

Inhaltsverzeichnis

1	Einführung	9
1.1.	HTML: wesentliche Technik seit der Entstehung des Internets	10
1.2.	CSS als wichtige Ergänzung zu HTML-Seiten.....	12
1.3.	HTML – eine Auszeichnungssprache	13
1.4.	Statische und dynamische Internetseiten	15
1.5.	HTML als Grundlage für die Programmierung mit PHP, JavaScript und anderen Sprachen	16
2	HTML: Entstehungsgeschichte und Versionen	18
2.1.	HTML: Die Urversion für reine Textauszeichnungen	18
2.2.	Zahlreiche Weiterentwicklungen im Laufe der Jahre	20
2.3.	XHTML: Eine Abwandlung von HTML	21
2.4.	HTML5: Umfangreiche Anpassungen für eine zeitgemäße Darstellung von Internetseiten	23
3	CSS: Unzählige Möglichkeiten für die Gestaltung einer Internetseite	25
3.1.	Das Layout der Internetseiten wurde immer wichtiger	25
3.2.	Einführung von CSS: Trennung von Inhalt und Design	27
3.3.	Das Zusammenspiel zwischen HTML und CSS	28
4	Die Vorbereitungsmaßnahmen für die Gestaltung von Internetseiten mit HTML und CSS	31
4.1.	Verschiedene Webbrowser für die Darstellung der Seiten	31
4.2.	Texteditor für die Gestaltung des Codes	33
4.3.	WYSIWYG-Editoren als Alternative.....	35
5	Der Aufbau einer HTML-Seite	38
5.1.	HTML-Seiten: aufgeteilt in head und body	38
5.2.	Wichtige Elemente innerhalb des head-Bereichs.....	41
5.3.	Die Doctype-Angabe.....	43
5.4.	Die Sprache der Seite angeben.....	45
5.5.	Die Seite validieren	46
5.5.	Übungsaufgabe: die Struktur einer HTML-Seite gestalten.....	48
6	Der body-Bereich für die Inhalte der Seite	51
6.1.	Überschriften für die einzelnen Bereiche der Seite	51
6.2.	Absätze in die Seite einfügen	54
6.3.	Zeilenumbrüche.....	58
6.4.	Verschiedene Textabschnitte hervorheben	61

6.5.	Einzelne Bereiche mit div- und span-Tags kennzeichnen.....	65
6.6.	Kommentare für ein besseres Verständnis des Codes	69
6.7.	Übungsaufgabe: Inhalte in die Seite einfügen.....	70
7	Links in die Seite einbinden	76
7.1.	Das a-Tag: Linktext und Linkziel.....	76
7.2.	Weitere Attribute für den Link vorgeben.....	79
7.3.	Links innerhalb der Seite setzen.....	80
7.4.	Übungsaufgaben: Seiten mit Links erstellen	84
8	Bilder und weitere Medien-Inhalte in die HTML-Seite einbauen	89
8.1.	Das img-Tag für den Einbau von Grafiken.....	89
8.2.	Breite und Höhe festlegen	92
8.3.	Einen Alternativtext vorgeben.....	95
8.4.	figure, figcaption: neue Möglichkeiten durch HTML5	96
8.5.	Sound-Dateien in die Seite integrieren.....	98
8.6.	Videos und Animationen einbinden	100
8.7.	Übungsaufgabe: Medien-Inhalte in die Seite einbauen	101
9	Listen mit HTML gestalten	108
9.1.	Ungeordnete Listen.....	108
9.2.	Geordnete Listen	110
9.3.	Description List: Listen mit weiterführenden Angaben zu den Einträgen.....	113
9.4.	Listen für die Navigation der Seite verwenden	115
9.5.	Übungsaufgabe: mit Listen arbeiten.....	118
10	Tabellen	125
10.1.	Die Tabelle anlegen.....	125
10.2.	Beschriftungen hinzufügen.....	127
10.3.	Zellen miteinander verbinden.....	129
10.4.	Weitere Einteilungen für die Tabelle vornehmen	133
10.5.	Das Layout mit Tabellen vorgeben: eine veraltete Technik	136
10.6.	Übungsaufgabe: Tabellen selbst gestalten.....	144
11	Formulare mit HTML erstellen	149
11.1.	Ein Formular mit dem form-Tag erstellen	149
11.2.	Textfelder erstellen.....	152
11.3.	Radiobuttons für verschiedene Auswahlmöglichkeiten	153
11.4.	Checkbox: Weitere Auswahlmöglichkeiten.....	155
11.5.	Auswahllisten.....	156
11.6.	Mehrzeilige Textfelder	159
11.7.	Versteckte Felder in das Formular einfügen	162
11.8.	Submit- und Reset-Button einfügen.....	162
11.9.	Ein Beispiel-Formular erstellen.....	166
11.10.	Übungsaufgabe: Seiten mit Formularen erstellen	170

12	Semantische Auszeichnungen in HTML5	174
12.1.	Warum ist es sinnvoll, die einzelnen Seitenelemente zu benennen?.....	174
12.2.	Die Struktur der Seite vorgeben: header, nav, main und footer	176
12.3.	Weitere Tags: section, article und aside.....	177
12.4.	Übungsaufgabe: Die einzelnen Bereiche der Seite kennzeichnen.....	178
13	CSS: die Kennzeichen dieser Sprache	183
13.1.	Spezifische Layoutvorgaben für verschiedene Elemente der Seite vorgeben	183
13.2.	Die einzelnen Elemente innerhalb der Seite positionieren.....	185
13.3.	Kommentare in CSS	186
14	Wo kann man die CSS-Vorgaben vornehmen?	188
14.1.	Angaben direkt in das HTML-Element einfügen.....	188
14.2.	CSS-Angaben im Kopfbereich vornehmen	190
14.3.	Separate CSS-Dokumente erzeugen.....	193
14.4.	Die Priorität der CSS-Angaben	195
15	Selektoren: Ein bestimmtes Element über CSS ansprechen	199
15.1.	Vorgaben für alle Tags eines Typs	199
15.2.	Klassen definieren und ansprechen	203
15.3.	Ein Element über seine ID ansprechen	207
15.4.	Auswahl von Tags und Klassen miteinander verbinden.....	210
15.5.	Pseudoklassen verwenden.....	212
15.6.	Übungsaufgabe: Seitenelemente ansprechen	215
16	Texte mit CSS formatieren	220
16.1.	Die richtige Schriftgröße auswählen	220
16.2.	Verschiedene Schriftarten.....	225
16.3.	Die Darstellung der Schriftzeichen	229
16.4.	Eine Textfarbe auswählen.....	231
16.5.	Exkurs: Farben in CSS	233
16.6.	Das Attribut text-decoration.....	237
16.7.	Bündigkeit der Absätze und Einrückungen	239
16.8.	Übungsaufgabe: Texte ansprechend gestalten	242
17	Hintergründe	247
17.1.	Einfache Hintergrundfarben.....	247
17.2.	Bilder als Hintergrund verwenden.....	248
17.3.	Den Hintergrund positionieren und fixieren.....	251
17.4.	Sich wiederholende Grafiken verwenden	253
17.5.	Farbverläufe gestalten	255
17.6.	Übungsaufgabe: ansprechende Hintergründe hinzufügen	258

18	Größen, Abstände und Positionen	265
18.1.	Die Höhe und die Breite eines Elements vorgeben: absolute und relative Werte	265
18.2.	Rahmen für die einzelnen Elemente vorgeben	273
18.3.	Innere und äußere Abstände definieren	282
18.4.	Das CSS Box Model	288
18.5.	Elementen absolute Positionen geben	292
18.6.	Weitere Alternativen für die Bestimmung der Position.....	296
18.7.	float und clear: Inhalte innerhalb eines Containers positionieren	298
18.8.	Darstellung zu großer Inhalte mit overflow anpassen.....	304
18.8.	Übungsaufgabe: Positionen und Abstände vorgeben.....	306
19	Das display-Attribut	313
19.1.	Zeilenelemente als Block- oder als Inline-Element darstellen.....	313
19.2.	Elemente unsichtbar machen	320
19.3.	Elemente mit flexibler Anzeige erzeugen	322
19.4.	Übungsaufgabe: Die Sichtbarkeit der Elemente beeinflussen.....	326
20	Animationen und spezielle Effekte: fortgeschrittene CSS-Techniken	332
20.1.	Transformationen	332
20.2.	Transitionen	338
20.3.	Animationen	342
20.4.	Übungsaufgabe: Die Seite durch Spezialeffekte interessant gestalten	347
21	Responsive Design	353
21.1.	Die Grundidee hinter responsive Design	353
21.2.	Das Meta-Tag Viewport.....	355
21.3.	Unterschiedliche Designs für Desktop-PCs und Smartphones.....	359
21.4.	Eine weitere Option für mittelgroße Geräte (Tablets) hinzufügen.....	368
21.5.	Bilder an die Bildschirmgröße anpassen	371
21.6.	Übungsaufgabe: Seiten mit responsive Design erstellen	374
22	Praxisbeispiel: Eine Website für ein Tierheim anlegen	383
22.1.	Die grundlegende Struktur und Layoutvorgaben für die Website vorgeben	384
22.2.	Grundlegende Vorgaben für die gesamte Seite	387
22.3.	Den Kopfbereich der Seite gestalten	389
22.4.	Die Vorgaben für den Hauptbereich.....	391
22.5.	Die Fußzeile gestalten.....	403
22.6.	Die weiteren Unterseiten hinzufügen	408
23	So geht es weiter	420

Alle Programmcodes aus diesem Buch stehen kostenfrei zum Download bereit. Dadurch müssen Sie Code nicht abtippen.



Außerdem erhalten Sie die eBook Ausgabe zum Buch im PDF Format kostenlos auf unserer Website:



www.bmu-verlag.de/html-css
Downloadcode: siehe Kapitel 23

Kapitel 1

Einführung

Das Internet hat das Leben der Menschen innerhalb von nur zwei Jahrzehnten enorm verändert. Ende der 90er Jahre hatten nur relativ wenige Privathaushalte einen Internetanschluss. Diese Technik setzte sich gerade erst langsam an Schulen, Universitäten und in einigen Unternehmen durch. Das Angebot beschränkte sich im Wesentlichen auf E-Mail-Dienste und Informationsseiten. Um das Internet zu nutzen, kam ein gewöhnlicher PC zum Einsatz. Die Anbindung erfolgte über die Festnetzleitung – manchmal über ISDN aber häufig auch noch auf analogem Weg.

Mittlerweile hat sich dieses Bild in allen Bereichen stark verändert. Einen der wesentlichen Unterschiede stellt die deutlich höhere Übertragungsgeschwindigkeit dar. Seit der flächendeckenden Einführung von DSL haben sich die Ladezeiten stark reduziert. Selbst umfangreichere Daten lassen sich damit in kürzester Zeit herunterladen. Das stellt die Voraussetzung für viele neue Dienste dar – von Online-Computerspielen bis hin zum Video-Streaming. Die höhere Übertragungsgeschwindigkeit führte daher zu einer deutlichen Ausweitung des Angebots. Darüber hinaus haben sich die Geräte, die für den Internetzugang infrage kommen, stark verändert. Vor 20 Jahren kamen hierfür meistens Desktop-PCs zum Einsatz, die keinerlei Mobilität boten. Heutzutage findet der Internetzugang immer häufiger über das Smartphone statt. Dieses kleine Gerät lässt sich bequem in der Hosentasche transportieren und ist daher stets einsatzbereit. Darüber hinaus ist mittlerweile eine schnelle Datenübertragung über das Mobilfunknetz möglich. All diese technischen Neuerungen führen dazu, dass das Internet für die meisten Menschen den ganzen Tag über präsent ist. Man kann jede freie Minute nutzen, um über das Internet mit anderen Menschen zu kommunizieren, um sich Informationen zu beschaffen oder um sich mit Spielen oder Videos die Zeit zu vertreiben.

