Selektoren behandelt. Dabei wurde näher auf die verschiedenen Arten von Selektoren und Kombinations- bzw. Verbindungsmöglichkeiten dieser eingegangen. Zum Schluss wurde das Selektieren von Elementteilen oder die Auswahl eines Elements über seinen Zustand mittels Pseudoelementen und Pseudoklassen behandelt.

Übungsaufgaben

Übung 3.1

Nennen sie die vier einfachen Selektoren, welche sie in diesem Kapitel kennengelernt haben.

Übung 3.2

Was ist der Unterschied zwischen einem ID-Selektor und einem Klassenselektor?

Übung 3.3

Fertigen sie einen Ausdruck aus Selektoren, welcher alle p Elemente auswählt, welche sich innerhalb eines h1 Elements befinden.

Übung 3.4

Schreiben sie einen Selektor, welcher die erste Zeile des Inhalts jedes p Elements auswählt.

Übung 3.5

Schreiben sie einen Selektor, welcher jedes zweite tr Element innerhalb eines table Elements auswählt.

4 Eigenschaften

Im vorherigen Kapitel wurde auf die verschiedenen Möglichkeiten eingegangen, Elemente je nach ihren Eigenschaften auszuwählen. In diesem Kapitel soll es darum gehen welche Eigenschaften bei den ausgewählten Elementen geändert werden können, was diese im Einzelnen bewirken und wie man sie notiert.

Nach Bearbeitung dieses Kapitels sollten sie in der Lage sein,

- *die verbreitetsten Eigenschaften und ihre Werte zu benennen.*
- *die Shorthand Notation verstehen und anwenden.*

4.1 Shorthand Notation

Gestaltungsregeln in CSS werden nach dem Schema Eigenschaft: Wert notiert. Oftmals kommt es allerdings vor, dass es zu einer Art von Eigenschaft mehrere Untereigenschaften gibt, die einzelne Teilaspekte der Eigenschaft ändern. So gibt es zur Eigenschaft margin (dem Abstand eines Elementes zu seinem umschließenden Element) die Untereigenschaften margin-left, margin-right, margin-top und margin-bottom, welche den Abstand in ihrer Richtung zum Elternelement angeben. Möchte man alle Abstände festlegen, müsste man jede der vier Untereigenschaften und ihren Wert in je eine Zeile schreiben. Hier kommt die Shorthandnotation zur Anwendung. Mit ihr gibt es die Möglichkeit margin: gefolgt von den vier Werten der einzelnen Untereigenschaften zu schreiben. Die Werte werden durch ein Leerzeichen getrennt. Die einzelnen Werte werden entsprechend ihrer Stelle im Uhrzeigersinn den Seiten bzw. Ecken eines Elements zugeordnet. Die erste Ecke ist die obere Linke, die erste Seite ist die obere.

Ein in der Praxis ebenfalls oft vorkommenden Fall ist, dass der gleiche Wert für mehrere Untereigenschaften verwendet werden soll. Möchte man alle Werte aller Untereigenschaften verändern so reicht es aus, einen einzigen Wert zu notieren, dieser wird dann für alle Untereigenschaften übernommen.

Bei der Angabe von zwei oder drei Werten gilt es zwischen Eigenschaften zu unterscheiden, welche die Ecken oder die Seiten eines Elementes beeinflussen.

Bezieht sich eine Eigenschaft auf die Seiten (wie z.B. Margin) gilt:

Bei der Angabe von zwei Werten, verändert der erste Wert die Seiten mit Horizontaler Ausrichtung und der zweite jene mit Vertikaler Ausrichtung. Bei der Angabe von drei Werten bezieht sich der erste Wert auf die obere Seite, der zweite auf die Seiten mit vertikaler Ausrichtung und der dritte auf die untere Seite.

Bezieht sich die Eigenschaft jedoch auf die Ecken eines Elements, so gilt:

Bei der Angabe von zwei Werten, verändert der erste Wert die linke obere und die rechte untere Ecke, und der zweite die linke untere und die rechte obere Ecke. Bei der Angabe von drei Werten bezieht sich der erste Wert auf die obere linke Ecke, der zweite auf die obere rechte und die untere linke Ecke und der dritte Wert auf die untere rechte Ecke.

Im Beispiel sieht man die Margin Eigenschaft mit und ohne Shorthand Notation.

