

Hochschule Darmstadt

Fachbereich Informatik

Entwicklung webbasierter Anwendungen



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

fbi

FACHBEREICH INFORMATIK

Hochschule Darmstadt

Fachbereich Informatik

Wiederholung



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

fbi

FACHBEREICH INFORMATIK

Webquellen und Software

Webquellen

- ⇒ MDN: JavaScript Guide
- ⇒ Für die Beantwortung konkreter Fragen (kein Tutorial) - stackoverflow

Standards

HTML-, CSS-, DOM-Standard
und HTML/CSS-Validator

validator.w3.org/

Freie Software, Dokus, Tutorials

Visual Studio Code

PHPStorm

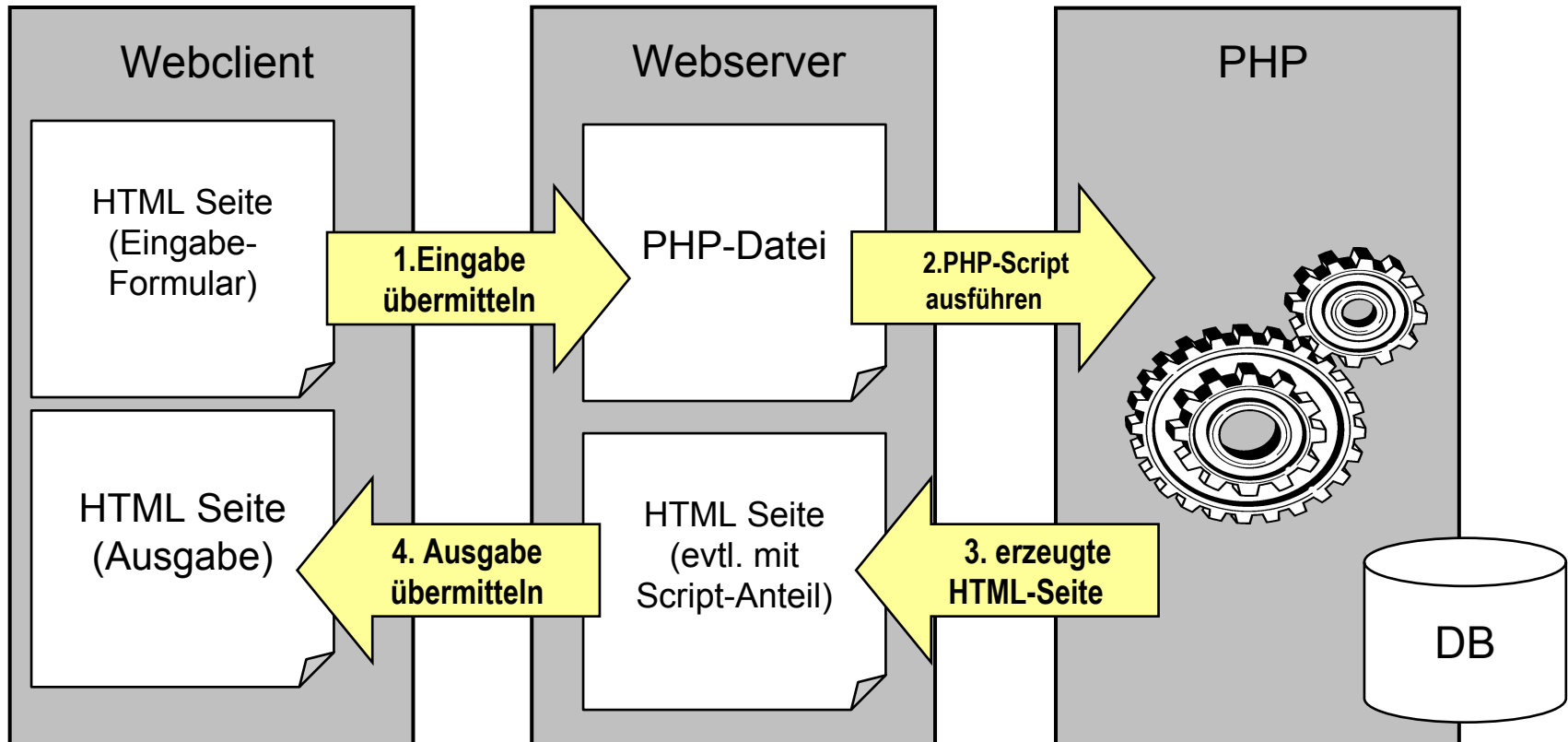
XAMPP (Webserver, MySQL, PHP)