Ein weiterer wichtiger Punkt ist die wirtschaftliche Bedeutung, die das Internet mittlerweile hat. Hier kann man mittlerweile beinahe alles bestellen – von Elektronik-Artikeln über Kleidung bis hin zu Lebensmitteln. Die Umsätze, die online erzielt werden, steigen bereits seit vielen Jahren enorm an und drängen den stationären Handel immer weiter zurück. Darüber hinaus sind immer mehr Produkte auf das Internet angewiesen. Ein Haushalt ohne Smart-TV oder automatisierten Haushalt-Geräten ist mittlerweile kaum mehr vorstellbar. Auch in der Industrie kommen immer häufiger vernetzte Maschinen zum Einsatz. Für all diese Anwendungen ist das Internet unverzichtbar.

HTML und CSS stellen zwei grundlegende Techniken dar, die für die Gestaltung von Internetseiten zum Einsatz kommen. HTML wurde bereits Ende der 80er Jahre entwickelt und stellte die Grundlage der ersten Internetseiten dar. CSS kam zwar erst einige Jahre später hinzu. Doch zählt diese Technik ebenfalls zu den grundlegenden Elementen für die Gestaltung einer Internetseite. Mittlerweile kamen zwar noch viele weitere Möglichkeiten hinzu. Für dynamisch gestaltete Seiten kommen beispielsweise häufig die Programmiersprachen PHP und JavaScript zum Einsatz. Beide bauen jedoch auf HTML auf. Um eine dynamische Seite mit diesen Programmiersprachen zu erstellen, benötigt man daher ebenfalls gute HTML- und CSS-Kenntnisse. Diese Techniken sind bis heute für die meisten Internetangebote unverzichtbar. Wer sich mit der Gestaltung von Internetseiten befassen will, sollte daher auf jeden Fall zunächst HTML und CSS lernen. Dieses Buch vermittelt die hierfür notwendigen Kenntnisse.

1.1. HTML: wesentliche Technik seit der Entstehung des Internets

Die Ursprünge des Internets gehen bereits bis in die späten 60er Jahre zurück. Damals entwarf die amerikanische Luftwaffe ein Projekt mit der Bezeichnung ARPANET (Advanced Research Projects Agency Network). Dieses diente dazu, verschiedene Universitäten, die im Auftrag der US-Armee forschten, miteinander zu vernetzen. Obwohl der Grundgedanke dieses Projekts sehr ähnlich wie beim heutigen Internet

ist, war die Umsetzung dennoch eine ganz andere. HTML kam im ARPANET beispielsweise nicht zum Einsatz. Die Entwicklung dieser Auszeichnungssprache erfolgte erst wesentlich später.

Die Entstehung von HTML geht auf ein wissenschaftliches Projekt zurück. Das Internet – und damit auch die hierfür verwendete Auszeichnungssprache HTML – entstand am Kernforschungszentrum CERN. Dieses verfügte über zwei Standorte – einen in Frankreich und einen in der Schweiz. Diese Aufteilung machte es jedoch schwierig, die Forschungsergebnisse, die die Mitarbeiter erzielten, den Kollegen am jeweils anderen Standort zugänglich zu machen. Kopien auf digitalen Datenträgern oder sogar auf Papier zu erstellen und über die Grenze hinweg zu transportieren, war mit einem erheblichen Aufwand verbunden.

Aus diesem Grund hatte der Physiker Tim Berners-Lee, der am CERN beschäftigt war, bereits 1980 eine neue Lösung für dieses Problem vorgeschlagen, in deren Rahmen die Daten digital übertragen werden sollten. Bis zur Umsetzung sollten jedoch noch viele Jahre vergehen. Erst 1989 wurde ein Team zusammengestellt, das konkrete Lösungen hierfür entwickeln sollte. Der Projektleiter war der bereits erwähnte Tim Berners-Lee. Für die Datenübertragung per Telefonleitung waren verschiedene Techniken notwendig. Berners-Lee entwickelte beispielsweise den Übertragungsstandard HTTP (Hyper Text Transfer Protocol), der für die Übermittlung der Daten zum Einsatz kam. Darüber hinaus schrieb er eine Server-Software, die die HTTP-Anfragen der Anwender beantworten konnte. Schließlich stellte sich noch die Frage, wie die Informationen dargestellt werden sollten. Es wäre sicherlich möglich gewesen, sie in reiner Textform zu präsentieren. Das wäre jedoch sehr unübersichtlich gewesen – insbesondere bei langen wissenschaftlichen Ausarbeitungen. Aus diesem Grund beschloss das Entwickler-team, hierfür eine Auszeichnungssprache zu gestalten. Deren wesentlicher Zweck bestand darin, einzelne Textelemente wie Überschriften oder Absätze deutlich zu machen und sie anschließend übersichtlicher darzustellen. Eine große Bedeutung hatten dabei auch Links. Diese erlaubten es, Dokumente, die für die Forschungsergebnisse ebenfalls von Bedeutung waren, einzubinden. Die Sprache, die Berners-Lee mit sei-

nem Team entwickelte, erhielt den Namen HTML (Hyper Text Markup Language). Die erste offizielle HTML-Spezifikation erschien 1992. Aufgrund der Leistungen rund um die Entwicklung von HTTP und HTML gilt Berners-Lee als der Vater des Internets.

1.2. CSS als wichtige Ergänzung zu HTML-Seiten

Die ersten Internetseiten, die Anfang der 90er Jahre mit reinem HTML entstanden, waren sehr schlicht gehalten. Sie bestanden vorwiegend aus reinem Text. Dieser war in Überschriften und Absätze gegliedert. Hinzu kamen Listen und Links. Die meisten Seiten bestanden ausschließlich aus schwarzem Text auf weißem Hintergrund. 1993 kam schließlich die Möglichkeit hinzu, Bilder und Grafiken in die Seiten einzubinden. Dieses Element diente in erster Linie dazu, wissenschaftliche Schaubilder zu integrieren. Es bot jedoch bereits die Möglichkeit, das Layout der Seite etwas ansprechender zu gestalten. Insgesamt bleibt jedoch festzuhalten, dass HTML nur sehr wenige Möglichkeiten bot, um ein ansprechendes Design zu erstellen.

Bereits nach kurzer Zeit stellte sich heraus, dass das Internet nicht nur für wissenschaftliche Publikationen von Interesse war. Darüber hinaus kam es schon nach einiger Zeit auch für die Gestaltung von Nachrichtenportalen, Informationsseiten und sogar für die ersten Online-Shops zum Einsatz. Durch diese Entwicklung bekam das Layout eine vollkommen neue Bedeutung. Die Besucher waren nun nicht mehr ausschließlich auf die Inhalte fokussiert. Darüber hinaus sollten sie durch eine ansprechende Darstellung dazu gebracht werden, die entsprechende Seite regelmäßig zu besuchen.

Es gab einige Ansätze, die versuchten, die bisherigen HTML-Tags für die Gestaltung ansprechender Seiten zu verwenden. Insbesondere Tabellen kamen häufig zum Einsatz, um der Seite eine Struktur zu verleihen. Allerdings führten diese Techniken nicht zu optimalen Ergebnissen. Daher wurde schnell klar, dass es notwendig war, die Gestaltungsmöglichkeiten zu erweitern. Einige Ansätze sahen vor, HTML um entsprechende Befehle zu erweitern. Diese konnten sich jedoch nicht durch-

setzen. Anstatt dessen wurde eine parallele Sprache für das Layout der Seite entwickelt.

Treibende Kraft hinter diesem Vorhaben war Håkon Wium Lie, der wie Berners-Lee am CERN arbeitete. Dieser stellte 1994 einen ersten Vorschlag für dieses Projekt vor – unter der Bezeichnung “Cascading HTML style sheets”. Parallel dazu entwickelte Bert Bos einen neuen Browser mit der Bezeichnung Argo, der ebenfalls ein Style-Sheet-Konzept umsetzen sollte. Aufgrund der Überschneidungen beschlossen die beiden Wissenschaftler, ihre Projekte zu vereinigen. Die Sprache sollte Cascading Style Sheets (CSS) heißen.

Nachdem die Entwickler ihre Ergebnisse auf Konferenzen in den USA in den Jahren 1994 und 1995 präsentiert hatten, wurde das World Wide Web Consortium (W3C) auf CSS aufmerksam. Diese Institution war mittlerweile dafür verantwortlich, allgemeine Web-Standards zu definieren. CSS fand schnell viele Anhänger. Die Sprache erweiterte nicht nur die Gestaltungsmöglichkeiten für Internetseiten erheblich. Darüber hinaus trennte sie das Layout vom Inhalt. Auf diese Weise erleichterte sie auch die Wartung der Seiten erheblich. Sie machte es möglich, Inhalte zu verändern, ohne dadurch das Design zu beeinflussen. Mittlerweile hat sich CSS neben HTML zur wichtigsten Web-Technologie entwickelt.

1.3. HTML – eine Auszeichnungssprache

Viele Menschen, die gerne mit der Gestaltung von Internetseiten beginnen möchten, gehen davon aus, dass es sich bei HTML um eine Programmiersprache handelt. Diese Annahme ist jedoch falsch. HTML ist eine Auszeichnungssprache. Dieses Kapitel stellt vor, worin die Unterschiede bestehen.

Eine Programmiersprache dient dazu, ein Computerprogramm zu gestalten. Zu diesem Zweck verfügt sie über Ein- und Ausgabebefehle. Das bedeutet, dass entweder der Anwender Eingaben machen kann oder, dass das Programm dazu in der Lage ist, Daten aus anderen Quellen einzulesen. Die Ergebnisse, die es erzeugt, kann es ausgeben – entweder

direkt über den Bildschirm oder indem es die Daten in eine Datei oder in eine Datenbank schreibt. Darüber hinaus erlaubt es eine Programmiersprache, Daten in Variablen zu erfassen. Man kann mit diesen daraufhin verschiedene Rechenoperationen durchführen. Auch Texte in Form von Zeichenketten lassen sich damit verarbeiten. Schließlich enthalten Programmiersprachen verschiedene Steuerungsanweisungen. Diese erlauben es beispielsweise, eine bestimmte Aktion an eine Bedingung zu knüpfen oder zu wiederholen.

HTML verfügt über kaum eine der genannten Funktionen. Diese Sprache ermöglicht es lediglich, bestimmte Stellen eines Texts auszuzeichnen – beispielsweise als Überschrift, als Zeile einer Liste oder als gewöhnlichen Absatz. Darüber hinaus ist es möglich, einige zusätzliche Angaben zum Dokument zu machen, wie etwa einen Titel vorzugeben oder wichtige Schlüsselbegriffe dafür zu nennen. Doch auch diese Bereiche verfügen lediglich über gewöhnlichen Text und die Tags, die für die Auszeichnung notwendig ist. Damit ist es jedoch weder möglich, einen Ablauf vorzugeben, noch eine Berechnung durchzuführen. HTML sorgt lediglich dafür, dass ein Text-Dokument eine Struktur erhält. Der Browser interpretiert diese Angaben daraufhin und sorgt für eine entsprechende Darstellung – beispielsweise indem er Überschriften größer gestaltet oder indem er Listen und Tabellen entsprechend der Vorgaben zusammenfügt.

