

Bedeutung und Funktion von HTML

Wirkung von CSS

Ein erstes Dokument erstellen und formatieren

Kapitel 1

Ziemlich beste Freunde: HTML und CSS

Alle Webseiten, die Sie online sehen – egal ob die Suchseite von Google, die Seite einer Online-Zeitung, ein Online-Shop oder ein Social-Media-Dienst wie Facebook –, bestehen im Wesentlichen aus zwei Dingen:

- ✓ aus HTML-Code, der die Struktur vorgibt und
- ✓ aus CSS-Code, der für die Formatierung zuständig ist

Sie sehen nun, was es genau damit auf sich hat, und schreiben Ihr erstes formatiertes Dokument.

Webseiten: vor dem Tunen ein Blick unter die Haube

Den HTML-Code jeder beliebigen Webseite können Sie direkt betrachten: Surfen Sie mit Firefox oder Chrome zur gewählten Webseite (das hier gezeigte Grundprinzip funktioniert auch mit jedem anderen Browser, aber die Optionen können sich an einer bisschen anderen Stelle verstecken). Sie sehen die Webseite. Lassen Sie sich dann den Quellcode anzeigen. Dafür klicken Sie in Firefox oder Chrome mit der rechten Maustaste ins Dokument und wählen SEITENQUELLTEXT ANZEIGEN, siehe Abbildung 1.1. Unter macOS halten Sie neben der linken Maustaste auch die `⌘`-Taste gedrückt.

In Abbildung 1.2 sehen Sie den Seitenquelltext, das heißt den HTML-Code der Seite.

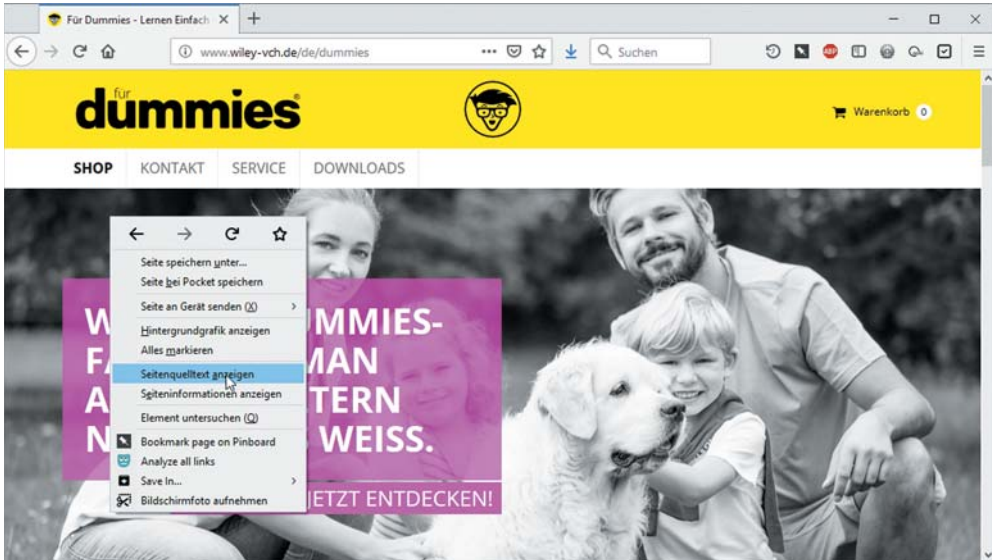


Abbildung 1.1: Quellcode aufrufen im Firefox

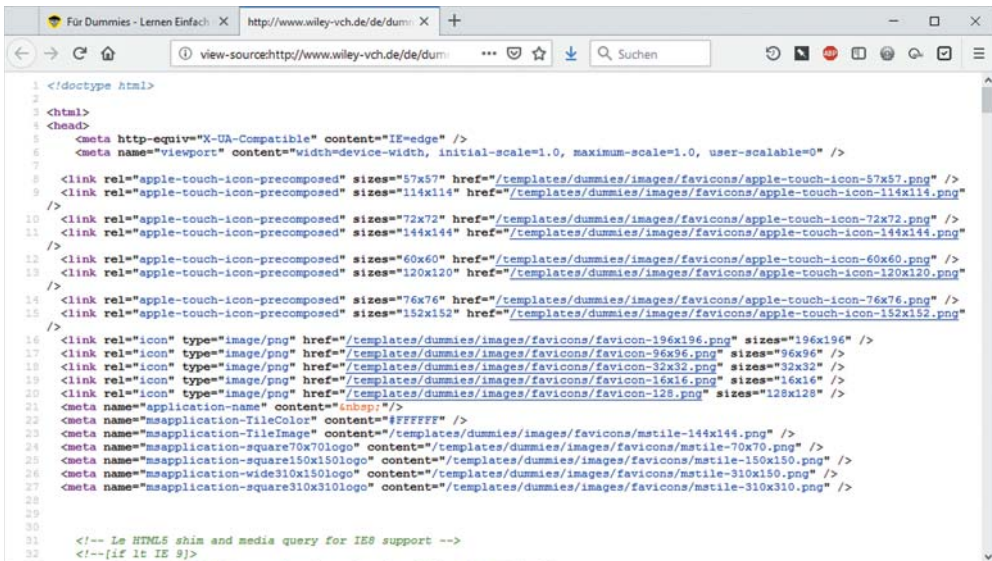


Abbildung 1.2: HTML-Quellcode

Kennzeichnend für den HTML-Code sind die spitzen Klammern, die um die einzelnen sogenannten *Elemente* stehen. Ganz oben erkennen Sie beispielsweise

```
<html>
```

Das bedeutet, dass jetzt HTML-Code beginnt.