<pre>p{ margin-left: 10px; margin-right: 10px; margin-top: 10px; margin-bottom: 10px; }</pre>	<pre>p{ margin: 10px; }</pre>
---	---------------------------------------

Abb. 4.1 Shorthand bei gleichem Datentyp

Nun gibt es aber noch den Fall, dass es für eine Eigenschaft mehrere einzelne Untereigenschaften gibt, die aber nicht Werte für einzelne Positionen der Eigenschaft angeben, sondern vollkommen unterschiedliche Aspekte. So gibt es für die Background Eigenschaft beispielsweise eine Untereigenschaft, welche die Farbe (background-color) angibt und eine welche den Pfad zu einem Hintergrundbild angibt (background-image). Auch hier lassen sich die einzelnen Werte hinter den Namen der Basiseigenschaft (in diesem Fall background) schreiben. Die Reihenfolge spielt dabei nur bei gleichen Datentypen eine Rolle, ansonsten ist sie beliebig. Gibt man den Wert für eine Untereigenschaft nicht an, wird der jeweilige Standardwert verwendet.

Im folgenden Beispiel werden die einzelnen Untereigenschaften der border-Eigenschaft einzeln und mit Shorthand Notation festgelegt.

<pre>p{ border-width: 2px; border-style: solid; }</pre>	<pre>p{ border:solid 2px; }</pre>
---	---

Abb. 4.2 Border bei unterschiedlichem Datentyp

4.2 Eigenschaften

Im Folgenden werden die verbreitetsten Eigenschaften und ihre mitverwendeten Untereigenschaften erklärt.

4.2.1 border

Mit der Border Eigenschaft lässt sich ein Rahmen um ein Element ziehen. Zu beachten ist, dass eine Border das Element verbreitert.

Über border-style lässt sich festlegen, welche Form der Rahmen haben soll. So gibt beispielsweise die Möglichkeit die Linie solide, gestrichelt oder doppelt darzustellen.

Möchte man die Breite angeben, kann dies über border-width unter Verwendung einer von CSS unterstützten Maßeinheit getan werden.

Die Farbe lässt sich über border-color setzen. Der Farbwert wird kann in mehreren, von CSS unterstützten, Notationsarten angegeben.

Die einzelnen Eigenschaften lassen sich auch gezielt auf verschiedene Seiten des Rahmens anwenden, wenn man sie dazuschreibt. So färbt das untere Beispiel nur die Rechte Seite des Rahmens rot ein.

```
p{  
  border-style: solid;  
  border-right-color: #ff0000;  
}
```

Abb. 4.3 Einzelne Untereigenschaften verändern

4.2.2 margin

Über Margin lässt sich der Außenabstand der Seiten eines Elements zum umschließenden Element angeben. Über margin-top, margin-bottom, margin-left und margin-right lässt sich dieser Abstand für jede Seite individuell angeben.

4.2.3 padding

Padding ist das Gegenstück zu Margin, denn es legt den Innenabstand zwischen dem Element und von ihm umschlossenen Elementen fest. Über padding-top, padding-bottom, padding-left und padding-right lassen sich ebenfalls wieder die einzelnen Seiten voneinander getrennt festlegen.

4.2.4 height und width

height und width geben die Höhe bzw. Breite eines Elements an. Die Angabe erfolgt in einem von CSS unterstützten Größenformat.

4.2.5 max-height, min-height, max-width, min-width

Über diese Eigenschaften kann die maximale bzw. minimale Größe eines Elements festgelegt werden.

4.2.6 outline

Outline verhält sich ähnlich wie border. Es besitzt die gleichen Untereigenschaften, unterscheidet sich jedoch dadurch, dass es die Maße des umrandeten Elements nicht vergrößert. Vorteil ist, dass die Website dadurch einfacher zu formatieren ist, Nachteil ist hingegen, dass es andere Elemente überdecken kann, da Eigenschaften wie margin oder padding ignoriert werden.

4.2.7 text

Über text lassen sich diverse Eigenschaften von Text ändern.

Text-align gibt die horizontale Position im umschließenden Element ein. So kann über die Werte left, right und center angegeben werden, ob der Text linksbündig, rechtsbündig oder zentriert angezeigt wird. Wird justify als Wert angegeben, werden alle Zeilen so gestreckt oder gestaucht, dass sie die gleiche Breite haben.

Eine andere verbreitete Untereigenschaft ist text-decoration. Mit den ihr zugehörigen Werten underline, overline und line-through kann Text entweder unterstrichen, überstrichen oder durchgestrichen werden. Mittels none können auch bereits vorhandene Striche entfernt werden.

Die Höhe einer Zeile wird durch line-height festgelegt.

Über direction kann die Richtung des Textflusses angegeben werden. Der Wert rtl (right to left) sorgt dafür, dass die Buchstaben von rechts nach links angeordnet werden. Gleichzeitig beginnt der Text auch auf der rechten Seite des umschließenden Elements, sofern dies nicht anderweitig festgelegt wurde.