1. Einleitung

Einsatz der Technologien im Zusammenhang

Es ist enorm wichtig zu verstehen, welche Inhalte in welchem Kontext erzeugt werden



- HTML
- CSS
- ECMA-Script
- DOM
- AJAX

- HTTP

- Server-Konfiguration

- CGI
- PHP

- MySQL



- Webseiten sollen auch für Menschen mit einer Behinderung zugänglich sein
 - ⇒ Sehschwächen (z.B. Rot-Grün-Blindheit) aber auch Hör-, Lern-, Lese-, motorische Schwächen uvm.
 - ⇒ insbesondere soll eine Webseite vorgelesen werden können
 - Screenreader lesen die Seite mit Text2Speech vor
 - Braille-Zeilen geben Blindenschrift zeilenweise aus
 - Tabellen werden von links nach rechts und von oben nach unten gelesen
 - Bilder und Videos sind als solche nicht darstellbar
 - Der Inhalt muss logisch gruppiert und angeordnet sein – und nicht nach der Anordnung auf dem Bildschirm
 - ⇒ Die Bedienung muss mit vereinfachten Tastaturen möglich sein

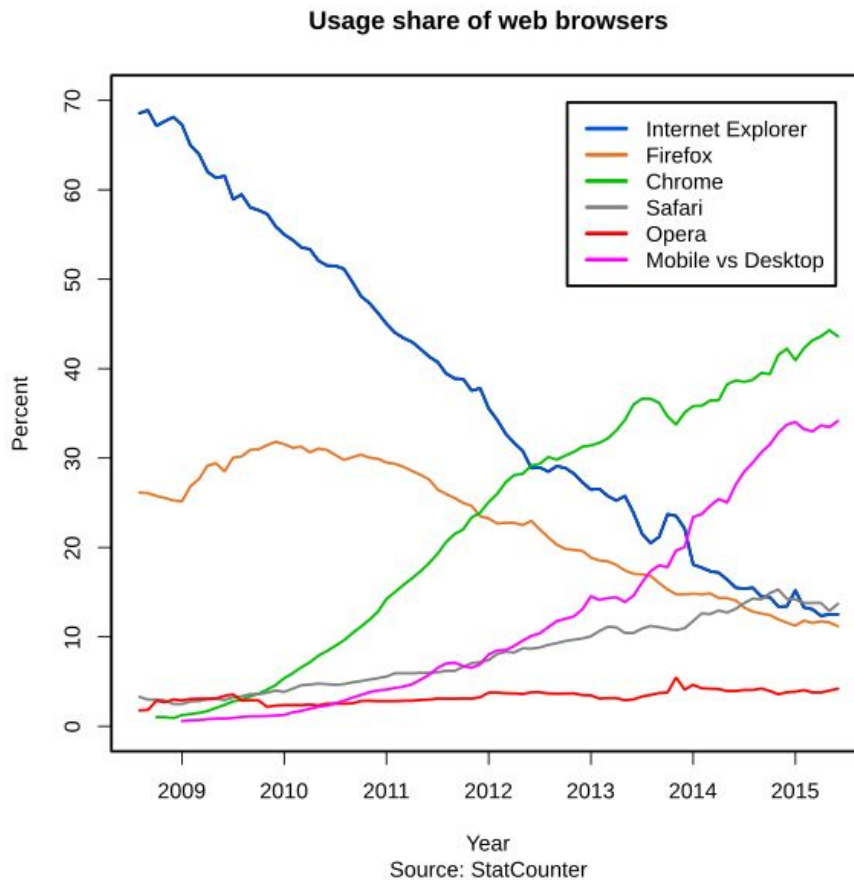


Textausgabe mit Braille-Zeile
Bild: SBV



www.computer-fuer-behinderte.de

Marktanteile und Trend der Web-browser



mobile / responsive

Source: https://upload.wikimedia.org/wikipedia/commons/thumb/8/86/Usage_share_of_web_browsers_%28Source_StatCounter%29.svg/600px-Usage_share_of_web_browsers_%28Source_StatCounter%29.svg.png