Danach folgt <head>. Das gibt an, dass der head (Kopf) des Dokuments anfängt. Im Kopf des Dokuments stehen Informationen über das Dokument – Hinweise zum Zeichensatz, eine Beschreibung oder Verweise auf Icons, die bei bestimmten Systemen benutzt werden.

Aufgebrezelt mit CSS

Die Webseite, die Sie im Browser betrachten können, sieht aber nur dank einer weiteren Technik, nämlich CSS (Cascading Stylesheets) so aus, wie sie aussieht. CSS ist für die Optik zuständig. Darüber bestimmen Sie:

- ✓ welche Schriftart benutzt werden soll und ob der Text fett, kursiv, unterstrichen oder gesperrt dargestellt werden soll,
- ✓ welche Farben Bereiche und Schriften haben sollen.
- ✓ Außerdem regelt CSS auch das grundlegende Layout, sorgt dafür, dass manche Bereiche nebeneinander, andere wiederum untereinander platziert werden.
- ✓ Zudem können Sie mit CSS Hintergrundbilder definieren, Farbverläufe festlegen, für elegante Übergänge beim Klicken auf einen Button oder sogar für Animationen sorgen.
- ✓ Dank CSS lassen sich auch responsive Layouts erstellen, die sich an die unterschiedlichen Bildschirmgrößen von Smartphone über Tablet bis Desktop anpassen.

Die Wirkung von CSS erkennen Sie gut, wenn Sie CSS einmal testweise deaktivieren. Firefox bietet dafür eine komfortable Möglichkeit. Dafür müssen Sie zuerst die Menüleiste anzeigen lassen. Klicken Sie, wie in Abbildung 1.3 gezeigt, in Firefox in den Bereich oben neben der Adresse mit der rechten Maustaste und wählen Sie MENÜLEISTE.

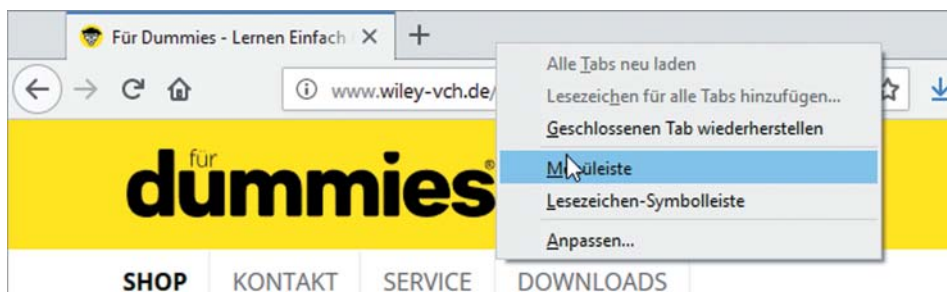


Abbildung 1.3: Menüleiste im Firefox anzeigen lassen

Dann können Sie bei einer beliebigen Webseite den CSS-Code ausschalten, indem Sie unter ANSICHT|WEBSEITEN-STIL|KEIN STIL wählen (Abbildung 1.4).

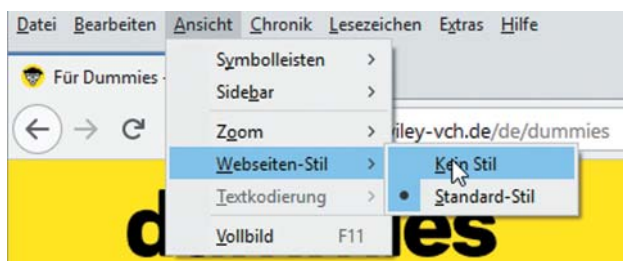


Abbildung 1.4: CSS deaktivieren

Nun präsentiert sich die Webseite sehr schlicht – Sie sehen zwar Bilder, Listen, Überschriften und Links, aber die Elemente sind nicht angeordnet, die Schrift ist die Standardschrift des Browsers, der Hintergrund ist weiß etc.

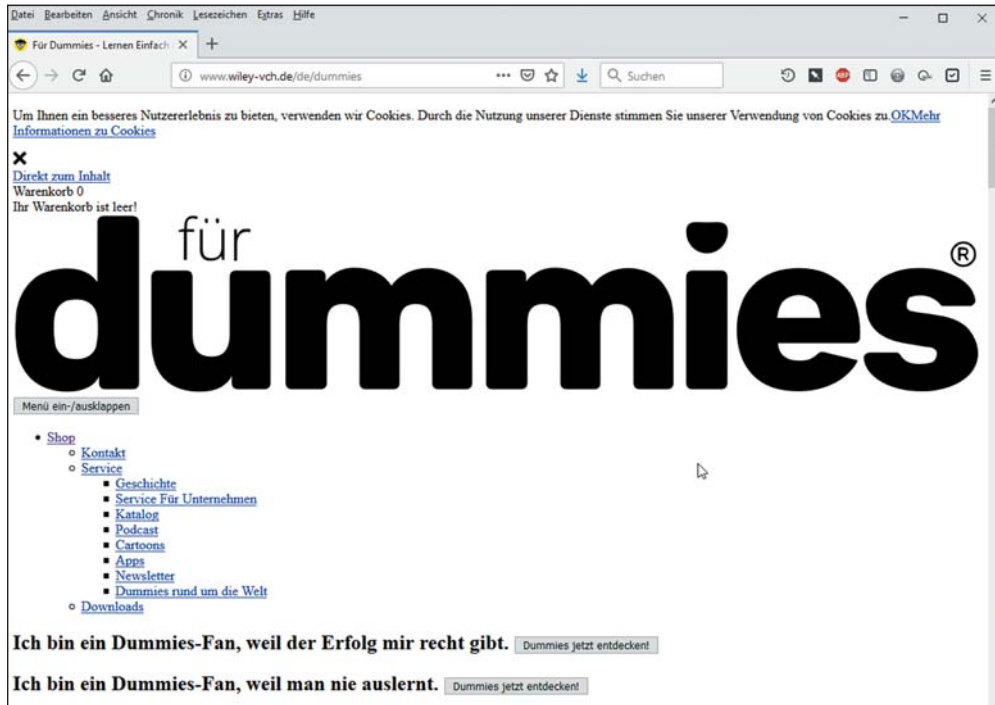


Abbildung 1.5: Webseite ohne CSS

Kurz gesagt ist CSS dafür zuständig, dass Seiten schön aussehen und dass es Spaß macht, die Texte zu lesen etc. Auch nur dank CSS sehen Seiten unterschiedlich aus.



