



openHPI Grundlegende HTML-Elemente

Prof. Dr. Christoph Meinel

Hasso-Plattner-Institut

Universität Potsdam

Dokumentenkopf und Meta-Tags (1/2)

- Kopf (Header) eines HTML-Dokuments enthält *Informationen über das Dokument* – **Meta-Informationen**, z.B.
 - Titel, Name des Autors, Beschreibung, etc.
- Meta-Information sind wichtig für automatische Verarbeitung des Dokuments, z.B. für dessen Aufnahme in den Suchindex einer Suchmaschine
- Kopf eines HTML-Dokuments wird umschlossen von `<head>...</head>`
- Folgende Tags können im HTML-Header verwendet werden:
 - **title:** Titel der Seite zur Anzeige in der Browser-Kopfleiste
 - **base:** Basis-URL für alle URLs in der HTML-Datei; alle *relativen URLs* werden dann (vorne) ergänzt um diese Basis-URL
 - **style:** Definition von CSS-Styles für das HTML-Dokument
 - **link:** Einbettung externer Ressourcen, z.B. CSS-Dateien
 - ...

Dokumentenkopf und Meta-Tags (2/2)

- Folgende Tags können in einem HTML-Header verwendet werden:
 - **script:** → JavaScript für das Dokument oder Einbettung einer externen JavaScript-Datei
 - **noscript:** alternativer Seiteninhalt für Browser, die kein JavaScript ausführen können
 - **meta:** Metainformationen zum Dokument, z.B.
 - `<meta name="author" content="Christoph Meinel">`
 - `<meta name="keywords" content="HTML, CSS, XML, ...">`
 - `<meta charset="UTF-8">` (Zeichenkodierung des Inhalts)
 - `<meta http-equiv="refresh" content="10">`
(Lädt nach 5 Sekunden die Seite neu)

- Eigentlicher Inhalt eines HTML-Dokuments befindet sich im **Body**
- Body wird eingefasst von `<body>...</body>`
- HTML unterscheidet zwei Kategorien struktureller Elemente:
 - **Block-Elemente** – bestimmen Block-basierte Strukturen im Body des Dokuments
 - können weitere Block-Elemente oder Inline-Elemente enthalten
 - zwei aufeinanderfolgende Block-Elemente werden grundsätzlich **untereinander** angeordnet und nehmen die **volle Breite** ein
 - **Inline-Elemente** – befinden sich im Fließtext
 - können weitere Inline-Elemente enthalten
 - können am Zeilenende umgebrochen werden
 - Mittels CSS kann für viele Elemente festgelegt werden, ob diese als Block- oder Inline-Element behandelt werden sollen

Die wichtigsten **grundlegenden Text- und Strukturelemente** sind:

- **<p>** – Paragraph, Absatz
- **<div>** – Division, Abschnitt
- **<h1>** bis **<h6>** – (Zwischen-)Überschriften absteigender Wichtigkeit
- **** – Abschnitte in Fließtext (Inline-Element)
- **<blockquote>** – Zitate
- **<i>**, **** – italics, emphasis, Hervorhebungen (Inline-Element, meist als schräggestellter Text dargestellt)
- **** – wichtiger Text (Inline-Element, meist fettgedruckter Text)

Die folgenden Tags kommen grundsätzlich **ohne schließenden End-Tag** aus:

- **<hr>** – horizontal rule, horizontale Linie
- **
** – break, Zeilenumbruch

Herzstück des WWW sind **Hyperlinks**

- Hyperlinks können voneinander getrennte, unabhängige Informationsressourcen miteinander verbinden
- HTML kennt verschiedene Typen von Hyperlinks:
 - **Unsichtbare Links**, z.B. zwischen HTML-Dokumenten und CSS-Stylesheets, werden im Hintergrund automatisch vom Browser bearbeitet – **<link>**
 - **Sichtbare Links** werden durch einen Mausklick des Nutzers aktiviert und führen zu anderen Informationsressourcen im Web, die über ihre URIs identifiziert werden – **<a>**

<a>-Element kann für verschiedene Zwecke genutzt werden:

- Markierungen (Ankerpunkte) – `<a name="ziel"></a>`
`ziel` definiert einen eindeutigen Bezeichner für einen Ankerpunkt innerhalb eines Dokuments, zu dem ein Browser springen kann
- Sprung zu einem Ankerpunkt – `<a href="ziel">Verlinkter Text</a>`
Klick auf den Link Text springt zu einem Ankerpunkt mit dem Namen `ziel`
- Sprung zu einem externen Ziel – `<a href="URL">Verlinkter Text</a>`
 - URI kann absolut oder relativ angegeben werden, z.B.
 - Sprung zu anderem Server mit absolutem URI:
`<a href="https://www.google.de">Google-Suche</a>`
 - Sprung zu HTML-Dokument auf gleichem Server mit relativem URI:
`<a href="/pages/imprint">Impressum</a>`

