

HTML 5

Elemente und Tabellen

Ingo Köster

Diplom Informatiker (FH)

Erstes HTML-Dokument

- › Hier ein ganz einfaches HTML-Dokument:

```
<!DOCTYPE html>
<html lang="de">
<head>
  <title>Hallo, HTML!</title>
</head>
<body>
  Hallo, Browser!
</body>
</html>
```

- › Der BODY eines HTML-Dokuments hat ein gemischtes Inhaltsmodell, d.h. er kann einfachen Text und/oder weitere Elemente enthalten

Absätze

- › Ein Absatz wird durch das Element `<p>` gebildet
- › Man kann das Trennen/Umbrechen von Worten in einem Absatz durch die Verwendung des Entity ` sp;` verhindern
- › Beispiel:

Bitte nicht` sp;`umbrechen.

- › Worttrennungen können durch das Entity `­` eingeleitet werden
- › Beispiel:
Trenn`­`strich

Absätze - Zeilenumbrüche

- › Da HTML jede Form von nicht druckbaren Zeichen (white space) unterdrückt, benötigt man einen „Marker“ für Zeilenumbrüche
- › Zeilenumbrüche werden in HTML durch das leere Element `
` angezeigt
 - › Es genügt auch einfach `
` zu verwenden
- › Beispiel:

```
<body>  
  <p>Hier fängt die erste Zeile an, <br />  
    hier startet dann die zweite Zeile.  
  </p>  
</body>
```

Überschriften

- › Für die Gliederung von Texten stehen sechs Überschriften zur Verfügung
- › Die Elemente heißen `<h1>` bis `<h6>`
- › Je höher die Zahl, desto „untergeordneter“ die Überschrift

Aufzählungen und Listen

- › Listen und Aufzählungen sind in HTML sehr flexibel einsetzbar
- › Sie können zur klassischen Strukturierung von Texten verwendet werden
- › Oft werden Listen auch zur Darstellung von Menüs und anderen Navigationselementen verwendet

Aufzählungen und Listen

- › HTML definiert folgende Elemente, um Listen und Aufzählungen zu realisieren:
- › OL (nummerierte) Aufzählung
- › UL Liste von Gliederungspunkten
- › LI Listenelement
- › DL Definitionsliste
- › DT Definitionsbegriff
- › DD Begriffsdefinition

Aufzählungen

- › Das Element `` kennt die Attribute `type` und `start`
- › Außerdem die Attribute `id`, `lang`, `class` und `dir`
- › Das Attribut `start` dient der Realisierung von Nummernfolgen
- › Das Attribut `type` legt das System der Nummerierung fest
 - › `type` ist veraltet: Nur noch mit CSS lösen!!
- › Mittels `reversed` eine absteigende Liste erzeugen

Aufzählungen

Wert für Type	Beschreibung	Auswirkung
1	Arabische Zahlen	1,2,3
A	Großbuchstaben	A, B, C
a	Kleinbuchstaben	a, b, c
I	Große römische Ziffern	I, II, III, IV, V
i	Kleine römische Ziffern	i, ii, iii, iv, v

Listen

- › Das Element `` hat das Attribute `type`
- › Außerdem die Attribute `id`, `lang`, `class` und `dir`
- › Das Attribut `type` legt das Symbol der Aufzählungspunkte fest
 - › `type` ist veraltet: Nur noch mit CSS lösen!!
- › Beispiel:

```
<ul type="square">  
  <li>ein Element</li>  
  <li>noch ein Element</li>  
</ul>
```

Begriffslisten

- › Ursprünglich zur Bildung von Glossaren in wissenschaftlichen Arbeiten eingeführt
- › Werden heute auch oft mittels CSS für andere Zwecke, wie Menüs verwendet

› Beispiel:

