

Datum:

Die abgebildeten Textauszüge stammen aus "HTML 5" v. C. Prevezanos, Burgthann 2015

Aufgabe

Lesen Sie die abgebildeten Texte durch. Führen Sie die dort beschriebenen Schritte parallel am PC durch.

Benutzen Sie dafür die Programme **Notepad++** oder **HTML-Kit**.

Speichern Sie Ihr Dokument unter der Bezeichnung "NACHNAME_Tags.html" (also z. B.: "Meier_Tags.html") auf dem Desktop.

2 Grundbefehle zum Erstellen von HTML-Dokumenten

Sie lernen in diesem Kapitel:

- die Grundlagen der HTML-Syntax
- die Struktur eines Dokuments zu erstellen
- den Body und Überschriften festzulegen
- weitere Bereiche gezielt zu deklarieren

Suchen Sie ein passendes Programm auf Ihrem PC und erstellen Sie ein neues Dokument mit der oben genannten Bezeichnung.

In diesem Kapitel starten wir mit dem Erstellen eigener HTML-Dokumente. Damit Sie die Beispiele direkt ausprobieren können, müssen Sie noch Folgendes wissen:

- Sie benötigen einen Texteditor zum Erstellen der Dateien. Das kann jedes kleine Schreibprogramm sein, welches in reinem ASCII-Text speichern kann. Es sind keine Formatierungen oder speziellen Dateiformate erlaubt. Wählen Sie beim Speichern immer das Format **TXT** oder **Nur-Text** aus. Die eigenen Formate von Microsoft Word, OpenOffice, LibreOffice, WordPerfect usw. dürfen nicht verwendet werden.

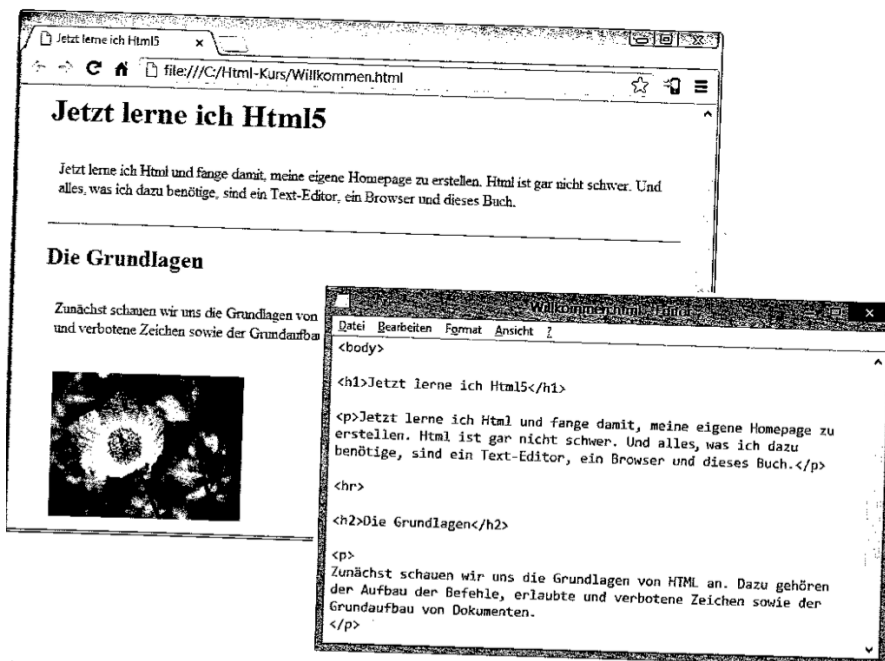


Abbildung 2.1: Ein Texteditor, ein Browser, dieses Buch und los geht es!

KAPITEL 2 So funktionieren HTML-Tags

- HTML-Dateien tragen immer die Endung **.html**. Die Endung **.htm** stammt aus der Zeit, als Windows nur das 8+3-Format kannte. Heute sollte nur noch die Bezeichnung **.html** verwendet werden. Die anderen Formate der HTML-Welt, wie z. B. **.shtml**, **.css**, **.xhtml**, brauchen Sie am Anfang nicht.
- Natürlich benötigen Sie auch einen Webbrowser, um Ihre HTML-Dateien sofort anzuschauen und zu überprüfen. Es spielt überhaupt keine Rolle, ob Sie Mozilla Firefox, Google Chrome, Internet Explorer oder einen anderen Browser verwenden. Es sollte aber die aktuelle Version sein, damit auch neue Befehle richtig interpretiert werden.

