



## 1 Grundlagen zu HTML

HTML (**H**yper**T**ext **M**arkup **L**anguage) ist eine Sprache zur Strukturierung von Texten, wobei aber auch die Möglichkeit besteht, Grafiken und multimediale Inhalte in Form einer Referenz einzubinden und in den Text zu integrieren.

Mit HTML können Überschriften, Textabsätze, Listen und Tabellen erzeugt werden. Sie können anklickbare Verweise (Hyperlinks) auf beliebige andere Web-Seiten oder Datenquellen im Internet erzeugen. Weiterhin bietet HTML Schnittstellen für Erweiterungssprachen wie CSS Stylesheets oder JavaScript an, mit deren Hilfe Sie HTML-Elemente nach Wunsch gestalten und formatieren oder Interaktion mit dem Anwender realisieren können.

HTML ist ein so genanntes Klartext-Format. HTML-Dateien können Sie mit jedem beliebigen Texteditor bearbeiten, der Daten als reine Textdateien abspeichern kann. Es gibt also keine bestimmte Software, die man zum Erstellen von HTML-Dateien benötigt.

HTML bedeutet **H**yper**T**ext **M**arkup **L**anguage. Es handelt sich dabei um eine Sprache, die mit Hilfe von SGML (Standard Generalized Markup Language) definiert wird. SGML ist als ISO-Norm 8879 festgeschrieben. Mittlerweile gibt es einen Ableger von HTML namens XHTML.

HTML ist eine so genannte Auszeichnungssprache (Markup Language). Sie hat die Aufgabe, die logischen Bestandteile eines textorientierten Dokuments zu beschreiben.

## 2. HTML-Versionen

Das W3-Konsortium ist für die Standardisierung von HTML verantwortlich und entscheidet somit, welche Befehle (Tags) zu den Sprachstandard von HTML gehören.

- HTML 1.0 enthielt nur Befehle für Elemente wie Überschriften, Textabsätze, Grafikreferenzen und Verweise (Links).
- HTML 2.0 wurde Ende 1995 der Sprachstandard. Er enthielt wenig neues, gilt aber als kleinster gemeinsamer Nenner.
- HTML 3.0 ist nie offiziell geworden und wurde weiter überarbeitet.
- HTML 3.2 wurde Anfang 1997 offizieller Sprachstandard. Ab dieser Version waren auch Tabellen offizieller Bestandteil von HTML.
- HTML 4.0 wurde Anfang 1998 offizieller Sprachstandard. Neben Frames gab es jetzt auch die Möglichkeit zur Einbindung von Cascading Style-Sheet (CSS) und Script-Sprachen in HTML. Nach ein paar Überarbeitungen liegt HTML jetzt in der Version 4.01 vor.
- XHTML 1.0 wurde Anfang 2000 verabschiedet und ist der Nachfolger von HTML 4.01. XHTML basiert nicht mehr auf SGML, sondern auf XML und ist dadurch syntaktisch kompatibel zu anderen XML-basierenden Sprachen (SVG, WML, ...). XHTML hat einen strengeren Syntax als HTML.
- XHTML 1.1 wurde Ende Mai 2001 verabschiedet. Der Hauptunterschied gegenüber Version 1.0 ist die Entfernung missbilligter Features.

## 3. HTML-Aufbau / Syntaxregeln

Die HTML-Steuerbefehle bezeichnet man als Tags. Diese Tags werden durch spitze Klammern <>, die Kleiner- und Größerzeichen, markiert. Fast alle HTML-Befehle besitzen einen einleitenden und einen abschließenden Tag. Auf alles was zwischen diesen beiden Tags steht, bezieht sich dieser Befehl.

Beispiel:

```
<h1>Dies ist eine Ueberschrift</h1>
```

<h1> ist der einleitende Tag, den abschließenden Tag erkennt man an dem Slash (/) und heißt hier </h1>. Alles was zwischen <h1> und </h1> steht wird also entsprechend formatiert.



In HTML ist die Groß- und Kleinschreibung egal, in XHTML dagegen sollen alle Tags klein geschrieben werden, darum ist zu empfehlen alle Tags klein zu schreiben.

Natürlich können auch mehrere Tags auf ein Element angewendet werden, dabei ist aber auf die richtige Verschachtelung der Tags zu achten.

Beispiel:

```
<h2>Dies <tt>ist ein <sup>formatierter</sup> Text</tt></h2>
```

Aussehen:

**Dies ist ein <sup>formatierter</sup> Text**

Hier erkennt man, dass der Tag der als letztes geöffnet wird, als erstes wieder geschlossen werden muss.

