

INP (2)

Prof. Dr.-Ing. S. Gössner

University of Applied Sciences Lippe & Höxter

Inhalt

- [INP \(2\)](#)
- [Inhalt](#)
- [HTML - Einführung](#)
- [Textauszeichnung](#)
- [Syntax](#)
- [Grundgerüst](#)
- [Beispiel](#)
- [Sonderzeichen & Zeichensätze](#)
- [HTML Module](#)
- [Überschriften \(headings\)](#)
- [Absätze \(paragraph\)](#)
- [vorformatierter Text](#)
- [Verweise \(links\)](#)
- [Listen](#)
- [Tabellen](#)
- [Grafiken](#)
- [Formularelemente](#)
- [CSS in 5 Min.](#)
- [Verweise](#)

HTML - Einführung

HTML ist ...

- die **H**ypertext **M**arkup **L**anguage
- von Web-Gründer [Tim Berners-Lee](#) 1990 am CERN entwickelt worden
- standardisiert ([W3C](#))
- die "lingua franca" des Web
- als Klartextformat plattform und softwareunabhängig
- mit der Version 4.01 am Ende der Entwicklung angelangt
- mit anderen Sprachen erweiterbar (CSS, Javascript).

Textauszeichnung

Ein *HTML*-Dokument verwendet sog. Auszeichnungen, um bestimmten Textabschnitten eine besondere Bedeutung zu geben.

```
[Überschrift]Zwei Frösche[Ende Überschrift]
[Paragraph]Auf der Insel Hondo in Japan lebten zwei Frösche.
Einer hauste in einem Graben außerhalb der Stadt Osaka,
während - der andere in einem kristallklaren Strom wohnte
nahe bei Kioto.[Ende Paragraph]
```

Derartig ausgezeichnete Text (*Markup*) wird von einem sog. *Parser* gelesen, interpretiert und auf syntaktische Korrektheit geprüft. Die Textauszeichnungselemente in HTML werden *Tags* genannt und sind in *spitze Klammern* eingeschlossen.

```
<h1>Zwei Frösche</h1>
<p>Auf der Insel Hondo in Japan lebten zwei Frösche.
Einer hauste in einem Graben außerhalb der Stadt Osaka,
während - der andere in einem kristallklaren Strom wohnte
nahe bei Kioto.</p>
```

Syntax

Ein *HTML*-Dokument besteht aus *Elementen*. Die allgemeine Syntax eines HTML-Elements lautet:

```
<element attr1="wert1" ... attrN="wertN">Inhalt</element>
```

Ein Element besteht aus:

- dem *Start-Tag* `<element . . .>`, das von einem spitzen Klammerpaar (`<>`) eingeschlossen ist und einen eindeutigen *Elementnamen* besitzt.
- den *Attributen* `attrI="wertI"`, die durch Leerzeichen voneinander getrennt sind und deren Reihenfolge irrelevant ist. Jedes *Attribut* setzt sich zusammen aus
 - dem *Attributnamen*, der innerhalb des jeweiligen Tags eindeutig sein muss.
 - dem '='-Zeichen
 - dem *Attributwert*, der von Anführungszeichen (") eingeschlossen ist.
- dem *Inhalt* des Elements, der aus Text besteht und/oder weitere HTML-Elementen einschliesst.
- dem *Ende-Tag*, ebenfalls eingeschlossen von einem spitzen Klammerpaar (`<>`) und einem dem *Elementnamen* vorangestellten Vorwärtsschrägstrich (*Slash*) `'/'`.

Inhaltslose *HTML*-Elemente können auf das Endetag verzichten und stattdessen die folgende Schreibweise verwenden.

```
<element attr1="wert1" ... attrN="wertN" />
```

Ein Element kann ein Anderes *vollständig* beinhalten.

```
<x> abc <y> def </y> ghi </x>
```

Überlappende Elemente sind nicht zulässig.

```
<x> abc <y> def </x> ghi </y>
```

Grundgerüst

Das Grundgerüst eines *HTML*-Dokuments

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional  
<html>  
  <head>  
    <title>Dokument Titel</title>  
  </head>  
  <body>  
    <!-- sichtbarer Inhalt des Dokuments -->  
  </body>  
</html>
```

- beginnt mit einer Definition des Dokumenttyps (*DTD*). Mittels dieser wird die Menge der zugelassenen Elemente einschliesslich der verwendbaren Attribute festgelegt.
- beinhaltet ein äusseres Element `html`, das sog. *Wurzelement* (*root*). Ausserhalb dessen dürfen offensichtlich keine Element notiert werden.
- besitzt einen Kopfbereich `head`, in dem nicht sichtbare Angaben zum Dokument stehen.
 - das Titelement `title` muss angegeben werden und wird
 - in der Titelzeile des Browsers angezeigt.
 - bei Verweisen, Lesezeichen (*bookmarks*) und Reitern (*tabs*) verwendet.
 - von Suchmaschinen ausgewertet.
 - Metaelemente `meta` machen üblicherweise Angaben hinsichtlich Autor, Erstellungsdatum, Schlagwörter und den verwendeten Zeichensatz.
 - Verweise `link`'s stellen logische Beziehungen zu anderen Dokumenten und Hilfsdateien her.
 - Skriptelemente `script` werden entweder mit vollständigem Inhalt oder als externe Verweise notiert.
- besitzt einen Rumpfbereich `body`, in dem der sichtbare Inhalt des Dokuments zusammengefasst ist.

Beispiel

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional  
<html>  
  <head>  
    <title>Zwei Frösche</title>  
    <meta http-equiv="content-type" content="text/html; c  
    <meta name="author" content="anonymous"/>  
    <meta name="created" content="2006-03-29"/>  
    <link rel="stylesheet" type="text/css" href="/css/ma  
  </head>  
  <body>  
    <h1>Zwei Frösche</h1>  
    <p>Auf der Insel Hondo in Japan lebten zwei Frösche.  
    Einer hauste in einem Graben außerhalb der Stadt O  
    während - der andere in einem kristallklaren Strom  
    nahe bei Kioto.</p>  
  </body>  
</html>
```