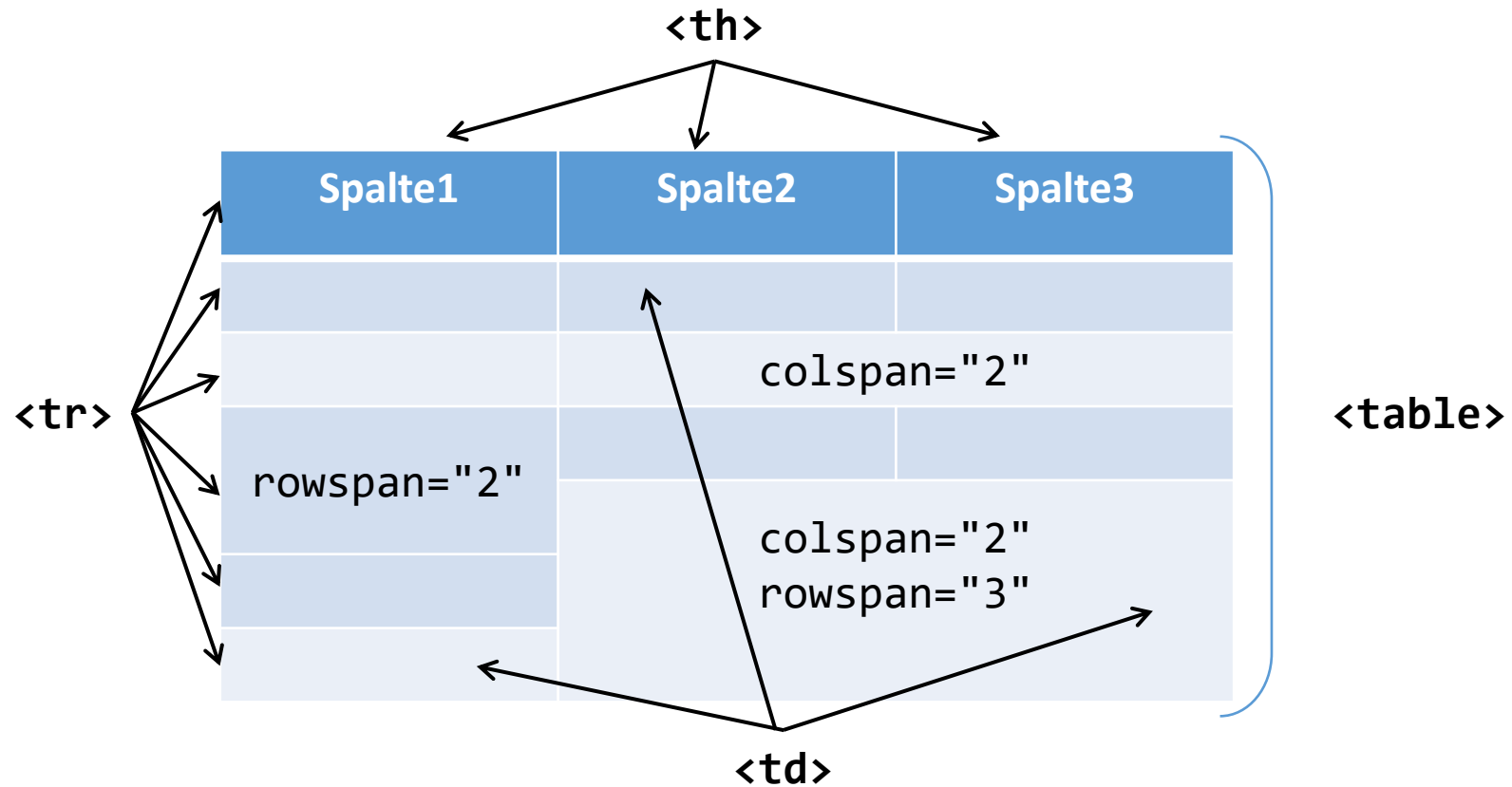
```
<dl>  
  <dt>HTML</dt>  
  <dd>Hypertext Markup Language</dd>  
  <dt>VB</dt>  
  <dd>Very Bad Language</dd>  
</dl>
```

Tabellen

Tabellen

- › Tabellen bestehen aus Zeilen und Spalten
- › Zeilen und Spalten können zusammengefasst werden
- › Tabellen können wiederum andere Tabellen enthalten
- › Anwendung:
 - › Präsentation von tabellarischen Daten
 - › Einfache Ausrichtung von Formularen

Tabellen



Tabellen

```
<table border="1">
<tr>
  <th>Spalte 1</th>
  <th>Spalte 2</th>
  <th>Spalte 3</th>
</tr>
<tr>
  <td>Wert 1</td>
  <td>Wert 2</td>
  <td>Wert 3</td>
</tr>
</table>
```

Spalte 1	Spalte 2	Spalte 3
Wert 1	Wert 2	Wert 3

Tabellen – Zellen zusammenfassen

```
<table border="1">
  <tr>
    <th>Spalte 1</th>    <th>Spalte 2</th>    <th>Spalte 3</th>
  </tr>
  <tr>
    <td>Wert 1</td>
    <td colspan="2">Wert 2 + Wert 3</td>
  </tr>
  <tr>
    <td>Wert 4</td>
    <td>Wert 5</td>
    <td rowspan="2">Wert 6 + Wert 8</td>
  </tr>
  <tr>
    <td>Wert 6</td>    <td>Wert 7</td>
  </tr>
</table>
```

