

# XHTML und CSS (Cascading Style Sheets)



**Sandro Bauer**

**02. September 2008**

**Studienstiftung des deutschen Volkes**

**Sommerakademie Greifswald 2008**

# Entstehung der Markupsprachen

---

## □ Geschichte des „World Wide Web“ als Teilnetz des Internet

### ■ Ziel

- Nutzung des Internet zur Publikation wissenschaftlicher Dokumente
- Möglichkeit der Einbindung von Bildern und anderen medialen Inhalten
- Ermöglichung der Hypertextfunktionalität

### ■ Entwicklung neuartiger Internetstandards

- Gründerväter des Internet: Tim Berners-Lee (heute Direktor des W3C-Konsortiums) und Marc Andreessen
- Konzeption von HTTP (Hypertext Transfer Protocol) und HTML (Hypertext Markup Language)

# Entstehung der Markupsprachen

---

## □ Entwicklung von HTML als Markupsprache

### ■ XMosaic als erster grafischer Webbrowser

### ■ Konzept der Auszeichnungssprachen

#### □ Komponenten von Textdokumenten

- Textdaten
- Struktur
- Formatierung

#### □ Typen von Auszeichnungen

- Prozedurale Auszeichnungen
- Deskriptive Auszeichnungen
- Visuelles Markup

### ■ Konzeption von HTML als SGML-konforme Sprache

### ■ HTML als Auszeichnungssprache

Historie
Entstehung der Markupsprachen
Versionsgeschichte
<b>XHTML</b>
Grundlegender Aufbau eines XHTML-Dokuments
XHTML als Nachfolger von HTML
Unterschiede zwischen HTML und XHTML
Modularisierung von XHTML
Ausblick: XHTML 2.0
XHTML als Weg zum semantischen Markup
<b>CSS</b>
Grundlegendes
Praktisches Beispiel
Fortgeschrittene Selektoren
<div> und <span>
Positionierung von Elementen
Arten von Stilvorlagen
media und @page

# Versionsgeschichte von HTML

---

- **HTML 1.0 (Inoffizielle Entwicklungsdefinition, „draft document“)**
  - Standard-Elemente wie Überschriften, Absätze, Links und Grafiken
- **HTML 2.0**
  - Erster offizieller HTML-Sprachstandard
  - Einführung von Formulartechniken mit POST- und GET-Übertragung
  - Sehr beschränkt in der Anwendung (z. B. fehlende Unterstützung für <frame> und <frameset>)
  - In der Praxis durch eigene HTML-Dialekte des Netscape-Browsers verdrängt → Zunehmende Nichtbeachtung der Standards durch Browsersoftware

„Then we descended into the Dark Ages for several years, because the Web exploded into a community that had no idea that such freedom was possible, but worried about putting on the remote screen exactly what they thought their information should look like. (...) Sad.“ (Robert Cailliau, CERN)

Historie
Entstehung der Markupsprachen
▶ Versionsgeschichte
<b>XHTML</b>
Grundlegender Aufbau eines XHTML-Dokuments
XHTML als Nachfolger von HTML
Unterschiede zwischen HTML und XHTML
Modularisierung von XHTML
Ausblick: XHTML 2.0
XHTML als Weg zum semantischen Markup
<b>CSS</b>
Grundlegendes
Praktisches Beispiel
Fortgeschrittene Selektoren
<div> und <span>
Positionierung von Elementen
Arten von Stilvorlagen
media und @page

# Versionsgeschichte von HTML

---

## □ HTML 3.0/3.2

- **Version 3.0 wird nicht veröffentlicht**
- **Neue Funktionalitäten in Version 3.2 (veröffentlicht im Januar 1997): Tabellen, Appleteinbindung, Textfluss um Grafiken**
- **Verhandlungen mit namhaften Browserherstellern → Integration proprietärer HTML-Erweiterungen von Microsoft und Netscape in die offizielle Spezifikation**
- **Ziel: Wiederherstellung eines einheitlichen Standards**
- **Problem: Trennung von Layout und Struktur immer weniger beachtet**

Die neue Spezifikation schließt Tables, Applets und Textfluß um Bilder ein und soll Abwärtskompatibilität zum derzeit gültigen HTML 2.0 gewährleisten. An der Entwicklung waren führende Firmen wie IBM, Microsoft, Netscape, Novell, SoftQuad, Spyglass und Sun Microsystems beteiligt. HTML 3.2 baut auf HTML 2.0 auf. Vorschläge wie HTML+ und HTML 3.0 werden damit anscheinend ungültig. Einige Tags wie <CENTER> und <FONT> werden offiziell, andere HTML 3.0 Tags wie <FIG> fallen weg.