Sonderzeichen & Zeichensätze

Zeichensätze

Heutige Browser arbeiten mit einer Zeichencodierung, die auf *Übersetzungstabellen* zugreift. Die Menge der Zeichen in einer Tabelle wird *Zeichensatz* genannt.

- ASCII (128 Basiszeichen)
- ISO 8859-1 (Latin-1) ... *meist voreingestellt.*
- UTF-8 (Unicode)

Im Kopfbereich des HTML-Dokuments kann der zu verwendende Zeichensatz eingestellt werden.

```
<head>  
  <meta http-equiv="content-type" content="text/html; char<br></head>
```

Sonderzeichen

Spezielle Zeichen, die

- nicht im aktuellen Zeichensatz vorhanden sind
- auf der Tastatur keine Entsprechung haben

können mittels folgender unterschiedlicher Notation

- `&name;` (`€`)
- `&#dezimalzahl;` (`€`)
- `&#xhexadezimalzahl;` (`€`)

Deutsche Umlaute sind in beiden Zeichensätzen enthalten, so dass auf deren früher übliche Darstellung (`ä` ...) heute verzichtet werden kann.

Hier ist eine [HTML-Zeichenreferenz](#).

spezielle HTML-Zeichen

HTML-Parser behandeln die Zeichen (`<` `>` `&`) als spezielle, Markup beschreibende Symbole. Wenn nun genau jene Zeichen innerhalb eines Textes verwendet werden, sind sie als

- `<`; ... (<)
- `>`; ... (>)
- `&`; ... (&)
- `"`; ... ("")

zu schreiben.

HTML Module

Strukturmodul

html, head, body, title

Textmodul

abbr, acronym, address, blockquote, br, cite, code, dfn, div, em, h1, h2, h3, h4, h5, h6, kbd, p, pre, q, samp, span, strong, var

Hypertextmodul

a

Listenmodul

dl, dt, dd, ol, ul, li

Formularmodul

button, fieldset, form, input, label, legend, select, optgroup, option, textarea

Tabellenmodul

caption, col, colgroup, table, tbody, td, tfoot, th, thead, tr

Bildmodul

img

Skriptmodul

noscript, script

weitere Module:

Object, Presentation, Edit, Bidirectional Text, Image Map, Events, Metainformation, Stylesheet, Link, Base

(siehe [XHTML 1.1](#))

Überschriften (headings)

```
<h1>Überschrift</h1>
```

```
<h2>Überschrift</h2>
```

```
<h3>Überschrift</h3>
```

```
<h4>Überschrift</h4>
```

```
<h5>Überschrift</h5>
```

```
<h6>Überschrift</h6>
```

```
<h3 align="center">zentrierte Überschrift</h3>
```

Überschrift

Überschrift

Überschrift

Überschrift

Überschrift

zentrierte Überschrift

vorformatierter Text

```
<pre>
|  x  |  y  |
|=====|=====|
| 1.0 | 2.4 |
| 2.0 | 6.3 |
|   : |   : |
| 8.0 |98.7 |
|=====|=====|
</pre>
```

x	y
1.0	2.4
2.0	6.3
:	:
8.0	98.7

Verweise (links)

Verweise sind ein **primäres Merkmal** üblicher Webdokumente. Links können auf

- externe Webdokumente (*entfernte Webserver, andere Domain*)
- interne Webdokumente (*dieselbe Domain*)
- lokale Stellen im aktuellen Quelldokument

verweisen.

externe Verweise

```
<a href="http://google.de">Suche</a>  
<a href="http://google.de" target="_blank">Suche im neuer
```

[Suche](#)

[Suche im neuen Fenster](#)

interne Verweise

```
<a href="/learn/inp/">INP</a>  
<a href="../lec01/">Suche im neuen Fenster</a>
```

[INP](#)

[Suche im neuen Fenster](#)

lokale Verweise

```
<a href="#mark">zur Markierung</a>  
...  
<p id="mark">Paragraph, auf dessen Beginn mittels id="ma  
werden kann.</p>  
<a href="../lec01/#e12">Folie 12, letzte Vorlesung</a>
```

[zur Markierung](#)

...

