

Einführung in HTML

1. Einführung

HTML bedeutet **H**yper**T**ext **M**arkup **L**anguage und ist eine sogenannte *Auszeichnungssprache*. Mit ihr wird die Struktur (und das Aussehen) einer Webseite bestimmt.

Eine einzelne Webseite besteht dabei aus vielen HTML-Elementen bzw. -Blöcken. Diese werden in sogenannten *Tags* eingeschlossen.

Der HTML-Code wird dabei als reine Textdatei geschrieben und lediglich mit der Dateierdung `.html` abgespeichert.

2. Einführung Tags

Tags dienen dazu, von ihnen eingeschlossene Texte/Inhalte zu gruppieren und zu formatieren. Tags treten dabei meist paarweise auf: ein beginnendes Tag und ein abschließendes Tag.

Ein Tag erkennt man an den eckigen Klammern, beispielsweise `<html>`. Die abschließende Version davon wird mit einem Schrägstrich geschrieben: `</html>`.

Für jedes Tag können auch noch weitere Attribute angegeben werden. Diese werden dann auch im beginnenden Tag innerhalb der spitzen Klammern geschrieben, z. B.: `<html lang="de">`.

3. Grundgerüst einer HTML-Datei

Jede HTML-Datei muss folgendes Grundgerüst enthalten:

```
1 <!DOCTYPE html>
2
3 <html lang="de">
4
5 <head>
6     <title>HTML ist toll</title>
7 </head>
8
9 <body>
10     Hallo Welt!
11 </body>
12
13 </html>
```

Listing 1: Grundgerüst einer HTML-Datei

`<!DOCTYPE html>` (Zeile 1) Die Angabe sorgt dafür, dass eine Seite im Browser korrekt verarbeitet und angezeigt wird.

`<html>...</html>` (Zeile 3-13) Das HTML-Element ist das Grundelement. In diesem Element kann als Attribut noch die Sprache des Textes auf der Seite angegeben werden. Innerhalb dieses `html`-Tags werden dann die weiteren Blöcke geschrieben.

`<head>...</head>` (Zeile 5-7) Im `head`-Element stehen Informationen über die Webseite. Es enthält sozusagen den Vorspann, der allerdings im Browser nicht erscheint. Eines der wichtigsten Elemente im `head`-Element ist `title`.

`<title>...</title>` (Zeile 6) Das `title`-Element enthält die Bezeichnung der jeweiligen Webseite und wird im Browserfenster angezeigt.

`<body>...</body>` (Zeile 9-11) Nach dem `head`-Element, also dem Kopf, folgt das `body`-Element. Alles, was zwischen dem Start- und End-Tag des `body`-Elements steht, wird später als sichtbarer Inhalt im Browser angezeigt.

4. Aufgabe

- Erstelle in einem Texteditor (z. B. Windows-Editor/Notepad, **nicht Word!**) eine HTML-Datei mit dem Titel *Grundgerüst einer HTML-Datei* und dem Inhalt *Informatik*.
- Speichere die Datei als `index.html` ab. Achte dabei auf die Dateieindung!
- Öffne die Datei in einem Browser.

5. Aufgabe

Zur Formatierung von Text können nachfolgende Tags benutzt werden. Teste diese indem du um den Text in deiner Datei zusätzlich ein Tag einfügst und beschreibe kurz die sichtbare Funktion:

<code>...</code> :	Fett	„bold“
<code>...</code> :	Fett	
<code><i>...</i></code> :	Kursiv	„italic“
<code>...</code> :	Kursiv	„emphasize“
<code><u>...</u></code> :	Unterstrichen	„underline“
<code><h1>...</h1></code> :	} Überschrift	„headline“
<code><h2>...</h2></code> :		
<code><h3>...</h3></code> :		
<code><sup>...</sup></code> :	Hochgestellt	„superscript“
<code><sub>...</sub></code> :	Tiefgestellt	„subscript“
<code>...</code> :	Durchgestrichen	„delete“

6. Aufgabe

Versuche, einen Text mit einem Zeilenumbruch anzeigen zu lassen. Wie funktioniert das?

Platz	Vorname	
1	Sarah	Rent
2		

```
< table >
```

```
< tr >
```

```
< th> Platz < /th>
```

```
< th> Vorname < /th>
```

```
< /tr>
```

```
< tr >
```

```
< td> 1 < /td>
```

```
< td> Sarah < /td>
```

```
< td> Rent < /td>
```

```
< /tr>
```

```
< tr >
```

```
< td> 2 < /td>
```

```
< /tr>
```

```
< /table >
```

Tabellen in HTML

1. Bestandteile einer Tabelle

Eine Tabelle wird in HTML mit dem Tag `<table>...</table>` eingeschlossen. Jede Zeile wird dabei mit `<tr>...</tr>` umschlossen.