Lediglich im Bereich der Ein- und Ausgabe bestehen einige Gemeinsamkeiten mit einer Programmiersprache. HTML erlaubt es, den Inhalt auf dem Bildschirm auszugeben. In sehr begrenztem Umfang ist es auch möglich, Werte vom Anwender abzufragen, indem man ein Formular mit Eingabefeldern erstellt. HTML kann diese Werte jedoch nicht selbst auswerten. Es ist lediglich möglich, sie an ein anderes Programm zu übermitteln.

Aus dieser Aufzählung wird deutlich, dass sich HTML stark von einer Programmiersprache unterscheidet. Das bedeutet auf der einen Seite, dass man auf diese Weise nur statische Darstellungen erzeugen kann, die keine wirkliche Interaktion mit dem Anwender erlauben. Auf der anderen Seite ist die Komplexität von HTML im Vergleich zu einer Pro-

grammiersprache ausgesprochen gering. Das macht es leicht, diese Auszeichnungssprache zu erlernen.

1.4. Statische und dynamische Internetseiten

1

Wenn der Anwender eine Internetseite aufruft, fordert der Browser eine Datei vom Server an. Deren Inhalte stellt er daraufhin dar, damit der Besucher diese aufnehmen kann. Dabei gibt es jedoch Unterschiede zwischen statischen und dynamischen Internetseiten. Um den Anwendungsbereich von HTML zu verstehen, ist es wichtig, auf dieses Detail kurz einzugehen.

Wenn eine Internetseite statisch ist, bedeutet das, dass eine Datei stets die gleichen Inhalte liefert. Die Internetseite sieht immer genau gleich aus – unabhängig davon, wer sie aufruft und zu welchem Zeitpunkt dies passiert. Die einzige Möglichkeit, um die Inhalte zu verändern, besteht darin, die Datei manuell zu bearbeiten und sie in der aktualisierten Version auf dem Server abzulegen.

Dynamische Internetseiten können ihre Inhalte hingegen verändern. Die Datei, die abgerufen wird, bleibt dabei jedoch stets die gleiche. Allerdings läuft hierbei ein Computerprogramm ab, das die Inhalte zusammenstellt. Auch das Layout lässt sich damit beeinflussen. Die entsprechenden Programme rufen beispielsweise Informationen aus Datenbanken ab und fügen diese dann in die Seite ein. Wenn sich die Inhalte der Datenbank verändern, ändert sich auch die angezeigte Internetseite – obwohl die Datei, die sie erzeugt, die gleiche bleibt. Darüber hinaus gibt es dynamische Internetseiten, die Eingaben des Nutzers aufnehmen und damit die Inhalte der Seite gestalten können. Wenn ein Webshop beispielsweise die Produkte auflistet, die der Kunde bestellt hat, handelt es sich dabei um ein dynamisches Angebot.

HTML dient lediglich der Erstellung statischer Internetseiten. Das bedeutet, dass die Inhalte, die sie präsentieren, stets gleich bleiben. Nur wenn man die zugrunde liegende Datei bearbeitet, ändern sich auch die Inhalte. Dynamische Angebote sind mit reinem HTML nicht mög-

lich. Hierfür ist es notwendig, eine Programmiersprache zu verwenden. Besonders häufig kommt PHP oder JavaScript für diese Aufgabe zum Einsatz.

1.5. HTML als Grundlage für die Programmierung mit PHP, JavaScript und anderen Sprachen

Im vorherigen Abschnitt wurde dargelegt, dass HTML der Erstellung statischer Internetseiten dient. Für dynamische Angebote sind Programmiersprachen – in erster Linie PHP und JavaScript – notwendig. Allerdings benötigen auch Webdesigner, die sich der Gestaltung dynamischer Internetseiten widmen wollen, sehr gute HTML-Kenntnisse.

Wenn man eine dynamische Seite serverseitig programmiert, dann muss man dafür ein entsprechendes Programm schreiben. Dieses ruft beispielsweise Informationen aus Datenbanken, Dateien oder von anderen Internetseiten ab. Auch die Eingaben des Besuchers können verwendet werden. Diese Informationen bereitet das Programm dann auf und gestaltet damit eine neue Seite.

Dabei muss man das Programm jedoch so gestalten, dass es eine gültige HTML-Seite erzeugt. Diese Seite übermittelt es dann an den Browser, sodass dieser sie darstellen kann. Wenn man beispielsweise PHP für diese Aufgabe verwendet, muss der Programmierer das Programm so gestalten, dass die einzelnen Ausgabebefehle HTML-Code erzeugen.

Auch für clientseitig programmierte dynamische Seiten spielt HTML eine wichtige Rolle. Hierbei sind die Befehle in eine HTML-Seite eingebaut und können die Inhalte der einzelnen HTML-Elemente beeinflussen.

Daraus ergibt sich, dass HTML auch für dynamische Seiten von großer Bedeutung ist. Hierfür ist es selbstverständlich notwendig, eine entsprechende Programmiersprache zu erlernen. Um die Seiten zu gestalten, sind jedoch auch gute HTML-Kenntnisse erforderlich. Wenn ein Programm HTML-Code ausgeben soll, muss man selbstverständlich

genau wissen, wie dieser aufgebaut ist. Auch wenn man clientseitige Scripte in eine HTML-Seite einbettet, sind für deren Gestaltung profunde Kenntnisse in dieser Auszeichnungssprache notwendig. Daher besteht der erste Schritt stets darin, diese zu erlernen. Wenn man später dynamische Angebote erstellen will, kann man auf diesen Kenntnissen aufbauen und die hierfür notwendigen Techniken zusätzlich erlernen.

Kapitel 2

HTML: Entstehungsgeschichte und Versionen

Bevor wir damit beginnen, die ersten Seiten mit HTML zu erstellen, sollen hier kurz die Entstehungsgeschichte und die verschiedenen Versionen von HTML etwas ausführlicher vorgestellt werden. Das ist nicht nur wichtig, um die Ursprünge und die Anwendungen dieser Auszeichnungssprache besser zu verstehen. Darüber hinaus ist das notwendig, um die Unterschiede der verschiedenen Versionen besser nachvollziehen zu können. Dieses Buch stellt vor, wie man Seiten mit HTML5 gestaltet. Internetangebote, die eine der früheren Versionen verwenden, können jedoch erhebliche Unterschiede dazu aufweisen. Als Webdesigner kommt es immer wieder vor, dass man Seiten nicht von Grund auf selbst erstellt, sondern die Entwürfe anderer Designer überarbeiten muss. Wenn diese in einer anderen Version erstellt sind, muss man den Code ebenfalls verstehen können.

2.1. HTML: Die Urversion für reine Textauszeichnungen

Die Entwicklung von HTML geht auf das Jahr 1989 zurück. In diesem Jahr begann Tim Berners-Lee mit seinem Projekt, Computer miteinander zu vernetzen. 1990 entwickelte er bereits eine Server-Software und einen Browser als Grundlage des Internets. Dabei kam bereits ein erster Vorläufer von HTML zum Einsatz. Diese Sprache orientierte sich an SGMLguid – einer Auszeichnungssprache, die am CERN für die Formatierung wissenschaftlicher Dokumente zum Einsatz kam. Nachdem die Testphase beendet war, veröffentlichte Berners-Lee einen Artikel mit der Bezeichnung HTML-Tags. Dieser stellte 18 verschiedene Tags vor. Dabei handelt es sich um die Urversion von HTML, die manchmal auch als HTML 1 bezeichnet wird. Allerdings hatte sie zum Zeitpunkt der Veröffentlichung keine Versionsnummer.

Bereits die Tatsache, dass die Urversion nur 18 verschiedene Tags enthielt, lässt darauf schließen, dass sie nur stark eingeschränkte Möglichkeiten bot. Außerdem kommt hinzu, dass einige dieser Tags lediglich interne Funktionen für den entsprechenden Browser hatten. Daher trugen sie nicht dazu bei, den eigentlichen Inhalt zu formatieren.

Allerdings waren in dieser Zusammenstellung bereits viele wichtige Tags enthalten. Die Mehrheit von ihnen wird auch heute noch von aktuellen HTML-Versionen unterstützt. Bei manchen von ihnen handelt es sich auch bei aktuellen Seiten um die am häufigsten verwendeten HTML-Tags.

Die 18 Tags der Urversion machten es beispielsweise möglich, einen Titel für ein Dokument zu bestimmen, Überschriften auszuzeichnen und gewöhnliche Absätze zu definieren. Außerdem unterstützte sie zwei verschiedene Arten von Listen. Darüber hinaus war es von Anfang an möglich, Links in das Dokument einzufügen, um Querverbindungen zu anderen Dokumenten zu erzeugen.

Der Aufbau der einzelnen Tags ist ebenfalls seit der Urversion der gleiche geblieben. Jedes Tag beginnt mit einem Kleinerzeichen (<). Danach steht der Name des Tags und darauf folgt ein Größerzeichen (>). Wenn man beispielsweise ein p-Tag für einen Absatz erstellen will, sieht dieses so aus: <p>. Die Groß- und Kleinschreibung spielte dabei ursprünglich keine Rolle. Die Tags <p> und <P> haben genau die gleiche Funktion. Es ist jedoch mittlerweile üblich, die Namen der Tags klein zu schreiben.

Die meisten Tags verfügen über ein öffnendes und ein schließendes Element. Dazwischen steht der Inhalt, der dargestellt werden soll. Für das schließende Tag muss man erneut den Namen des Tags in Kleiner- und Größerzeichen schreiben. Allerdings ist dabei noch ein Schrägstrich integriert, der anzeigt, dass es sich hierbei um das schließende Tag handelt. Um den Absatz zu beenden, ist daher folgendes Zeichen notwendig: </p>

Zwischen diesen beiden Tags steht ein beliebiger Text. Dieser wird dann vom Browser als ein zusammenhängender Absatz präsentiert:

<p>Hier steht ein Absatz</p>

Darüber hinaus gibt es jedoch auch einige Tags, die kein schließendes Element haben. Dabei handelt es sich um Inhalte, die keinen Text enthalten. Wenn man beispielsweise einen einfachen Zeilenumbruch oder ein Bild einfügen will, soll hierbei kein Text auf der Seite angezeigt werden. Daher ist auch kein Text vorhanden, den die Tags umschließen könnten. Aus diesem Grund ist für derartige Elemente nur ein einziges Tag notwendig.

2.2. Zahlreiche Weiterentwicklungen im Laufe der Jahre

Wenn man die im vorherigen Abschnitt beschriebenen Möglichkeiten mit einer modernen Internetseite vergleicht, dann wird schnell klar, dass es im Laufe der Zeit zu zahlreichen Änderungen gekommen sein muss.

Die erste umfangreiche Erweiterung des HTML-Wortschatzes fand bereits kurz nach der Veröffentlichung der ursprünglichen Version statt. 1993 erschien ein weiterer Artikel, der neue Tags vorstellte. Von besonderer Bedeutung war dabei die Einführung des `img`-Tags. Dieses erlaubte es, Fotos, Schaubilder oder weitere Grafiken in die Seiten einzufügen. Zu dieser Zeit stand bei der Entwicklung von HTML nach wie vor die Darstellung wissenschaftlicher Schriften im Vordergrund. Diese verwendeten jedoch häufig Schaubilder, um ihre Ergebnisse zu veranschaulichen. Das machte es notwendig, eine Möglichkeit für die Einbindung grafischer Elemente bereitzustellen. Diese Entwicklung erlaubte es jedoch auch, die Seiten für den Betrachter deutlich anschaulicher zu gestalten. Sie war ein wichtiger Schritt, um das Internet zu einem beliebten Informations- und Unterhaltungsmedium zu machen. Darüber hinaus wurden in dieser neuen Version viele weitere Auszeichnungen implementiert. Beispielsweise war es fortan möglich, den Text fett oder kursiv darzustellen.

