

HTML und CSS

HTML ist eine Sprache zur **Strukturierung** von Daten.

HTML = **H**ypertext **M**arkup **L**anguage

Hypertext bedeutet ganz allgemein, dass es Referenzen innerhalb eines Textes oder zwischen verschiedenen Texten gibt, die es erlauben, einen Text nicht ausschließlich linear von vorne bis hinten zu lesen. Es existieren **Referenzen** von einer Textstelle zu einer anderen. Als Hypertextsystem kann man schon das Inhaltsverzeichnis eines Buchs oder Fuß- und Endnoten betrachten.

Markup Language (Auszeichnungssprache) bedeutet, dass in einen Text maschinenlesbare **Elemente** eingefügt werden, die die Struktur des Textes abbilden. In HTML sind die maschinenlesbaren Elemente dadurch gekennzeichnet, dass sie von spitzen Klammern begrenzt werden.

CSS ist ein Standard zur Beschreibung der **Darstellung** von HTML-Dokumenten.

Beide Standards werden betreut vom **w3c** (w3c.org) und **WHATWG** (whatwg.org); Der aktuelle HTML-Standard ist **HTML5**, der aktuelle CSS-Standard ist **CSS3**.

HTML-Syntax

Als **Syntax** bezeichnet man die Gesamtheit der Regeln, nach denen man Anweisungen in einer Programmier-, Skript- oder Auszeichnungssprache zusammensetzen kann.

HTML besteht aus einer Reihe von vordefinierten Elementen. (Im Ggs. zu XML, wo man die Elemente selbst definieren kann). HTML lässt nur in ganz geringem Umfang eigene Erweiterungen zu, nämlich mit **data-** zusammengesetzte Attributnamen.

Die Gesamtheit der HTML-Elemente und aller Inhalte eines Dokuments bezeichnet man als **Quelltext**.

Die Elemente sind in **hierarchisch** angeordnet. Das heißt, es gibt ein Element, das alle anderen Elemente einschließt, das ist das **Root-Element**. Das Root-Element hat ein oder mehrere **Child-Elemente**, und diese Child-Elemente können ihrerseits wieder Child-Elemente haben. Die Elemente müssen konsequent verschachtelt sein. Daraus ergibt sich eine in mehreren Ebenen aufgebaute Struktur, die auch als **DOM (Document Object Model)** bezeichnet wird.

In HTML heißt das Root-Element **html** und hat genau zwei Child-Elemente, nämlich **head** und **body**.

head ist für Meta-Informationen und eventuell für **style-** und **script-**Elemente da, **body** für die Inhalte, die für die User von Interesse sind.

Die ganze Elementstruktur bezeichnet man auch als **DOM (Document Object Model)**

Die **Elemente** in HTML bestehen aus **Tags**.

Tags haben einen Namen, z. B. „**p**“ oder „**a**“ oder „**h1**“ oder „**div**“.

Die meisten Elemente bestehen aus einem **öffnenden Tag**, einem **schließenden Tag** und dem **Inhalt**. Im schließenden Tag steht vor dem Tag-Namen ein **Schrägstrich**.

Einige Tags sind leer, zum Beispiel **img** oder **br**.

```
<p>Mein Hut, der hat drei Ecken.</p>

```

Das Element ist im ersten Fall **p** (**paragraph**), der Inhalt ist der Text zwischen den beiden Tags, im zweiten Fall **img** und das Bild, das im Browser angezeigt wird wird mit dem Wert des Attributs **src** referenziert.

Elemente kann man verschiedenen Gruppen zuordnen:

Inline-Elemente: Das sind solche Elemente, die sich wie Text auf der Zeile verhalten. Dazu gehören u. a. **img**, **a**, **em** oder **code**.

Block-Level-Elemente: Das sind solche Elemente, die ohne Rücksicht auf den Inhalt die ganze Breite des Viewports einnehmen und so hoch sind, wie der Inhalt es verlangt. Dazu gehören u. a. **h1**, **p**, **div**, oder **header**.

