

HTML - Einführung

HTML ist die Abkürzung für „Hypertext Markup Language“. Es handelt sich dabei um eine plattformunabhängige Dokumentbeschreibungssprache, in der Web-Seiten geschrieben sind.

HTTP ist die Abkürzung für "Hypertext Transfer Protocol", also das Übertragungsverfahren mit dem Webseiten durch das Internet übermittelt werden.

HTTPS ist die Abkürzung für "Hypertext Transfer Protocol Secure". Es handelt sich um http mit einer Verschlüsselung.

TAG ist das englische Wort für Auszeichner oder Etikett. HTML und alle verwandten Sprachen (XHTML, XML, etc.) arbeiten mit tags in der Form:

```
<Anweisung> ... </Anweisung>
```

Dabei muss jedes tag <...> auch wieder mit </...> geschlossen werden (Start- und Endtag). In der Praxis, ist es bei manchen tags nicht notwendig diese wieder zu Schließen und es gibt einige Kurzschreibweisen. Theoretisch muss ein tag aber immer geschlossen werden.

Das HTML-Grundgerüst (Fettgedruckte Elemente)

```
<!DOCTYPE html>  
<html>
```

```
  <head>
```

```
    <title> Titel der Seite </title>
```

```
  </head>
```

```
  <body>
```

```
    // Innerhalb des Browserfensters sichtbarer Bereich
```

```
    //
```

```
    // Beispiel:
```

```
    <h3>Ich bin eine Überschrift</h3>
```

```
    <p>
```

```
      Ich bin ein Beispieltext mit <br>  
      zwei Zeilen.
```

```
    </p>
```

```
  </body>
```

```
</html>
```

IF9	HTML
-----	------

Basis-tags

<pre><h1> ... </h1> ... <h6> ... </h6></pre>	Überschrift-tags (H eading)
<pre><p> ... </p></pre>	Textabschnitt-tag (P aragraph)
<pre>
</pre>	Zeilenumbruch (L ine b reak)
<pre> sichtbarer Text </pre>	Hyperlink Attribut „href“ gibt das Ziel des Links an
<pre></pre>	Image-tag Attribut „src“ gibt die darzustellende Bilddatei an

Listen

<pre> ... </pre>	Unordered List
<pre><ol type="1"> ... </pre>	Ordered List Attribut „type“ gibt die Art der Nummerierung an. (z.B. 1, A, a, I, i)
<pre> ... </pre>	List Item (Eintrag in einer Liste)

Beispiel:

<pre> Kaffee Tee Milch </pre>	<pre><ol type="A"> Player1 Anon dc </pre>
---	---

Tabellen

<code><table border="0"></code> ... <code></table></code>	Tabellen-tag mit „border“-Attribut (Rahmen Ein/Aus)
<code><tr> ... </tr></code>	Eine Tabellenzeile (T able r ow)
<code><td> ... </td></code>	Eine Tabellenspalte (T able d ata)
<code><th> ... </th></code>	Eine Tabellenspaltenüberschrift (T able h eading)

Speziell bei den tags `<td>` und `<th>` kann das Attribut „colspan="x"“ eingefügt werden. Eine Spalte wird um die angegebene Anzahl x verbreitert und Zählt so als 2, 3, 4, ... Spalten.

Beispiel:

```
<table>
  <tr>
    <th>Nachname</th>
    <th>Vorname</th>
    <th>2. Vorname</th>
    <th>Telefonnummer</th>
  </tr>
  <tr>
    <td>Dummy</td>
    <td>Crash</td>
    <td>Test</td>
    <td>1337</td>
  </tr>
  <tr>
    <td>Testaburger</td>
    <td colspan="2">Wendy</td> //die Zellen Vorname und 2.Vorname
    <td>0815</td> //werden zu einer Zelle verbunden
  </tr>
</table>
```

Formularelemente

<pre><form action="mailto:beispiel@mail.de"> ... </form></pre>	Formular-tag
--	--------------

<pre><input type="text" value="wert" name="test"></pre>	einfaches Textfeld
---	--------------------

<pre><input type="password" value="wert" name="test"></pre>	Passwort Textfeld
---	-------------------

<pre><input type="submit" value="text" name="test"></pre>	Senden Knopf
---	--------------

<pre><input type="reset" value="text" name="test"></pre>	Reset Knopf
--	-------------

<pre><input type="radio" name="auswahl" value="wert1"> <input type="radio" name="auswahl" value="wert2"></pre>	Radio Button
--	--------------

Die Besonderheit von Formularelementen in Form des <input>-tags ist, dass sie sich durch das „type“-Attribut unterscheiden und es sich nicht um verschiedene tags handelt.

<pre><select name="test"> <option value="elem1">1.Eintrag</option> <option value="elem2">2.Eintrag</option> <option value="elem3">3.Eintrag</option> </select></pre>	Drop-down Liste (Ausklappbare Liste)
--	---

<pre><textarea name="textbereich" rows="10" cols="30"> Beispieltext </textarea></pre>	Textbereich
---	-------------

Div-tag

```
<div> ... </div>
```

Beim Design von Webseiten hat das div-tag eine besondere Rolle. Das div-Element ist dazu gedacht, mehrere Elemente wie Text, Grafiken, Tabellen usw., in einen gemeinsamen Bereich einzuschließen. Ansonsten hat es keine Eigenschaften. Es ist dazu gedacht, Bereiche zu erzeugen, die mit Hilfe von CSS formatiert werden können. Div steht für **division** (Bereich).

id-Attribut (Identifikationsattribut)

Es ist möglich, jedem Html-tag eine id zuzuweisen. Diese id muss einzigartig sein. Hat man einem Element mittels id-Attribut eine id gegeben, kann man es mittels CSS oder JavaScript ansprechen und anpassen.

