

Basics

HTML5 & CSS3



Hyper Text Markup Language

- Text oder Hypertext
- Auszeichnen der Inhalte
- Semantik (=Bedeutungslehre)
- Interpretersoftware (=Browser)
- W3C Standards
- Dokumenttyp-Definition (DTD)



!DOCTYPE (DTD)

Welchen Standard verwenden?

- HTML4

`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`

- XHTML1.0 Transitional oder Strict

`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`

- XHTML Mobile Profile

`<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN" "http://www.wapforum.org/DTD/xhtml-mobile10.dtd">`

- HTML5

`<!DOCTYPE html>`



der HTML-Tag

Programmieren im Texteditor

- Tag Beginn: `< ... >`
- Tag Ende: `</ ... >`
- leerer Tag HTML: `< ... >`
- leerer Tag XHTML: `< ... />`
- Erweiterungen: `< ... attribut = " WERT " >`

Beispiel:

`<p id="test" align="right">Das ist Text</p>`



XHTML Grundgerüst

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <head>  
    <title>Titel des Dokuments</title>  
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  </head>  
  
  <body>  
  
  </body>  
</html>
```



HTML5 Grundgerüst

Hinweis: Validieren Sie Ihren Code! (<http://validator.w3.org>)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Titel des Dokuments</title>
</head>

<body>
</body>
</html>
```



XHTML vs HTML

XHTML ist eine XML Anwendung daher gelten Regeln aus XML

- alle Tags und Attribute müssen kleingeschrieben werden
- leere Tags `< ... />`
- alle Attributwerte in doppelten Anführungszeichen
- XML Namespace im Wurzelement `<html xmlns="...">`
- Anker mit id
- lang und xml:lang
- Inhalte von Scriptbereichen auskommentieren
`/* <![CDATA[*/`
`/*]]> */`
- strengere Verschachtelungsregeln (z.B. im `<a>` kein weiteres `<a>`)



Strict vs Transitional

Welcher Standard ist besser?

- Strict bedeutet eine strikte Trennung von Inhalt und Formatierung (HTML und CSS), bei Transitional können Formatierungen auch im HTML stehen (z.B. `<p align="left">`)
- DOCTYPE-Switch führt nur bei Strict den Browser zur Ausgabe im „Full-Standard-Mode“ (weniger Browserunterschiede)
- im Strict ist das Attribut `target` bei Hyperlinks verboten (öffnen von Links im neuen Fenster nicht ohne Javascript möglich)



Unterschiede HTML4 und 5

ACHTUNG: HTML5 ist noch im „draft“-Status!

- keine DTD in HTML5
- neue Elemente (z.B.: section, article, nav, video, ...)
- neue Attribute vor allem beim Tag „input“
- Semantikänderung bei Elementen (z.B. strong, em)
- Änderungen von Attributen (z.B. <table border="">)
- Entfernte Elemente (z.B. center, font, acronym)

Änderungen zu HTML4: <http://www.w3.org/TR/html5-diff/>
komplette Standard: <http://www.w3.org/TR/html5/>



Head-Inhalte

Zusatzinformationen zum Dokument

- `<title>` (Seitentitel, SEO, Bookmarks)
- Charset mittels `<meta>`
- Metadaten für Suchmaschinen
`<meta name="keywords" content="....">`
`<meta name="description" content="....">`
- Suchmaschinenindizierung
`<meta name="robots" content="index, follow">`
- Stylesheets (`<style>` oder `<link>`)
- Javascript einbetten mit `<script>`
- Favicon mit `<link rel="icon" href="..." type="...">`

Inline – Block Konzept

- Seite (<body>) beinhaltet Blöcke (=Boxen, Blockelemente)
- Blöcke können verschachtelt werden
Achtung: Einschränkungen (z.B. <p><p> ist nicht erlaubt)
- Blöcke stehen untereinander, sind so hoch wie ihr Inhalt und nehmen sich die maximal zur Verfügung stehende Breite
- Auf Blöcke kann das Boxmodell angewendet werden! (siehe Abschnitt CSS)
- Der Inhalt (Text) wird innerhalb von Boxen in Zeilen ausgegeben.
- Zeilenumbrüche können durch den Tag
 erzwungen werden.
- Inline Elemente können fast beliebig verschachtelt werden, jedoch darf innerhalb eines Inline-Elements kein Block-Element kommen