Die Farbe des Textes lässt sich über color angeben. Der Wert muss in einem von CSS unterstützten Format angegeben werden.

4.2.8 display

Mittels display lässt sich das Verhalten eines Elements im Seitenfluss festlegen.

Der Wert none sorgt dafür, dass das Element und seine Kind Elemente gar nicht angezeigt werden und auch sonst keinen Einfluss auf andere Elemente haben.

Inline als Wert bindet das Element normal in den Seitenfluss ein. Für diese Elemente lassen sich keine Höhe oder Breite festlegen, und auch margin und padding werden für die horizontalen Seiten nicht beachtet.

Durch block wird festgelegt, dass das Element normal in der Seitenfluss eingebunden wird. Zusätzlich erzeugt es einen Zeilenumbruch. Höhe, Breite sowie alle margins und paddings können angegeben werden und werden auch beachtet.

Inline-block kombiniert die inline und block Werte. Es lässt aber Höhen- und Breitenangaben sowie alle margin- und Paddingwerte zu und erzeugt keinen Zeilenumbruch.

4.2.9 position

Über die position Eigenschaft lässt sich die Position eines Elements im Seitenfluss anpassen.

Über den Standardwert static wird das Element ganz normal in den Seitenfluss eingebunden.

Verwendet man relative lässt sich über die vier Untereigenschaften top, bottom, left und right die position relativ zur eigentlichen Position des Elements festlegen. Beachtenswert ist, dass die dadurch entstehende freie Fläche im Seitenfluss nicht durch andere Elemente benutzt werden kann.

Über fixed lässt sich unter Verwendung der Untereigenschaften top, bottom, left und right die Position eines Elements relativ zum Viewport (also z.B dem Bildschirm) anpassen. Das bewirkt, dass das Element auch beim Scrollen an der gleichen Stelle des Displays bleibt.

Wird absolute verwendet, wird das Element, an der über top, bottom, left und right festgelegten Stelle, relativ zu nächsten umschließenden positionierten Element platziert.

4.2.10 z-index

Wenn sich mehrere Elemente überdecken lässt sich mittels z-index festlegen, welches Element vorne steht. Der Wert ist hierbei eine ganze Zahl, wobei Elemente mit einem höhere z-index Elemente mit einem niedrigeren überdecken.

4.2.11 overflow

Wenn der Inhalt eines Elements über die Grenzen seines Elternelements herausragt, bietet sich die Nutzung von overflow an.

Der Standardwert ist visible. Er bewirkt, dass der Inhalt Sichtbar ist und nicht abgeschnitten wird.

Der Wert hidden legt fest, dass der überstehende Inhalt versteckt wird.

Möchte man den Inhalt in einer scrollbaren Box anzeigen, lässt sich dies über visible bewerkstelligen.

Zusammenfassung

Dieses Kapitel hat sich mit den einzelnen Eigenschaften beschäftigt, welche man HTML Elementen zuweisen kann. Zuerst wurde die Shorthandnotation und ihre Besonderheiten eingegangen. Im Anschluss wurde auf die weitverbreitetsten Eigenschaften eingegangen und diese wurden näher erläutert.

Übungsaufgaben

Übung 4.1

Beschreiben sie die Vorteile der Shorthandnotation gegenüber der normalen Notation.

Übung 4.2

In welcher Reihenfolge müssen die Werte bei der Shorthandnotation angegeben werden, wenn es sich um unterschiedliche Datentypen handelt.

Übung 4.3

Schreiben sie CSS Code, welcher ein beliebiges Element mit einer 2 Pixel breiten Linie umrandet. Verwenden sie hierbei die Shorthand Notation.

Übung 4.4

Verschieben sie jedes Element X gegenüber seiner normalen Position im Seitenfluss 20px nach rechts. Andere Elemente sollen nicht beeinflusst werden.

Übung 4.5

Begrenzen sie die Maße des Elements p auf mindestens 200 Pixel und maximal 500 Pixel und stellen sie sicher, dass der Inhalt nicht über die gesetzten Grenzen hinausragt.

5 Größenangaben

In diesem Kapitel werden die verschiedenen Möglichkeiten behandelt, wie Größenangaben in CSS verwendet werden. Es wird auf die Unterschiede zwischen relativen und absoluten Größenangaben eingegangen und jeweils Vor- und Nachteile genannt.

Nach der Bearbeitung dieses Kapitels sollten sie in der Lage sein,

- *die unterschiedlichen Größeneinheiten und ihre Bedeutung zu verstehen.*
- *den Unterschied zwischen relativen und absoluten Maßeinheiten erklären zu können.*

Wenn in CSS eine Größe angegeben soll, kann sich einer Vielzahl an Maßeinheiten bedient werden, wobei zu beachten ist, dass nicht jede Eigenschaft jede Maßeinheit unterstützt.