Bei allen bisherigen HTML-Versionen handelte es sich nicht um offizielle Standards, sondern um persönliche Publikationen von Berners-Lee. Dennoch hatten sie für die ersten Entwicklungsschritte des Internets

eine große Bedeutung. Der erste allgemeingültige HTML-Standard wurde erst 1995 veröffentlicht – unter der Bezeichnung HTML 2.0. Damit bestand erstmals ein verbindliches Regelwerk für die Verwendung von HTML. Darüber hinaus wurden einige neue Funktionen eingeführt. Insbesondere die Verwendung von Formularen ist dabei hervorzuheben.

Die eigentlich geplante Version HTML 3.0 wurde niemals veröffentlicht. Daher kam es erst im Januar 1997 zu einer neuen Ausführung: HTML 3.2. Diese führte unter anderem Tabellen ein. Außerdem bot sie die Möglichkeit, Applets einzubinden. Dabei handelt es sich um kleine Programme, die in der Regel in Java programmiert wurden. Diese ließen sich in die Seite integrieren und konnten dynamische Inhalte erzeugen.

Im Dezember des gleichen Jahres erschien bereits HTML 4.0. Diese Version zeichnete sich in erster Linie dadurch aus, dass sie Stylesheets (CSS) unterstützte. Damit war es möglich, präzise Layouts zu erstellen. Das führte zu Internetseiten, die optisch deutlich ansprechender wirken und außerdem leichter zu pflegen sind. CSS entwickelte sich zu einer der wesentlichen Technologien des Internets und stellt deshalb auch einen wichtigen Teil dieses Buchs dar. Daher ist der Übergang zu HTML 4.0 als ein wichtiger Umbruch zu bewerten, der bis heute die meisten Internetangebote prägt.

2.3. XHTML: Eine Abwandlung von HTML

Im Jahre 2000 erschien mit XHTML eine weitere Möglichkeit für die Gestaltung von Internetseiten. Diese Abkürzung steht für Extensible Hypertext Markup Language. Bereits der Name lässt vermuten, dass hierbei eine enge Beziehung zu HTML besteht. Nicht nur der Anwendungsbereich ist genau der gleiche. Auch die Syntax ist sehr ähnlich. Wer sich bereits gut mit HTML auskennt, sollte daher keine Schwierigkeiten damit haben, Dokumente in XHTML zu verstehen. Dennoch gibt es dabei einige Besonderheiten zu beachten.

XHTML ist eine Neuformulierung von HTML unter Beachtung der XML-Syntax. Hierbei handelt es sich um eine weitere Auszeichnungs-

sprache, die jedoch einen deutlich weiteren Anwendungsbereich hat. XML dient in erster Linie dazu, Informationen so abzuspeichern, dass sie einfach maschinell zu verarbeiten sind. Auch XML verwendet Tags, die mit einem Kleinerzeichen beginnen und mit einem Größerzeichen beendet werden. Allerdings kann man hierbei die Namen der Tags frei wählen und an die Bedeutung des jeweiligen Inhalts anpassen.

Eine der grundsätzlichen Regeln von XML besteht darin, dass hierbei jedes Tag wieder geschlossen werden muss. Das führt zu einem der wesentlichen Unterschiede zwischen XHTML und gewöhnlichem HTML. Die vorherigen Versionen beinhalteten einige Tags, die nur für sich alleine standen, ohne dass sie geschlossen werden mussten. Ein Beispiel hierfür ist das `br`-Tag, das einen Zeilenumbruch erzeugt. In XHTML darf hingegen kein Tag alleine stehen. Daher ist es auch hierbei notwendig, ein schließendes Tag einzufügen: `
</br>`. Eine weitere Möglichkeit besteht darin, den Schrägstrich, der das schließende Tag kennzeichnet, bereits in das ursprüngliche Tag einzubauen: `
`. Wenn man diesen Strich am Ende des Tags einfügt, gilt es ebenfalls als beendet.

HTML ist außerdem nicht sehr streng bei der Auslegung der Syntaxregeln. Da viele Internetseiten von Personen erstellt wurden, die nur begrenzte Kenntnisse in diesem Bereich vorwiesen, kam es dabei zu zahlreichen Fehlern. Die Anwender empfinden es jedoch sehr unangenehm, wenn anstatt der Inhalte Fehlermeldungen auf der Seite erscheinen. Aus diesem Grund sind die gängigen Browser so gestaltet, dass sie die Inhalte trotz der Syntaxfehler darstellen. Es ist sogar häufig möglich, das End-Tag wegzulassen, obwohl dieses eigentlich nach den offiziellen Regeln erforderlich wäre. XHTML ist in dieser Hinsicht jedoch wesentlich strenger. Wenn in einem XHTML-Dokument ein Tag nicht geschlossen wird, wird dieser Fehler im Browser angezeigt.

Ein weiterer Unterschied besteht darin, dass XHTML ausschließlich klein geschriebene Tags akzeptiert. Bei gewöhnlichem HTML kann man diese auch groß schreiben. Darüber hinaus gibt es einige Unterschiede hinsichtlich der Angabe der Attribute eines Tags.

XHTML war über mehrere Jahre hinweg von großer Bedeutung für die Gestaltung von Internetseiten. Mittlerweile ist diese Technik jedoch überholt. Das bedeutet für die Leser dieses Buchs, dass sie sich mit dieser HTML-Abwandlung nicht weiter befassen müssen. Wenn man jedoch einmal auf eine ältere Internetseite trifft und den Quellcode bearbeiten muss, ist es möglich, dass es sich dabei um XHTML handelt. Damit in diesem Fall keine Verwirrung auftritt, ist es wichtig, die wichtigsten Grundlagen dieser Technik zu verstehen.

2.4. HTML5: Umfangreiche Anpassungen für eine zeitgemäße Darstellung von Internetseiten

Die bisher vorgestellten HTML-Versionen entstanden zu einer Zeit, zu der das Internet noch eine vergleichsweise geringe Bedeutung hatte. Im Jahre 2000 verfügte nur ein relativ kleiner Teil der Haushalte über einen Internetanschluss. Die Anschlüsse waren außerdem sehr langsam. Die Angebote bestanden vorwiegend aus Text und aus einigen Bildern.

In den darauffolgenden Jahren kam es jedoch zu einer technischen Revolution in diesem Bereich. Die Zahl der Internetanschlüsse stieg rasant an. Auch die Geräte, mit denen die Anwender auf das Internet zugriffen, änderte sich. Schließlich erweiterte sich die Bandbreite für die Datenübertragung erheblich. Das machte es möglich, auch umfangreiche Inhalte zu übermitteln – beispielsweise Animationen oder Videos.

HTML 4 war für diese Techniken jedoch genauso wenig geeignet wie XHTML. Das führte dazu, dass es hierfür notwendig war, den Browser mit Plug-Ins auszustatten. Sehr beliebt war beispielsweise die Verwendung von Flash. Damit war es möglich, interaktive Animationen zu programmieren. Auch für die meisten Audio- und Video-Inhalte waren Plug-Ins erforderlich. Das brachte jedoch zahlreiche Probleme mit sich. Für die Anwender war es mühsam, immer wieder neue Programme zum Browser hinzuzufügen. Die Entwickler konnten sich nicht sicher sein, welche Plug-Ins die Nutzer installiert hatten. Deshalb war es im-

mer notwendig, alternative Formate anzubieten. Das erhöhte den Aufwand bei der Gestaltung der Seiten erheblich.

Diese Probleme führten dazu, dass der Wunsch entstand, diese Elemente in HTML einzubinden. Das machte es notwendig, HTML von Grund auf zu überarbeiten. In der Zwischenzeit erschienen zwar noch einige weitere Fassungen von XHTML. Doch liefen die Arbeiten für die neue HTML-Version bereits seit vielen Jahren auf Hochtouren. Es sollte jedoch noch einige Zeit dauern, bis sie endgültig vorgestellt wurde. Im Jahr 2014 war es endlich soweit und HTML5 erschien. Dabei handelt es sich um einen Zusammenschluss von HTML 4 und XHTML. Darüber hinaus wurde der Funktionsumfang deutlich vergrößert.

Neben einer erheblich verbesserten Unterstützung von Multimedia-Inhalten zeichnet sich HTML5 auch dadurch aus, dass diese Sprache eine semantische Auszeichnung der Seitenelemente erlaubt. Das bedeutet, dass das Menü, die Kopfleiste oder der Fuß der Seite nun nicht mehr nur durch ihre Positionierung erkenntlich sind. Darüber hinaus können diese Elemente direkt ausgezeichnet werden.

Hierbei handelt es sich nur um einen kurzen Überblick über die wichtigsten Neuerungen, die HTML5 mit sich brachte. Darüber hinaus bringt diese neue Version noch viele weitere Möglichkeiten mit sich. Außerdem vereinfacht sie den Code in vielen Bereichen. HTML5 ist die Antwort des W3C auf die technische Entwicklung, die das Internet erlebt hat. Auch dieses Buch verwendet HTML5. Auf diese Weise erlaubt es den Lesern, moderne Internetseiten mit vielfältigen Funktionen zu erstellen.

Kapitel 3

CSS: Unzählige Möglichkeiten für die Gestaltung einer Internetseite

CSS (Cascading Style Sheets) ist neben HTML der zweite wesentliche Bestandteil dieses Buchs. Hierbei handelt es sich um eine Stylesheet-Sprache. CSS ist demnach ebenfalls keine Programmiersprache. Vielmehr handelt es sich dabei um eine Layout-Vorgabe, die die Gestaltungsweise der einzelnen Elemente vorgibt. CSS ist jedoch keine eigenständige Technik. Diese Vorgaben werden stets auf ein HTML-Dokument angewendet. Das bedeutet, dass man zunächst mit HTML die Inhalte strukturieren muss. Danach kann man per CSS die Größe, die Position, die Farbe und viele weitere Details bestimmen.

3.1. Das Layout der Internetseiten wurde immer wichtiger

Als das Internet entstand, diente dieses Medium der Verbreitung wissenschaftlicher Texte. Die Vermittlung der Inhalte stand hierbei im Vordergrund. Dabei ist lediglich eine übersichtliche Gestaltungsweise wichtig, um die benötigten Informationen möglichst schnell zu finden. Das Layout spielt dabei hingegen eine untergeordnete Rolle. Bunte Farben oder ansprechende Fotos waren nicht erforderlich. Die verwendeten Grafiken dienten ebenfalls der Wissensübermittlung – beispielsweise indem ein Schaubild in den Text eingefügt wird. Die ersten HTML-Versionen waren ausschließlich für diesen Verwendungszweck bestimmt und unterstützten daher kaum Möglichkeiten, um das Layout zu beeinflussen.

Bereits kurze Zeit später wurde jedoch klar, dass das Internet noch viel mehr Potenzial bot. Auf diese Weise lassen sich nicht nur wissenschaftliche Texte in Sekundenschnelle in aller Welt abrufen. Das Gleiche gilt auch für Nachrichten, Warenbestellungen und für viele weitere Informationen. Dabei handelt es sich jedoch um kommerzielle Angebote.

Diese unterscheiden sich in vielen Aspekten von den wissenschaftlichen Texten, für die HTML ursprünglich geschaffen wurde. Beispielsweise ist hierbei nicht nur der Inhalt von Bedeutung. Darüber hinaus spielt das Layout eine wichtige Rolle. Der Besucher ist hier immer auch Kunde. Wenn er sich aufgrund einer ansprechend gestalteten Seite wohlfühlt, trägt das erheblich zum Erfolg des Angebots bei. Das führte dazu, dass das Layout eine immer größere Bedeutung gewann.

