

1.3 HTML/HTML-Grundlagen

1.3.1 Meine erste Datei im Browser

HTML-formatierter Text besteht aus ASCII-Text und so genannten Markup-Befehlen. Der ASCII-Text wird 1:1 übernommen. Die Markup-Befehle werden ausgeführt, falls sie dem Browser bekannt sind.

1.3.1.1 Exkurs: Editor+Browser+Quelltext-Ansicht In Schritt eins erstellt man mit dem Editor die Datei `/home/schueler/hello1.html`. Man sollte den Editor nicht schließen, sondern bei Bedarf speichern (mit `Strg` - `S`).

```
1 Das ist das Haus vom Nikolaus.
```

- in Linux: `gedit /home/schueler/hello1.html`
- in Wind.: `edit C:\tmp\hello1.html`

In Schritt zwei wird der Browser gestartet, man lädt die Datei `/home/schueler/hello1.html`. Dazu gibt man in die Adresszeile ein:

- in Linux: `file:///home/schueler/hello1.html`
- in Wind.: `file:///C:\tmp\hello1.html`

Auch den Browser sollte man nicht schließen, sondern nur (bei Bedarf) die Datei neu laden mit `Strg` - `R`.

1.3.2 Mein erster HTML-Befehl

1.3.2.1 Befehle In HTML nennt man die Befehle Tags. Sie befinden sich in spitzen Klammern: `<befehl>` oder `<befehl>`. Mit `<befehl>` wird eine Eigenschaft begonnen und mit `</befehl>` wird sie beendet.

Ein HTML-Bereich beginnt mit dem öffnenden Tag und endet mit dem schließenden Tag: `<befehl>bla bla bla</befehl>`

Man kann Bereiche schachteln: `<a>` ist erlaubt, `<a>` dagegen nicht. **Immer der Bereich, der zuletzt begonnen wurde, muss zuerst wieder geschlossen werden.** Andernfalls kommt der Browser durcheinander und zeigt Unsinn an.

Beispiel-Befehle sind ``, `<i>`, `<tt>`, `<u>`, `<blink>`, `<sub>`, `<sup>`. Sie werden gezeigt in `src/hello2.html`:

```
1 Das ist das <b>Haus</b> vom Nikolaus.
```

1.3.2.2 Schachtelung Befehle können kombiniert (=geschachtelt) werden wie die Blöcke in einer Programmiersprache Intelligente Editoren rücken den Text entsprechend ein (wie in Java, C, ...).

```
1 Das ist
2 <b>
3   das
4   <u>
5     Haus
6   </u>
7   vom
8 </b>
9 Nikolaus.
```

Man kann so viele Befehle schachteln, wie man will:

```

1 Das
2 <b>
3   ist
4   <u>
5     das
6     <i>
7       <blink>
8         Haus
9         <tt>
10          vom
11          <sup>
12            Nikolaus .
13          </sup>
14        </tt>
15      </blink>
16    </i>
17  </u>
18 </b>

```

1.3.2.3 Besonderer Befehl: Kommentar Auch Kommentare können eingebunden werden:

```

1 Das ist das Haus <!-- stimmt gar nicht --> vom Nikolaus .

```

Das Problem dabei ist, dass der Kommentar in der Quelltext-Ansicht sichtbar bleibt.

1.3.3 Mein erstes HTML-Dokument

1.3.3.1 Grundstruktur von HTML bis Version 2.0 Die Grundstruktur einer HTML-Datei zeigt die folgende Datei:

```

1 <html>
2   <head>
3     <title>Eine neue Erkenntnis</title>
4   </head>
5   <body>
6     Ohne Flei&szlig; kein Preis .
7   </body>
8 </html>

```

Das `<html>`-Tag dient zur Abgrenzung von anderen Formaten (SGML, XML, ...). Das `<head>`-Tag markiert den HTML-Kopf:

- Was hier steht, gilt für das ganze Dokument.
- Was hier steht, dient nicht dazu, angezeigt zu werden...

Das `<body>`-Tag markiert den HTML-Rumpf:

- Was hier steht, gilt nur für den Bereich, in dem es steht.
- Was hier steht, wird angezeigt.