Für jede einzelne Zelle gibt es die Tags `<th>...</th>` (für Kopfzellen) bzw. `<td>...</td>` (für Inhaltzellen).

2. Aufgabe: Tabelle erstellen

Erstelle eine Webseite mit folgender Tabelle:

Augenfarbe	Braun	Blau	Grau	Grün	Summe
Haarfarbe					
Braun	119	84	54	29	286
Blond	7	94	10	16	127
Schwarz	68	20	15	5	108
Rot	26	17	14	14	71
Summe	220	215	93	64	592

3. Aufgabe: Tabellenattribute

Informiere dich, welche Attribute es für das `<table>`-Tag gibt, teste diese und beschreibe deren Funktion kurz:

[illegible]

4. Aufgabe: Abwechselnde Zeilenfarben

Zur besseren Übersicht werden die Zeilen einer Tabelle oft mit unterschiedlichen Grautönen hinterlegt. Wie lässt sich das erreichen? Ändere deine Tabelle entsprechend ab.

5. Aufgabe: Formatierte Tabelle erstellen

Erstelle eine Webseite mit folgender Tabelle:

Mannschaft	SP	S	U	N	T	GT	P
VfB Stuttgart	19	12	2	5	33	21	38
Hannover 96	19	10	5	4	33	23	35
Braunschweig	19	10	5	4	31	21	35
Union Berlin	19	9	5	5	28	21	35
Dynamo Dresden	19	8	7	4	27	21	31
SV Sandhausen	19	8	6	5	28	17	30
Heidenheim	19	8	5	6	28	19	29
Würzburger Kickers	19	7	7	5	34	33	28

Füge außerdem noch eine Überschrift **Tabelle der 1. Bundesliga** hinzu und eine kurze Beschreibung *Tabellenstand vom 10.2.2017*

6. Aufgabe: Verknüpfungen

Schön wäre es, wenn man bei einem Klick auf die Mannschaftsnamen direkt auf die Homepage der Mannschaft weitergeleitet würde. Informiere dich, wie man eine Verknüpfung (bzw. einen *Link*) erzeugen kann und verknüpfe die Mannschaftsnamen in der ersten Spalte mit der jeweiligen Homepage.

Mit welchem Tag ist das möglich:

Ein Nachteil der Links ist, dass sich die Mannschaftsseite in dem Fenster (bzw. in dem Tab) öffnet, in dem auch unsere Homepage offen ist. Oft wollen wir erreichen, dass bei einer Verknüpfung auf eine externe Seite diese in einem neuen Fenster/Tab geöffnet wird. Wie funktioniert das?

Bilder in HTML

1. Bilder einbinden

Um die Tabelle aus letzter Woche optisch etwas ansprechender zu gestalten fügen wir am Anfang jeder Zeile noch eine Spalte ein mit dem Logo des jeweiligen Vereins.

Hierzu musst du zunächst die Logos im gleichen Ordner wie deine HTML-Datei abspeichern. Anschließend kannst du mit dem ``-Tag das Bild einfügen:

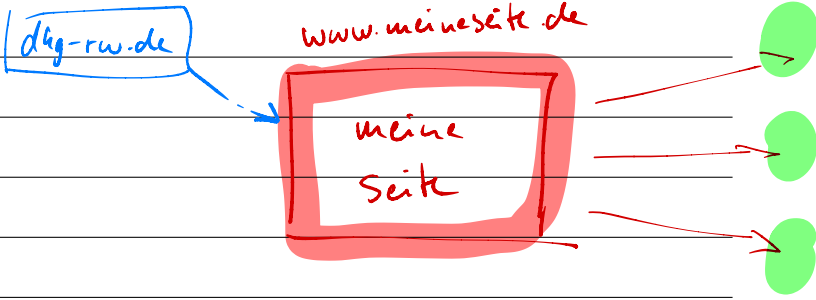
```

```

Listing 1: Einbinden von Bildern

Neben dem `src`-Attribut, bei welchem man die einzubindende Bilddatei angibt gibt es noch nachfolgende Attribute. Beschreibe deren Funktion:

`align`: Ausrichtung
`alt`: Alternativtext
`border`: Rahmen
`height`: Höhe
`width`: Breite