Ein erster Ansatz, um die Darstellung der Seiten zu verbessern, bestand darin, bestehende HTML-Tags dafür zu verwenden. Das wichtigste Beispiel hierfür sind Tabellen. Diese hatten eigentlich die Aufgabe, Informationen strukturiert auf der Seite auszugeben. Allerdings boten sie auch die Möglichkeit, der Seite eine Struktur zu geben. Es war möglich, für die einzelnen Zellen eine spezifische Größe vorzugeben. Darüber hinaus erlaubt es HTML, Zellen miteinander zu verbinden. Auf diese Weise lassen sich verschiedene Blöcke mit Texten oder Bildern gestalten und genau an der gewünschten Stelle auf der Seite anbringen. Damit erreicht man eine ansprechende Strukturierung und ein deutlich verbessertes Erscheinungsbild. Allerdings wird die Tabelle dabei zweckentfremdet, da sie nicht mehr für ihre eigentliche Aufgabe zum Einsatz kommt.

Darüber hinaus entstanden neue Tags, mit deren Hilfe sich das Layout beeinflussen ließ. Für die Vorgabe der Seitenstrukturen wurden beispielsweise Frames eingeführt. Ein weiteres wichtiges Beispiel ist das font-Tag. Mit dessen Hilfe konnte man unter anderem die Schriftart, die Schriftgröße und die Farbe eines Texts verändern. Das bot bereits zahlreiche Möglichkeiten für eine individuelle und ansprechende Gestaltung. Eine große Bedeutung erlangte auch das Attribut bgcolor. Dieses konnte man auf viele verschiedene HTML-Elemente anwenden, um den entsprechenden Bereichen eine ansprechende Hintergrundfarbe zu verleihen. Darüber hinaus entstanden einige weitere Tags und Attribute, mit denen man das Layout festlegen konnte.

Diese Vorgehensweise stieß allerdings auf erheblichen Widerstand. Zahlreiche Entwickler forderten, die Layout-Vorgaben nicht innerhalb von HTML umzusetzen, sondern eine separate Stylesheet-Spra-

che hierfür zu verwenden. Auf diese Weise entstand CSS. Diese Technik stellt eine Alternative zu den eben vorgestellten HTML-Attributen dar. Seit der Vorstellung der ersten offiziellen Version im Jahre 1996 setzten immer mehr Entwickler diese Technik um. Sie bot nicht nur deutlich mehr Möglichkeiten als die bisherigen HTML-Tags. Darüber hinaus erlaubt CSS eine deutlich effizientere Umsetzung. Das führte dazu, dass nur noch wenige Webentwickler das Layout per HTML festlegten. Ab HTML 4 wird die Verwendung von Tags und Attributen, die das Layout vorgeben, nicht mehr empfohlen. HTML5 unterstützt sie überhaupt nicht mehr. Daher verwenden alle modernen Internetangebote ausschließlich CSS für die Gestaltung des Layouts.

3.2. Einführung von CSS: Trennung von Inhalt und Design

Einer der wesentlichen Gründe, der für die Einführung von CSS und gegen die Umsetzung der Layoutvorgaben innerhalb von HTML sprach, war die Trennung zwischen Inhalt und Design. Diese erleichtert die Arbeitsabläufe bei der Gestaltung und der Pflege einer Internetseite erheblich.

Ein wichtiger Grund für die Trennung der beiden Bereiche besteht darin, dass für die Inhalte und für das Layout häufig unterschiedliche Personen verantwortlich sind – zumindest bei größeren Projekten. Bei einem Nachrichtenportal erstellen die Inhalte beispielsweise Redakteure und Journalisten. Um der Seite ein ansprechendes Layout zu geben, werden hingegen meistens spezialisierte Designer eingestellt. Es kommt hinzu, dass der HTML-Code sehr häufig über serverseitige Programme erzeugt wird. Für deren Erstellung sind Programmierer zuständig, die sich mit den Vorgaben für das Design nur ungern auseinandersetzen. Im Gegenzug verfügen die meisten Designer nur über geringe Programmierkenntnisse. Die konsequente Trennung der verschiedenen Bereiche vereinfacht die Zusammenarbeit erheblich. Auf diese Weise kann sich jede Fachkraft auf seinen eigenen Bereich konzentrieren, ohne dass das die Arbeit der Kollegen beeinträchtigt.

Ein weiterer Grund, der für die Verwendung von CSS spricht, ist die einfachere Pflege der Internetseiten. In der Regel erfährt das Layout über einen längeren Zeitraum hinweg keine Änderungen, wenn es einmal erstellt ist. Die Inhalte werden hingegen regelmäßig aktualisiert. Diese Aufgabe wird deutlich einfacher, wenn man sie getrennt von den Layout-Vorgaben erledigen kann.

Hin und wieder kommt es jedoch auch zu einer Überarbeitung des Designs der Seite. Bei dieser Aufgabe wirkt sich die Trennung zwischen den Inhalten und dem Design besonders positiv aus – insbesondere wenn man die CSS-Vorgaben in einem separaten Dokument abgespeichert hat. Wäre das Layout innerhalb der HTML-Tags definiert, müsste man jede einzelne Seite überarbeiten, um das Design anzupassen. Wenn man ein separates CSS-Dokument erstellt hat, muss man die Änderungen jedoch nur ein einziges Mal umsetzen. Sie werden dann automatisch auf jede einzelne Seite übertragen, die das entsprechende Stylesheet verwendet. Das führt zu einer erheblichen Arbeitersparnis.

Aufgrund dieser Vorteile setzte sich CSS sehr schnell durch. Mittlerweile zählt diese Technik zu den Kernsprachen des Internets und hat daher eine sehr große Bedeutung. Die Festlegung des Layouts per HTML ist nicht nur kompliziert und arbeitsintensiv. Seit der Einführung von HTML5 ist diese Vorgehensweise überhaupt nicht mehr möglich. Daher ist es zwingend notwendig, CSS zu verwenden, wenn man ein ansprechendes Design für eine Seite erstellen will.

3.3. Das Zusammenspiel zwischen HTML und CSS

In einem der vorherigen Abschnitte wurde bereits erwähnt, dass CSS keine eigenständige Sprache darstellt. Wenn man ein CSS-Stylesheet erstellt, ist es nicht möglich, dieses für sich alleine auszuführen. Die Angaben ergeben nur dann Sinn, wenn sie auf ein HTML-Dokument angewendet werden. Das bedeutet, dass es bei der Darstellung der Seite immer zu einer Interaktion zwischen den beiden Sprachen kommt. Aus diesem Grund ist es sinnvoll, das Zusammenspiel zwischen HTML und CSS etwas genauer zu untersuchen.

HTML zeichnet verschiedene Elemente der Seite aus. Wie bereits erwähnt, kann es sich dabei beispielsweise um Überschriften, um Links oder um Absätze handeln. Darüber hinaus gibt es jedoch noch viele weitere Möglichkeiten, die wir in diesem Buch ebenfalls noch kennenlernen werden. Wenn man diese Elemente ohne CSS-Vorgaben in eine Internetseite einfügt, werden sie dem Standard-Layout des Browsers entsprechend dargestellt. Dafür gibt es keine allgemeingültigen Vorgaben. Die Umsetzung hängt vom gewählten Browser ab. Allerdings sind die Unterschiede zwischen den verschiedenen Angeboten nur äußerst gering. Es ist beispielsweise üblich, dass die Überschriften größer als gewöhnlicher Text dargestellt werden. Links erscheinen bei fast allen Browsern in blauer Farbe.

Diese Standardeinstellungen sorgen jedoch in der Regel für ein Design, das recht langweilig wirkt. Die einzelnen Elemente werden dabei einfach untereinander aufgeführt. Der Hintergrund ist weiß und der Text schwarz. Wenn man eine Seite auf diese Weise gestaltet, werden nur wenige Besucher dort verweilen. Deshalb ist es sinnvoll, per CSS eigene Design-Vorgaben zu gestalten.

Wenn man eine Änderung per CSS vornimmt, muss sich diese immer auf ein HTML-Element beziehen. Ohne diesen Bezug hat die Vorgabe keinerlei Effekt. Um ihn herzustellen, kommen mehrere Möglichkeiten infrage. Sehr einfach – und daher bei Anfängern recht beliebt – ist es, die Style-Vorgaben direkt in das Tag zu schreiben. Auf diese Weise wird sofort klar, auf welches Element sie sich beziehen. Wenn man diese Methode wählt, geht jedoch der wesentliche Vorteil von CSS – die Trennung von Inhalt und Layout – verloren. Daher ist es besser, die CSS-Vorgaben in einem separaten Dokument oder zumindest im Kopfbereich der HTML-Seite anzubringen. Das macht es jedoch notwendig, genau festzulegen, auf welches HTML-Element sie sich beziehen.

Hierfür ist es möglich, Vorgaben für alle HTML-Elemente der Seite zu machen. Wenn man beispielsweise für alle Bereiche eine grüne Schrift verwenden will, dann kann man die entsprechende Anweisung so gestalten, dass sie für alle Elemente gültig ist. In vielen Fällen ist es jedoch erwünscht, dass das entsprechende Layout nur auf ganz bestimmte

Elemente angewandt wird. Es ist beispielsweise möglich, Vorgaben zu machen, die für alle Absätze oder für alle Überschriften gültig sind. Darüber hinaus kann man individuelle Klassen definieren. Wenn man eine Style-Vorgabe für eine Klasse macht, wird diese auf alle entsprechend ausgezeichneten Elemente angewandt – unabhängig davon, um welchen Typ es sich dabei handelt. Schließlich ist es möglich, einem Element eine spezifische ID zu verleihen. Diese Bezeichnung darf nur ein einziges Mal auf der Seite verwendet werden. Das macht es möglich, Vorgaben für ein ganz bestimmtes Element zu machen. Die vielfältigen Möglichkeiten, um die verschiedenen HTML-Elemente anzusprechen, erlauben eine effiziente und präzise Steuerung der Layout-Vorgaben.

Kapitel 4

Die Vorbereitungsmaßnahmen für die Gestaltung von Internetseiten mit HTML und CSS

Nachdem die vorherigen Kapitel einen Überblick über die wichtigsten Grundlagen von HTML und CSS gegeben haben, ist es nun an der Zeit, selbst die ersten Seiten zu gestalten. Schließlich soll dieses Buch nicht nur die entsprechenden Sprachen beschreiben, sondern dem Leser deren Verwendung beibringen. Daher stellen praktische Übungen und die Erstellung eigener Internetseiten einen wesentlichen Teil des Inhalts dar.

Wenn man eine eigene Webseite erstellen will, benötigt man dafür jedoch das richtige Handwerkszeug. Ohne dieses ist es nicht möglich, den Code zu erstellen und auszuführen. Von großer Bedeutung ist es beispielsweise, einen passenden Webbrowser auf dem Computer zu installieren – am besten in unterschiedlichen Ausführungen. Darüber hinaus ist ein Texteditor notwendig, um den Quellcode zu erzeugen. Hilfreich kann auch die Verwendung eines sogenannten WYSIWYG-Editors sein. Dieser verfügt über eine grafische Benutzeroberfläche, über die man verschiedene HTML-Elemente in die Seite einfügen kann, ohne den Code dafür selbst zu erstellen. Dieses Kapitel stellt die entsprechenden Werkzeuge vor und zeigt, wie man diese installiert.