Streng genommen ist die Seite auch ohne CSS schon etwas gestaltet – Aufzählungspunkte sind eingerückt und Links blau unterstrichen. Das liegt daran, dass alle Browser grundlegende CSS-Formatierungen anwenden.

Bevor Sie weiterlesen, sollten Sie unbedingt CSS wieder aktivieren, indem Sie ANSICHT|WEBSEITEN-STIL|STANDARD-STIL wählen. Falls Sie das vergessen, wundern Sie sich ansonsten sicher, dass Ihre selbst geschriebenen CSS-Formatierungen nicht funktionieren (ich spreche da aus leidvoller Erfahrung).

Ausgezeichnet mit HTML

HTML steht für *Hypertext Markup Language*. Markup Language bedeutet Auszeichnungssprache, Hypertext bezieht sich darauf, dass Webseiten über Links (in den Urzeiten des Web noch Hyperlinks genannt) verbunden sind.



HTML wird vom World Wide Web Consortium (W3C) und der WHATWG betreut. Das sind Institutionen mit vielen Arbeitsgruppen. In diesen Arbeitsgruppen sitzen Mitarbeiter großer Firmen und Browserhersteller und arbeiten an der Weiterentwicklung der Technologien. Für HTML ist also nicht eine Firma zuständig, sondern es ist ein offener Standard. Seit 2017 ist HTML 5.2 aktuell.

Das Grundprinzip von HTML besteht darin, dass Bereiche ausgezeichnet werden (englisch *markup* = auszeichnen). Durch HTML sagen Sie etwa »das, was jetzt folgt, ist ein X und hier ist dieses X wieder vorbei.« So können Sie beispielsweise

```
<p>Absätze auszeichnen</p>
```

- ✓ `<p>` im Beispiel heißt »Jetzt kommt ein Absatz«. `p` ist dabei die Abkürzung für *paragraph* – englisch Absatz.
- ✓ `</p>` bedeutet: Der Absatz ist jetzt vorbei.
- ✓ Der Text zwischen `<p>` und `</p>` wird dann vom Browser als Absatz interpretiert.

Man spricht dabei auch vom *Start-Tag* (*tag* [tæg] = Auszeichnung, Label) `<p>` und vom *End-Tag* `</p>`. Das End-Tag sieht genauso aus wie das Start-Tag, aber es besitzt einen zusätzlichen Slash (Schrägstrich) / nach der öffnenden spitzen Klammer.

Neben `p` stehen viele weitere Elemente zur Verfügung, beispielsweise `h1` für die Hauptüberschrift. `h1` steht dabei für *header 1*:

```
<h1>Eine wichtige Überschrift</h1>
```

Zudem gibt es z. B. die Möglichkeit, einzelne Wörter über `em` (`em` = *emphasized* = betont) hervorzuheben:

```
<p>Er stolperte über <em>einen schlechten Witz</em>. </p>
```

HTML definiert, welche Elemente wo wie stehen dürfen. HTML zu lernen, bedeutet die Elemente zu lernen samt ihrer Verschachtelungsregeln.

Browser: the two and only

Neben einem Texteditor (dazu gleich mehr) zur Erstellung der Webseiten brauchen Sie auch einen Browser, um die Webseiten zu betrachten. Hier können Sie fürs Erste den von Ihrem Betriebssystem bereits installierten Browser einsetzen. Danach sollten Sie sich aber unbedingt noch weitere Browser installieren:

- ✓ Firefox (<https://www.mozilla.org/de/firefox/>) steht für alle Betriebssysteme zur Verfügung. Er bietet viele nützliche Features für Webentwickler und wird in diesem Buch hauptsächlich verwendet. Entscheidend für die Darstellung von HTML/CSS ist die *Rendering Engine*, bei Firefox heißt diese Gecko bzw. seit neuestem Quantum. Es gibt weitere auf Gecko/Quantum basierende Browser wie beispielsweise SeaMonkey.
- ✓ Chrome (<https://www.google.com/intl/de/chrome/>) von Google ist derzeit der beliebteste Browser. Ihn sollten Sie auf jeden Fall ebenfalls installieren. Chrome besteht größtenteils aus Chromium (einem Open-Source-Browser, der aber nicht für Endnutzer

gedacht ist). Chromium basiert auf der Rendering Engine Blink. Weitere Browser, beispielsweise Opera, verwenden ebenfalls Chromium. Ab Version 19 wird auch Edge auf Chromium basieren.

- ✓ Edge: Dies ist der Standardbrowser unter Windows. Früher war es Internet Explorer. Er hat die Rendering Engine EdgeHTML. Im Dezember 2018 hat allerdings Microsoft angekündigt, ebenfalls die Rendering Engine Chromium zu nutzen.
- ✓ Safari ist ein Webbrowser von Apple, der inzwischen nur für macOS zur Verfügung steht. Er basiert auf KHTML (HTML-Rendering) und KJS (JavaScript-Implementation), die von Apple im gemeinsamen Projekt WebKit weiterentwickelt werden. Safari wird auch auf iOS benutzt.