Es gibt relative und absolute Maßeinheiten. Der Unterschied ist, dass relative Längeneinheiten an eine Eigenschaft des Endgeräts oder Dokuments (z.B. Schriftgröße oder Bildschirmbreite) gebunden sind, wohingegen absolute Maßeinheiten unabhängig vom Dokument oder Endgerät immer gleich lang sind.

Relative Maßeinheiten sind Absoluten Maßeinheiten vorzuziehen, da bei absoluten Werten aufgrund der Vielzahl an Endgeräten nicht mehr gewährleistet ist, dass die festgelegten Größen noch zur Bildschirmgröße passen.

Zu den Absoluten Maßeinheiten zählen die SI-Einheiten cm (Zentimeter), mm (Millimeter) und das englische in (Inch). Des Weiteren gibt es noch pt(point) und pc (pica). Ein Point ist ein 72tel eines Inches und ein pica entspricht 12 points. Der Pixel gehört zu den absoluten und relativen Maßeinheiten. Absolut ist es, weil eine von ihm beschriebene Distanz auf einem individuellen Endgerät immer gleich groß ist. Relativ ist es, da er nicht immer nur einem physikalischen Gerätepixel entspricht, sondern bei höherauflösenden Geräten über ein von Gerät zu Gerät unterschiedliches Verhältnis auf mehrere Gerätepixel skaliert wird.

Relative Maßeinheiten umfassen unter anderem em und ex. Diese bieten sich für die Formatierung von Schrift an, da ein em der Höhe eines großen M auf dem Endgerät entspricht. Ein ex ist die Höhe des kleinen x. Großer Vorteil davon ist, dass so vom Endnutzer individuell für das Gerät eingestellte Schriftgrößen als Basis für die Formatierung verwendet werden können. Über vh (viewport-height) und vw (viewport-width) lässt sich ein Hundertstel der Höhe bzw. Breite des sichtbaren Bildausschnitts, also des Viewports, verwenden. Über das Prozent Zeichen kann eine Größe relativ zur jeweiligen Größe des Elternelements angegeben werden. So hat ein Element mit einer Breite von 80%, 80% der Breite seines Elternelements.

Alle Maßeinheiten lassen sich durch direktes Vorstellen einer reellen Zahl mit dieser multiplizieren.

Im Beispiel wird die Schriftgröße des Dokuments auf das Doppelte von em festgelegt.

```
body{  
    font-size: 2em;  
}
```

Abb. 5.1 Schriftgröße mittels em

Zusammenfassung

Dieses Kapitel beschäftigte sich mit den unterschiedlichen Möglichkeiten Größenangaben in CSS anzugeben. Es wurde auf die Unterschiede zwischen relativen und absoluten Größenangaben eingegangen und ihre Besonderheiten erläutert.

Übungsaufgaben

Übung 5.1

Beschreiben sie den Unterschied zwischen relativen und absoluten Maßeinheiten bzw. Größenangaben.

Übung 5.2

Welche Einheit bietet sich für die Festlegung von Schriftgrößen an und warum?

Übung 5.3

Warum sollten absolute Angaben bei Elementmaßen im Allgemeinen vermieden werden?

6 Farben

In diesem Kapitel werden die verschiedenen Möglichkeiten behandelt, wie sich Farbwerte in CSS integrieren lassen. Die Möglichkeiten werden nach Notationsart aufgeschlüsselt und näher erklärt.

Nach der Bearbeitung dieses Kapitels sollen sie in der Lage sein,

- *die verschiedenen Notationsarten für Farbwerte zu nennen.*
- *ihre Besonderheiten aufzuzählen und verstehen.*

6.1 Technik

Der klassische Bildschirm stellt Farben als Mischung aus den drei Primärfarben (Rot, Grün und Blau) dar. Dabei handelt es sich um die additive Farbmischung, da die Lichtwellen der einzelnen Grundfarben gemischt werden, und somit der Eindruck einer Mischfarbe aus den verwendeten Primärfarben entsteht. Um die drei Primärfarbenwerte anzugeben bzw. zu speichern werden pro Wert 8 Bit bzw. ein Byte verwendet. Pro Byte gibt es 256 mögliche Werte. Diese werden im Fall von Farben als Zahl im Bereich von 0-255 angegeben. Die Null steht für keine Intensität der Farbe, die 255 für volle Intensität. So ergeben sich durch die drei Grundfarben alleine $256 * 256 * 256 = \sim 16.700.000$ mögliche Kombinationen bzw. Farben.