Semantische Elemente (semantic elements): Das sind Elemente, deren Tagnamen etwas über ihre Funktion im Dokument oder im Layout aussagen, z. B. **header**, **footer** oder **nav**.

Ersetzte Elemente (replaced elements): Das sind Elemente, die keinen Text strukturieren, sondern die durch eine graphische Repräsentation ersetzt werden. Dazu gehören **img** oder **input**.

Man betrachtet das Element als **Elementknoten (element node)** im **DOM (Document Object Model)** und den Text als **Textknoten (text node)**.

Elemente können **Attribute (Attributknoten/attribute nodes)** haben. Attribute stehen innerhalb des Start-Tags.

```
<div class="warning">Wenn du den Rubikon überschreitest...</div>
<h1 lang="de" class="x-beliebig">Erstes Kapitel</h1>
```

Ein Attribut(knoten) besteht aus **Namen** und **Wert**, bzw. stellt ein **Name-Wert-Paar (name-value pair)** dar. Der **div** im Beispiel hat ein **class**-Attribut mit dem Wert „warning“, das **h1**-element hat ein **lang**-Attribut mit dem Wert „de“ und ein **class**-Attribut mit dem Wert „x-beliebig“, Einige wenige Attribute können wahlweise ohne Wert oder mit den booleschen Literalen **true** und **false** verwendet werden.

Manche Attribute lassen nur vordefinierte Werte zu (wie z. B. **lang**). Das Attribut **lang** erlaubt nur die zweistelligen ISO 639-1 Sprachcodes. Andere Attribute (wie **class**) erlauben (fast) beliebige Zeichenketten als Wert.

Zwischen dem Namen eines Attributs und seinem Wert steht ein Ist-gleich-Zeichen.

Der Wert eines Attributs wird in der Regel von Anführungszeichen begrenzt; möglich sind doppelte und einfache Anführungszeichen, man muss aber wenigstens einfache oder doppelte Anführungszeichen paarweise verwenden. Man kann die Anführungszeichen auch weglassen, wenn man die Zeichen **<**, **>**, **"**, **'**, **=** oder Leerschrittzeichen nicht verwendet.

In einen HTML-Quelltext können **Kommentare** eingefügt werden.

```
<!--  
  Dieses Element ist nur auf großen und mittelgroßen  
  Monitoren zu sehen. Auf kleinen Monitoren wird die  
  Eigenschaft display auf none gesetzt.  
-->
```

Alles, was zwischen **<!--** und **-->** steht wird vom Browser ignoriert (Eine Ausnahme hiervon würden **bedingte Kommentare** bzw. **conditional comments** darstellen. Die kann man inzwischen aber als obsolet betrachten).

Kommentare können nicht verschachtelt werden!

CSS-Syntax

CSS definiert für jedes HTML-Element eine Reihe von **Properties**, die jeweils eine bestimmte Gruppe von **Werten** haben kann. So hat jedes **div**-, **p**-, **h1**- oder auch das **body**-Element u. a. die Properties **font-family**, **font-size**, **color** und **background-color**.

font-family kann als Wert den Namen einer bestimmten **Schriftart (font)** wie **Arial**, **Times New Roman** oder **Verdana**, aber auch eine **generische Schriftart (generic font)** wie **cursive**, **monospace** oder **sans serif** haben.

font-size kann als Wert einen **absoluten** oder einen **relativen Schriftgrad** haben: **16px** (absolut) oder **1.5em** (relativ).

background-color ist die **Hintergrundfarbe** eines Elements und kann als Wert einen der **HTML-Farbnamen**, eine **RGB-Angabe** in hexadezimaler Schreibweise (z. B. **#fa23ad**) oder eine Angabe in RGB, **HSL** in dezimaler oder Prozent-Schreibweise haben.

color ist die **Vordergrundfarbe** (bei Elementen, die Text als Inhalt haben mithin die Textfarbe) und kann die gleichen Werte haben wie **background-color**:

Eine Property die numerische Angaben als Wert hat kann in relativen oder absoluten Größen angegeben werden:

```
font-size: 1.5em /* relativ zum Standard-Schriftgrad */
width: 60% /* relativ zum Parent-Container */
margin-left: 7px /* absolut in Pixel */
width: 700mm /* absolut in Millimeter */
```

Als Dezimalzeichen wird nicht, wie in Europa üblich, ein Komma verwendet, sondern wie im Angelsächsischen Raum üblich, ein Punkt!