Einzelne Browser wie die verschiedenen Browser, die Chromium nutzen, unterscheiden sich oft in den zusätzlichen für den Endnutzer zur Verfügung gestellten Funktionen. Wenn Sie Webseiten testen, ist für Sie hingegen nur die Rendering Engine relevant. Zwei Browser, die auf derselben Rendering Engine (inklusive Version) basieren, zeigen keinen prinzipiellen Unterschied in der Darstellung. Bei Details spielt auch das Betriebssystem eine Rolle.

Bei der Unterstützung für neue Features in HTML und CSS unterscheiden sich die Browser teilweise, d. h., dass eine bestimmte Funktionalität in einem Browser zur Verfügung steht, in einem anderen aber noch nicht. Wo das relevant ist, finden Sie jeweils Hinweise im Text. Die aktuellen Informationen dazu lesen Sie unter <https://caniuse.com/>.



Seit Microsoft entschieden hat, dass Edge als Basis keine eigene Rendering Engine mehr haben soll, sondern stattdessen Chromium genutzt wird, droht im Web eine Monokultur bezüglich der Rendering Engine. Auf den ersten Eindruck klingt das attraktiv – eine Rendering Engine heißt, dass es weniger Unterschiede zwischen den Browsern gibt. De facto sind allerdings solche Monokulturen sehr problematisch – und die letzte Browsermonokultur war keine einfache Zeit. Damit es nicht irgendwann nur noch eine Browserengine gibt, ist es wichtig, sich nicht nur auf den am häufigsten genutzten Browser – Chrome – zu konzentrieren, sondern Webseiten immer auch im Firefox zu testen.

Fundamental: HTML-Grundstruktur

Alle HTML-Dokumente basieren auf demselben Fundament, nämlich folgender Grundstruktur:

```
<!doctype html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <title>Aller Anfang ist leicht</title>
</head>
<body>
  <p>Und dann geht es so weiter.</p>
</body>
</html>
```

Die erste Zeile `<!doctype html>` ist die Dokumenttypangabe, im Beispiel diejenige von HTML5. Die Dokumenttypangabe verrät dem Browser, welche Version von HTML er zu erwarten hat.

Sie sehen keine direkte Auswirkung, ob Sie die Dokumenttypangabe schreiben oder nicht. Bei komplexeren Dokumenten mit CSS kann die Darstellung aber ohne Dokumenttypangabe nicht wie gewünscht aussehen. Also: Wenn Sie die Dokumenttypangabe immer am Anfang schreiben, sind Sie auf der sicheren Seite.



Falls Sie bei anderen HTML-Dokumenten in der ersten Zeile eine lange recht kryptische Angabe wie die folgende sehen:

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

... dann ist das der Dokumenttyp von HTML4 und das ist heute veraltet. Sie sollten es nicht mehr einsetzen.

In der nächste Zeile steht:

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

Das ist das Start-Tag von HTML und sagt dem Browser, dass der HTML-Code loslegt. Außerdem erhält das Start-Tag die Information, dass das Dokument auf Deutsch verfasst ist – dazu gleich mehr.

Mit der darauffolgenden Zeile:

```
<head>
```

... beginnt der Kopfbereich. HTML-Dokumente bestehen aus einem Kopf (head) und einem Rumpf (body). Im Kopfbereich stehen Informationen über das Dokument, etwa der Zeichensatz und der Seitentitel. Im body befindet sich hingegen alles, was Sie auf der Webseite sehen – also etwa Texte, Bilder, Aufzählungen, Verlinkungen. Im Beispiel sind zwei Elemente im Kopfbereich. Zuerst einmal die Zeichensatzangabe:

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

Durch diese Zeichensatzangabe ist sichergestellt, dass Sie Sonderzeichen wie Äs, Üs, ßs und Konsorten direkt schreiben können und dass sie richtig dargestellt werden. Das zweite Elemente im Kopfbereich bezeichnet den Seitentitel:

```
<title>Aller Anfang ist leicht</title>
```

Der Seitentitel ist wichtig, da er an drei zentralen Stellen zu sehen ist:

- ✓ im Tab oben im Browser
- ✓ als Standard, wenn man eine Webseite in die Lesezeichen/Favoriten aufnimmt
- ✓ in der Trefferliste von Suchmaschinen

Der Seitentitel ist damit eine zentrale Information und sollte dem Besucher verraten, worum es in der Webseite geht. Ein Beispiel sehen Sie in Abbildung 1.6.

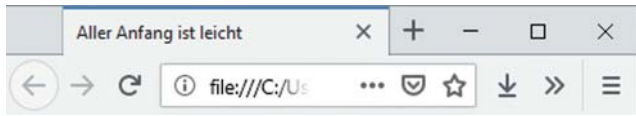


Abbildung 1.6: Seitentitel im Tab

Nach dem Seitentitel wird head geschlossen und body beginnt:

```
</head>
<body>
```

Im body steht das, was direkt im Browserfenster zu sehen ist.

```
<p>Und dann geht es so weiter.</p>
```

Im Beispiel ist das ein Absatz. Hier können Sie weitere Elemente wie Überschriften, Listen, Bilder, Links etc. unterbringen.

Zum Schluss müssen Sie noch body und html schließen.

```
</body>
</html>
```

HTML-Dokument erstellen

Um das Dokument zu erstellen und auszuprobieren, brauchen Sie einen Texteditor. Ein Textverarbeitungsprogramm wie Word ist hingegen nicht geeignet. Betriebssysteme bringen normalerweise einen Texteditor mit, aber es ist besser, wenn Sie von Anfang an einen Editor verwenden, der ein bisschen komfortabler ist. Kostenlos und für alle Betriebssysteme zur Verfügung steht beispielsweise der Editor Atom (<https://atom.io/>).