1.3.3.2 Was kann man im Kopf festlegen? Das `<title>`-Tag gibt den Titel des Dokuments an. Im Dokument selbst hat es keine Auswirkung, siehe `src/hello6.html`:

```

1 <html>
2   <head>
3     <title>Eine neue Erkenntnis</title>
4   </head>
5   <body>
6     Ohne Flei&szlig; kein Preis.
7   </body>
8 </html>

```

1.3.4 Mehr Formatierungsbefehle

HTML erkennt die Zeilenumbrüche im ASCII-Text nicht an, siehe `src/hello7.html`:

```

1 <html>
2   <head>
3     <title>Ein tolles Thema</title>
4   </head>
5   <body>
6     Hier steht ganz viel Text drin.
7
8     Und t&auml;glich wird es mehr.
9   </body>
10 </html>

```

Der Zeilenumbruch erfolgt über den Befehl `
`, siehe `src/hello8.html`:

```

1 <html>
2   <head>
3     <title>Ein tolles Thema</title>
4   </head>
5   <body>
6     Hier steht auch ganz viel Text
7     drin.
8     <br />
9     Und hier ist eine zweite Zeile , von der ersten durch einen Befehl getrennt.
10  </body>
11 </html>

```

Soll in einem Bereich kein Zeilenumbruch stattfinden, so kann man das über die Kombination `<nobr>...<wbr>...</nobr>` festlegen, siehe `src/hello9.html`. Wenn man die eigenen Zeilenumbrüche behalten will, empfiehlt sich vorformatierter Text (auch für Quelltext-Darstellung zu gebrauchen: `<pre>...</pre>`, siehe `src/hello10.html`).

Einen Absatz kann man mit den Tags `<p>Absatz</p>` eingrenzen, siehe auch `src/hello11.html`:

```

1 <html>
2   <head>
3     <title>Text mit mehreren Abschnitten</title>
4   </head>
5   <body>
6     <p>
7       Dieses ist der erste Abschnitt. Hiermit f&auml;ngt der Text an.
8     </p>

```

```

9      <p>
10     Dieses ist der zweite Abschnitt.
11     </p>
12     <p>
13     Und dieses ist der dritte Abschnitt. Jeder Abschnitt sollte
14     vom vorhergehenden durch einen Zeilenumbruch, einen vertikalen
15     Abstand oder durch eine Einrückung getrennt sein.
16     </p>
17     </body>
18 </html>

```

Einen zentrierten Bereich erreicht man über `<center>...</center>`, siehe `src/hello12.html`. Eine waagerechte Linie erreicht man mit dem `<hr />`-Tag. Kapitelüberschriften gibt es in sechs Größen von h1 (größte) bis h6 (kleinste): `<h1>Kapitelüberschrift</h1>`, `<h2>Unterkapitelüberschrift</h2>` bis `<h6>...</h6>`.

1.3.5 Sonderzeichen

Zeichen, die im ASCII nicht vorkommen, wie Umlaute oder das Ohm-Zeichen, müssen gesondert codiert werden:

Zeichen	Codierung
ä	<code>&auml;</code>
ö	<code>&ouml;</code>
ü	<code>&uuml;</code>
Ä	<code>&Auml;</code>
Ö	<code>&Ouml;</code>
Ü	<code>&Uuml;</code>
ß	<code>&szlig;</code>

Außerdem gibt es Steuerzeichen, die zwar im ASCII vorkommen, aber innerhalb von HTML-Dateien zur schöneren Formatierung verwendet werden. Auch sie müssen codiert werden:

Zeichen	Codierung
Zeilenende	<code>
</code>
Leerzeichen	<code>&nbsp;</code>

Zuguterletzt gibt es noch Zeichen, die nur deshalb codiert werden müssen, weil sie in Befehlen vorkommen:

Zeichen	Codierung
<	<code>&lt;</code>
>	<code>&gt;</code>
&	<code>&amp;</code>
"	<code>&quot;</code>
'	<code>&apos;</code>