4.1. Verschiedene Webbrowser für die Darstellung der Seiten

Wenn man eine Internetseite besuchen will, ist es notwendig, dafür einen Browser aufzurufen. Diese Software dient dazu, die HTML-Seiten aus dem Internet abzurufen. Allerdings zeigt sie nicht deren Quellcode an. Sie interpretiert diesen und sorgt auf diese Weise für eine ansprechende Darstellung.

Wenn wir unsere eigenen Internetseiten mit HTML und CSS gestalten, dann möchten wir selbstverständlich wissen, wie diese später auf den Besucher wirken. Nur so erkennen wir, ob wir die Vorstellungen, die wir für unser Layout entwickelt haben, richtig umgesetzt haben. Hierfür rufen wir sie ebenfalls in einem gewöhnlichen Webbrowser auf. Dieser stellt unsere Seiten auf genau die gleiche Weise dar, wie die Inhalte, die wir aus dem Internet abrufen.

Da ein Browser zur Standard-Ausstattung eines Computers gehört, ist es in der Regel nicht notwendig, diesen neu zu installieren. Lediglich Leser, die noch über keinen Browser verfügen, sollten dies nun nachholen. Hierfür gibt es mehrere kostenfreie Angebote im Internet. Sehr beliebt sind beispielsweise der Browser Chrome von Google und Firefox von Mozilla. Diese stehen unter folgenden Links zum Download bereit:

<https://www.google.com/intl/de/chrome/>

<https://www.mozilla.org/de/firefox/new/>

Dabei handelt es sich um die am häufigsten verwendeten Browser weltweit. Es steht dem Leser jedoch auch frei, jeden beliebigen anderen Browser zu verwenden.

Die Darstellung einer HTML-Seite ist in den verschiedenen Browsern beinahe identisch. Allerdings kann es dabei zu einigen feinen Unterschieden kommen. In den meisten Fällen sind diese so gering, dass sie kaum wahrnehmbar sind. Allerdings kommt es auch gelegentlich vor, dass ein kleiner Unterschied große Auswirkungen hat. Insbesondere der Browser Internet Explorer von Microsoft, der mittlerweile jedoch veraltet ist, zeichnete sich durch einige Besonderheiten gegenüber den übrigen Angeboten aus. Beispielsweise wurden hierbei die Abstände zwischen den verschiedenen Elementen anders berechnet. Die Unterschiede waren so gering, dass sie in der Regel kaum auffielen. Manchmal kam es jedoch vor, dass aufgrund dieses Unterschieds die Elemente nicht mehr in den dafür vorgesehenen Bereich passten. Das konnte das komplette Layout zerstören.

Wenn man eine Internetseite veröffentlicht, weiß man nicht, welchen Browser die Besucher verwenden. Das Ziel besteht jedoch darin, dass die Seite für alle Anwender ansprechend angezeigt wird. Aus diesem Grund ist es wichtig, die Darstellung mit verschiedenen Browsern auszuprobieren. Wenn man dabei feststellt, dass auf einem Browser die Darstellung nicht den Wünschen entspricht, kann man den Code vor der Veröffentlichung noch abändern. Falls man plant, die Seiten zu veröffentlichen, ist es daher sinnvoll, alle gängigen Browser zu installieren, um sie einem umfassenden Test zu unterziehen.

Die Screenshots in diesem Buch zeigen in der Regel die Darstellung mit dem Browser Chrome. Wer Wert darauf legt, dass die eigene Anzeige exakt mit diesen Bildern übereinstimmt, sollte daher ebenfalls Chrome installieren. Da die Unterschiede jedoch nur minimal sind, kann man aber auch problemlos einen anderen Browser verwenden.

4.2. Texteditor für die Gestaltung des Codes

Wenn man den Code für eine Webseite erstellt, dann besteht dieser aus reinem Text. Um diesen Text zu erstellen, ist jedoch ein geeignetes Programm notwendig. Auf den meisten Computern ist bereits ein Textverarbeitungsprogramm installiert – beispielsweise Word oder der Open Office Writer. Nun könnte man auf die Idee kommen, diese Programme für die Erstellung des Quellcodes zu verwenden. Das ist jedoch nicht möglich. Der Quellcode darf nur reinen Text enthalten. Die genannten Programme fügen jedoch noch vielfältige weitere Informationen hinzu – von der Schriftart über die Farben bis hin zu den Zeilenabständen. Der Browser kann Dateien mit diesen Zusatzinformationen nicht lesen.

Aus diesem Grund ist es notwendig, einen Texteditor zu verwenden. Dieser speichert das Dokument als reinen Text ab. Ein Beispiel hierfür ist der Microsoft-Editor, der auch unter dem Namen Notepad bekannt ist. Dieser ist auf allen Windows-Systemen standardmäßig installiert und ermöglicht es, den Quellcode im entsprechenden Format zu erstellen. Allerdings bietet er nur einen minimalen Funktionsumfang.

Es gibt zahlreiche weitere Texteditoren, die noch viele zusätzliche Funktionen anbieten. Beispielsweise rücken diese ineinander geschachtelte Tags automatisch ein. Das sorgt für eine übersichtlichere Darstellung. Außerdem stellen Sie Tags, Attribute und Inhalte in verschiedenen Farben dar. Auch das macht es einfacher, den Quellcode zu überblicken. Schließlich bieten viele Texteditoren eine automatische Vervollständigung. Wenn man ein Tag erstellt, fügt die Software automatisch das zugehörige Schluss-Tag hinzu. Das kann viel Zeit beim Erstellen der Seiten sparen. Hinzu kommen viele weitere Funktionen, die eine einfache und effiziente Codierung unterstützen.

```
1 <html>
2   <head>
3     <title>Seitentitel</title>
4   </head>
5   <body>
6     <h1>Überschrift</h1>
7     <p>Absatz</p>
8     <input type = "button" value = "Hier klicken">
9   </body>
10 </html>
```

Screenshot 1: Die Darstellung des Codes in einem Texteditor

Aus den genannten Gründen ist es empfehlenswert, einen etwas höherwertigen Texteditor herunterzuladen und auf dem Computer zu installieren. Hierfür gibt es unzählige Gratis-Angebote. Im folgenden werden einige beliebte Beispiele für die Erstellung von HTML-Seiten vorgestellt. Bei der Auswahl wurde besonderer Wert darauf gelegt, dass die Software kostenlos erhältlich und einfach zu verwenden ist. Der Leser kann dabei frei entscheiden, welche Software er installiert. Selbstverständlich ist es auch möglich, einen Texteditor zu verwenden, der nicht in dieser Liste auftaucht.

- ▶ Visual Studio Code:

<https://code.visualstudio.com/download>

- ▶ Bluefish:

<http://bluefish.openoffice.nl/index.html>

- ▶ Sublime Text:

<http://www.sublimetext.com/>

- ▶ phase 5:

<https://www.phase5.info/>

- ▶ Geany:

<https://www.geany.org/download/releases>

4.3. WYSIWYG-Editoren als Alternative

Für die Beispiele und Aufgaben in diesem Buch sind die im vorherigen Abschnitt vorgestellten Text-Editoren vollkommen ausreichend. Allerdings gibt es noch eine weitere Alternative, die der Vollständigkeit halber ebenfalls vorgestellt werden soll: WYSIWYG-Editoren.

Diese Abkürzung steht für “What you see is what you get”. Dieser Ausdruck beschreibt bereits anschaulich, was diese Editoren auszeichnet. Sie verfügen über eine grafische Benutzeroberfläche, auf der der Anwender verschiedene Elemente per Mausklick einfügen kann. Die Darstellung sieht beinahe genau gleich aus, wie wenn man die entsprechende Seite in einem Browser aufruft. Der Editor erzeugt den Code, der dieser grafischen Darstellung entspricht, automatisch.

WYSIWYG-Editoren sind zum einen für Personen ohne HTML-Kenntnisse gedacht, die dennoch ihre eigenen Internetseiten gestalten möchten. Anhand der grafischen Benutzeroberfläche ist es ohne jegliche Vorkenntnisse möglich, eine Seite zu erstellen. Für die Leser dieses Buchs ist diese Form der Anwendung jedoch nicht von Interesse – schließlich soll hier ja gerade die Erstellung des HTML-Codes vermittelt werden. Wenn man nun auf die Idee kommt, dass man aufgrund dieser Technik darauf verzichten könnte, HTML zu lernen, ist man jedoch nicht gut beraten. Sollte man ausschließlich die grafische Benutzeroberfläche dieser Editoren verwenden, ohne selbst Hand an den Code anzulegen,

entstehen in der Regel keine Seiten, die einen professionellen Eindruck erwecken. Daher eignet sich diese Vorgehensweise nur für private Projekte.

Dennoch gibt es auch viele professionelle Webdesigner, die mit WYSIWYG-Editoren arbeiten. Der Grund dafür liegt zum einen darin, dass diese eine effiziente Arbeitsweise unterstützen. Ein HTML-Element mit der Maus auf die Oberfläche zu ziehen, benötigt in der Regel weniger Zeit als die entsprechenden Tags in den Quellcode zu schreiben. Auf diese Weise lässt sich die Grundstruktur schnell und einfach erstellen. Diese Editoren verfügen jedoch auch über eine Code-Ansicht. Daher kann man den Quellcode ebenfalls bearbeiten. Das macht es möglich, die grundlegenden Elemente zunächst über die grafische Oberfläche einzufügen und die Feinheiten anschließend im Quellcode zu bearbeiten. Das bedeutet jedoch, dass für die Erstellung der Seiten ebenfalls gute HTML-Kenntnisse notwendig sind.

Ein weiterer Vorteil der WYSIWYG-Editoren besteht darin, dass man die Darstellung der Seite sehr schnell überprüfen kann. Wenn man den Code verfasst, ist es wichtig, die Seite zu öffnen, um herauszufinden, ob man dabei die gewünschten Ergebnisse erzielt hat. Wenn man sie jedes Mal in einem Webbrowser laden muss, benötigt das viel Zeit. WYSIWYG-Editoren ermöglichen jedoch in der Regel eine zweigeteilte Ansicht. In der einen Hälfte erscheint der Quellcode, sodass man diesen bearbeiten kann. In der anderen Hälfte kann man die Ergebnisse direkt in Echtzeit überprüfen.

Wenn man sich nun die Frage stellt, welche Software man zu diesem Zweck verwenden kann, dann ist an erster Stelle sicherlich Dreamweaver von Adobe zu nennen (<https://www.adobe.com/de/products/dreamweaver.html>). Diese Software bietet ausgesprochen viele Funktionen und ermöglicht eine sehr effiziente Gestaltung der Webseiten. Unzählige Webdesigner in aller Welt verwenden dieses Programm. Allerdings ist dieses nicht kostenfrei. Insbesondere wenn man ausschließlich private Projekte umsetzen will, sollte man sich genau überlegen, ob sich die hohen Ausgaben lohnen.

Darüber hinaus gibt es jedoch auch viele kostenfreie WYSIWYG-Editoren. Diese bieten in der Regel zwar nicht ganz so viele Funktionen wie Dreamweaver. Für Anfänger sind sie aber dennoch ausreichend. Unter den kostenfreien Editoren sind insbesondere die folgenden beiden Angebote sehr beliebt:

- ▶ BlueGriffon:

<http://bluegriffon.org/>

- ▶ Mobirise:

<https://mobirise.com/>

Für dieses Lehrbuch ist es jedoch nicht notwendig, einen WYSIWYG-Editor zu installieren. Wie bereits erwähnt reicht ein gewöhnlicher Texteditor vollkommen aus. Diese Möglichkeit soll lediglich als zusätzliche Option vorgestellt werden. Es bleibt dem Anwender überlassen, diese Programme herunterzuladen und auszuprobieren. Über die Code-Ansicht ist es dabei ebenfalls möglich, alle Aufgaben und Übungen in diesem Buch zu erledigen.