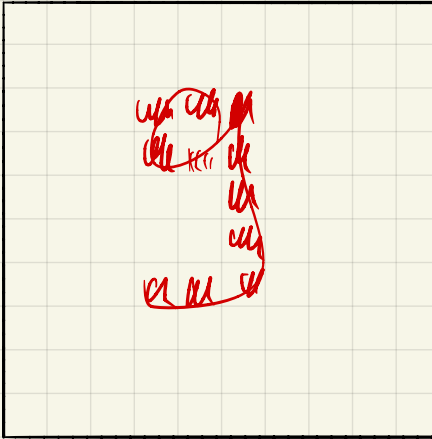


2. relative und absolute Ortsangabe

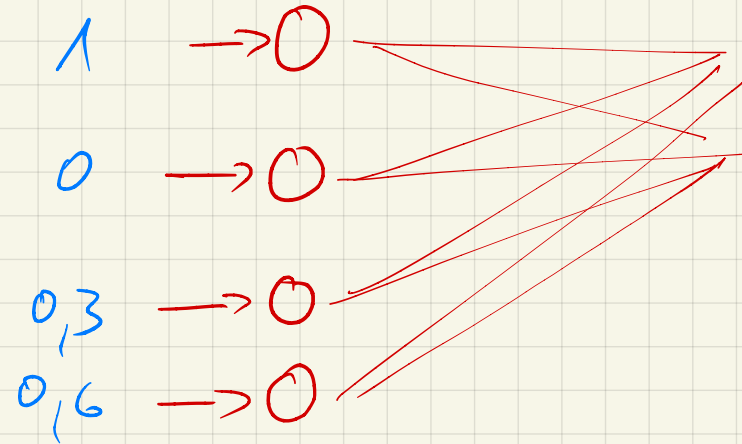
Ein wichtiger Punkt bei der Verlinkung und bei Einbindung von Bildern ist die Ortsangabe. Diese kann entweder *relativ* zur aktuellen HTML-Datei oder *absolut* angegeben werden.

Relative Pfadangaben werden meist eingesetzt, wenn eine andere Datei auf dem selben Computer bzw. Server geöffnet werden soll. Absolute Pfadangaben können hingegen dann eingesetzt werden, wenn eine externe Quelle geöffnet werden soll.

Wie funktionieren die relativen und absoluten Angaben?



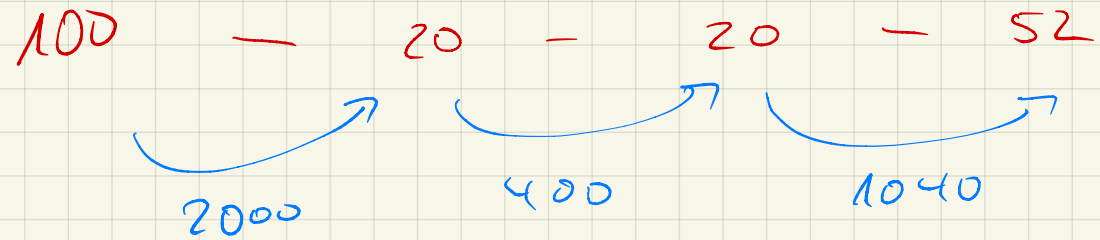
Eingabe



⋮

Ausgabe

- 0 0,41
- 1 0,02
- 2 0,17
- 3
- 4
- 5
- 6
- 7
- 8
- 9



+20
+20

+20
+20

2520

CSS

Einführung

CSS – Cascading Style Sheets – ist eine Layout- und Formatierungssprache, um strukturelle Inhalte wie HTML zu formatieren. Das Aussehen der semantisch deklarierten Abschnitte im HTML-Dokument wird damit genauer bestimmt, verändert und erweitert. Ziel davon ist es, die Struktur einer Webseite von deren Layout zu trennen.

Mit CSS geben wir sozusagen zentrale **Formatvorlagen** an, die dann einem Tag und dessen Inhalt zugewiesen werden können.