Block-Elemente (XHTML1)

in Klammer finden Sie die semantische Bedeutung

- `<div>` (Bereich)
- `<p>` (Absatz)
- `<h1>` bis `<h6>` (Überschriften)
 - `<h1>` (Seitenüberschrift, normalerweise 1x pro Dokument)
 - `<h2>` (Abschnittsüberschrift)
 - `<h3>` (Kapitelüberschrift)
 - `<h4>` (Unterkapitel)
 - `<h5>` und `<h6>` (werden häufig für Sidebars und Nebeninhalte verwendet)
- `<blockquote>` (Zitat)
- `<form>` (Formular)
- ``, `` (Listen, ungeordnet oder geordnet)
- `` Listenpunkt
- `<address>` (Adressbereich)
- `<pre>` (vorformatierter Text)



Inline-Elemente (XHTML1)

- `` (allgemeiner Inline-Bereich, ohne semantische Bedeutung)
- `<a>` (Anker)
- `` (Hyperlink, Anker mit Ziel)
- `` (hervorgehoben)
- `` (stark hervorgehoben)
- `<input>` (Eingabefeld)
- `<cite>` (Zitat)
- `<button>` (Button)

spezielles Element ``

Inline-Block-Elemente, Pflichtattribute `src` und `alt`



Pfade in HTML

- absolute Pfade
`http://subdomain.domain.tld/ordnername/ordnername/datei.extension`
- relative Pfade (ausgehend vom Ordner in dem sich die aktuelle Datei befindet)
 - `datei.ext`
 - `subordner/datei.ext`
 - `../datei.ext`
 - `../ordner/datei.ext`
- relativ vom Serverroot `/ordner/ordner/datei.ext`
- Ankerpunkte (`URL#name`)
- dynamische GET-URLs (`URL?variable=wert&variable=wert`)



Farben in HTML bzw. CSS

- vordefinierte Werte als symbolische Konstanten:
black, white, red, green, ...
- Hex-Codes:
#05af34, #000000
Kurzform #c5a bedeutet #cc55aa
- RGB
rgb(134,56,0)
- RGBA (+Alphakanal, ab CSS3!)
rgba(134,56,0,0.65)



Tabellen

Anm.: Es gibt kein Element für Spalte!

- `<table summary="...">`
- `<caption>`
- `<thead>`, `<tbody>`, `<tfoot>`
- `<tr>` (=Zeile)
- `<th scope="...">` (=Kopfzelle)
- `<td>` (=Zelle)
zusammengefasste Zellen (`<td rowspan="..." colspan="...">`)



Formulare (HTML4, XHTML)

- `<form method="" action="">`
- `<input type="text">`
- `<input type="password">`
- `<input type="checkbox">`
- `<input type="radio">`
- `<input type="hidden">`
- `<input type="file">`
- `<select><option value="file">`
- `<textarea>`
- `<input type="submit">` oder `<button></button>`



CSS

Cascading Stylesheets (geschachtelte Gestaltungsvorlagen)

- `p { text-align:left; color:red; }` CSS-Regel
- `p` { text-align:left; color:red; } Selektor
- `p { text-align:left; color:red; }` Deklarationsblock
- `p { text-align:left; color:red; }` Deklaration
- `p { text-align:left; color:red; }` Eigenschaft
- `p { text-align:left; color:red; }` Wert



CSS einbinden

Anm.: unterschiedliche Ausgabemedien können berücksichtigt werden (all, aural, print, handheld, screen, ...)

- Inline CSS
`< ... style="text-align:left">`
- im Dokument (meist im `<head>`)
`<style>CSS-Regeln</style>`
- externe CSS Datei (beliebig.css)
`<link rel="stylesheet" href="..." type="text/css">`



kaskadieren?

Mehrere CSS Regeln können bei einem Element wirken!