HTML5 und ordentlicher Code

- HTML5 wurde abwärtskompatibel und browserfreundlich definiert
 - ⇒ für viele Sprachkonstrukte, die unter HTML4 einen Fehler erzeugten, ist jetzt das Browserverhalten definiert – und es sind keine Fehler mehr
 - viele fehlende Tags werden automatisch erkannt
 - schließende Tags können oft weggelassen werden
 - Groß/Kleinschreibung der Tags ist nicht festgelegt usw.
 - ⇒ Praktische Konsequenzen
 - Code der gegen viele Regeln eines sauberen Entwurfs verstößt, kann trotzdem HTML5-konform sein
 - um HTML5-Code auf einen ordentlichen Stil zu überprüfen, reicht der normale Konformitätscheck (HTML-Validator) nicht mehr aus
 - ⇒ Lösung
 - HTML5 sollte in professionellen Projekten zusätzlich mit einem statischen Code-Analyse-Tool auf die Einhaltung von Implementierungsrichtlinien geprüft werden (vgl. Lint für C++)

Hochschule Darmstadt

Fachbereich Informatik

2.1.1 HTML Grundlagen



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

fbi

FACHBEREICH INFORMATIK

Tags & Attribute

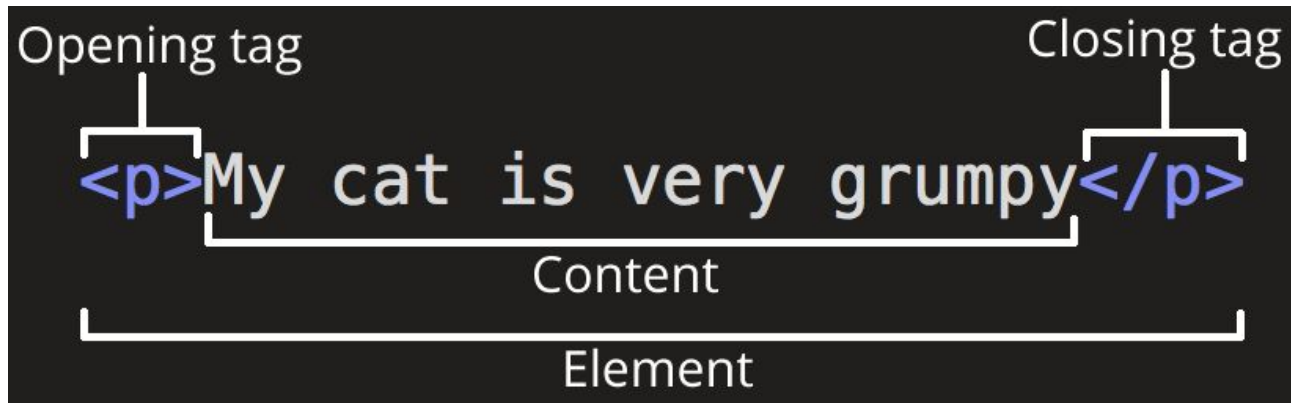


Diagram illustrating an HTML element with an attribute:

```
<p class="editor-note">My cat is very grumpy</p>
```