2.1 So funktionieren HTML-Tags

Wie Sie bereits wissen, stellt HTML eine Auszeichnungssprache dar. Die Inhalte des Dokuments werden dabei anhand von Markierungen als Elemente deklariert bzw. strukturiert. Diese HTML-Befehle werden meist als **Tags** bezeichnet. Ein HTML-Tag ist also einfach nur ein HTML-Befehl. Die Bezeichnung kommt aus dem Englischen und steht für Etikett, Marke oder auch Auszeichner (to tag = auszeichnen). Das erklärt auch gleich noch mal die Funktionsweise von HTML, indem Textelemente ausgezeichnet werden. Die Syntax von HTML-Befehlen ist sehr einfach und es gibt dabei auch nur wenig zu beachten. Deshalb werden Sie das Prinzip sofort problemlos verstehen und anwenden können.

2.1.1 Die Syntax von HTML-Tags

Die Syntax von HTML-Befehlen basiert auf dem Prinzip »Öffnen und Schließen«. Das bedeutet, dass ein Tag immer an einer bestimmten Stelle geöffnet und wieder geschlossen werden muss. Daraus erkennen Sie sofort, dass es für jeden HTML-Befehl mindestens zwei Auszeichnungen gibt – dort, wo er beginnt, und dort, wo er endet. In einem Beispiel könnte dies die Überschrift für Ihren Text sein. Sie geben dabei an, wo die Überschrift beginnt und wo sie endet. Der HTML-Tag zum Öffnen steht also direkt vor der Überschrift und der HTML-Tag zum Schließen direkt hinter der Überschrift.

befehl_überschrift_anfang

Überschrift

befehl_überschrift_ende

Genau so funktioniert das auch bei allen anderen Elementen innerhalb einer HTML-Datei. Sie geben also z. B. an, wo der lange Fließtext beginnt und aufhört, welcher Teil stellt eine Liste dar, welches Wort bildet ein Zitat usw. Das Prinzip ist bei allen HTML-Tags gleich.

Vergessen Sie niemals, einen HTML-Befehl zu schließen! Ein solches Versäumnis würde dazu führen, dass die gesamte Struktur durcheinander gerät und die Seite falsch oder gar nicht angezeigt würde. Gewöhnen Sie sich gleich von Anfang an eine gründliche Arbeitsweise an. Bei aufwendigen Dokumenten kann man leicht die Übersicht verlieren und dann wird die Fehlersuche zur Geduldsprobe.

HTML-Befehle stehen immer in spitzen Klammern. Nur dann werden sie vom Browser als solche erkannt. Vergessen Sie die spitzen Klammern, ignoriert der Browser den Befehl und zeigt ihn als Fließtext an.

```
<befehl_1>
<befehl_2>
<befehl_3>
```

Die Befehle zum Öffnen und Schließen eines HTML-Tags sind gleich. Allerdings wird dem Befehl zum Schließen ein Querstrich (Slash) vorangestellt, um ihn als End-Tag zu deklarieren. Das macht die Handhabung deutlich einfacher, denn Sie müssen sich nicht für jede HTML-Funktion zwei Befehle merken.

```
<befehl>
Text, Text, Text
</befehl>
```

Es spielt übrigens keine Rolle, ob Sie die HTML-Befehle und die Inhalte zusammen in eine Zeile setzen oder jeweils eine neue Zeile beginnen. Das stört weder die Syntax von HTML noch den Browser. Am Anfang kann es übersichtlicher wirken, wenn Sie jeweils eine neue Zeile verwenden, aber notwendig ist es nicht. In diesem Buch wird das variiert, je nachdem, was gerade praktischer oder übersichtlicher ist.

```
<befehl>
Text, Text, Text
</befehl>
```

```
<befehl>Text, Text, Text</befehl>
```

Achten Sie immer auf die Groß- und Kleinschreibung. Die Webserver und Browser verarbeiten zwar sowohl groß- als auch kleingeschriebene Befehle problemlos, HTML5 schreibt aber explizit eine Kleinschreibung für alle HTML-Tags vor. Gewöhnen Sie sich das sofort an, dann entsprechen Ihre Dokumente immer dem Standard. In HTML4 gab es diese Vorgabe nicht und oftmals wurden sämtliche Tags in Großbuchstaben geschrieben – vor allem von HTML-Editoren. Das sieht aber merkwürdig aus und ist heute nicht mehr akzeptabel. Schreiben Sie alle HTML-Befehle klein.

Falsch:

```
<BEFEHL>Text, Text, Text</BEFEHL>
```

Richtig:

```
<befehl>Text, Text, Text</befehl>
```

Wie fast überall existieren auch bei den HTML-Befehlen ein paar Ausnahmen. Im Widerspruch zum festen Prinzip »Öffnen und Schließen« gibt es ein paar Elemente, die nicht so funktionieren. Sie haben keinen Start- oder End-Befehl, sondern stehen für sich alleine. Das ist allerdings die große Ausnahme und es kommt nur bei sehr wenigen Befehlen vor. Bleiben Sie in jedem Fall beim strikten Öffnen und Schließen von Befehlen und sehen Sie die wenigen Ausnahmen als die berühmte Bestätigung der Regel an.