Aber welche Formatierung gilt nun?

- Gilt Regel für das Element und das Ausgabemedium?
- Spezifität einer CSS Regel (Inline-CSS ist immer am Stärksten!)
- !important
- bei gleichwertigen Regeln gilt die zuletzt angegeben (im Quellcode weiter unten stehende Regel)



Selektoren

- Universalselektor * {...}
- Typ- oder Tagselektor (z.B. p {...})
- Klassenselektor
HTML: <p class="hinweis">
CSS: .hinweis { ... }
- ID-Selektor
HTML: <p id="hinweis">
CSS: #hinweis { ... }
- Pseudoklassenselektor (z.B. :focus {...})
- Attributabhängige Selektoren
CSS: [Attribut=WERT]



Selektoren kombinieren

- p.hinweis (Absatz mit Klasse)
- body #wrapper (verschachtelte Elemente: Das Element mit der ID wrapper muss irgendwo innerhalb des Tags body stehen!)
- p > a (nur wenn der Hyperlink direkt innerhalb des Absatzes steht)
- p * a (der Hyperlink ist min. zwei Ebenen tiefer als der Absatz)
- p + ul (Die Liste folgt direkt auf einen Absatz)



CSS Größenwerte

- Eigenschaft: **Wert Einheit**;
- Kommawerte mit Punkt ! (z.B.: 0.65)
- Einheiten Screen:
 - px (Pixel)
 - % (Prozent)
 - em (relative Größe, 1em = 100% Schriftgröße)
- Einheiten Druck:
 - cm (Zentimeter)
 - mm (Milimeter)
 - pt (Punkt, 1pt = 3,528 mm)
 - % (Prozent)
- Einheiten CSS3
 - deg (Grad, Kreis = 360deg)
 - rad (Radiant, 1rad = 57,296deg)
 - s (Sekunden)
 - ms (Millisekunden)



CSS Eigenschaften Schrift/Text

- font-family: Schriftart;
- font-size: Schriftgröße;
- color: Schriftfarbe;
- font-weight: normal, bold, bolder, lighter, 100-900;
- font-style: normal, italic;
- font-variant: normal, small-caps; (=Kapitälchen)
- line-height: Zeilenhöhe;
- text-decoration: none, underline, line-through, overline;
- text-transform: lowercase, uppercase, none;
- text-align: left, right, center, justify;
- letter-spacing: Zeichenabstand;
- word-spacing: Wortabstand;



CSS Eigenschaften Schrift/Text

- font-family: Schriftart;
- font-size: Schriftgröße;
- color: Schriftfarbe;
- font-weight: normal, bold, bolder, lighter, 100-900;
- font-style: normal, italic;
- font-variant: normal, small-caps; (=Kapitälchen)
- line-height: Zeilenhöhe;
- text-decoration: none, underline, line-through, overline;
- text-transform: lowercase, uppercase, none;
- text-align: left, right, center, justify;
- letter-spacing: Zeichenabstand;
- word-spacing: Wortabstand;



das Boxmodell

- margin: Außenabstand;
- border: Dicke Art (solid, dotted, dashed) Farbe;
- padding: Innenabstand
- Contentbox (innerhalb des paddings entsteht eine unsichtbare Box)
- width: Breite der Contentbox;
- height: Höhe der Contentbox;
- overflow: hidden, auto, visible;
- background: Hintergrund (siehe nächste Seite)
- position: Positionierung von Elementen;
- float: schwebende Elemente



der Hintergrund

background: fasst alle Eigenschaften zusammen (mit Leerzeichen getrennt die Werte angeben)

- background-color: Farbe;
- background-image:
 - url(Pfad zur Datei, relativ von CSS Datei aus)
 - in CSS3 auch Farbverläufe
- background-position: „horizontaler Abstand von links“ „vertikaler Abstand von oben“ (z.B. 10px 25px) oder mit „left, center, right“ und „top center bottom“
- background-repeat: no-repeat, repeat-x, repeat-y, repeat;
- background-attachment: scroll, fixed;

Vielen Dank für
Ihre Aufmerksamkeit!