6.2 RGB und RGBA

Möchte man eine Farbe in RGB Notation angeben, schreibt man `rgb(x,y,z)`. X, y und z werden hierbei durch die entsprechenden Werte von 0-255 für Rot, Grün und Blau ersetzt. Der Unterschied zu RGBA, mit der Notation: `rgba(x,y,z,a)`, ist, dass hier als vierter Wert noch ein Alphawert (0.0-1.0) angegeben werden kann. Dieser bestimmt die Transparenz der Farbe, also zu welchem Prozentwert die darunterliegenden Inhalte zu sehen sind. Der Wert 1.0 steht für komplette Opazität, 0.0 für maximale Transparenz.

6.3 HEX

Eine kürzere Notation für die einfachen drei Primärfarben ist die HEX Schreibweise. Die einzelnen Primärfarbenwerte werden Hexadezimal codiert und in RGB Reihenfolge hinter eine Raute geschrieben. Pro Farbwert werden immer 2 Stellen verwendet. Bsp: `#00ff00`

6.4 HSL und HSLA

Bei der Verwendung von HSL lässt sich die Farbe über andere Parameter angeben. Die Schreibweise ist `hsl(hue, saturation, lightness)`. Hue gibt eine Gradzahl von 0-360 auf dem Farbkreis an. Der Wert 0 steht für rot, 120 für Grün und 240 für Blau. Werte dazwischen beschreiben dementsprechend einen Mischwert zwischen den beiden nächsten Primärfarben. Über den zweiten Wert saturation lässt sich die Sättigung der Farbe einstellen. Der Wert wird von 0-100 als Prozentwert angegeben, und beschreibt auf dem Farbkreis den Abstand zwischen Kreismittelpunkt und gewählter Gradzahl. Über den letzten Wert lightness lässt sich die Helligkeit anpassen. Die Angabe des Werts erfolgt wie bei der Sättigung. 0% steht für keine Helligkeit, also Schwarz, der Wert 100% für maximale Helligkeit, also weiß. Bei Angabe von 50% wird das Aussehen der Farbe nicht beeinflusst.

Bei der Verwendung von HSLA ist alles wie bei HSL, nur dass sich als letzter Wert noch eine Zahl von 0.0 – 1.0 angeben lässt, welche die Transparenz der Farbe angibt. Bei 0.0 ist die Farbe komplett undurchsichtig, bei 1.0 komplett durchsichtig.

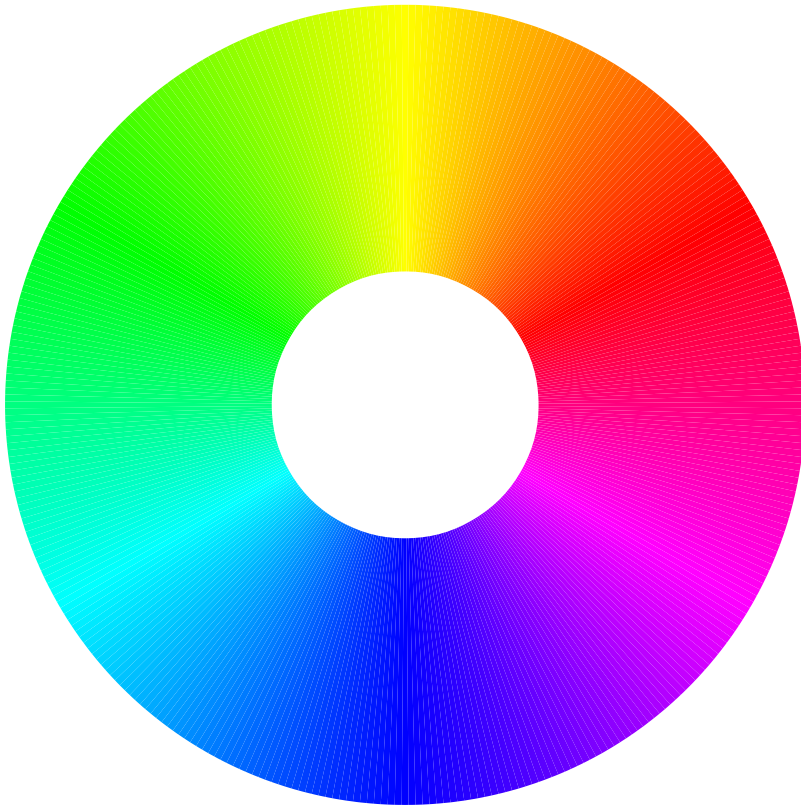


Abb. 6.1 Ein Farbkreis mit Unterstufen

User: NI74, https://upload.wikimedia.org/wikipedia/commons/6/6d/RGB_color_wheel_360.svg, gemeinfrei, 27.11.19

6.5 Farbnamen

In CSS sind einzelne Farbwerte bereits als Farben vordefiniert und lassen sich über ihren Namen abrufen und verwenden. So gibt es z.B. black, blue oder yellow.