## 4. Einfacher Seitenaufbau

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

Diese HTML-Tags zaubern eine leere, meist weiße Seite in den Browser und somit auf den Monitor.

Dabei umschließt der <html>-Tag die HTML-Seite und teilt dem Browser mit, was für Daten kommen (Programm, Bild, Seite, ...).

### 4.1 Bedeutung und Inhalte von <head>

Der <head>-Bereich ist der Seitenkopf, er geht von <head> bis </head> und beinhaltet Informationen zur Seite. Diese Informationen können z.B. sein: der Seitentitel, der Autor, globale Formatierungen, Herausgeber, Schlüsselwörter, ... Im Seitenkopf stehen also Informationen, die auf der Seite selbst nicht angezeigt werden (Titel wird im Browser angezeigt, aber nicht auf der Seite), es sind sozusagen die Eigenschaften dieses Dokumentes.

Beispiele:

<meta name="generator" content="HTML-Editor">	Angabe über den verwendeten Editor
<meta name="content-language" content="de">	Angabe über die Sprache des Dokumentes
<meta name="author" content="Karl Fahrtmann">	Angabe über den Autor des Dokumentes
<meta name="publisher" content="Netzwelt Katlenburg-Lindau e.V.">	Angabe über den Herausgeber
<meta name="keywords" content="Stichwort1, Stichwort2, Stichwort3">	Auflistung von Stichwörtern
<meta name="description" content="Kurzbeschreibung">	Kurzbeschreibung des Seiteninhaltes

### 4.2 Bedeutung und Inhalte von <body>

Der <body>-Bereich geht von <body> bis </body> und enthält darzubietenden Inhalt der Seite. Dieser Inhalt kann bestehen aus Text/Tabellen, Bildern, Videos, Klängen, Geräuschen, Musik, Sprache.

Für die Gestaltung der Seite stehen eine ganze Reihe HTML-Tags zur Verfügung.

Der <body>-Bereich beschreibt also den Inhalt der Seite und die Art der Darstellung dieses Inhaltes.



## 5 Grundlegende Tags

### 5.1 Textabsätze <p> und Zeilenumbrüche <br>

Beim Erstellen von Webseiten reicht es nicht aus den Text durch Leerzeichen, Tabulatoren und Zeilenumbrüche (RETURN) zu formatieren, denn ein Browser fasst mehrere Leerzeichen zu eins zusammen und entfernt Zeilenumbrüche die mit RETURN eingefügt wurden sind.

Normaler Text sollte in HTML in <p>-Tags (p=paragraph) eingeschlossen sein, so erzeugt man einen Absatz. Jeder neue Absatz muss wieder in neuen <p>-Tags eingeschlossen werden. In HTML wird es geduldet, wenn an der Stelle an der der Absatz erscheinen soll, nur ein einfaches <p> gesetzt wird. In XHTML ist dies nicht mehr erlaubt, es muss immer ein schließendes Tag vorhanden sein. Darum meine Empfehlung, <p>-Tags immer schließen.

Für einen einfachen Zeilenumbruch in einem Text muss an der entsprechenden Textstelle ein <br> (br=break) geschrieben werden. Im Quelltext der Seite kann direkt hinter dem <br>-Tag weitergeschrieben werden, aber der Übersicht halber sollte der Text in einer neuen Zeile fortgesetzt werden.

Beispiele für <p> und <br>:

```
<p>Dies ist der 1. Absatz.</p>
<p>Und hier ist der 2. Absatz,<br>mit zwei Zeilen!</p>
```

Und so sieht das Ergebnis aus:

Dies ist der 1. Absatz.

Und hier ist der 2. Absatz,  
mit zwei Zeilen!

Am Ende des Browserfensters wird natürlich ein automatischer Zeilenumbruch erzeugt, es sei denn ein Element das nicht umbrochen werden kann ist breiter als das Browserfenster. Wollen Sie verhindern das der Umbruch zwischen zwei bestimmten Wörtern geschieht, so muss das Leerzeichen durch ein geschütztes Leerzeichen ersetzt werden.

Ein geschütztes Leerzeichen wird &nbsp; geschrieben (nbsp = no breaking space), dass &-Zeichen und das Semikolon sind mit einzugeben. Aus Wort1 Wort2 wird somit Wort1&nbsp;Wort2. Anstelle von &nbsp; kann auch &#160; geschrieben werden.