Spalte 1	Spalte 2	Spalte 3
Wert 1	Wert 2 + Wert 3	
Wert 4	Wert 5	Wert 6 + Wert 8
Wert 6	Wert 7	

Kopf, Rumpf und Fuß einer Tabelle

- › Eine Tabelle kann logisch in drei Bereiche eingeteilt werden:
 - › Kopf
 - › Rumpf
 - › Fuß
- › Die Einteilung ist nur logisch, es wird nicht automatisch eine andere Formatierung der Tabelle angewendet
- › Per CSS kann man jedoch ein entsprechendes Layout generieren

Tabellen unterteilen

```
<table border="1">
  <thead>
    <tr>
      <th>Spalte 1</th>   <th>Spalte 2</th>   <th>Spalte 3</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Wert 1</td>   <td>Wert 2</td>   <td>Wert 3</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Wert 4</td>   <td>Wert 5</td>   <td>Wert 6</td>
    </tr>
  </tfoot>
</table>
```

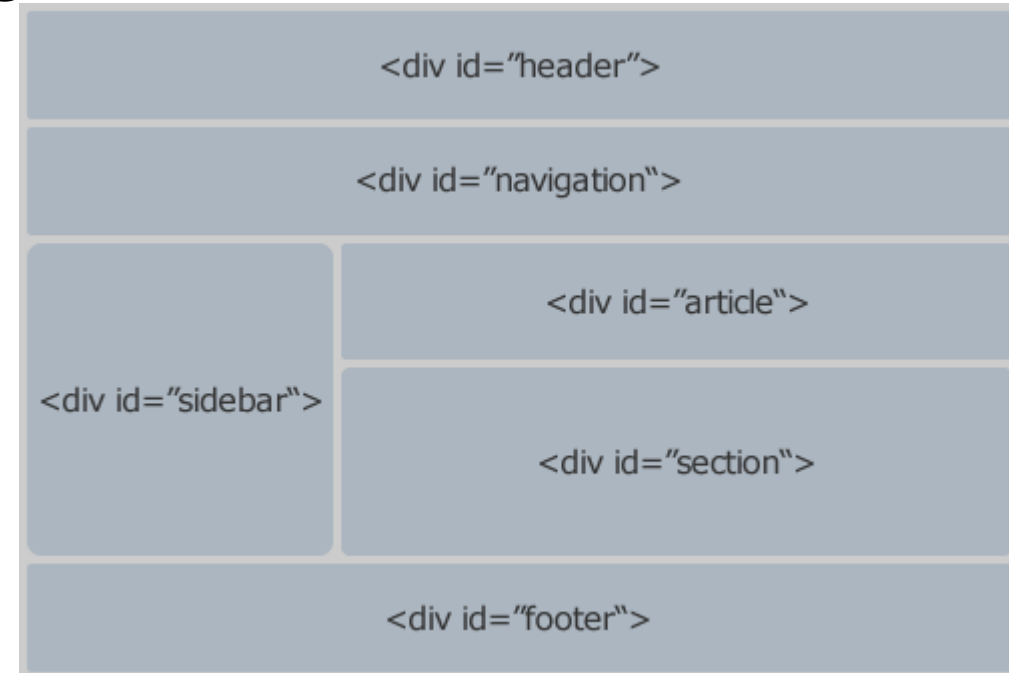
Container

Der DIV-Container

- › Mit HTML 4 wurden zwei Container-Elemente eingeführt:
 - › `<div>`
 - › ``
- › `<div>` ist der Container für Block-Elemente
- › `` ist der Container für Inline-Elemente
- › Anwendungen für `<div>`:
 - › Logische bzw. semantische Gruppierung von Inhalten
 - › Gruppierung von Inhalten zur Formatierung mit CSS

Der DIV-Container

- › Klassisches Anwendungsbeispiel für DIV-Elemente war/ist die Strukturierung von Web-Seiten für das Layout
- › Jeder DIV bekommt eine ID oder CSS-Klasse zugewiesen und wird dann entsprechend positioniert und formatiert
- › Alternative: Semantisches Markup



Semantische Elemente

<section>

<header>

<nav>

<article>

<aside>

<footer>



Semantische Elemente

- › Vorschlag der Elemente erfolgte nach einer Analyse vieler Web-Seiten und deren Struktur
- › Soll die Nutzung von DIV-Containern zur semantischen Strukturierung ablösen
- › Der Browser nimmt keinerlei automatisches Layout vor, es muss immer noch selbst mittels CSS gestaltet werden
- › Hilft gut strukturierte Web-Seiten zu erstellen, erleichtert SEO-Aufgaben