Füllen Sie Ihr Dokument mit den Inhalten:

```
<body>
```

```
<h1> Hallo Welt! </h1>
```

```
<h2> Autor: IHR NAME
</h2>
```

```
<br />
<hr />
```

```
<h3> Dies ist mein erster Versuch,
eine HTML-Seite aufzubauen!
</h3>
```

```
</body>
```

So müsste Ihr Ergebnis aussehen!

Hallo Welt!

Autor: IHR NAME

Dies ist mein erster Versuch, eine HTML-Seite aufzubauen!

KAPITEL 2

So funktionieren HTML-Tags

2.1.2 Der Umgang mit Leerzeichen und Tabstopps

HTML kennt keine Formatierungen oder Zeichenausrichtungen im herkömmlichen Sinne. Leerzeichen, Absätze und Tabulatoren werden vollständig ignoriert. Sie müssen sich also keine Mühe geben, Ihren Text mit solchen Elementen zu formatieren. Im Browser wird das alles als ein langer Fließtext ausgegeben.

```
Text, Text, Text
Text
    Text
    Text
        Text
```

Ein solcher Text wird im Browser so angezeigt:

```
Text, Text, Text Text Text Text Text
```

Die einzige Ausnahme stellt hier das Leerzeichen da. Es wird im Fließtext als solches erkannt und angezeigt. Allerdings erfolgt das nur einmal zur Worttrennung. Mehrere Leerzeichen in Folge werden ignoriert und als ein einziges angezeigt.

Sie können diese Elemente aber durchaus dazu verwenden, um Ihren HTML-Code übersichtlicher zu gestalten. Strukturieren Sie Bereiche wie die Überschrift, den Fließtext, Unterelemente usw. durch Absätze, Leerzeichen oder auch Tabulatoren. Je größer und umfangreicher ein HTML-Dokument wird, desto schneller geht die Übersicht verloren. Eine gute Strukturierung ist praktisch, sieht im Texteditor besser aus und beeinflusst die Verarbeitung auf dem Server oder im Browser in keiner Weise.

```
<befehl>Überschrift</befehl>
```

```
<befehl>
Fließtext, Fließtext, Fließtext
</befehl>
```

```
    <befehl>
    Neuer Absatz mit Text
    </befehl>
```

```
        <befehl>Text</befehl>
        <befehl>Text</befehl>
        <befehl>Text</befehl>
```

```
<befehl>Text</befehl>
```

2.1.3 Verbotene Zeichen und Symbole

In einem HTML-Dokument vermischen sich die Inhalte und die Befehle selbstverständlich. Aus diesem Grund müssen Sie in Ihren Inhalten, also dem Text, den Bildern, den Links usw., bestimmte Zeichen vermeiden. Sie werden sonst fälschlicherweise als Befehle interpretiert. Das kann leicht das gesamte HTML-Dokument durcheinanderbringen.

Fügen Sie auf Ihrer Website entsprechende Beispiele ein. Nutzen Sie dazu nur die tags h3 und h4. Als Inhalt stellen Sie dar, welche Unterrichtsfächer Sie haben.

Grundbefehle zum Erstellen von HTML-Dokumenten

KAPITEL 2

Auf keinen Fall dürfen Sie die spitzen Klammern verwenden. Mathematische Formeln mit »größer als« und »kleiner als« dürfen Sie damit nicht beschreiben. Ebenso wenig ist es möglich, damit Pfeile oder andere Symbole darzustellen. Möchten Sie solche Sonderzeichen verwenden, gibt es dafür spezielle Codes, die im nächsten Abschnitt vorgestellt werden.

Verboten:

10 > 8

Siehe hier -> und dort <-

Das finde ich <<SUPER>>

Bei der Verwendung von Anführungszeichen und Apostrophen müssen Sie ebenfalls besonders vorsichtig sein. Innerhalb von Fließtext hat HTML selbst damit zwar keine Probleme, aber es können sich später bei der Verwendung von Skriptsprachen welche ergeben. In Programmier- bzw. Skriptsprachen wie JavaScript, PHP, Perl usw. werden Anweisungen, Schleifen, Variablen und Ähnliches in Anführungszeichen oder Apostrophe gesetzt. Arbeiten Sie unsauber und vergessen z. B. die Anführungsstriche wieder zu schließen, kann dies fälschlicherweise als Teil des Programms angesehen werden. Verwenden Sie diese Zeichen also mit Vorsicht oder besser gleich, wie im nächsten Abschnitt beschrieben, mit der entsprechenden HTML-Codierung.

In Ordnung:

Ich möchte "HTML" lernen.

Problematisch:

Ich möchte 'HTML' lernen.