- Attribute:** `class="editor-note"`



Grundgerüst einer (ordentlichen) HTML5-Datei

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Text des Titels</title>
  </head>
  <body>
    <p>Eigentlicher Inhalt</p>
  </body>
</html>
```



Dokumenttyp

Zeichensatz

Titel für
Browserfenster



Schreibregeln

■ Leerzeichen, Tabulator und Zeilenvorschub sind Trenner

- ⇒ Anzahl spielt keine Rolle, außer in Attributwerten
- ⇒ Ausnahme: in `<pre>` Abschnitten (=preformatted)



■ Einrückung dient nur der Lesbarkeit

- ⇒ wird vom Browser ignoriert

■ Kommentare

`<!-- das ist ein Kommentar -->`


■ Sonderzeichen und Umlaute können kodiert werden

| | | | | | | | |
|---|-------------------------|---|-------------------------|---|--------------------------|---|-------------------------|
| < | <code>&lt;</code> | > | <code>&gt;</code> | & | <code>&amp;</code> | " | <code>&quot;</code> |
| ä | <code>&auml;</code> | Ä | <code>&Auml;</code> | ö | <code>&ouml;</code> | Ö | <code>&Ouml;</code> |
| ü | <code>&uuml;</code> | Ü | <code>&Uuml;</code> | ß | <code>&szlig;</code> | € | <code>&euro;</code> |




Tags (Marken) und Attribute

■ Tags (Marken) markieren Abschnitte im Text

- ⇒ Tag-Name steht in spitzen Klammern
- ⇒ gleicher Name für öffnendes und schließendes Tag
- ⇒ schließendes Tag kenntlich an zusätzlichem /
- ⇒ Der Name des Tags wird kleingeschrieben 
- ⇒ `<h2>Willkommen in unserem Hotel</h2>`

■ öffnende Tags können zusätzliche Attribute enthalten

- ⇒ Attribute haben einen Namen und in der Regel einen Wert
- ⇒ Attributwerte werden in Anführungszeichen geschrieben 
- ⇒ `<h2 id="hallo">Willkommen in unserem Hotel</h2>`

■ mit Tags markierte Abschnitte können verschachtelt sein

`<h2>Willkommen in unserem Hotel</h2>`

Sonderfall Standalone-Tags

- es gibt einige wenige "Standalone-Tags"
 - ⇒ leere Elemente = Abschnitte ohne Inhalt
 - ⇒ werden durch einen / vor der schließenden Klammer des öffnenden Tags hervorgehoben



Willkommen `
`

auf unserer Homepage

oder

`<meta charset="UTF-8" />`

- alternative Schreibweisen (HTML5)