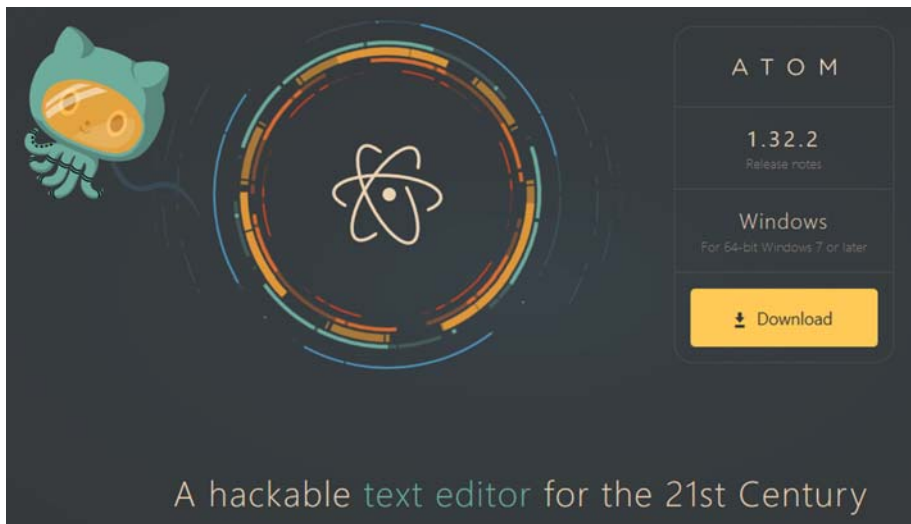


Abbildung 1.7: Webseite des Atom-Editors – das Maskottchen ist schon mal knuddelig.

Am besten laden Sie sich Atom herunter und installieren ihn. Sie können natürlich auch einen anderen Texteditor nehmen. Wenn Sie bereits einen Liebblingseditor haben, bleiben Sie unbedingt dabei.

Weitere Editoren

Falls Ihnen Atom nicht zusagt, hier ein paar Tipps für weitere Editoren:

- ✓ Notepad++ (<https://notepad-plus-plus.org/>) ist ein schöner, einfacher Editor, der aber nur für Windows zur Verfügung steht
- ✓ Brackets hingegen gibt es für alle Betriebssysteme. Sie finden ihn unter <http://brackets.io/>.
- ✓ Sublime Text (<https://www.sublimetext.com/>) steht auch für MacOS, Windows und Linux zur Verfügung. Sie können ihn kostenlos testen, werden dann aber immer wieder aufgefordert, ihn käuflich zu erwerben.
- ✓ Visual Studio ist eine integrierte Entwicklungsumgebung, das heißt, es bietet mehr Funktionen. Sie finden dieses von Microsoft entwickelte Tool unter <https://visualstudio.microsoft.com/de/vs/>.



Statt sich einen Editor zu installieren, können Sie auch eine Online-Editorumgebung wie CodePen (<https://codepen.io/>) oder JSFiddle (<https://jsfiddle.net/>) nutzen.

Geben Sie dann den Code ein und speichern Sie das Dokument als *grundgeruest.html* ab (Abbildung 1.8).

```
grundgeruest.html
1 <!doctype html>
2 <html lang="de">
3 <head>
4   <meta charset="UTF-8">
5   <title>Aller Anfang ist leicht</title>
6 </head>
7 <body>
8   <p>Und dann geht es so weiter.</p>
9 </body>
10 </html>
```

Abbildung 1.8: HTML-Grundgerüst in Atom

Zwei Dinge sind beim Speichern und bei der Wahl des Dateinamens wichtig:

1. Das Dokument braucht die Endung *html* oder *htm*. *html* ist üblicher.
2. Damit Sie keine Probleme bei der Weiterverarbeitung haben, empfiehlt es sich, **keine Leerzeichen und keine Sonderzeichen** wie »Ä«s und Co. im Dateinamen zu benutzen. Im Beispiel heißt die Datei deswegen nicht *grundgerüst.html*, sondern *grundgeruest.html*.



Unter Windows werden manchmal bei bekannten Dateien die Dateierendungen ausgeblendet. Das heißt, Sie sehen dann die Endung *.html* nicht, was irritierend ist. Deswegen ist es besser, wenn Sie sich die Endungen anzeigen lassen. Das geht so:

- ✓ Öffnen Sie den Windows-Explorer.
- ✓ Wählen Sie ganz rechts OPTIONEN.
- ✓ Im nun angezeigten Fenster klicken Sie auf den Tab ANSICHT.
- ✓ Hier finden Sie die Option ERWEITERUNGEN BEI BEKANNTEN DATEITYPEN AUSBLENDEN. Entfernen Sie das Häkchen und wenden Sie die Änderung durch OK an (siehe Abbildung 1.9).

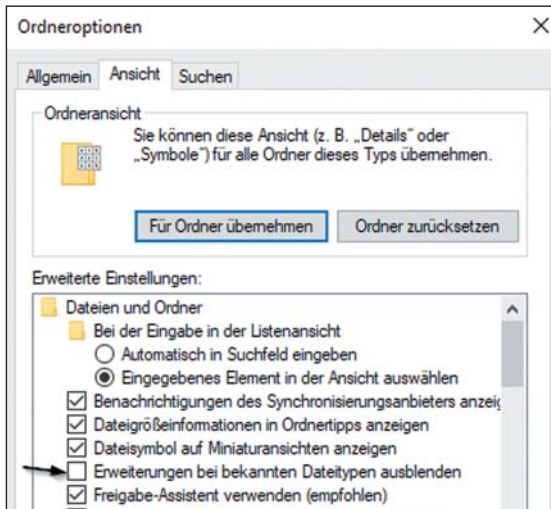


Abbildung 1.9: Windows: Damit die Endungen zu sehen sind

Dann können Sie das Dokument aufrufen. Dafür klicken Sie im Ordner, in dem Sie die Datei abgespeichert haben, doppelt auf den Dateinamen. Wie in Abbildung 1.10 zu sehen, öffnet sich das Dokument in Ihrem Standardbrowser.