Paragraph, auf dessen Beginn mittels id="mark" verwiesen werden kann.

[Folie 12, letzte Vorlesung](#)

Listen

HTML unterstützt drei verschiedene Arten von Listen

- Aufzählungslisten
- nummerierte Listen
- Definitionslisten

Aufzählungsliste (*unordered list*)

```
<ul>
  <li>eins</li>
  <li>zwei</li>
</ul>
```

- eins
- zwei

Mit dem Attribut `type=" [circle|square|disc] "` kann der *Bullet*-Type festgelegt werden.

nummerierte Liste (*ordered list*)

```
<ol>
  <li>eins</li>
  <li>zwei</li>
</ol>
```

1. eins
2. zwei

Mit dem Attribut `type=" [a|A|i|I] "` kann eine alphabetische oder römische Nummerierung jeweils *klein/gross* festgelegt werden. Das Attribut `start=" [zahl] "` setzt den Startwert auf einen von "1" abweichenden Wert fest.

verschachtelte Listen

```
<ol>
  <li>eins
    <ul>
      <li>eins-eins</li>
      <li>eins-zwei</li>
    </ul>
  </li>
  <li>zwei</li>
</ol>
```

1. eins
 - eins-eins
 - eins-zwei
2. zwei

Definitionslisten

```
<dl>
  <dt>HTML</dt>
  <dd>Hypertext Markup Language</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheets</dd>
</dl>
```

HTML

Hypertext Markup Language

CSS

Cascading Style Sheets

Tabellen

Einfache Tabelle

```
<table border="1">
  <tr>
    <th>Kernobst</th>
    <th>Steinobst</th>
  </tr>
  <tr>
    <td>Apfel</td>
    <td>Pflaume</td>
  </tr>
  <tr>
    <td>Birne</td>
    <td>Kirsche</td>
  </tr>
</table>
```

Kernobst	Steinobst
Apfel	Pflaume
Birne	Kirsche

Die Tabellenelemente in *HTML* sind

```
<table align="[left | center | right]" ... A
  border="[number]" ... B
  cellpadding="[number]" ... C
  cellspacing="[number]" ... D
  rules="[none | groups | rows | cols | all]" ... E
  width="[number]"> ... F

<tr align="[left | center | right | justify]" ... G
  valign="[top | middle | bottom | baseline]"> ... H

<th|td align="[left | center | right | justify]" ... I
  valign="[top | middle | bottom | baseline]" ... J
  colspan="[number]" ... K
  rowspan="[number]"> ... L
```

Der Aufbau [komplexer Tabellen](#) ist anschaulich bei *SELFHTML* beschrieben. Ebenso deren [Gestaltung mittels CSS](#).

Grafiken

Grafiken

- können nicht direkt eingebettet werden. Sie werden über externe Dateien referenziert.
- unterstützte Grafikformate sind *gif*, *jpeg*, *png*.
- können je nach Grösse eine nicht unerhebliche Ladezeit verursachen.

Beispiel

```
  
<a href="http://de.wikipedia.org/wiki/Hauptseite">  
    
</a>
```



Formularelemente

-ToDo-

CSS in 5 Min.

CSS ist ...

- Cascading Style Sheets
- ähnlich den Formatvorlagen eines Textverarbeitungsdokuments.
- eine Sammlung von Vorschriften zur Festlegung des Aussehens von XML/XHTML Elementen.
- eine Steuereinheit für Seitenlayout, Textfluss und Print-Layout.
- ein Hilfsmittel für Webseitengestaltung und Corporate Design.

Externe CSS Definition

Syntax

```
<link rel="stylesheet" media="{target}" href="{url}" />
```

Beispiel

```
<html>
  <head>
    <title>Webdokument</title>
    <link rel="stylesheet" media="screen" href="seite.css" />
    <link rel="stylesheet" media="print" href="druck.css" />
    <link rel="stylesheet" media="projection" href="slide.css" />
  </head>
  <body>
  </body>
</html>
```

```
Selector {property1: value1; ...; propertyN: valueN; }
```

; Selector: Auswahl der gewünschten Elemente (body, h1, table, ...).

property

festzulegende Eigenschaft (color, border-style, ...)

value

Eigenschaftswert (red, solid, ...)

zentrale Einbindung in Dokument

Beispiel

```
<html>
  <head>
    <title>Webdokument</title>
    <style type="text/css">
      body { font-weight:bold; font-family:Tahoma,sans-se
      h1, h2 { color:blue; font-style:italic; }
    </style>
  </head>
  <body>
  </body>
</html>
```

lokale Stilfestlegung in Elementen

Beispiel

```
<html>
  <head>
    <title>Webdokument</title>
  </head>
  <body style="font-weight:bold; font-size:1.2em;">
    <h1 style="color:blue; font-style:italic;">...</h1>
  </body>
</html>
```

SelfHTML bietet eine gute Übersicht über alle CSS 2.1 Eigenschaften.

[CSS Kurzreferenz](#)

Verweise

- [SelfHtml - HTML](#)
- [SelfHtml - CSS](#)
- [HTML @ W3C](#)
- [CSS @ W3C](#)
- [kostenloser HTML-Editor](#)