Kommentare werden begrenzt durch `/*` und `*/`.

Styles einbinden

Um Styles auf HTML-Elemente anzuwenden gibt es 3 Möglichkeiten:

1. Inline-Styles

Inline-Styles sollte man möglichst nicht verwenden. Wenn es aber keine andere Möglichkeit gibt, dann kann man im HTML-Element ein `style`-Attribut mit einer oder mehreren Deklarationen als Wert verwenden.

```
<h1 style="color:cornflowerblue; background:wheat; font-size:2em">
  Die Geschichte vom Rotkäppchen und dem bösen Wolf
</h1>
```

Die einzelnen Deklarationen sind dann durch ein Semikolon separiert, und die ganze Liste muss zwischen Anführungszeichen stehen.

2. Interne Stylesheets

Interne Stylesheets stehen innerhalb des HTML-Dokuments zwischen `<style>` und `</style>`. Das `style`-Tag steht zusammen mit den Metatags im HTML-Header.

```

<html>
  <head>
    <meta charset="utf8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" >
    <meta name="keywords" content="sonne, warm">
    <title>mein erstes html-dokument</title>
    <style>
      h1 {
        background-color: wheat;
        color: cornflowerblue;
      }
    </style>
  </head>
  <body>
    <h1>Ein angenehmer tag im januar</h1>
    <ul>
      <li>eier</li>
      <li>milch</li>
    </ul>
    für januar ist es heute angenehm warm.
  </body>
</html>

```

3. Externe Stylesheets

Externe Stylesheets sind separate Dateien und werden mit dem Element **link** eingebunden:

```

<link rel="stylesheet"
      href="css/stylesheet.css" type="text/css">
<link
      href="https://fonts.googleapis.com/
              icon?family=Material+Icons"
      rel="stylesheet">

```

Das Attribut **href** ist obligatorisch. Externe Stylesheets können entweder auf der gleichen Website gehostet werden wie im ersten Beispiel oder von einer anderen Website geladen werden wie im zweiten.

Wenn das Stylesheet lokal gehostet wird, dann muss die Datei die Dateinamenserweiterung **.css** haben. Wenn die Datei in der Webroot liegt, dann ist der Wert des Attributs **href** der Dateiname, ansonsten der Pfad relativ zur Webroot + Dateiname. im ersten Beispiel liegt die Stylesheetdatei in einem Unterordner der Webroot, der **css** heißt.

Deklarationen

Eine Deklaration ist ein Name-Wert-Paar:

Der Name einer Property, gefolgt von einem Doppelpunkt, gefolgt von einem passenden Wert.

```
position: relative;
left: 100px;
top: 50px;
font-family: verdana, helvetica, sans-serif;
font-size: 1.2em
border-left: 1px solid red;
```

Die Property **position** hat eine **Zeichenfolge** als Wert. Wenn der Wert eine Zeichenfolge ist, gibt es normalerweise eine vordefinierte Liste mit zulässigen Werten. Im Fall von position sind das: **static**, **absolute**, **relative**, **fixed** und **sticky**.