`
` oder `
</br>`

Strukturierung von Text

alle außer
 und

erzeugen einen Block

■ Überschriften

<h1> Überschrift der höchsten Gliederungsebene

<h6> Überschrift der niedrigsten Gliederungsebene

heading1 ... heading6

■ Abschnitte

<p> Textabsatz

<div> allgemeiner Block

div = division = Bereich

 Inline-Element

kein Block

"Aufhänger" für CSS

■ Aufzählungen (nummeriert oder auch nicht)

, ,

ordered list, unordered list, list item

■ Zeilenumbruch erzwingen und verhindern

 expliziter Zeilenumbruch (standalone tag)

kein Block

 geschütztes Leerzeichen – verhindert Zeilenumbruch

­ soft hyphen – Bindestrich bei Bedarf

z.B. 3.
Kapitel

Strukturierung von Webseiten



■ Der Text innerhalb des `<body>`-Tags kann gegliedert werden

- ⇒ `<section>` Abschnitt - ein logischer Bereich einer Webseite (z.B. der News Bereich)
- ⇒ `<article>` Artikel
ein Textabschnitt, der eigenständig einen Inhalt abdeckt (z.B. eine einzelne News)
- ⇒ `<body>` kann mehrere `<section>`s und `<article>`s enthalten – auch verschachtelt
- ⇒ `<nav>` Navigationsbereich – enthält Verknüpfungen zur Navigation
- ⇒ `<header>` und `<footer>` Kopf- / Fußzeilenbereich
ein Bereich mit Überschriften bzw. Logos, Datum usw. für das Gesamtdokument (im `<body>`) oder `<section>`s und `<article>`s
- ⇒ `<aside>` Blöcke an der Seite
z.B. für Seitenleisten (außer Navigation), Zitate, Anmerkungen

`<section>`s und `<article>`s haben eine Überschrift (`<h2>...<h6>`)!



Strukturierung von Webseiten am Beispiel



h_da FB INFORMATIK

body

heading von body

heading der section (<h2>) section

// h_da Fachbereich Informatik / Forschung & Partner

section

nav

Navigation

- Organisation
- Studieninteressierte
- Studium
- Labore
- Verschiedenes
- International
- Forschung & Partner
- Online Belegsysteem
- Intern

Direktlinks

- Stundenpläne
- Personen
- Sitemap
- Suche
- RSS-Feed
- Fachschaft

article

heading des article (<h3>) article