Kapitel 5

Der Aufbau einer HTML-Seite

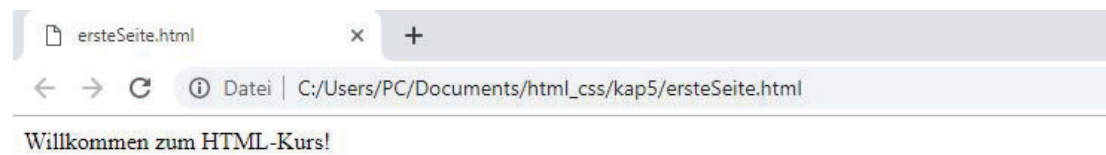
Wenn ein passender Texteditor installiert und ein Browser auf dem Rechner verfügbar ist, sind die Vorbereitungsmaßnahmen abgeschlossen und wir können damit beginnen, die ersten eigenen Internetseiten zu erstellen. Die ersten Beispiele stellen den grundlegenden Aufbau einer Seite vor. Die meisten Elemente, die dabei eingeführt werden, müssen auf jeder korrekt gestalteten HTML-Seite vorhanden sein.

5.1. HTML-Seiten: aufgeteilt in head und body

Um die erste Internetseite zu erstellen, ist es notwendig, den Texteditor zu öffnen, der im Rahmen der Vorbereitungsmaßnahmen installiert wurde. Hier muss man dann eine leere Seite aufrufen. In diese können wir dann einen beliebigen Text einfügen – beispielsweise “Willkommen zum HTML-Kurs!”. Nun müssen wir die Seite noch richtig abspeichern. Dazu können wir einen beliebigen Namen auswählen. Wichtig ist es lediglich, dass wir nach einem Punkt die Dateierweiterung `html` hinzufügen. Dieses Beispiel soll den Namen `ersteSeite.html` erhalten.

Alternativ dazu ist es möglich, die Dateierweiterung `htm` zu verwenden. Diese wurde eingeführt, da manche Betriebssysteme keine Dateierweiterungen mit mehr als drei Zeichen unterstützten. Diese Alternative wird zwar nach wie vor von allen gängigen Systemen unterstützt, allerdings ist sie nicht mehr gebräuchlich. Alle modernen Betriebssysteme erlauben die Verwendung von vier Buchstaben als Dateierweiterung. Daher sollten die Dateien in der Regel mit der Erweiterung `html` abgespeichert werden.

Nun können wir den Ordner öffnen, in dem wir die Datei abgespeichert haben. Wenn wir sie doppelt anklicken, öffnet sich automatisch der Standard-Browser und gibt die entsprechende Seite aus:



Screenshot 2: Die Ausgabe der ersten Seite.

An diesem Beispiel sieht man, dass es ganz einfach ist, eine HTML-Seite zu erstellen. Obwohl wir hierfür kein einziges HTML-Tag verwendet haben, wird der eingegebene Text im Browser angezeigt. Allerdings handelt es sich hierbei um kein korrektes HTML-Dokument. Dieses widerspricht in vielen Bereichen den hierfür gültigen Regeln. Dass der Text trotzdem richtig angezeigt wird, liegt darin begründet, dass die meisten Browser sehr tolerant gegenüber ungültigem HTML-Code sind. Selbst wenn man dabei schwerwiegende Fehler macht, werden die Inhalte angezeigt. Das soll störende Fehlermeldungen beim Surfen im Internet vermeiden.

Dennoch besteht unser Ziel darin, korrekte HTML-Seiten zu erstellen. Aus diesem Grund ist es wichtig, sich zunächst mit dem grundlegenden Aufbau der Seite zu befassen. Eine HTML-Seite kann man sich wie Container-Boxen in unterschiedlichen Größen vorstellen, die man ineinander schachteln kann. Die größte Box, in der sich alle weiteren Elemente befinden, ist immer das `html`-Tag. Dieses zeigt an, dass der Inhalt der Seite in HTML verfasst ist. Alle übrigen Bestandteile der Seite stehen innerhalb der `html`-Tags. Dabei gibt es lediglich eine Ausnahme, die später in diesem Kapitel vorgestellt wird. Wenn wir die HTML-Tags einfügen, sieht der Code für die Seite so aus:

```
<html>
  Willkommen zum HTML-Kurs!
</html>
```

Die Struktur einer HTML-Seite sieht außerdem vor, dass diese in zwei Bereiche aufgeteilt sein muss. Zum einen gibt es den Kopfbereich,

der in `head`-Tags steht. Dieser Bereich ist für weiterführende Angaben vorgesehen, die nicht direkt auf der Seite angezeigt werden. Hier stehen beispielsweise der Seitentitel, eine Kurzzusammenfassung der Seite, Angaben zum verwendeten Zeichensatz und viele weitere wichtige Informationen. Da wir jedoch noch nicht wissen, wie wir die entsprechenden Informationen einfügen können, lassen wir den `head`-Bereich vorerst leer. Um die Strukturen der Seite deutlich zu machen, sollen die entsprechenden Tags trotzdem in den Code eingefügt werden.

Die Inhalte der Seite stehen hingegen innerhalb der `body`-Tags. Hier sind alle Elemente aufgeführt, die auf der Seite angezeigt werden sollen. Da der Text, den wir bereits in die Datei eingefügt haben, auf diese Weise ausgegeben werden soll, muss er innerhalb der `body`-Tags stehen. Der Code sieht dann wie folgt aus:

```
<html>
  <head>
  </head>
  <body>
    Willkommen zum HTML-Kurs!
  </body>

</html>
```

Sowohl beim `body`- als auch beim `head`-Tag ist es wichtig, zu beachten, dass diese genau ein Mal auf der Seite vorkommen müssen. Wenn man sie mehrere Male in den Code einfügt, führt das zu einem fehlerhaften HTML-Dokument.

Mit diesen Nachbesserungen ist unsere Seite zwar immer noch nicht vollkommen fehlerfrei, doch sind nun immerhin die grundlegenden Strukturen vorhanden. In den folgenden Abschnitten werden die weiteren Elemente vorgestellt, die jede HTML-Seite enthalten muss.

Die Darstellung ändert sich durch diese Maßnahmen übrigens nicht. Wenn man die Seite erneut aufruft, wird sie auf genau die gleiche Weise angezeigt, wie im letzten Screenshot.

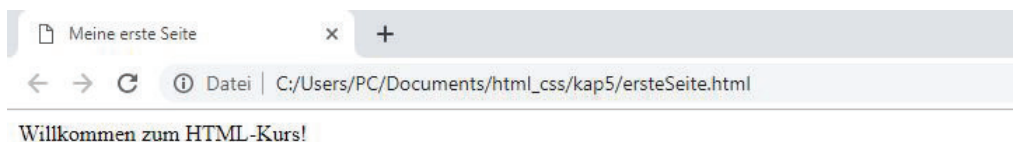
5.2. Wichtige Elemente innerhalb des head-Bereichs

Im bisherigen Beispiel haben wir den gesamten head-Bereich leer gelassen. Hier stehen in der Regel jedoch viele wichtige Angaben. Diese sind allerdings für den Besucher der Seite größtenteils nicht direkt zu erkennen. Das führt dazu, dass viele Anfänger ihnen nur wenig Bedeutung beimessen. Dennoch enthalten sie zahlreiche Informationen, die sehr wichtig für den Erfolg einer Internetseite sind.

Die wichtigste Angabe innerhalb des head-Bereichs ist der Titel der Seite. Dieser steht im `title`-Tag:

```
<html>
  <head>
    <title>
      Meine erste Seite
    </title>
  </head>

  <body>
    Willkommen zum HTML-Kurs!
  </body>
</html>
```



Screenshot 3: Die Seite mit der Anzeige des Titels

Hierbei handelt es sich um die einzige Angabe innerhalb des `head`-Tags, die der Besucher der Seite direkt erkennen kann. Der Text, den wir hier eingefügt haben, erscheint zwar nicht im Inhaltsbereich der Seite. Doch ist er ganz oben als Beschriftung der entsprechenden Registerkarte zu erkennen.

Wenn man die Seite später veröffentlicht und die Absicht hat, möglichst viele Besucher anzulocken, spielt der Titel ebenfalls eine wichtige Rolle.

Wenn man eine Anfrage bei einer Suchmaschine stellt, dann wird der Titel hier angezeigt. Falls dieser ansprechend und interessant erscheint, zieht das deutlich mehr Besucher an. Darüber hinaus ist es sinnvoll, an dieser Stelle wichtige Suchbegriffe einzufügen, damit die Seite bei einer entsprechenden Anfrage auch angezeigt wird.

Eine weitere wichtige Angabe ist der verwendete Zeichensatz. In der Informatik gibt es mehrere Zusammenstellungen von Schriftzeichen, die jeweils unterschiedliche Symbole enthalten. Damit der Browser den Inhalt richtig darstellen kann, ist es wichtig, dass er dafür den korrekten Zeichensatz verwendet. Normalerweise empfiehlt es sich, den Zeichensatz UTF-8 zu verwenden, der auch als Unicode bezeichnet wird. Dieser enthält alle Umlaute sowie das scharfe S. Auch die Buchstaben fast aller anderen Sprachen der Welt sind darin enthalten. In HTML5 muss folgender Befehl für die Bestimmung des Zeichensatzes eingefügt werden:

```
<meta charset="UTF-8">
```

Auf älteren Seiten findet man häufig auch folgende Angabe:

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

Diese kam in älteren HTML-Versionen zum Einsatz und erfüllte dort genau die gleiche Aufgabe. Mittlerweile ist sie jedoch überholt.

Darüber hinaus ist es möglich, eine kurze Beschreibung der Seite einzufügen. Dafür ist folgender Befehl notwendig:

```
<meta name="description" content="Die erste HTML-Seite in diesem Lehrbuch">
```

Außerdem kann man wichtige Schlüsselbegriffe eingeben:

```
<meta name="keywords" content="HTML, Lehrbuch, Internetseite">
```

Diese beiden Angaben spielen in erster Linie für die Suchmaschinenoptimierung eine wichtige Rolle. Sie führen dazu, dass die Inhalte der

Seite in den Suchmaschinen besser dargestellt werden und auf diese Weise mehr Besucher anlocken. Allerdings spielt das `keyword`-Tag mittlerweile keine Rolle mehr. Deshalb lassen es viele Webentwickler heutzutage weg.

Schließlich ist es möglich, den Autor der Seite zu benennen. Doch ist dies ebenfalls nicht unbedingt notwendig und hat keinen großen Einfluss auf das Projekt. Dennoch soll der entsprechende Befehl der Vollständigkeit halber vorgestellt werden:

```
<meta name="author" content="Max Mustermann">
```

5

Mit Ausnahme des Titels begannen alle Tags, die in diesem Kapitel vorgestellt wurden, mit dem Ausdruck `meta`. Das bedeutet, dass es sich hierbei um weiterführende Informationen handeln, die nicht direkt mit den Inhalten in Verbindung stehen. Aufgrund dieser Gemeinsamkeit werden sie auch als `meta`-Tags bezeichnet.

Bei den hier vorgestellten Beispielen handelt es sich um die wichtigsten `meta`-Tags. Darüber hinaus gibt es jedoch auch einige weitere Möglichkeiten, die man bei Bedarf ebenfalls in die Seite einfügen kann. Da dies jedoch in der Regel nicht notwendig ist, werden diese hier nicht alle erläutert.