<a>-Element kann für verschiedene Zwecke genutzt werden:

- Sprung zu externer Ressource und zu einem Ankerpunkt kann auch kombiniert werden, z.B.

```
<a href="https://open.hpi.de/pages/faq#wo-finde-ich-die-  
kursmaterialien">
```

FAQ-Eintrag auf openHPI

```
</a>
```

Klick auf den Link öffnet FAQ-Seite und springt zu angegebener Frage

- **<a>-Tags** kennt neben **href** auch weitere **Attribute**, z.B.
 - **target:** bestimmt, wo verlinkte Ressource geöffnet werden soll
 - `<a href="URL" target="_blank">Verlinkter Text</a>`
Öffnet Link in neuem Fenster oder Tab (je nach Browser-Einstellung)
 - `<a href="URL" target="_self">Verlinkter Text</a>`
Öffnet Link in gleichem Fenster oder Tab (Default-Target)

Bilder (1/2)

Wichtigste „verlinkte Ressource“ in HTML-Dokumenten sind **Bilder**

- Haben eigenen Tag: **** (ohne Endtag)
- Wichtige Attribute:
 - **src:** URL, wo das Bild liegt (relativ oder absolut)
 - **alt:** alternativer Text, der angezeigt wird, wenn das Bild nicht geladen oder angezeigt werden kann, z.B. wegen schlechter Verbindung oder mit Screen Readern
 - ➔ sollte **immer** angegeben werden
 - **height / width:** Höhe und Breite, in der das Bild auf der HTML-Seite angezeigt werden soll
 - Wird nur ein Wert angegeben, wird der andere automatisch entsprechend der Original-Seitenverhältnisses vom Browser berechnet
 - Alternativ: **style="height: Höhe; width: Breite"**

Bilder (2/2)

Beispiel

- Bilder können auch mit einem Link versehen werden

```
<a href="https://open.hpi.de" target="_blank">  
    
</a>
```

Ein Klick auf das Bild öffnet den Link in einem neuen Fenster

HTML kennt verschiedene Typen von Listen

■ Ungeordnete Listen

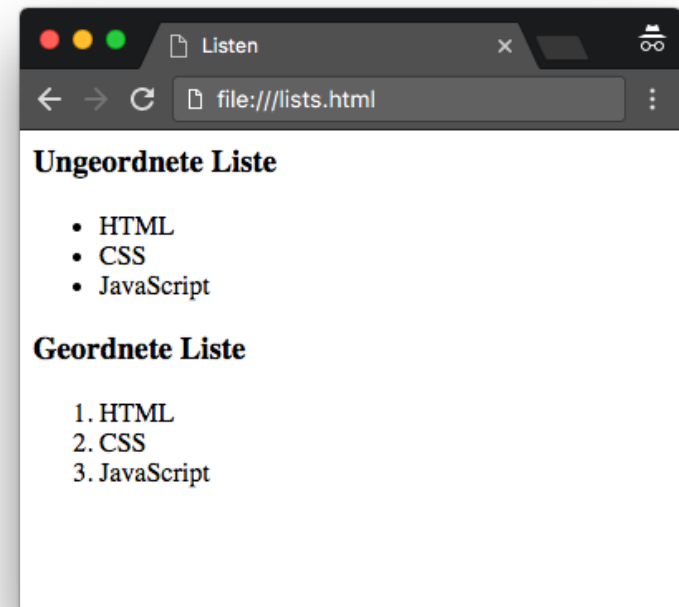
- Nicht-nummeriert mit Aufzählungszeichen
- Werden durch ****-Tags erzeugt
- Listenelemente stehen in ****-Tags

■ Geordnete Listen

- Engl. „ordered list“, nummeriert
- Werden durch ****-Tags erzeugt
- Listenelemente wieder in ****-Tags

■ ...

```
<h3>Ungeordnete Liste</h3>
<ul>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ul>
```