**(heise.de, 10.05.1996)**

<b>Historie</b>
Entstehung der Markupsprachen
▶ Versionsgeschichte
<b>XHTML</b>
Grundlegender Aufbau eines XHTML-Dokuments
XHTML als Nachfolger von HTML
Unterschiede zwischen HTML und XHTML
Modularisierung von XHTML
Ausblick: XHTML 2.0
XHTML als Weg zum semantischen Markup
<b>CSS</b>
Grundlegendes
Praktisches Beispiel
Fortgeschrittene Selektoren
<div> und <span>
Positionierung von Elementen
Arten von Stilvorlagen
media und @page

# Versionsgeschichte von HTML

---

## □ HTML 4.0/4.01

- Letzte Version des „klassischen HTML“ vor der Redefinition durch XHTML 1.0
- Anpassung an die neue Rolle des Internet als E-Commerce- und Kommunikationsplattform
- Große Anzahl neuer Funktionalitäten
  - Einbindung clientseitiger Skriptsprachen (v. a. JavaScript)
  - Cascading Stylesheets (CSS) zur Trennung von Inhalt und Design
  - Framesets
  - Eingebundene Objekte
  - Internationalisierung (Unterstützung von Unicode)
  - Verbesserungen der Barrierefreiheit (Braille, akustische Informationen, Manövrieren per Tastendruck)
- Bereitstellung verschiedener DTDs

# Grundlegender Aufbau eines XHTML-Dokuments

## XML-Deklaration

An dieser Stelle wird deutlich: XHTML basiert ausdrücklich auf XML und nicht mehr länger auf SGML!

Einleitungstag

Kopfbereich des XHTML-Dokuments (Dokumenttitel, Stylesheets, Metadaten, Skripte)

Rumpfbereich des XHTML-Dokuments (Raum für den eigentlichen Inhalt des Dokuments, z. B. Text, Bilder, Objekte, div-Container etc.)

```
<?xml version="1.0" ?>
<!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</title>
</head>
<body>
</body>
</html>
```

Verweis zur Dokumenttypdeklaration (DTD)

Angabe des verwendeten XML-Namensraums

# Grundlegender Aufbau eines XHTML-Dokuments

---

## □ Sprachelemente in XHTML-Dokumenten (Auswahl):

### ■ Meta-Tags

- liefern Informationen für Browser, Server und Suchmaschinen
- dienen der Einordnung und Katalogisierung von Webseiten (z. B. stichwortgebundene Suche in Suchmaschinen)
- **Verbreitete Meta-Attribute:**
  - **description:** Kurzbeschreibung der Webseite, die in Suchmaschinen wiedergegeben wird
  - **author:** Copyright-Angabe
  - **keywords:** erleichtert Suchmaschinen die Zuordnung zu bestimmten Suchbegriffen
  - **date:** Veröffentlichungsdatum (z. B. zur Versionskontrolle)
- **Konzepte zur Standardisierung der Meta-Angaben:** Dublin Core, RDF



# Grundlegender Aufbau eines XHTML-Dokuments

---

## □ Weitere Beispiele:

### ■ Metatags zur Browsersteuerung

- `<meta http-equiv="expires" content="0" />`

Neuabruf des Seiteninhalts bei jedem Seitenaufruf (keine  
Verwendung des Browsercaches)

- `<meta http-equiv="refresh" content="5;  
URL=http://www.bayern.de" />`

(im Bsp.) Weiterleitung zur Adresse `http://www.bayern.de` nach 5  
Sekunden Wartezeit

- Zugriffssteuerung mit PICS-Labeln (nicht näher behandelt)
- Angabe der Zeichencodierung (wichtig zur Unterscheidung  
zwischen ISO- und Unicode-Codierung)