Die Forschung an der Hochschule Darmstadt zielt darauf ab eine wissenschaftlich fundierte Hochschulausbildung auf hohem Niveau zu garantieren. Aufgrund des besonderen Praxisbezugs der Fachhochschulen wird keine Grundlagenforschung betrieben, sondern vielmehr angewandte Forschung und Entwicklung, häufig in Form von Projekten in Kooperation mit externen Unternehmen.

heading des article (<h3>) article

Mit dem Institut für Angewandte Informatik (aida) und dem Zentrum für Angewandte Informatik (Z.A.I.) gehören zwei Institute dem Fachbereich Informatik an.

aida entwickelt in Zusammenarbeit mit Partnerunternehmen Forschungsprototypen und evaluiert neue Informatik-Technologien im Hinblick auf ihre praktische Einsetzbarkeit.

Das Zentrum für Angewandte Informatik e. V. ist ein gemeinnütziger Verein, der zur Förderung der Forschung und Lehre auf dem Gebiet der Angewandten Informatik an der Hochschule Darmstadt gegründet wurde.

Partnerhochschulen

Der Fachbereich Informatik der Hochschule Darmstadt kooperiert zur Zeit mit mehr als 15 ausländischen Hochschulen und hat damit die notwendige Infrastruktur für die Mobilität von Studenten und Dozenten sowie weitergehende transnationale Zusammenarbeit geschaffen.

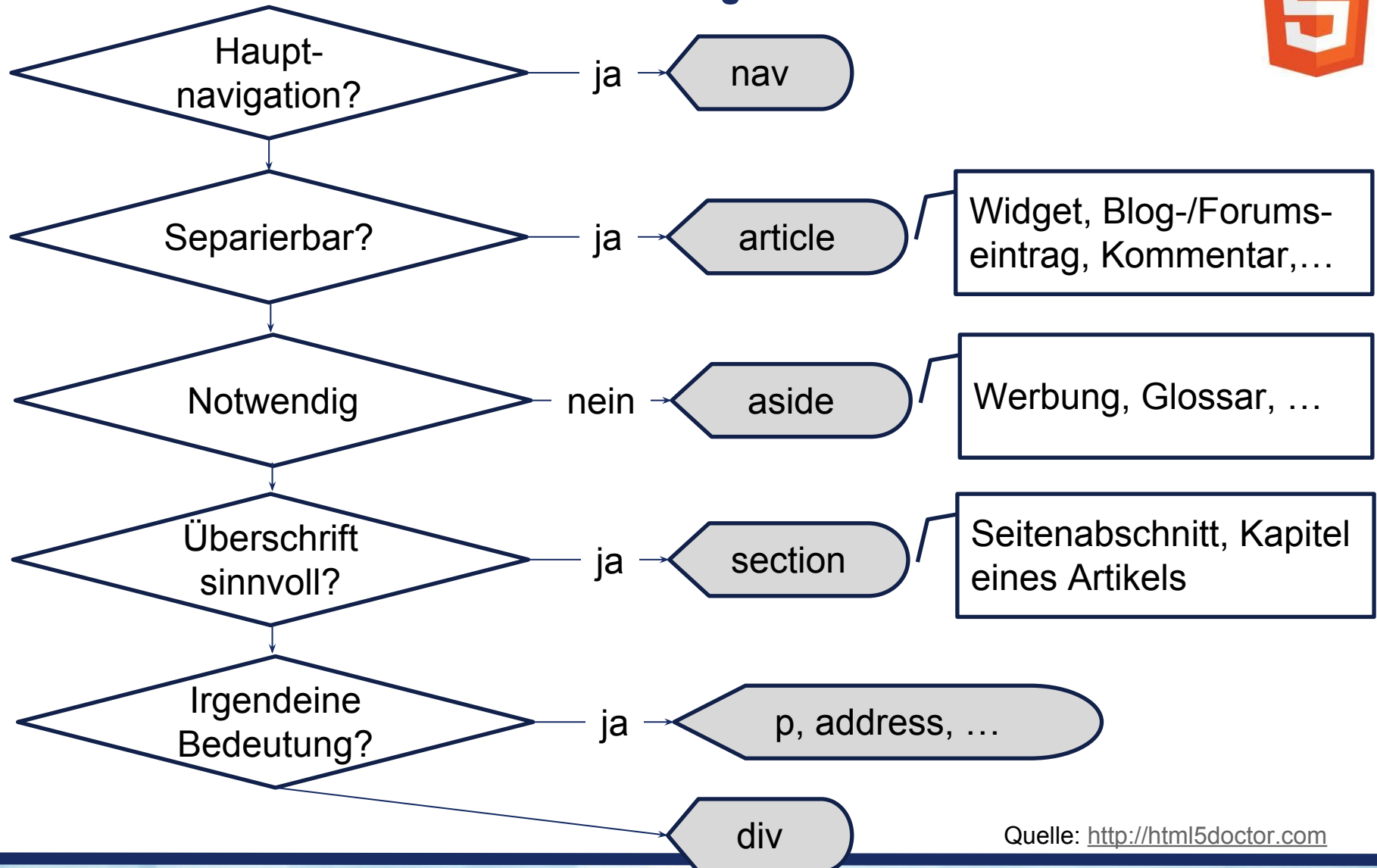
Copyright Fachbereich Informatik Hochschule Darmstadt / Forschung & Partner - Darmstadt / Rhein-Main / Hessen
[Bild: Hochschule Darmstadt] [Logo: Hochschule Darmstadt] [Verantwortung: Dekanat des FB Informatik]

[Impressum]

footer von section

footer (von body)

Welches Strukturelement ist das Richtige?

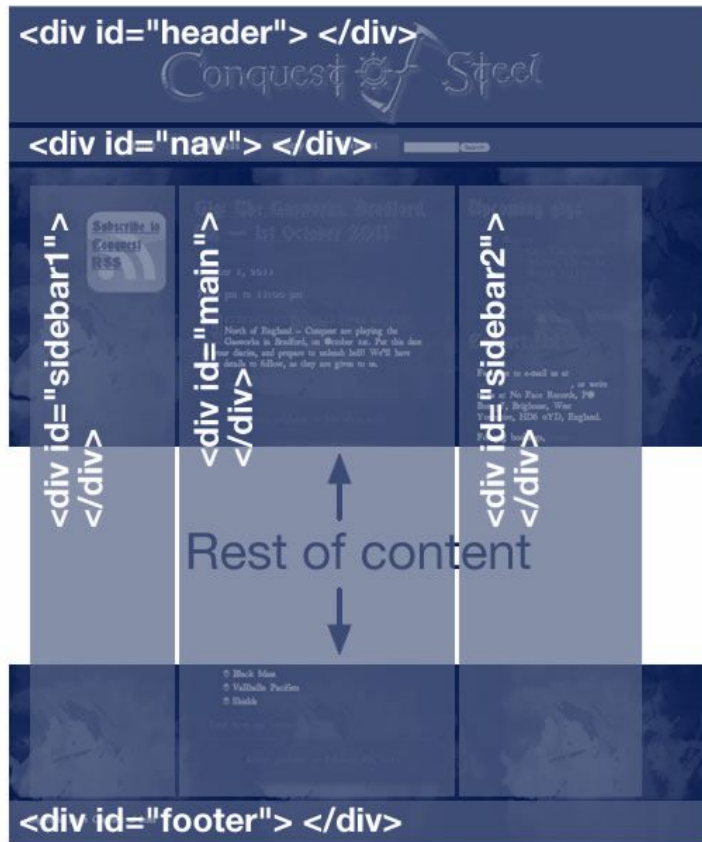


Quelle: <http://html5doctor.com>

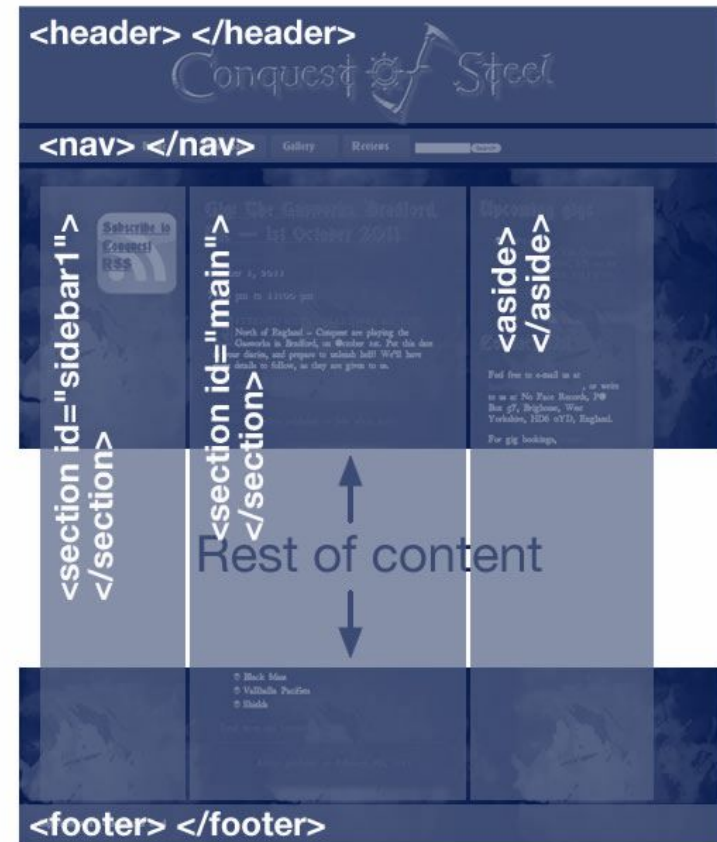
2.1.1 HTML Grundlagen

Vergleich HTML4 und HTML5

■ HTML4



■ HTML5



Quelle: http://www.w3.org/wiki/HTML_structural_elements

Universalattribute

■ können zu jedem Tag hinzugefügt werden

- ⇒ `id` dateiweit eindeutiger Bezeichner für Skripte
- ⇒ `class` Name der zugehörigen Style Sheet Klasse
- ⇒ `title` Erläuterung zum Element, erscheint als Tooltip
- ⇒ `style` eingebettete Style Sheet Attribute (siehe CSS)
- ⇒ `lang, dir` Landessprache und Textlaufrichtung