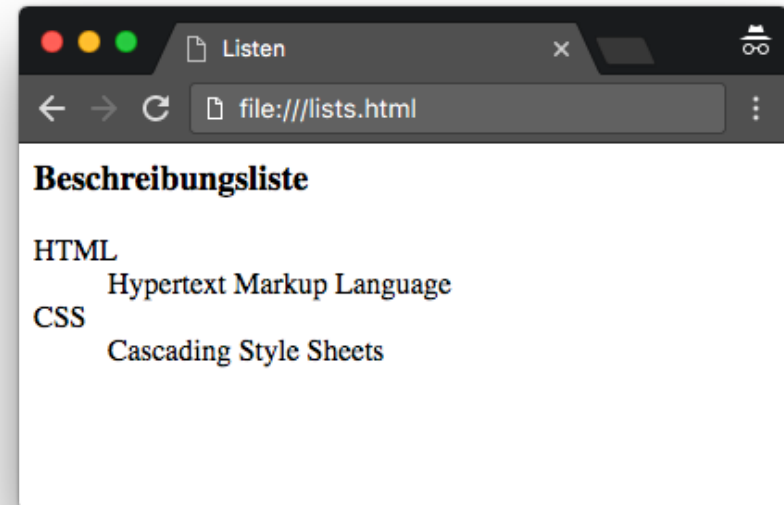


HTML kennt verschiedene Typen von Listen

■ Beschreibungslisten

- Engl. „description lists“
 - Werden durch **<dl>**-Tag erzeugt
 - Listenelemente bestehen aus
 - Begriffen (terms) – **<dt>**
 - Erläuterungen (descriptions) – **<dd>**
- Bei allen Listentypen kann die Darstellung mit CSS angepasst werden, z.B.
- Form des Aufzählungszeichens
 - Art der Nummerierung

```
<h3>Beschreibungsliste</h3>
<dl>
  <dt>HTML</dt>
  <dd>Hypertext Markup Language</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheets</dd>
</dl>
```

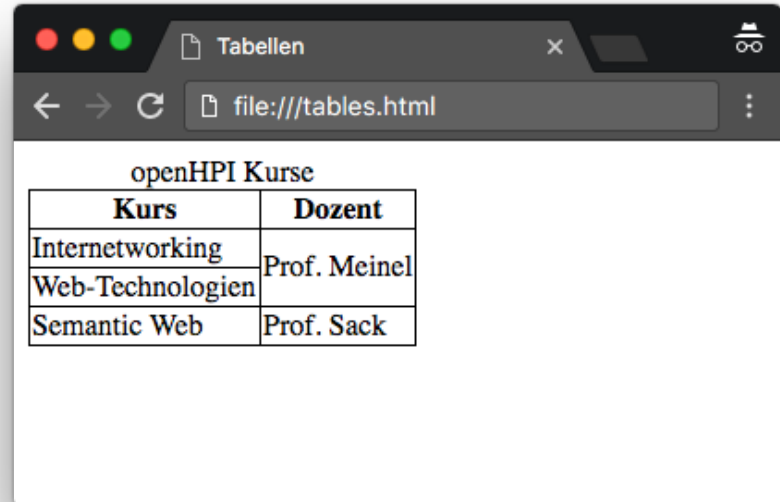


- Tabellen in HTML werden durch die Tags `<table>...</table>` erzeugt
- Tabellen bestehen aus rechteckigen Zellen, die in Zeilen und Spalten organisiert sind
 - `<tr>` erzeugt eine neue Zeile (table row) in einer Tabelle
 - `<td>` erzeugt eine neue Zelle in einer Tabellenzeile
 - `<th>` erlaubt die Auszeichnung spezieller Zellen für den Tabellenkopf (table header)
- Innerhalb einer Tabelle können Tabellenkopf und Tabellenfuß auch explizit festgelegt werden
 - `<thead>`, `<tbody>` und `<tfoot>` bezeichnen dann die einzelnen Bereiche innerhalb der Tabelle
 - Bereiche enthalten dann jeweils Zeilen und Spalten
 - Unterteilung erlaubt differenziertere Behandlung bei Skripting und Formatierung

- **Benachbarte Zellen** können mit speziellen Attributen für `<td>` **zusammengefasst** werden
 - Nebeneinanderliegende Zellen: **colspan**
`<td colspan="2">Zelle geht über 2 Spalten</td>`
 - Übereinanderliegende Zellen: **rowspan**
`<td rowspan="2">Zelle geht über 2 Zeilen</td>`
- Tabellen können mit Beschriftung versehen werden
 - **<caption>**-Tag wird direkt als erstes innerhalb der Tabelle eingefügt
 - Ohne weitere Formatierung wird Caption oberhalb der Tabelle angezeigt
- Formatierung der Tabelle (Rahmen, Abstände, Ausrichtung, usw.) erfolgt mittels CSS