Zusammenfassung

In diesem Kapitel wurden die Möglichkeiten zur Farbwertabgabe in CSS beschrieben. Zunächst wurde beschrieben, wie Farben aus technischer bzw. physikalischer Sicht entstehen und repräsentiert werden. Darauffolgend ging es um die einzelnen Notationsmöglichkeiten und ihre Unterschiede.

Übungsaufgaben

Übung 6.1

Wie lassen sich Farben über ihre RGB Werte noch angeben, wenn nicht die `rgb(r,g,b)` Notation verwendet werden soll?

Übung 6.2

Was unterscheidet die `rgb` Notation von der `hsl` Notation?

Übung 6.3

Welche Bedeutung hat der letzte Parameter bei der `rgba` bzw. `hsla` Schreibweise?

7 Responsives Webdesign

In diesem Kapitel werden die Grundelemente von responsiven Webdesign behandelt. Es wird auf die einzelnen Elemente bzw. Prinzipien eingegangen und diese werden näher beschrieben.

Nach der Bearbeitung dieses Kapitels sollten sie in der Lage sein,

- *die Eigenschaften einer responsiven Website aufzuzählen.*
- *beschreiben können, warum das Entwickelt einer responsiven Website sinnvoll ist.*
- *Media Querys zu verwenden.*

Das Aufkommen von immer mehr Geräten mit unterschiedlicher Bildschirmgröße und unterschiedlichem Seitenverhältnis hat einen Wandel im Webdesign bedingt. Vor einigen Jahren war es noch ausreichend, wenn händisch nur für den normalen Standbildschirm optimiert wurde, heute hingegen kommt die Hauptnutzerzahl ohnehin von mobilen Endgeräten unterschiedlichster Größen. Um eine Website auf allen bzw. möglichst vielen Geräten ohne Informations- oder Interaktionsverlust, also ohne ausblenden von Interaktionsmöglichkeiten oder Medien, darzustellen, bedient man sich des responsiven Webdesigns.

Hauptwerkzeuge für das responsive Webdesign sind CSS, Javascript und HTML.

Grundsätzliche Vorgehensweisen des responsiven Webdesigns sind die Verwendung von relativen Maßeinheiten für Größenangaben, das verwenden eines responsiven Layouts und bei Bedarf das Verwenden von Media Querys.

7.1 Layout

Wichtiger Aspekt eines responsiven Layouts ist, dass sich die Elemente in ihren Maßen an die Maße des Elternelements anpassen. Das gilt sowohl für einfache Elemente die z.B. Text beinhalten, sowie auch für Elemente die andere Medientypen, wie z.B. ein Bild beinhalten.

7.2 Media Query

Media Querys bieten die Möglichkeit die normalen CSS Gestaltungsregeln samt Selektor an bestimmte Bedingungen zu knüpfen.

Die allgemeine Syntax einer Media Query ist wie folgt:

```
@media (Bedingung X) [and (Bedingung Y) , (Bedingung Z)]{  
    CSS Gestaltungsregeln  
}
```

Der Teil in den eckigen Klammern ist optional. Die Klammern dienen lediglich der Strukturierung des Beispiels. Jede Media Query beginnt mit dem Schlüsselwort @media gefolgt von geklammerten Bedingungen. Möchte man mehrere Bedingungen verwenden, kann man diese über das Wort and verbinden. Ein Komma wirkt wie ein logisches Oder.

Im Folgenden soll es einen kurzen Abriss über die meistverwendeten Eigenschaften geben, welche als Bedingung für Querys verwendet werden.

7.2.1 Bedingungen

7.2.1.1 Mediatyp

Als Bedingung kann auch ein Mediatyp verwendet werden. Dieser legt fest, um welche Geräteart es sich bei einem Endgerät handelt. Screen steht hierbei für einen handelsüblichen Standbildschirm, print für einen Drucker und handheld für ein mobiles Endgerät. Viele mobile Endgeräte geben sich allerdings als screen aus, um vorzubeugen, dass sie eine reduzierte Version der Website bekommen. Wird der Mediatype nicht angegeben, geht CSS davon aus, dass die Regeln auf alle Endgeräte angewendet werden sollen.