```
<h2 id="JB007" class="mycssstyleclass" title="mytooltip" style="color:red" lang="de" dir="ltr">
```

Hallo

```
</h2>
```



Weitere HTML5-Attribute



- Oft genutzt: das data-* attribut, um eigene Datentypen zu generieren
 - mit einem Namen ohne Großbuchstaben
 - Bsp.: data-preis, data-key, data-xxx
 - In HTML oder auch über das DOM
 - Wird wichtig, wenn man z.B. HTML Daten per JavaScript auslesen möchte

```
<ul>
  <li data-student-type="erstie">Erstsemesterstudent</li>
  <li data-student-type="slacker">Student im 18. Semester</li>
  <li data-student-type="master">Masterstudent</li>
</ul>
```

Pizza-Service Beispiel:

```
<ol>
  <li data-preis="4.99">Pizza Salami</li>
  ...
</ol>
```



Logische Formatierung



- markiert Bedeutung von Textabschnitten
- macht keine Aussage über visuelles Erscheinungsbild
 - ⇒ das wird erst per CSS definiert
 - ⇒ für Sprachausgabe muss stattdessen das akustische Erscheinungsbild definiert werden...
- ein paar Beispiele:
 - `` emphatisch (betont)
 - `` stark betont
 - `<samp>` Beispiel
 - `<dfn>` Definition
 - `<cite>` inline-Zitat (z.B. für Eigennamen; oft kursiv)
 - `<q cite="URL">` Zitat mit Quellenangabe
 - `<blockquote>` Zitat als Block gesetzt

Block

Grenzfall: physische Formatierung

- definiert das visuelle Erscheinungsbild
- nicht verpönt, aber besser zu vermeiden – es gibt oft andere Tags mit einer echten Bedeutung!
- ein paar Beispiele:
 - `` **fette Schrift (bold)**
 - `<i>` *kursive Schrift (italic)*
 - `<sup>` **Schrift**^{hochgestellt}
 - `<sub>` **Schrift**_{tiefgestellt}
 - `<pre>` vorformatierter Text (nur für Quellcode)

Block

Struktur einer Tabelle (1)

Grundidee:

| <table> | | | | |
|----------|---------------|---------------|---------------|-------|
| <tr> | <th> </th> | <th> </th> | <th> </th> | </tr> |
| <tr> | <td> </td> | <td> </td> | <td> </td> | </tr> |
| <tr> | <td> </td> | <td> </td> | <td> </td> | </tr> |
| </table> | | | | |

nach S. Münz: SELFHTML

Struktur einer Tabelle (2)

nur zur Darstellung
tabellarischer Daten

Tabellen-Überschrift

zeilenweise (tr = table row)