Voilà – Ihr erstes Dokument. Zugegeben ist noch nicht sehr viel zu sehen – aber es ist ein vollständiges HTML-Dokument. Gratulation!



Abbildung 1.10: Das erste HTML-Dokument im Browser



Ein großer Vorteil des Editors Atom ist die große Anzahl an Packages, mit dem Sie ihn erweitern können. Wählen Sie dafür in Atom FILE|SETTINGS|INSTALL. Dann können Sie nach Packages suchen. Wenn Sie beispielsweise im Suchfeld »browser« eingeben, finden Sie unter anderem *open-in-browsers* aufgeführt. Ein Klick auf INSTALL installiert dieses Paket. Danach können Sie Ihr HTML-Dokument direkt von Atom aus im Browser öffnen. Das Package ergänzte in der Statusleiste eine Liste mit Browsericons, über die Sie das Dokument öffnen können – vorher das Speichern nicht vergessen!

Alles klar mit Attributen

Einige Sachen wurden noch nicht erklärt, zum Beispiel die folgende Angabe:

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

Damit legen Sie die Sprache des Dokuments fest, im Beispiel Deutsch. Das hilft automatisierten Tools, die Inhalte korrekt zu interpretieren. Syntaktisch gibt es aber eine Besonderheit. `lang` ist ein sogenanntes *Attribut* (auf Deutsch so viel wie »Beifügung«), das auf den *Attributwert* `de` gesetzt wird. Über Attribute können Sie zusätzliche Informationen für Elemente bereitstellen. Attribute haben folgende Besonderheiten:

- ✓ Attribute bestehen üblicherweise aus dem Attributnamen (hier: `lang`) und dem Attributwert (hier: `de`).
- ✓ Dazwischen steht ein Gleichheitszeichen.
- ✓ Der Attributwert steht in Anführungszeichen.

Attributangaben stehen dabei nur beim Start-Tag, sie werden nicht beim End-Tag wiederholt.

- ✓ Korrekt: `<html lang="de"> </html>`
- ✓ Falsch: `<html lang="de"></html lang="de">`

Auch im folgenden Fall steht ein Attribut

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

Hier heißt das Attribut `charset` (Zeichensatz) und der Attributwert ist `UTF-8`. UTF-8 ist der Standardzeichensatz und erlaubt es, alle üblichen (und unüblichen) Zeichen zu benutzen.

In manchen Fällen sind Attribute besonders wichtig, z. B. bei Verlinkungen. Wenn Sie einen Link auf die Seite »für dummies« erstellen wollen, d. h. auf die URL `http://www.wiley-vch.de/de/dummies`, schreiben Sie

```
<a href="http://www.wiley-vch.de/de/dummies ">Reihe für dummies</a>
```

Hier steht beim Attribut `href` (`href = hypertext reference`) die Internetadresse, die aufgerufen werden sollen, wenn der Besucher auf den Link klickt. Das `<a ...>` und `` umfasst den Linktext, also den anklickbaren Bereich.

Leere Elemente: leer, aber nicht hohl

Sie haben das Grundprinzip von HTML erlernt. Sie wissen, dass Elemente aus Start- und Endtag bestehen wie

- ✓ `<html>` und `</html>` für Anfang und Ende des HTML-Dokuments
- ✓ `<head>` und `</head>` für Anfang und Ende des Kopfbereichs
- ✓ `<title>` und `</title>` für Anfang und Ende des Seitentitels
- ✓ `<body>` und `</body>` für Anfang und Ende des Dokumentrumpfs
- ✓ `<p>` und `</p>` für Anfang und Ende eines Absatzes

Nicht in dieses Schema passt die Zeichensatzangabe:

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

Bei der Zeichensatzangabe fehlt nämlich das End-Tag. Elemente, die nur aus einem Start-Tag bestehen, nennt man »leere« Elemente. Sie heißen leer, weil sie keinen Text beinhalten und keine anderen HTML-Elemente. Sie haben aber trotzdem eine Funktion – diese besteht beim `meta`-Element darin, dass es Attribute aufnehmen kann. Im Beispiel kennzeichnet dieses `meta`-Element den Zeichensatz.

Übrigens können Sie leere Elemente auch mit einem zusätzlichen / (Slash) ausstatten. Dieser steht dann vor der schließenden spitzen Klammer. Die folgende Schreibweise ist also ebenfalls möglich:

```
<meta charset="utf-8" />
```

Ich verwende im Folgenden aber die Variante ohne den zusätzlichen Slash, weil sie kürzer ist.

Kontrolle ist besser

Das HTML-Grundgerüst müssen Sie nur einmal schreiben und können es dann für die nächsten Dokumente immer wieder kopieren. Es empfiehlt sich, das Grundgerüst einmal einzugeben, weil Sie sich dadurch das Grundprinzip besser einprägen. Bevor Sie aber das Grundgerüst als Basis für alle weiteren Dokumente nehmen, sollten Sie überprüfen, ob es korrekt ist.

Dass der Browser es richtig darstellt, heißt nicht unbedingt, dass der Code fehlerfrei ist. Browser sind nämlich bis zu einem gewissen Grad fehlertolerant. Browser bemühen sich bei Fehlern, zu erraten, was Sie gemeint haben könnten, und stellen das dar. Je komplexer allerdings die Dokumente werden, desto wichtiger ist es, keine Fehler zu machen, weil die Browserfehlertoleranz an ihre Grenzen kommt; das heißt, Fehler führen oft zu einer ungewünschten Darstellung.