Wenn man dafür sorgen will, dass die Seite auch auf Smartphones und Tablets ansprechend dargestellt wird, ist es darüber hinaus wichtig, das `viewport-meta`-Tag einzufügen. Dieses wird jedoch erst im weiteren Verlauf dieses Buchs vorgestellt.

5.3. Die Doctype-Angabe

In Kapitel 5.1 wurde gesagt, dass alle Elemente der Seite innerhalb der `html`-Tags stehen und dass es dabei nur eine einzige Ausnahme gibt. Dabei handelt es sich um die Doctype-Angabe. Diese steht immer ganz zu Beginn des Dokuments – also noch vor dem öffnenden `html`-Tag. Daher handelt es sich hierbei genau genommen nicht um ein Element

dieser Auszeichnungssprache. Die Doctype-Angabe stellt eine Deklaration dar, die dem Browser anzeigt, welche Sprache für die Gestaltung der Seite verwendet wird.

In HTML5 ist die Doctype-Angabe recht einfach aufgebaut:

```
<!DOCTYPE html>
```

Diese Angabe sollte am Anfang aller Internetseiten stehen, die wir verfassen. Bei älteren Seiten, die etwa in HTML 4.01 oder in XHTML verfasst sind, finden sich jedoch auch andere Doctype-Angaben. Diese sind nicht nur komplizierter aufgebaut. Darüber hinaus gibt es dabei auch viele verschiedene Möglichkeiten – je nachdem, welche genaue HTML-Spezifikation man verwendet. Eine mögliche Doctype-Angabe sieht beispielsweise so aus:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

Da moderne Seiten in HTML5 verfasst werden sollten, ist es jedoch nicht mehr sinnvoll, derartige Doctype-Angaben einzufügen. Daher ist es auch nicht notwendig, die genauen Details zu beschreiben. An dieser Stelle ist es lediglich wichtig, zu wissen, dass man bei älteren Seiten gelegentlich auf derartige Angaben stößt und dass sich diese auf ältere HTML-Versionen beziehen.

Wenn man nun auch die Doctype-Angabe in die Seite eingefügt hat, sieht der Code dafür so aus:

```
<!DOCTYPE html>
<html>
  <head>
    <title>
      Meine erste Seite
    </title>
    <meta charset="UTF-8">
    <meta name="description" content="Die erste HTML-Seite in diesem Lehrbuch">
    <meta name="keywords" content="HTML, Lehrbuch, Internetseite">
    <meta name="author" content="Max Mustermann">
  </head>
```

```

<body>
  Willkommen zum HTML-Kurs!
</body>
</html>

```

5.4. Die Sprache der Seite angeben

Darüber hinaus ist es in HTML möglich, die Sprache anzugeben, in der die Seite verfasst ist. Das ist für die automatisierte Verarbeitung der Inhalte und für die in vielen Browsern verfügbare Rechtschreibprüfung sinnvoll.

Um die Sprache vorzugeben, kommt jedoch kein eigenes Tag zum Einsatz. Anstatt dessen nutzt man hierfür ein Attribut. Attribute sind zusätzliche Angaben, die für den Inhalt des Tags gelten, in dem sie stehen. Daraus geht hervor, dass man ein Attribut immer zu einem Tag hinzufügen muss.

Nun stellt sich die Frage, in welches Tag man das Attribut für die Sprache einfügen soll. Die Sprachvorgabe soll immer für alle Inhalte gelten – sowohl für die Angaben im `head`- als auch im `body`-Bereich. Daher ist es erforderlich, die Angabe in das `html`-Tag zu setzen, das alle weiteren Elemente umfasst.

Das Sprach-Attribut wird mit dem Begriff `lang` eingefügt. Danach folgt – wie bei allen Attributen üblich – ein Gleichheitszeichen. Anschließend steht innerhalb von Anführungszeichen der Wert, den es annehmen soll. Für Seiten auf Deutsch kommt beispielsweise die Abkürzung `de` zum Einsatz. Das Kürzel `en` steht für englischsprachige Seiten, `fr` für Französisch und `es` für Spanisch. Darüber hinaus gibt es für jede gängige Sprache der Welt ebenfalls eine passende Bezeichnung. Wenn wir nun eine Seite in deutscher Sprache verfassen wollen, sieht das `html`-Tag dafür so aus:

```
<html lang = "de">
```

Damit ist der Code unserer ersten Seite vollständig. Nachdem alle notwendigen Angaben eingefügt wurden, sieht er so aus:

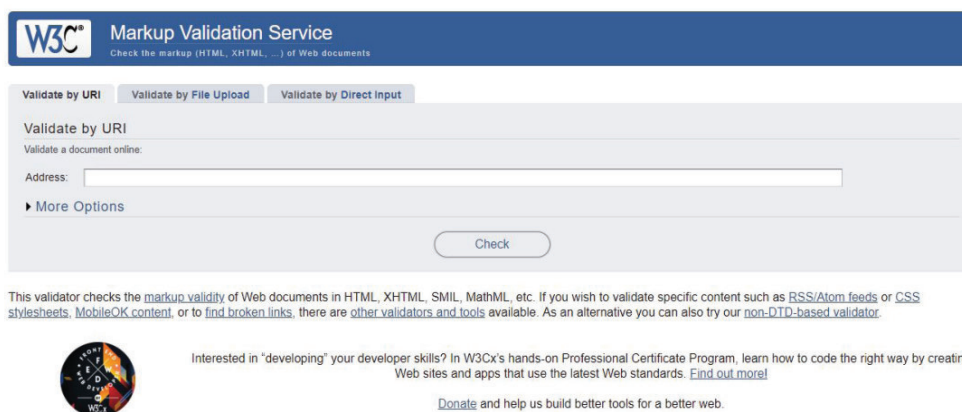
```
<!DOCTYPE html>
<html lang = "de">
  <head>
    <title>
      Meine erste Seite
    </title>
    <meta charset="UTF-8">
    <meta name="description" content="Die erste HTML-Seite in
    diesem Lehrbuch">
    <meta name="keywords" content="HTML,Lehrbuch,Internetseite">
    <meta name="author" content="Max Mustermann">
  </head>
  <body>
    Willkommen zum HTML-Kurs!
  </body>
</html>
```

5.5. Die Seite validieren

Wenn man eine HTML-Seite erstellt hat, ist es sinnvoll, zu überprüfen, ob sie allen formalen Anforderungen entspricht. Diese Überprüfung müssen wir jedoch nicht von Hand vornehmen. Hierfür gibt es sogenannte Validatoren. Von besonderer Bedeutung ist dabei der offizielle W3C-Validator. W3C ist die gleiche Organisation, die auch die offiziellen HTML-Standards verabschiedet. Daher stellt der Validator dieser Organisation die beste Möglichkeit dar, um die Seite auf Fehler zu überprüfen.

Dieser Validator ist unter folgender Adresse verfügbar:

<https://validator.w3.org/>



Screenshot 4: Der offizielle HTML-Validator

Hier kann man nun eine beliebige Internetadresse eingeben, um die Seite zu validieren. Da unsere Seite jedoch auf keinem Server verfügbar ist, scheidet diese Methode aus. Daher ist es notwendig, auf die Registerkarte mit der Aufschrift “Validate by File Upload” zu klicken. Nun kann man eine Datei vom eigenen Rechner hochladen. Daraufhin muss man nur noch die Datei auswählen, die wir gerade erstellt haben und auf den Button mit der Aufschrift “Check” klicken.

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for uploaded file ersteSeite.html

Checker Input

SHOW source outline image report

Check by file upload ▾ Keine ausgewählt

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Used the HTML parser.
Total execution time 1 milliseconds.

Screenshot 5: Das Ergebnis der Überprüfung

Die Überprüfung zeigt folgendes Ergebnis an: “Document checking completed. No errors or warnings to show.” Das bedeutet, dass unsere Seite korrekt erstellt wurde.

Um zu überprüfen, welche der Elemente, die wir eingefügt haben, für eine formal korrekte HTML-Seite notwendig sind, kann man diese einzeln entfernen und die Seite daraufhin erneut überprüfen. Dabei stellt man beispielsweise fest, dass das `title`- und das `head`-Tag genauso wie die `doctype`-Angabe und das `charset`-Tag unbedingt erforderlich sind. Wenn man hingegen die Keywords oder die Angabe des Autors entfernt, führt das zu keiner Fehlermeldung. Das bedeutet, dass diese Inhalte optional sind.

5.5. Übungsaufgabe: die Struktur einer HTML-Seite gestalten

Damit sich die beschriebenen Inhalte besser einprägen, steht am Ende der meisten Kapitel eine kleine Übungsaufgabe. Diese greift im Wesentlichen die Themen auf, die in den vorherigen Abschnitten erläutert wurden. Doch werden auch die Kenntnisse aus den vorherigen Kapiteln vorausgesetzt.

In der Regel sollte es möglich sein, die entsprechenden Aufgaben ohne weitere Hilfestellung zu erledigen. Sollten für die Lösung dennoch einmal Kenntnisse erforderlich sein, die noch nicht vermittelt wurden, werden diese kurz erklärt.

Am Ende wird immer eine Musterlösung angeboten. Es ist jedoch empfehlenswert, dass Sie zunächst versuchen, die Aufgaben selbstständig zu lösen. Die Musterlösung dient dem Abgleich der fertigen Ergebnisse. Sollten Sie jedoch alleine nicht weiterkommen, können Sie hier ebenfalls nachschauen. Dabei müssen Sie außerdem beachten, dass es in der Informatik meistens nicht eine einzige richtige Lösung gibt. Häufig führen mehrere Wege ans Ziel. Das bedeutet, dass Sie in erster Linie selbst überprüfen müssen, ob Ihre Lösung alle Anforderungen erfüllt. Diese kann selbst dann richtig sein, wenn sie von der angegebenen Musterlösung abweicht.

In diesem Kapitel sollen Sie folgende Aufgabe lösen:

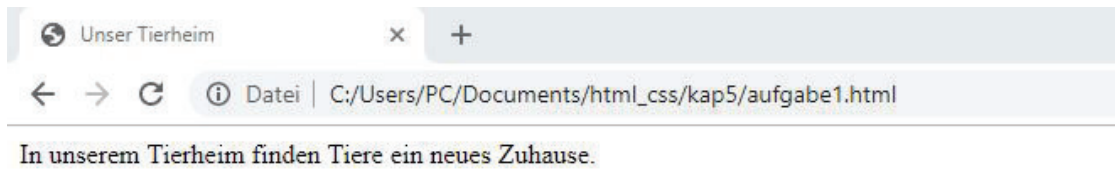
Erstellen Sie eine einfache Internetseite für ein Tierheim. Die Inhalte können Sie dabei frei wählen. Validieren Sie den Code, den Sie erstellt haben. Die Seite soll so gestaltet werden, dass weder Fehler- noch Warnmeldungen erscheinen. Optionale Elemente im `head`-Bereich sind nicht erforderlich.

Lösung:

```
<!DOCTYPE html>
<html lang = "de">
  <head>
    <title>
      Unser Tierheim
    </title>
    <meta charset="UTF-8">
  </head>

  <body>
    In unserem Tierheim finden Tiere ein neues Zuhause.
  </body>
</html>
```

5



Screenshot 6: So sieht die fertige Seite aus.

Alle Programmcodes aus diesem Buch stehen kostenfrei zum Download bereit. Dadurch müssen Sie Code nicht abtippen.



Außerdem erhalten Sie die eBook Ausgabe zum Buch im PDF Format kostenlos auf unserer Website:



www.bmu-verlag.de/html-css
Downloadcode: siehe Kapitel 23