```
<table>
  <caption>Meine Tabelle</caption>
  <tr>
    <th>Kopfzelle: 1. Zeile, 1. Spalte</th>
    <th>Kopfzelle: 1. Zeile, 2. Spalte</th>
  </tr>
  <tr>
    <td>Datenzelle: 2. Zeile, 1. Spalte</td>
    <td>Datenzelle: 2. Zeile, 2. Spalte</td>
  </tr>
  <tr>
    <td>Datenzelle: 3. Zeile, 1. Spalte</td>
    <td>Datenzelle: 3. Zeile, 2. Spalte</td>
  </tr>
</table>
```

beliebig viele Zeilen und Spalten

Struktur einer Tabelle (3)

■ Spaltenbreite definieren

- ⇒ falls der Browser die Spaltenbreite nicht automatisch bestimmen soll, kann über CSS die Breite festgelegt werden

■ Zellen verbinden

- ⇒ Spalten verbinden `<td colspan="3">...</td>`
- ⇒ Zeilen verbinden `<td rowspan="2">...</td>`

Grafiken einbinden

■ Das ``-Tag erlaubt das Einbinden von Bildern

standalone tag

⇒ für Bilddateien der Formate GIF, PNG, JPG und seit HTML5 SVG

⇒ notwendige Attribute:

`src` Quelldatei

`alt` Ersatztext, der ausgegeben wird, wenn die Grafik nicht angezeigt werden kann (Screenreader, Ladeprobleme)

Dient die Grafik nur zur Veranschaulichung

(z.B. Bild einer Pizza neben dem Namen der Pizza)

wird `alt=""` gesetzt

⇒ optionale Attribute

`width`, `height` Breite und Höhe in Pixeln

`title` Text, der als Tooltip angezeigt wird

⇒ Beispiel

```

```

Bilddateien

Per JavaScript
gezeichnete Grafiken
mit `<canvas>`

- GIF oder PNG für Grafiken (z.B. Screenshots, ClipArts)
 - ⇒ GIF: 256 Farben Palette, komprimiert
 - ⇒ Nun: PNG: komprimiert, verlustfrei
 - ⇒ Freistellen (transparenter Hintergrund) möglich
- JPEG für Fotos
 - ⇒ Echtfarben, komprimiert, verlustbehaftet
 - ⇒ Freistellen nur bei PNG
- SVG (Scalable Vector Graphics) für technische Zeichnungen
 - ⇒ XML-basiertes Format für 2D-Vektorgrafiken
 - ⇒ ohne Qualitätsverlust skalierbar
- Möglichkeiten zur Speicherplatzersparnis
 - ⇒ Hintergrund mit kleiner Grafik "kacheln"
 - ⇒ Größe und Farbtiefe reduzieren
 - ⇒ Beschränkung auf wenige Grafiken



Hauptproblem
Übertragungszeit!

Audio und Video



- Audio und Video gehören erst seit HTML5 zum Standard
 - ⇒ Vor HTML5 waren Multimediadateien nur über Plugins (z.B. FlashPlayer) abspielbar (und die Plugins waren nicht betriebssystemübergreifend verfügbar)
 - ⇒ Seit HTML5 gibt es "nur noch" Uneinigkeit über die vom Browser zu unterstützenden Formate (wegen Lizenzrechten)
 - ⇒ Damit die AV-Datei von allen Browsern abgespielt wird, müssen derzeit 2 Formate bereitgestellt werden

- mp3 und ogg für Audio bzw. ogv und mp4 für Video

```
<audio autoplay controls>
```

```
  <source src="myAudio.mp3" />
```

```
  <source src="myAudio.ogg" />
```

```
</audio>
```

```
<video width="320" height="240" poster="bild.jpg" control
```

```
  <source src="myVideo.mp4" />
```

```
  <source src="myVideo.ogv" />
```

Ihr Browser unterstützt leider kein Video-Tag

```
</video>
```

Der Browser wählt
ein Format, das er
unterstützt!

- ⇒ Abspielen erfolgt im "streaming mode" wegen der Übertragungszeit

Meta-Angaben

- Anweisungen für WWW-Server, WWW-Browser und automatische Suchprogramme ("Robots")

- eine kleine Auswahl von Meta-Angaben:

```
<meta name="description" content="Autovermietung" />
```

```
<meta name="author" content="B. Kreling" />
```

```
<meta name="keywords" content="Hotel,Urlaub,Meer" />
```

```
<meta name="robots" content="noindex" />
```

```
<meta name="date" content="2001-02-06" />
```

```
<meta name="language" content="de" />
```

```
<meta http-equiv="refresh" content="5">
```

lädt die aktuelle
Seite nach 5
Sekunden erneut

- Setzen des verwendeten Zeichensatzes

```
<meta charset="UTF-8">
```