Für eine schnelle Überprüfung öffnen Sie das Dokument in Firefox und lassen sich den Seitenquelltext anzeigen (Abbildung 1.11). Dafür klicken Sie mit der rechten Maustaste in das Dokument im Firefox und wählen SEITENQUELLTEXT ANZEIGEN.



```

1 <!doctype html>
2 <html lang="de">
3 <head>
4   <meta charset="UTF-8">
5   <title>Aller Anfang ist leicht</title>
6 </head>
7 <body>
8   <p>Und dann geht es so weiter.</p>
9 </body>
10 </html>
11

```

Abbildung 1.11: Seitenquelltext-Ansicht des Dokuments im Firefox

An den Farben erkennen Sie, wenn das Dokument korrekt ist – die Elementnamen erscheinen lila, der Text schwarz. Wenn an einer Stelle die Farbgebung anders ist, deutet das auf einen Fehler hin.

Abbildung 1.12, Abbildung 1.13 und Abbildung 1.14 zeigen ein paar Beispiele für mögliche Fehler: Im folgenden Fall ist das `title`-Element nicht geschlossen. Nach »ist leicht« müsste `</title>` statt `<title>` stehen.



```

1 <!doctype html>
2 <html lang="de">
3 <head>
4   <meta charset="UTF-8">
5   <title>Aller Anfang ist leicht<title>
6 </head>
7 <body>
8   <p>Und dann geht es so weiter.</p>
9 </body>
10 </html>
11

```

Abbildung 1.12: Ein kleiner Fehler

Rote Stellen weisen ebenfalls auf Fehler hin. Gerade am Anfang passiert es leicht, dass eine spitze Klammer in die falsche Richtung zeigt. Im folgenden Beispiel heißt es statt `<title>` folgendermaßen: `>title<`.



```

1 <!doctype html>
2 <html lang="de">
3 <head>
4   <meta charset="UTF-8">
5   >title>Aller Anfang ist leicht</title>
6 </head>
7 <body>
8   <p>Und dann geht es so weiter.</p>
9 </body>
10 </html>
11

```

Abbildung 1.13: Einmal spitze Klammer vertauscht

Ganz typisch am letzten Beispiel ist, dass die rote Hervorhebung **nach dem Fehler** auftaucht. Sie erscheint erst dann, wenn der Browser merkt, dass etwas nicht stimmt – und das kann an späterer Stelle sein.

Der effektivste Fehler, also der Fehler mit der größten Auswirkung, besteht darin, ein Anführungszeichen bei Attributwerten zu vergessen. Dieser Fehler ist effektiv, weil er sich im Gegensatz zu den anderen Fehler auch auf die Ansicht im Browser auswirkt – Sie sehen dann eventuell sogar gar nichts.



```

1 <!doctype html>
2 <html lang="de">
3 <head>
4   <meta charset="UTF-8"
5   <title>Aller Anfang ist leicht</title>
6 </head>
7 <body>
8   <p>Und dann geht es so weiter.</p>
9 </body>
10 </html>
11

```

Abbildung 1.14: Ein kleines Anführungszeichen hat eine große Wirkung.

Im Beispiel fehlt das Anführungszeichen hinter UTF-8 vor der schließenden spitzen Klammer.

CSS – auswählen, formatieren und los

HTML definiert die Struktur der Inhalte, aber ohne CSS sind Seiten langweilig. CSS ist für die Formatierung von Webseiten zuständig. Mit HTML bestimmen Sie, dass etwas ein Absatz ist, mit CSS legen Sie fest, wie dieser Absatz dargestellt werden soll.

CSS steht übrigens für Cascading Stylesheets, also etwa »kaskadierende Formatvorlagen«. Kaskadierend deswegen, weil es sein kann, dass mehrere Formatierungen wasserfallartig auf ein Element wirken und genau festgelegt ist, welche dann gelten. CSS wird ebenfalls wie HTML vom W3C betreut. Inzwischen ist Version CSS3 aktuell.



Es gibt einen wesentlichen Unterschied zwischen CSS 2.1 (der Vorversion) und CSS3: CSS 2.1 ist in einer umfangreichen Spezifikation beschrieben, in der alle Features von CSS aufgeführt sind. Seit CSS3 gibt es nicht mehr ein Dokument, sondern mehrere Dokumente für alle spezifischen CSS-Eigenschaften: Eine einzelne Spezifikation beschreibt etwa nur, wie man Rahmen erstellt, eine andere, wie man mit Schriften arbeitet. So können die einzelnen Bereiche separat weiterentwickelt und von den Browserherstellern implementiert werden.

Geschweifte Klammern, Doppelpunkt & Strichpunkt

CSS folgt eigenen Regeln. HTML ist die Welt der spitzen Klammern – daran erkennen Sie sofort, dass Sie ein HTML-Dokument vor sich haben. CSS sieht anders aus, wie das folgende Beispiel demonstriert:

```
body {
  background-color: yellow;
  color: blue;
}
```

In CSS regieren geschweifte Klammern, Doppelpunkte und Strichpunkte.

Am Anfang steht im Beispiel `body`. `body` ist ein *Selektor* (Auswähler). Dieser Ausdruck wählt aus, auf welches Element oder welche Elemente die Formatierungen angewendet werden sollen. Hier schreibt man häufig den Namen von HTML-Elementen – allerdings ohne spitze Klammern. Im Beispiel soll das `body`-Element formatiert werden. Da das `body`-Element den auf der Webseite sichtbaren Code enthält, wird damit die gesamte Seite formatiert. Dann folgt eine öffnende geschweifte Klammer, die am Ende wieder geschlossen wird. Alles, was innerhalb der geschweiften Klammern steht, gilt für das angegebene Element. Im Beispiel sind es zwei Deklarationen. Die erste Deklaration

```
background-color: yellow;
```

... legt als Hintergrundfarbe Gelb fest. Bei Deklarationen steht zuerst die *Eigenschaft* (`background-color`), dann folgen ein Doppelpunkt und der zugewiesene *Wert* (`yellow`). Ein Strichpunkt beendet eine Regel.