Diese Geben wir im `<head>...</head>` Bereich an:

```
<head>
  ...
  <style>
    /* hier werden die Formate definiert */
  </style>
</head>
```

Listing 1: Formatvorlagen mit CSS

1. Formatierung aller Überschriften `h1`

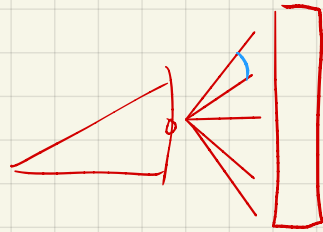
Formate können beispielweise für bestimmte Tags definiert werden. Hier sollen zunächst alle Überschriften `h1` formatiert werden. Dazu legen wir im `<styles>`-Bereich die Vorlage an:

```
<style>
  h1 {
    font-size: 30px;
  }
</style>
```

Listing 2: Vorlage für Überschriften

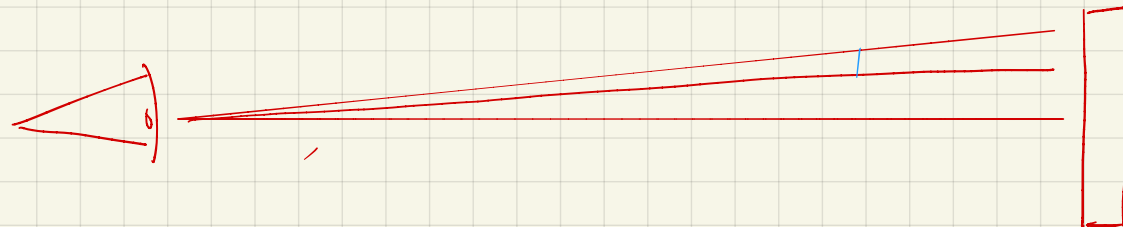
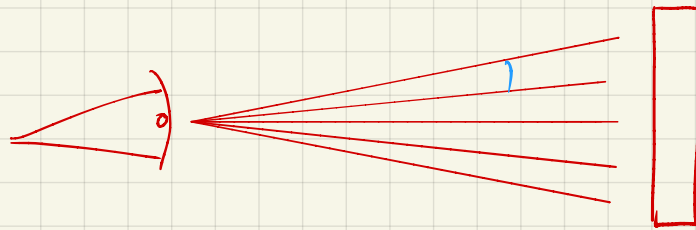
Ein Unterschied zur direkten Formatierung der Elemente ist, dass hierbei sämtliche Größenangaben mit einer Einheit angegeben werden müssen. Mögliche Einheiten sind:

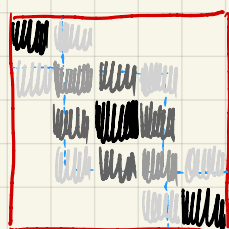
mm:	Millimeter
cm:	Zentimeter
px:	Pixel
pt:	Punkte
in:	1 inch = 2.54 cm
em:	Relativ zur Breite eines "m"
ex:	Relativ zur Breite eines "n"
%:	Relativ zur übergeordneten Größe



$$\tan^{-1}(0,0002) = 0,01^\circ$$

Diagram illustrating the geometry of the light source and screen. A light source on the left emits rays towards a vertical rectangular screen on the right. The distance from the source to the screen is labeled as 3m. The height of the screen is labeled as 1,20m. The angle of the rays is labeled as $0,01^\circ$.





Für die Formatierung von Text können (unter anderem) folgende Eigenschaften benutzt werden:

`font-size:` _____
`font-family:` _____
`font-variant:` _____
`font-size:` _____
`font-weight:` _____
`line-height:` _____
`color:` _____
`background-color:` _____
`letter-spacing:` _____
`text-decoration:` _____
`text-shadow:` _____
`text-align:` _____

Diese Eigenschaften sind zumeist selbsterklärend (bei ausreichenden Englischkenntnissen). Überlege dir zunächst, was diese Eigenschaften bewirken könnten und probiere diese anschließend aus.

2. Formatierung von Klassen

Neben der Formatierung aller `h1`-Tags können wir auch Formatklassen anlegen und diese dann nur bestimmten Elementen zuordnen. Formatklassen beginnen mit einem Punkt gefolgt von einem (fast) beliebigen Namen:

```
<style>
  .farbig {
    color: red;
  }
</style>
```

Listing 3: Formatklasse „farbig“

Diese Klasse muss allerdings dann noch einem Element zugewiesen werden:

```
<h1 class="farbig">Das ist eine farbige Überschrift</h1>
```

Listing 4: Formatklasse „farbig“ einem `h1`-Element zuweisen

Diese Klasse können wir natürlich auch mehreren Elementen zuordnen, mit dem Vorteil, dass man Änderungen dann nur noch an einer einzigen Stelle anpassen muss.

Aufgabe: lege eine Tabelle an (oder nimm die Tabelle von letztem Mal ohne direkte Formatierungen) und erzeuge damit abwechselnd eingefärbte Zeilen. Erstelle dazu eine Formatierungsklasse mit einer Hintergrundfarbe und weise diese Klasse dann jeder zweiten Zeile zu.