

# HTML

Es gibt zwei Möglichkeiten, um Seiten für das Internet zu schreiben.

1. Es gibt Programme, bei denen du nichts oder nur wenig über **HTML** wissen musst. In diesen Programmen siehst du die Seiten gleich so, wie sie auch im Internet erscheinen. Wenn du mit so einem Programm arbeiten möchtest, kannst du im Handbuch nachlesen, wie es funktioniert.
2. Es besteht die Möglichkeit, die Seiten richtig zu schreiben (programmieren). Du lernst bestimmte Befehle mit denen du festlegst, wie deine Seite aussehen soll.

Was du alles brauchst, wenn du **HTML** lernen willst.

1. Einen Computer auf dem du die Seiten programmierst.
2. Ein Programm das "Browser" genannt wird. Das benötigst du, um das Ergebnis deiner Programmierung anschauen zu können. (z.B. Firefox, Opera, Netscape Navigator oder MS Internet Explorer)
3. Ein Programm mit dem du Texte schreiben kannst. Ein einfacher Editor reicht voll und ganz aus.

## Warum Handarbeit?

### Vorteile

- volle Kontrolle auch über Details
- besser lesbarer Code (kommt auf den Schreiber an)
- es ist keine besondere Software nötig, HTML kann mit allem erstellt werden (selbst mit Notepad oder Kate)
- sauberer Code (sofern der Ersteller ein wenig Ahnung hat)
- besondere Kniffe sind per Hand einfacher realisierbar
- it's cool, man

### Nachteile

- Kenntnisse von HTML müssen vorhanden sein
- am Anfang zeitaufwendiger

## Was ist HTML?

HTML ist eine Seitenbeschreibungssprache, mit der dem Browser "gesagt" wird, wie die Daten angezeigt werden: welche Einheiten (Buchstaben, Wörter, Sätze, Absätze) z.B. fett gedruckt werden, in bestimmten Farben dargestellt werden etc.

Dazu werden über das Internet HTML-Seiten übertragen. Das Bedeutsame dabei ist, dass die HTML-Seiten in Klartext übertragen werden und so jeder schauen kann, wie das eigentlich funktioniert. Einfach mal im Browser auf "Ansicht -> Quellcode anzeigen" gehen. Das zunächst Verwirrende ist, dass zwischen den Informationen noch etwas auftaucht.

Es gibt also nicht nur Informationen, sondern auch Anweisungen, wie diese Informationen angezeigt werden sollen. Das geschieht über HTML-TAGs.

# TAGs

Die eigentlichen **Befehle** werden in HTML durch sogenannte **TAGs** realisiert.

## Beispiel:

```
<b>fett darzustellender Text</b>
```

Dies bewirkt eine Ausgabe im HTML-Browser von

<b>fett darzustellender Text</b>
--------------------------------------

Dabei sind drei Dinge zu beachten:

1. Die TAGs sitzen immer zwischen Kleiner- und Größerzeichen < .. >
2. Zwischen den < > befindet sich die Anweisung - hier b - und wie so vieles im Leben ist das eine Abkürzung und das Ganze von dem englischen "bold" (fett). Dementsprechend sind alle Befehle nur Abkürzungen von englischen Begriffen. So lassen sich die Befehle gut merken (vorausgesetzt, man kann Englisch).
3. Alle HTML-TAGs immer klein schreiben. Auswirkungen werden zwar erst in der Zukunft sichtbar, aber wenn man sich etwas falsches angewöhnt, bekommt man das schlecht wieder los.
4. Alles hat ein Anfang und ein Ende. So auch praktisch fast jeder HTML-TAG.

## Anfangs- und End-TAGs

Weiterhin gibt es einen Anfangs-TAG und einen End-TAG. Sonst müsste unser Text mit obigem Beispiel ja immer fett sein. Der End-TAG ist fast identisch mit dem Anfangs-TAG, mit dem einzigen Unterschied, dass nach dem Kleinerzeichen (<) ein Schrägstrich kommt (siehe Beispiel).

## Grundgerüst / HTML-Seitenaufbau

Jede HTML-Seite hat ein Grundgerüst, das aus zwei Teilen besteht.

Die folgende Minimalversion funktioniert bereits, muss aber für den professionellen Bedarf noch erweitert werden. Das Grundgerüst (siehe Beispiel) ist zum Lernen und Testen aber ausreichend.

## Beispiel:

```
<html>
  <head>
    <title>
    </title>
  </head>

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

Die zwei Bereiche sind der Head-Bereich und der Body-Bereich. Im Head-Bereich werden alle Angaben gemacht, die für den normalen Surfer nicht im Ausgabefenster des Browsers sichtbar sind. Eine Ausgabe des obigen Codes führt zu einem leeren Browser-Bild. Im Body-Bereich stehen alle Texte, Bildverweise und HTML-TAGs, die der Surfer dann sieht.

**Zum sauberen Programmieren gehört das Einrücken. Dabei sollen entsprechende Anfangs- und Endtags untereinander stehen.**

## Body-Bereich

Alles was im Body-Bereich eingetragen wird, zeigt der Browser an. Zunächst machen wir alle Eintragungen zwischen `<body>` und `</body>`. Es soll die Ausgabe von einem Text erfolgen.

Gib dazu wie im folgenden Ausdruck den Code exakt ein (ohne Zeilennummer!):

### Beispiel:

```
1. <html>
2. <head>
3. <title>
4. </title>
5. </head>
6.
7. <body>
8. Hier kommt unser erster Test. Das klassische Beispiel,
9. das in jedem B u c h
10.
11. zu
12.
13. Programmierung
14.
15. z         u
16.
17. finden
18. ist!
19.
20. Hallo Welt.
21. </body>
22.
23. </html>
```

Was zeigt der Browser an?

Erstaunlich, oder? Alle Umbrüche sind beseitigt und mehrere Leerzeichen werden zu einem Leerzeichen zusammengefasst. Nicht wundern - denn dies ist in HTML völlig normal.