7.2.1.2 height und width

Mittels height und width wird die Höhe bzw. Breite des Viewports abgefragt. Über die Präfixe min- und max- kann ebenfalls eine obere bzw. untere Grenze für die Bedingung angegeben werden. Angabe des Wertes erfolgt in einer von CSS unterstützten Größeneinheit.

7.2.1.3 device-width und device-height

Über device-width und device-height lässt sich eine Breite bzw. Höhe als Bedingung setzen. Hier lassen sich ebenfalls die Präfixe min- und max- verwenden.

7.2.1.4 Orientation

Über orientation lässt sich die Ausrichtung des Geräts bestimmen. Ist die Breite des Viewports größer als dessen Höhe wird der Wert landscape angenommen. Gilt hingegen das Gegenteil, wird portrait verwendet.

7.2.1.5 aspect-ratio

Mittels aspect-ratio wird das Seitenverhältnis des Viewports überprüft. Die Präfixe min- und -max sind erneut zulässig.

7.2.1.6 Resoluion

Zeigt die Website Inhalte an, welche eine gewisse Mindestauflösung zur störungsfreuen Betrachtung benötigen, kann dies mit resolution überprüft werden. Der Wert kann in dpi (Dots per inch), dpcm (dots per centimeter) oder dppx (Dots per pixel) angegeben werden.

Erfüllen mehrere Media Querys ihre Bedingungen und verändern in ihren angehängten Gestaltungsregeln die gleichen Eigenschaften eines Elements, werden die letzten Änderungen übernommen. Im Normalfall entspricht dies den Anweisungen, welche am weitesten unten im CSS Code stehen.

Gängige Praxis für das herausfinden geeigneter Stellen, an denen das Design angepasst werden soll (sogenannter Breakpoints), ist das Verändern der Abmessungen der Website. Bei den Maßen, bei

denen die Website anfängt kaputt zu gehen bzw. zu brechen (engl. break - für Breakpoint) werden die Media Querys gesetzt. Dies wird dem händischen Ausrechnen dieser Position vorgezogen, da dies bei komplexeren Websites deutlich komplizierter wird, und durch ausprobieren garantiert jeder Effekt mitberücksichtigt wird.

In der folgenden Abbildung wird der Hintergrund der Website schwarz eingefärbt, wenn sie eine Breite von mindestens 500 Pixel besitzt und ein Bildschirm ist.

```
@media screen and (min-width:500px){  
  body{  
    background-color:black;  
  }  
}
```

Abb. 7.1 Media Query mit CSS Regel

7.3 Mobile First

Da zum heutigen Zeitpunkt die Mehrzahl der Seitenaufrufe über mobile Endgeräte stattfindet, bietet es sich an die Websites dafür zu optimieren. In der Praxis bezieht sich Mobile First darauf, dass der allgemeine CSS Code für das mobile Endgerät ausgelegt ist, und die Seite über Media Querys an eine Ausgabe am herkömmlichen Bildschirm angepasst wird. Somit wird verhindert, dass das mobile Endgerät erst den CSS Code für herkömmliche Desktops laden und über die Media Querys an die eigenen Bedürfnisse anpassen muss. Das genaue Gegenteil ist der Fall. Das reduziert den Rechenaufwand des Endgeräts, schont dadurch den Akku und spart zusätzlich gegebenenfalls Datenvolumen, da viele größere Bilder auf mobilen Endgeräten gar nicht oder in kleinerer Form angezeigt und somit auch heruntergeladen werden.

Zusammenfassung

In diesem Kapitel wurden zuerst die einzelnen Beweggründe für das responsive Webdesign genannt um daraufhin auf die Grundelemente von jenem einzugehen. Es wurden die Grundeigenschaften eines responsiven Layouts genannt und das Anpassen einer Website via Media Querys beschrieben. Zuletzt wurde noch auf das Designprinzip Mobile First und dessen Vorteile eingegangen.

Übungsaufgaben:

Übung 7.1

Was ist das Hauptziel des responsiven Webdesigns?

Übung 7.2

Wofür werden Media Querys verwendet?

Übung 7.3

Was ist der Hauptzweck des Mobile First Designprinzips?

Übung 7.4

Schreiben sie eine Media Query, welche nur Geräten mit einer Bildschirmbreite von mindestens 500 Pixeln wählt, welche sich zusätzlich noch im Querformat befindet.

8 Anhang

8.1 Lösungen der Aufgaben

1

1.1

Hauptmotivation hinter Cascading Style Sheets (CSS) und ähnlichen Stylesheet-Sprachen ist, die Trennung von Inhalt und Darstellung.

1.2

Eine CSS Gestaltungsregel setzt sich aus einem Selektor und einem Deklarationsblock zusammen.