Probleme vorprogrammiert:

Ich möchte "HTML lernen.

Ich möchte HTML ' lernen.

Bisher durften HTML-Dokumente lediglich Zeichen aus dem ASCII-Zeichensatz beinhalten. Er erlaubt nur die einfachen Buchstaben von A–Z, die Ziffern von 0–9 sowie ein paar wenige Sonderzeichen. Damit sind z. B. keine Buchstaben aus dem Französischen, Russischen oder auch deutsche Umlaute möglich. Das hat sich mit HTML5 geändert und durch den neuen Standardzeichensatz UTF-8 stehen Ihnen viel mehr Möglichkeiten zur Verfügung. Deutsche Umlaute und die meisten europäischen Sonderzeichen sind überhaupt kein Problem mehr.

Trotzdem sollten Sie weiterhin vorsichtig sein. »Einfache« Sonderzeichen beherrscht inzwischen fast jedes Programm, aber sobald es zu exotisch wird, reagieren viele mit merkwürdigen Fehlern. Vermeiden Sie also Sonderzeichen im Quelltext möglichst. Eine Übersicht der erlaubten ASCII-Zeichen finden Sie in »Anhang D – Sonderzeichen« ab Seite 417.

Vermeiden Sie:

á è é à î Ñ Æ

Θ Ω Φ Ψ

Ф Д Ч И Ъ Ы

Vollziehen Sie diese Beispiele praktisch nach!

KAPITEL 2 So funktionieren HTML-Tags**2.1.4 ü – Sonderzeichen im Text verwenden**

Mit der Einführung von UTF-8 als Standardzeichensatz werden die meisten alltäglichen Sonderzeichen von den Browsern selbstständig erkannt und verarbeitet. Trotzdem gibt es noch eine ganze Reihe von Sonderzeichen, die sich nicht auf diese Weise verarbeiten lassen. Damit diese meist schwierigen oder exotischen Sonderzeichen trotzdem möglich sind, bietet HTML eine eigene Codierung dafür. Diese Codierung funktioniert ganz unabhängig vom gewählten Zeichensatz und der Sprache, sodass sie von jedem Browser dargestellt werden kann.

Die Codierung ist bei allen Sonderzeichen ähnlich: Eingeschlossen von einem kaufmännischen Et (= &) und einem Semikolon (= ;) geben Sie den Code für das Sonderzeichen ein.

&CODE1; &CODE2; &CODE3;

Bei der Wahl der Codes hat man sich an den tatsächlichen Namen der Zeichen orientiert, wodurch man sich diese deutlich besser merken kann. Ein deutsches »ä« wird auch als »a mit Umlaut« bezeichnet. Und genau so lautet auch der Code in verkürzter Form.

ä = ä
ü = ü
ö = ö
ß = ß

Der Code unterscheidet hierbei Groß- und Kleinschreibung. Wird der erste Buchstabe des Codes großgeschrieben, erscheint auch das Sonderzeichen groß.

Ä = Ä
Ü = Ü
Ö = Ö
&Szlig; = ß (identisch, da es kein großes ß gibt)

Sie setzen diese Sonderzeichen einfach ohne Leerzeichen direkt in Ihren Satz bzw. in das betreffende Wort. Das mag im Quellcode am Anfang verwirrend und unübersichtlich wirken, funktioniert aber problemlos.

```
<p>
Demn&auml;chst erfahren Sie mehr &uuml;ber HTML, z.B. wie Sie Aufz&auml;
hlungen erstellen, Abs&auml;tze formatieren, Bilder einf&uuml;gen usw.
</p>
```

Verwenden Sie als Zeichencodierung ISO-8859-1, ist das spezielle Codieren von deutschen Sonderzeichen überflüssig. Trotzdem wurden sie hier erläutert, weil sie sich sehr gut als Beispiel eignen. Im Alltag müssen Sie das nicht tun, wenn Sie den Zeichensatz mit angeben. Es gibt aber eine ganze Reihe von Sonderzeichen, die sich nicht direkt verarbeiten lassen. Oft gibt es dafür gar keine Taste auf der Tastatur. Typische Beispiele sind mathematische Formeln, internationale Währungszeichen, Wirtschaftszeichen wie ®, ™, © usw. Ebenso dürfen Sie keine spitzen Klammern verwenden, weil sie ein HTML-eigenes Kommando sind. Alle diese Zeichen müssen Sie zwangsläufig wie hier beschrieben codieren.

Fügen Sie auf Ihrer Website unterhalb der Unterrichtsfächer ein:

- eine Linie
- folgenden Text:
Er sagte: "Nenn' mir die Maße und Preise für Gartenhäuschen!".
Ich erinnerte mich an einen Wert > 2,80 m und < 3,20. Preis: ø 1.200 €