### 5.2 Überschriften

In HTML gibt es sechs verschiedene Überschriftenebenen, die dazu genutzt werden können um Hierarchieverhältnisse in Dokumenten dazustellen. Eine Überschrift erster Ordnung erzeugt man mit <h1>Text</h1>. Das h steht für heading (=Überschrift) und die Zahl für die Überschriftenebene. Dabei ist 1 die höchste und 6 die niedrigste Ebene.

Es ist darauf zu achten, dass der abschließende Tag die gleiche Nummer hat wie der einleitende Tag. Außerdem erzeugt eine Überschrift einen Absatz davor und dahinter und darf somit auch nicht in <p>-Tags gesetzt werden.



## 5.3 Textauszeichnungen

Textauszeichnungen dienen dazu Textabschnitte zu formatieren. Wie bei fast allen Tags steht am Anfang des zu formatierenden Textes der einleitende und am Ende des Textes der abschließende Tag. Mit physischen Textauszeichnungen wird das Aussehen des Textes beschrieben, mit logischen dagegen wird die Textfunktion beschrieben.

<code>&lt;i&gt;...&lt;/i&gt;</code>	kursiv (italic)	normaler und <i>kursiver Text</i>
<code>&lt;b&gt;...&lt;/b&gt;</code>	fett (bold)	normaler und <b>fetter Text</b>
<code>&lt;tt&gt;...&lt;/tt&gt;</code>	Fernschreiber (Teletyper)	normaler und Fernschreiber- bertext
<code>&lt;big&gt;...&lt;/big&gt;</code>	größer formatiert	normaler und größerer Text
<code>&lt;font size +/-x&gt;...&lt;/font&gt;</code>	Schriftgrößenänderung	x steht für ganze Zahlen
<code>&lt;small&gt;...&lt;/small&gt;</code>	kleiner formatiert	normaler und kleinerer Text
<code>&lt;sup&gt;...&lt;/sup&gt;</code>	hochgestellt	normaler und <sup>hochgestellter Text</sup>
<code>&lt;sub&gt;...&lt;/sub&gt;</code>	tiefgestellt	normaler und <sub>tiefgestellter Text</sub>
<code>&lt;p align=right&gt;...&lt;/p&gt;</code>	rechtsbündig	alternativ: left, center
<code>&lt;blink&gt;...&lt;/blink&gt;</code>	blinkend	Blinkender Text

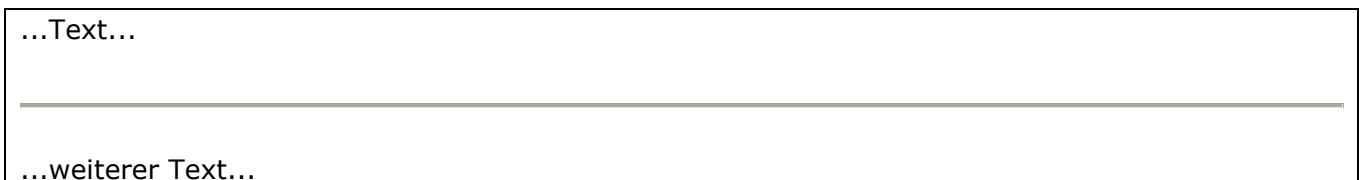
## 5.4 Trennlinien

In HTML können horizontale Trennlinien `<hr>` eingefügt werden zur optischen Trennung von Elementen. Jede Trennlinie erzeugt einen Absatz und darf somit nicht in `<p>`-Tags eingeschlossen werden. Trennlinien bestehen in HTML nur aus einem Tag, es gibt also kein abschließendes Tag.

Beispiel:

```
...Text...<hr>...weiterer Text...
```

Aussehen:



In XHTML dagegen muss immer ein abschließender Tag vorhanden sein. Eine Schreibweise ist `<hr />`, somit wird dieser Tag auch von nicht-XHTML fähigen Browsern verstanden.

## Aufgaben:

1. Entpacken Sie die Datei **selfhtml.zip** in Ihr Verzeichnis und verschaffen Sie sich einen ersten Überblick über dieses Tutorial.
2. Starten Sie einen Texteditor. Entwerfen Sie den in Punkt 4 gegebenen Aufbau eines html-Dokuments. Erstellen Sie im Bereich `<body>` einen beliebigen Text (z. B. den Klassiker „hello world“) und formatieren Sie diesen mehrfach mit den gegebenen Tags.

