

Computergrundlagen

HTML — Hypertext Markup Language

Axel Arnold

Institut für Computerphysik
Universität Stuttgart

Wintersemester 2013/14



Ausgabemedien

Papier

- feste Seitengröße
- fixe Reihen(Seiten-)folge
- Kapitel-, Seitennummern
- einmal gedruckt, für immer fest
- Beschrieben in \LaTeX

Bildschirm

- freie Seitenlänge- und Breite
- keine Reihenfolge
- Verweise auf Einheiten (Kapitel/Formeln/Seiten)
- Darstellung per Browser zur Laufzeit
- Beschrieben in **HTML**
- Interaktivität durch Formulare

HTML

Hypertext Markup Language

- 1990 erfunden durch Tim Barners-Lee
- für den Dokumentenaustausch am CERN
- Hypertext: mehr als nur Text
- heute oft aus anderen Quellen erzeugt:
Wiki, CMS, ...



T. Barners-Lee
wikipedia.org

Browser

- Darstellung von u. a. HTML
- Netscape (bis 2008), Firefox, Internet Explorer, Google Chrome
- Lynx nichtgraphisch - Testen aufgrund Barrierefreiheit



Grundlegender Aufbau

```
<!DOCTYPE html>
<html>
  <!-- Header (Preamble) -->
  <head>
    <title>Titel der Seite</title>
  </head>
  <!-- und der Textkoerper -->
  <body>
    Hallo
    <em>Welt</em>! <!-- mit Betonung! -->
    <!-- wie LaTeX: Anzahl Leerzeichen ist egal -->
  </body>
</html>
```

- Baum von Elementen
- Elemente haben Form `<tag>...</tag>`
- Kommentare: `<!-- ein Kommentar -->`



Der Textkörper: body

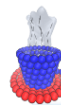
Logisches Markup

- Darstellung hängt vom Browser (oder CSS) ab
-

```
<h1>Ueberschrift</h1>  
  <h2>Unterueberschrift</h2>  
  ...  
    <h6>kleinste Ueberschrift</h6>  
<p>Ein Absatz</p>  
<p>und noch einer,  
  mit betontem <em>Wort</em>  
</p>
```

Visuelles Markup (veraltet)

- kursiv: **<i>**, fett: ****, zentriert: **<center>**
- besser CSS benutzen!



Listen

``

` vergleichbar itemize in LaTeX `

` noch ein Item `

``

``

` vergleichbar enumerate in LaTeX `

` zweites Item `

``

`<dl>`

`<dt>Liste von Beschreibungen</dt>`

`<dd>Paare von Begriffen und Beschreibungen</dd>`

`<dt>Aufzählung</dt>`

`<dd>einfache nummerierte Liste</dd>`

`</dl>`



Tabellen

```
<table border="1">
<tr> <!-- erste Zeile 1 0 -->
  <td>0</td> <td>1</td>
</tr>
<tr> <!-- zweite Zeile 0 1 -->
  <td>1</td> <td>0</td>
</tr>
<tr> <!-- dritte Zeile zeigt einen Pfeil -->
  <td></td> <td></td>
</tr>
</table>
```

- nur ganz mit Grenzen oder ohne
- Elemente können beliebiges HTML enthalten
- wird oft für Layout missbraucht



Verweise

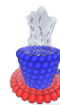
` Linktext `

` ICP `

- Linktext kann komplexes HTML sein, z. B. ein Bild
- man weiß nicht, wieviele Links auf eine Seite zeigen
- außer man fragt Google...

Bilder

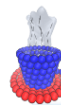
- ``
- im selben Verzeichnis oder voller Pfad
- alt-Text wichtig für Lynx und Screenreader



Header: Metainformationen

```
<head>
  <title>Titel der Seite</title>
  <meta name="author" content="Axel Arnold">
  <style type="text/css">
    body { background-color:black; color:white; }
  </style>
</head>
```

- Information im Header wird nicht dargestellt
- kann aber von z. B. Google-Crawlern ausgewertet werden
- Layoutbeschreibung in CSS
- kann auch JavaScript enthalten



Layout: Cascading Style Sheets (CSS)

```
body {  
  font-family: sans-serif;  
  background: #808080;  
  margin: 20%;  
}  
p { font-weight: bold; text-align: center; font-size: 2em; }  
a:visited { text-decoration: underline; color: #ff0080; }  
li { font-style: italic }
```

- Beschreiben, wie HTML dargestellt werden soll
- im `<head>` als
- `<style type="text/css">..</style>`
- oder Verweis auf externe Datei mit `<link rel="stylesheet" type="text/css" href="simple.css"/>`
- Was mit CSS alleine geht: <http://www.csszengarden.com>

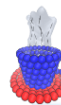


Elemente und Zustände

```
<head> <style type="text/css">
  p { font-weight: bold; text-align: center; font-size: 2em; }
  a:link      { color: red; }
  a:visited   { color: green; text-decoration: underline; }
  li { font-style: italic }
</style> </head>
```

```
<body> <p>
  Absatz mit <a href="index.html">Link</a> und
  <ul>
    <li> Liste. </li>
  </ul>
</p> </body>
```

-
- Layout aller HTML-Elemente nach Typ
 - Elemente können einen Zustand haben, z. B. Links



Class und Id

```
<head> <style type="text/css">
.stichwort { color: blue; }
p.emph     { color: purple; }
#thema     { font-size: 2em; }
</style> </head>
```

```
<body>
<p id="Thema"> Thema sind CSS class und id. </p>
<p> Normaler Absatz mit
  <span class="stichwort">Bereich</span>. </p>
<p class="emph"> Hervorgehobener Absatz. </p>
</body>
```

- Markieren einzelner Elemente per `id="tag"` (`#tag` in CSS)
- Markieren von Klassen von Elementen über `class="class"` (`.class` in CSS)
- Einschränken auf bestimmte HTML-Elemente: `element.class`



Interaktivität: Formulare

```
<form action="search.php">
  <legend>Auf dem ICP Suchen</legend>
  <input type="text" name="search" value="" />
  <input type="submit" name="fulltext" value="Suche" />
  <ul>
    <li><input type="radio" name="r" value="alle"> Alle </li>
    <li><input type="radio" name="r" value="cg"> Nur CG </li>
  </ul>
</form>
```

- ruft bei Klick auf „Suche“ die Seite **search.php?search=test&fulltext=Suche&r=alle** auf
- die Parameter müssen auf Serverseite verarbeitet werden, daher meist Programmiersprache wie PHP
- mehrere Formulare auf einer Seite möglich
- heute oft JavaScript