Beispiel

```
<table>
  <caption>openHPI Kurse</caption>
  <tr>
    <th>Kurs</th>
    <th>Dozent</th>
  </tr>
  <tr>
    <td>Internetworking</td>
    <td rowspan="2">Prof. Meinel</td>
  </tr>
  <tr>
    <td>Web-Technologien</td>
  </tr>
  <tr>
    <td>Semantic Web</td>
    <td>Prof. Sack</td>
  </tr>
</table>
```



Kurs	Dozent
Internetworking	Prof. Meinel
Web-Technologien	
Semantic Web	Prof. Sack

Etwas CSS für bessere Lesbarkeit:

```
<style>
  table, td, th {
    border: 1px solid black;
    border-collapse: collapse;
  }
</style>
```

- Zur Übertragung von Daten vom Browser zum Server, die über bloße Anforderung von Dokumenten hinaus gehen, gibt es HTML **Formulare**
- Formulare können für unterschiedliche Zwecke eingesetzt werden:
 - Anforderung ähnlich strukturierter Informationen, z.B.
 - Filter für Daten, Paginierung, ...
 - Spezifikation von Suchanfragen in großen Datenmengen
 - Übermittlung eigener Datensätze an einen Server
 - Individuelle Nutzerinteraktion, z.B. in Online Shops
 - Übermittlung von Dateien an einen Server
 - ...

- Formulare werden mit dem **<form>**-Tag erzeugt:
`<form action="/login" method="POST">...</form>`
 - **action:** Gibt das Ziel des Formulars an – also eine URL, an die die Formulardaten mit HTTP gesendet werden
 - **method:** Gibt die HTTP-Methode an, mit der das Formular gesendet wird – also z.B. GET oder POST
- Es gibt eine **Reihe grundlegender Eingabefelder**, z.B.
 - **<input type="text">** – Texteingabefeld
 - **<input type="password">** – Texteingabefeld für Passwörter, Eingabe wird nicht angezeigt
 - **<input type="radio">** – „Radio Buttons“, runde Knöpfe für die Auswahl einer von mehreren Optionen
 - **<input type="checkbox">** – Boxen mit Häkchen für Mehrfachauswahl von Optionen

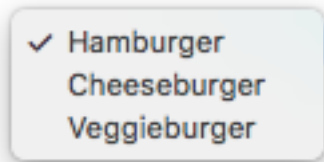
☒ Wahr
☐ Falsch

☒ HTML
☒ CSS

- Es gibt eine **Reihe grundlegender Eingabefelder**, z.B.

- Auswahlfeld zum Ausklappen („Dropdown-Liste“)

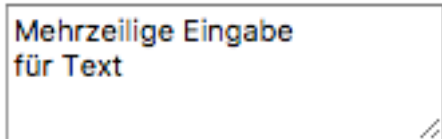
```
<select name="bestellung">  
  <option>Hamburger</option>  
  <option>Cheeseburger</option>  
  <option>Veggieburger</option>  
</select>
```



✓ Hamburger
Cheeseburger
Veggieburger

- Textfeld für mehrzeiligen Text

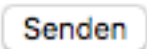
```
<textarea rows="5" cols="20"></textarea>
```



Mehrzeilige Eingabe
für Text

- Button zum Absenden des Formulars

```
<input type="submit" value="Senden" />
```



Senden

➔ Führen später weitere Formularfelder ein; auch im Kontext der Webprogrammierung geht es um Verarbeitung von Formulardaten

HTML-Entitäten

- **Entitäten** werden bei Markup-Sprachen verwendet, um häufig vorkommende Zeichenketten zu verwalten und wiederzuverwenden
- HTML-Entitäten haben das Format **&Kennung;**
- Entitäten werden zum einen für Sonderzeichen verwendet, die für das Markup vorgesehen sind, z.B.
 - `<` – `<`; bzw. `>` – `>`;
 - `&` – `&`;
- Außerdem werden Entitäten für sprachspezifische Sonderzeichen genutzt, z.B.
 - Deutsche Umlaute: `ä` – `ä` / `Ä` – `Ä` / `ß` – `ß` / ...
 - Accents: `Á` – `Á` / `ê` – `ê` / `ò` – `ò` / ...
 - Griechisches Alphabet: `π` – `π` / `Ω` – `Ω` / ...
 - Andere, z.B. `©` – `©` / `½` – `½` / `€` – `€` / ...