Die zweite Regel:

```
color: blue;
```

... funktioniert genauso: Dieses Mal heißt die Eigenschaft `color` und legt die Textfarbe fest. Zugewiesen wird nach einem Doppelpunkt die Farbe Blau (`blue`). Die CSS-Eigenschaften sind englisch und bestehen immer aus einem Wort, das auch Bindestriche beinhalten kann. Die Werte können englische Wörter sein wie im Beispiel `yellow` oder `blue`; sie können aber auch aus Längenangaben bestehen wie `10px` (10 Pixel) und bei manchen Angaben sind auch mehrere Werte möglich oder notwendig.

Stylesheets bestehen aus einer großen Anzahl solcher Regeln. Abbildung 1.15 zeigt einen Ausschnitt aus einem Stylesheet.

```
40 ▾ article,aside,details,figcaption,figure,footer,header,hgroup,main,menu,nav,section,summary {
41     display:block
42 }
43 ▾ audio,canvas,progress,video {
44     display:inline-block;
45     vertical-align:baseline
46 }
47 ▾ audio:not([controls]) {
48     display:none;
49     height:0
50 }
51 ▾ [hidden],template {
52     display:none
53 }
54 ▾ a {
55     background-color:transparent
56 }
57 ▾ a:active,a:hover {
58     outline:0
59 }
60 ▾ b,optgroup,strong {
61     font-weight:700
62 }
63 ▾ dfn {
64     font-style:italic
65 }
66 ▾ h1 {
67     margin:.67em 0
68 }
69 ▾ mark {
70     background:#ff0;
71     color:#000
72 }
```

Abbildung 1.15: Ausschnitt aus einem Stylesheet

CSS zu lernen, bedeutet zum einen, sich mit den Selektoren zu befassen – d. h. damit, wie Sie HTML-Elemente auswählen –, und zum anderen die Eigenschaften in ihren Zusammenwirkungen kennenzulernen.

Das erste Stylesheet

Stylesheets müssen mit dem HTML-Dokument verknüpft werden, damit sie eine Auswirkung haben. Wie das geht, sehen Sie nun. Erstellen Sie zuerst eine neue Datei mit dem Namen *stil.css*, die den gezeigten Inhalt aus Abbildung 1.16 hat:


```

1  body {
2    background-color: yellow;
3    color: blue;
4  }

```

Abbildung 1.16: CSS-Code

Wichtig ist die Endung `css`. Speichern Sie dieses Dokument im selben Ordner, in dem Sie Ihr HTML-Dokument abgespeichert haben.

Um dieses Dokument mit dem HTML-Dokument zu verknüpfen, müssen Sie wie in Abbildung 1.17 in Ihrem HTML-Dokument innerhalb von `<head>` und `</head>` eine zusätzliche Zeile ergänzen, nämlich:

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

```

1  <!doctype html>
2  <html lang="de">
3  <head>
4    <meta charset="UTF-8">
5    <title>Aller Anfang ist leicht</title>
6    <link rel="stylesheet" href="stil.css">
7  </head>
8  <body>
9    <p>Und dann geht es so weiter.</p>
10 </body>
11 </html>

```

Abbildung 1.17: HTML-Dokument mit Link auf Stylesheet-Datei

Das Attribut `rel` steht für *relation*. `rel="stylesheet"` verrät dem Browser, dass die angegebene Datei ein Stylesheet für das aktuelle Dokument ist. Wo die Datei zu finden ist, steht bei der Angabe `href="stil.css"`. Damit es klappt, muss die CSS-Datei `stil.css` heißen und im selben Ordner wie die HTML-Datei gespeichert sein. Bevor Sie das HTML-Dokument neu im Browser öffnen, müssen Sie es speichern. Wenn alles korrekt ist, sehen Sie wie in Abbildung 1.18 blauen Text auf gelbem Hintergrund.



Abbildung 1.18: Gewinnt noch keinen Schönheitswettbewerb, aber zeigt deutlich: CSS wirkt.

Falls es nicht funktioniert, kontrollieren Sie, ob der Link auf die CSS-Datei und die Datei selbst funktionieren, also:

- ✓ Hat die CSS-Datei denselben Namen, den Sie in Ihrem HTML-Dokument angegeben haben?
- ✓ Stimmt die Endung? Sie muss `.css` lauten.
- ✓ Befindet sich die Datei im selben Ordner wie das HTML-Dokument?
- ✓ Kontrollieren Sie dann am besten auch den CSS-Code!

Außerdem können Sie sich alle Listings auf der Verlagsseite unter www.wiley-vch.de/publish/dt/books/ISBN978-3-527-71585-5 herunterladen und diese Dateien mit Ihren vergleichen.

Kurzgefasst

HTML und CSS sind die Basis aller Webseiten. HTML legt die Struktur fest und CSS die Formatierung. In HTML kennzeichnen Sie Inhalte über Start-Tags (z. B. `<p>`) und End-Tags (`</p>`). Elemente können Attribute für weitere Informationen haben, wie beispielsweise `<html lang="de">`.

Damit CSS wirkt, muss es mit dem HTML-Dokument verknüpft werden, zum Beispiel über ein `link`-Element. In CSS schreiben Sie zuerst den Selektor und in geschweiften Klammern die Angaben, die gelten sollen. Zwischen Eigenschaften und Werten steht in CSS ein Doppelpunkt, also z. B. `color: green;`. Eine Deklaration wird mit einem Strichpunkt beendet.