`<meta http-equiv="content-type" content="text/html;  
charset=ISO-8859-1" />`

- Missbrauch in WYSIWYG-Editoren zur Kennzeichnung der benutzten  
Software („generator“)

# Grundlegender Aufbau eines XHTML-Dokuments

---

## ■ Logische Linkbeziehungen

```
<link rel="author" title="Impressum" href="../../impr.html" />  
<link rel="contents" title="Inhalt" href="../../content.html" />  
<link rel="index" title="Stichwortvz." href="/glossar.htm" />  
<link rel="top" title="SELFHTML" href="../../index.htm" />  
<link rel="up" title="HTML-Kopfdaten" href="index.htm" />
```

- Möglichkeit zur internen Repräsentation von Hypertext-Abhängigkeiten
- Abbild einer logischen Seitenhierarchie
- Möglichkeit zur automatischen Erstellung virtueller Landkarten von Internetseiten („Sitemaps“)
- Automatische Erstellung vollwertiger Inhaltsangaben Im Zusammenspiel mit seiteninternen Metaangaben
- Mangelnde Browserunterstützung

# Grundlegender Aufbau eines XHTML-Dokuments

## ■ Tags zur Textstrukturierung

- Überschriften (<h1>, <h2>, ... bis <h6>)
- Absätze (<p>)
- Zeilenumbrüche (<br />)
- Unsortierte Listen (<ul><li>...</li><li>...</li></ul>)
- Sortierte Listen (<ol><li>...</li><li>...</li></ol>)
- Definitionslisten (<dl><dt>...</dt><dd>...</dd>...</dl>)
- Logische Textauszeichnungen
  - Zitate
  - Adressen
  - Präformatierter Text
  - Emphasen
  - Quelltext
  - Definitionen
  - Abkürzungen

### Wichtig:

**Keine Aussage über visuelle  
Gestaltung und Positionierung!  
XHTML-konforme Tags zur  
Textstrukturierung treffen  
semantische, nicht  
gestalterische Aussagen!**

Historie
Entstehung der Markupsprachen
Versionsgeschichte
<b>XHTML</b>
▶ Grundlegender Aufbau eines XHTML-Dokuments
XHTML als Nachfolger von HTML
Unterschiede zwischen HTML und XHTML
Modularisierung von XHTML
Ausblick: XHTML 2.0
XHTML als Weg zum semantischen Markup
<b>CSS</b>
Grundlegendes
Praktisches Beispiel
Fortgeschrittene Selektoren
<div> und <span>
Positionierung von Elementen
Arten von Stilvorlagen
media und @page

# Grundlegender Aufbau eines XHTML-Dokuments

---

## ■ Allgemeine block- und inline-Elemente (<div> und <span>)

- <div>: Generisches Blockelement
- <span>: Generisches Inlineelement
- Zweck: einheitliche Formatierung und Positionierung zusammengehöriger Elemente mit CSS
- Erkennbare inhaltliche Gliederung des body-Bereichs
- Kritik:
  - Häufig unübersichtliche Verschachtelungen von <div>-Containern über mehrere Hierarchieebenen hinweg ([Beispiel](#))
  - Fehlende semantische Strukturierung durch <div>-Häufungen
  - CSS-Formatierungen mit <span>-Elementen unhandlicher als die ursprünglichen HTML-Elemente wie <b>, <i>, etc.

# Grundlegender Aufbau eines XHTML-Dokuments

## ■ Tabellen

- Syntax (Quelle: [de.selfhtml.org](http://de.selfhtml.org)):

```

<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>
    
```

**Tabellenlayout  
ist nicht Sache  
von XHTML!**

- **Bestimmung in XHTML:** Darstellung originär tabellarischen Datenmaterials
- Einsatz zur Gestaltung von Layouts ist gemäß XHTML-Spezifikation unbedingt zu vermeiden
- Oft vergessen: Möglichkeit zur Angabe von Tabellenüberschriften (<caption>, Tabellenheadern (<thead> und -footern (<tfoot>)  
→ Weitergehende logische Gliederung tabellarischer Daten (hilfreich z. B. für Printmedien)
- Möglichkeit zur Verbindung mehrerer Zellen analog zu Tabellenkalkulationen (colspan/rowspan-Attribute)