Beispiel:

```
<table id="tabelle01">
  <tr>
    <td id="dieZelleInDerEinTestTextSteht">Testtext</td>
  </tr>
</table>
```

class-Attribut

Das class-Attribut ermöglicht wie das id-Attribut den Zugriff auf bestimmte Elemente. Anders als id, können mehrere Elemente das gleiche class-Attribut besitzen.

Beispiel:

```
<div class="textFormat01">
  lorem ipsum dolor sit amet <br>
  <a href="https://duckduckgo.com" class="textFormat01">Suchmaschine</a>
</div>
```

CSS (Cascading Style Sheets)

CSS ist eine Sprache innerhalb von HTML, die das Design bestimmt. Meist wird CSS in eine externe Datei ausgelagert. Diese Datei wird so im Head einer HTML-Datei eingebunden:

```
...
<head>
  <link rel="stylesheet" type="text/css" href="/meineDatei.css">
  ...
</head>
...
```

Eine CSS-Datei kann beispielsweise so aussehen:

```
body {
  background-color: #060606;
}

h1 {
  color: #d8da3d;
  font-style: normal;
  font-size: 14px;
}

#dieZelleInDerEinTestTextSteht {
  color: #10563d;
}

. textFormat01 {
  border-style: solid;
  border-width: 5px;
}
```

Mit dem „Selektor“ vor der geschweiften Klammer, werden die gewünschten Elemente ausgewählt. h1 wählt z.B. alle <h1> - tags aus.

Für ein „id“-Attribut muss eine # vor dem Selektor stehen.

Für ein „class“-Attribut muss ein . vor dem Selektor stehen.

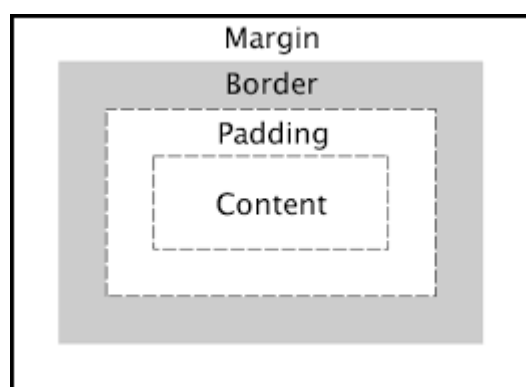
Innerhalb der Klammern können die ausgewählten Elemente dann formatiert werden.

Allgemein sieht ein CSS-Befehl folgendermaßen aus:

```
Selektor {
  Eigenschaft1: Wert;
  Eigenschaft2: Wert;
  ...
}
```

CSS (Cascading Style Sheets)

Farben	background-color: #000000; (#RRGGBB) color: #000000; (#RotGrünBlau)
Rahmen	border-style: solid; (double / inset / outset / hidden) border-color: #000000; border-width: 5px;
Außenabstand	margin-top: 100px; margin-bottom: 100px; margin-right: 150px; margin-left: 80px;
Innenabstand	padding-top: 50px; padding-right: 30px; padding-bottom: 50px; padding-left: 80px;
Höhe/Breite	height: 200px; width: 50%;
Schrift	font-family: "Arial"; (Verdana / Courier New) font-size: 40px; font-weight: normal; (bold)
Position	position: absolute; (static / fixed / relative) top: 10px; (bottom) left: 50px; (right) z-index: -1;



Registrieren

Geburtstag

 Tag Monat Jahr Weiblich Männlich

		Mo	Di	Mi	Do	Fr	
1	7.45 - 8.30	Englisch	Biologie	Deutsch	Musik	Mathematik	
	8.35 - 9.20						
3	9.40 - 10.25	Klassenleiterstunde	Kath./Ev. Religion	Prakt. Philos.	Sport	Lions Quest	Deutsch
4	10.30 - 11.15	Lernzeit				Englisch	Sport
5	11.25 - 12.10	Deutsch	Mathematik	Lernzeit	Mathematik	Politik	
6	12.15 - 13.00	M i t t a g s p a u s e					
7	13.05 - 13.50	Kunst	Lernzeit	Englisch	Lernzeit		
8	13.55 - 14.40		Erdkunde	Physik	AG		
9	14.45 - 15.30	Sport					

JavaScript / DOM (Document Object Model)

```

...
<head>
  <link rel="stylesheet" type="text/css" href="./meineDatei.css">
  <script type="text/javascript" src="./meineDatei.js"></script>
  ...
</head>
...

```

var meineVariable = *Element / Zahl / Text* ;

```

function meineFunction()      {
  //   Quelltext der function
  //
  //   z.B.:
  //   alert("Hallo Welt");
}

```

document.getElementById("meineDivBoxId") ermöglicht Zugriff auf ein Html Element

element.addEventListener(event, function, false) hängt ein Ereignis an ein Element

Liste der Maus-Events: **click, dblclick, mouseover, mouseleave, mouseenter**
 Liste der Tastatur-Events: **keydown, keypress, keyup**

Kontrollstrukturen:

```

if (Bedingung) {
  // dieser Bereich wird ausgeführt, wenn die Bedingung erfüllt ist
} else {
  //dieser Bereich wird ausgeführt, wenn die Bedingung nicht erfüllt ist
}

```