Die Property left bestimmt den Abstand (je nachdem welchen Wert **position** hat) vom linken Rand des Viewports, vom Parent-Element oder von der Position im statischen Elementfluss. **top** bestimmt den Abstand von oben.

font-family hat eine Zeichenfolge oder eine Komma-separierte Liste von Werten als Wert. „Verdana“ ist ein **spezifischer Font** (eine **spezifische Schriftart**). „Helvetica“ ist eine Gruppe von spezifischen Fonts, die sich durch ähnliche Eigenschaften auszeichnen, „sans-serif“ ist eine **generische Schriftart (generic font)**. Alle Fonts, die zu einer generischen Schriftart gehören, haben ebenfalls ähnliche Eigenschaften. Da man sich nicht darauf verlassen kann, dass eine ganz bestimmte Schriftart auf jedem Rechner verfügbar ist, ist es üblich zuerst eine spezifische Schrift anzugeben, dann eine Gruppe von ähnlichen Schriftarten und zuletzt einen generischen Font wie **cursive** oder **sans-serif**.

font-size kann als Wert entweder absolute oder relative Angaben haben. Absolute Angaben sind numerische Werte mit einer Einheit, z. B. **mm** (Millimeter), **pt** (Punkt) oder **px** (Pixel)

Selektoren

Ein Style besteht aus einer oder mehreren Deklarationen.

Ein Stylesheet besteht aus einer oder aus mehreren **Regeln**.

Eine Regel besteht aus einem **Selektor** und einer oder mehreren **Deklarationen** in geschweiften Klammern.

```
h1 {
    /* selektor */
    font-family: verdana, sans-serif; /* Deklaration */
    font-size: 18pt;                 /* Deklaration */
    background: wheat;
    color: cornflowerblue;
}
```

Mit den Selektoren werden bestimmte Elemente oder Gruppen von Elementen referenziert, auf die die betreffenden Regeln angewendet werden sollen.

Element-Selektoren entsprechen dem Namen eines Elements. also etwa h2, footer, oder a. Eine Regel mit einem solchen Selektor spricht alle Elemente im Dokument mit dem gleichen Elementnamen an.

Class-Selektoren beginnen mit `.` und sprechen alle Elemente im Dokument mit dem gleichen `class`-Attribut an:

```
...
<style>
  .warning {
    border: 3px solid red;
  }
</style>
...
<h1 class="warning">Vorsicht!</h1>
...
<p class="warning">
  Es darf keine zwei gleichen ID-Attribute in einem HTML-
  Dokument geben!
</p>
...
```

Ein `class`-Attribut kann mehrere Werte haben:

```
<p class="hide-small hide-medium"> ... </p>
```

Ein ID-Selektor beginnt mit `#` und spricht ein ganz bestimmtes Element mit dem entsprechenden `id`-Attribut an.

```
...

#sidebar {
  width: 30%;
  float: left;
}

...

<div id="sidebar">
  ...
</div>

...
```

Pseudo-Klassen- oder Pseudo-Element-Selektoren können mit Element-Selektoren und Klassen-Selektoren kombiniert werden, um unterschiedliche Zustände eines Elements oder von Elementen mit einem bestimmten Klassen-Attribut anzusprechen, die es so nicht gibt. Z. B gibt es eine Pseudoklasse `:hover`, das mit einem Element wie `a` oder `div` kombiniert werden kann, damit es darauf reagiert, wenn der Mauszeiger darüber bewegt wird:

```
a {
  text-decoration: none;
  color: navy;
}
a:hover {
  color: purple;
}
```

```
<html>

<head>

  <meta charset="utf8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0" >

  <meta name="keywords" content="sonne, warm">

  <title>mein erstes html-dokument</title>

</head>

<body>

  <h1>Ein angenehmer tag im januar</h1>

  <ul>

    <li>eier</li>

    <li>milch</li>

  </ul>

  für januar ist es heute angenehm warm.

</body>

</html>
```

Ein besonders wichtiges element in html ist a. a steht für anchor und bedeutet "hyperlink". ein hyperlink braucht mindestens das attribut href. hyperlink ist ein inline-element.

es gibt mehrere möglichkeiten, hyperlinks zu benutzen:

- externer link: ein anderes dokument von einer anderen website wird im browsertab angezeigt

- ein anderes dokument von der gleichen website wird geladen

- sprungmarke innerhalb des gleichen dokumentes. das sprungziel muss ein html-element mit

einem id-attribut sein.