2

2.1

`<p style="Eigenschaft: wert"></p>`

2.2

Der Wartungsaufwand der Extern Methode ist geringer, da der CSS Code in einer eigenen Datei steht, und von Überall nachgeladen werden kann. Die Inlinemethode verschlechtert zudem die Lesbarkeit des eigentlichen HTML Codes.

3

3.1 ID Selektor, Typ Selektor, Globalselektor, Attributselektor, Class Selektor

3.2 Über den ID Selektor wird das Element gewählt, welches die ID hat. Eine ID ist einzigartig. Über den Klassenselektor werden alle Elemente gewählt, die zu einer Klasse gehören. Dies können beliebig viele sein.

3.3

```
h1 p {  
}
```

3.4

```
p::first-line {  
}  
3.4 p:nth-child(even) {  
}
```

4

4.1

Die Shorthandnotation verkleinert den Schreibaufwand für die gleiche Menge an Anweisungen erheblich und verbessert die Übersichtlichkeit.

4.2

Die Reihenfolge bei unterschiedlichen Datentypen ist beliebig.

4.3

```
p{  
border: solid 2px;  
}
```

4.4

```
X {  
position:relative;  
right:10px;  
}
```

4.5

```
p{  
max-height: 500px;  
min-height: 200px;  
overflow:hidden;  
}
```

5

5.1

Absolute Größenangaben beschreiben Werte, welche von ihrer Umgebung unabhängig sind. Bei relativen Größenangaben ist der Wert immer tatsächliche Wert immer abhängig von anderen Eigenschaften des individuellen Endgeräts.

5.2

Es bietet sich em an, da somit die personalisierten Einstellungen des Endgeräts zur Schriftgröße verwendet werden. Alternativ kann pt verwendet werden, weil dies die Standardeinheit für Schriftgrößen ist.

5.3

Weil durch die Vielzahl an Endgerätegrößen nicht gewährleistet werden kann, dass ein so bemessenes Element auf jedem Endgerät wie gewünscht dargestellt wird.

6

6.1

Über die Hex Schreibweise. Die einzelnen RGB Werte werden hierbei in RGB Reihenfolge jeweils zweistellig und hexadezimal codiert angegeben.

6.2

Bei der rgb Notation wird der Farbwert über Angabe der drei Primärfarbenwerte bestimmt. Bei der hsl Notation gibt man eine Gradzahl auf dem Farbkreis gefolgt von Sättigung und Helligkeit an.

6.3

Er gibt die Transparenz der Farbe (alpha) an. Bei 0.0 ist die Farbe komplett undurchsichtig, bei 1.0 komplett durchsichtig.

7

7.1

Das Hauptziel ist es Websites zu erstellen, welche sich an die individuellen Bedürfnisse (z.B. Bildschirmbreite, Eingabeart) des Endgeräts anpassen.

7.2

Mit Media Querys werden bestimmte Eigenschaften des Endgeräts abgefragt, und auf deren Werten basierend CSS Gestaltungsregeln auf die Website angewendet.

7.3

Die Schonung der Ressourcen Mobiler Endgeräte (Datenvolumen, Akku, Prozessorzeit etc.) durch reduzieren von nicht benötigtem Inhalt.

7.4

@media handheld and (min-width:500px), handheld and (orientation: landscape)

8.2 Literaturverzeichnis

[Fuc19] Paul Fuchs: HTML5: und CSS3 für Einsteiger: Der leichte Weg zur eigenen Webseite; BMU Verlag; 2019; ISBN: 978-3966450256

[Mül15] Peter Müller: Einstieg in CSS Webseiten gestalten mit HTML und CSS; Rheinwerk Verlag; 2015; ISBN 978-3-8362-3683-6

[Duc11] Jon Duckett: HTML and CSS: Design and Build Websites; Wiley Verlag; 2011; ISBN 978-1118008188

8.3 Abbildungsverzeichnis

Abb. 1.1	Allgemeiner Selektoraufbau & konkretes Beispiel	3
Abb. 2.1	CSS mit der Inline Methode	5
Abb. 2.2	CSS mit der External Methode	5
Abb. 3.1	Typselektor	7
Abb. 3.2	Adjacent Selektor	8
Abb. 3.3	Text dem Inhalt eines Elements voranstellen.....	9
Abb. 4.1	Shorthand bei gleichem Datentyp.....	12
Abb. 4.2	Border bei unterschiedlichem Datentyp	12
Abb. 4.3	Einzelne Untereigenschaften verändern	13
Abb. 5.1	Schriftgröße mittels em	16
Abb. 6.1	Ein Farbkreis mit Unterstufen	19
Abb. 7.1	Media Query mit CSS Regel	23