# Grundlegender Aufbau eines XHTML-Dokuments

## ■ Verweise („Links“) – die Lebensadern des Internets

- Syntax: `<a href="www.bayern.de">Bayern</a>`
- Unterscheidung verschiedener Verweistypen
  - absolute Pfadangabe
  - relative Pfadangabe
  - dokumentinterne Verweise
- Problematik des target-Attributs
- Typisierte Verweise
- Manuelle Festlegung von Tastenkürzeln und Tab-Reihenfolge möglich
- Visuelle Effekte (z. B. Unterstreichen von Links im Fokus) früher oft mit JavaScript (heute durch CSS abgedeckt)  
`<tr onmouseover="style.backgroundColor='red'"  
onmouseout="style.backgroundColor='black'"> ...`

# Grundlegender Aufbau eines XHTML-Dokuments

---

## ■ Grafiken

- Syntax: ``
- Größe und visuelle Effekte werden vollständig durch CSS festgelegt
- Alternativtext ist in XHTML erforderlich
- Image-Maps

## ■ Formulare

- Ein- und mehrzeilige Textfelder, Auswahl von Einträgen in Dropdownboxen, Mehrfachauswahllisten, Radiobuttons, Checkboxes, Buttons, Dateiupload
- Übertragung zum Webserver mit POST- (direkte Übertragung an das festgelegte Formularziel) oder GET-Methode (z. B. `index.htm?var1=abc&var2=def&var3=ghi`)

# Grundlegender Aufbau eines XHTML-Dokuments

---

## ■ Einbindung von Objekten

### □ früher zahllose verschiedene Elemente

- <applet> für Java-Applets
- <img> für Bilddateien
- <bgsound> für Hintergrundklänge
- <embed> (Netscape-Eigenkonstruktion)
- <marquee> (Lauftext, Microsoft-Eigenkonstruktion)
- usw.

### □ Einführung einer einheitlichen Syntax durch XHTML 1.0:

```
<object data="ein_object.svg" type="image/svg+xml">
```

```
<param name="src" value="ein_object.svg" />
```

Alternativtext

```
</object>
```



<b>Historie</b>
Entstehung der Markupsprachen
Versionsgeschichte
<b>XHTML</b>
Grundlegender Aufbau eines XHTML-Dokuments
▶ XHTML als Nachfolger von HTML
Unterschiede zwischen HTML und XHTML
Modularisierung von XHTML
Ausblick: XHTML 2.0
XHTML als Weg zum semantischen Markup
<b>CSS</b>
Grundlegendes
Praktisches Beispiel
Fortgeschrittene Selektoren
<div> und <span>
Positionierung von Elementen
Arten von Stilvorlagen
media und @page

# XHTML als Nachfolger von HTML

- XML als einfacher zu handhabende Untermenge von SGML
- XHTML als Neudefinition von HTML 4.0 auf der Grundlage von XML (statt früher SGML)
  - XHTML 1.0 enthält (in der transitional-Variante) alle Elemente von HTML 4.01 (vollständige Kompatibilität)
  - Verarbeitung mit nicht XHTML-fähigen Browsern möglich
  - Geringere Fehlertoleranz durch Forderung der Wohlgeformtheit gemäß XML-Richtlinien
  - Endgültige Eliminierung von Layout-Elementen aus der Beschreibungssprache XHTML

## Historie

Entstehung der  
Markupsprachen

Versionsgeschichte

## XHTML

Grundlegender Aufbau eines  
XHTML-Dokuments

XHTML als Nachfolger von  
HTML

▶ Unterschiede zwischen HTML  
und XHTML

Modularisierung von XHTML

Ausblick: XHTML 2.0

XHTML als Weg zum  
semantischen Markup

## CSS

Grundlegendes

Praktisches Beispiel

Fortgeschrittene Selektoren

<div> und <span>

Positionierung von Elementen

Arten von Stilvorlagen

media und @page

# Unterschiede zwischen HTML und XHTML

- Strengeres Einhalten des (X)HTML-Gerüsts (z. B. keine Möglichkeit des Weglassens des <head>-Bereichs mehr)
- Obligatorische Kleinschreibung von Element- und Attributnamen (case-sensitive) – früher übliche Schreibweisen wie <HTML> sind nicht XHTML-konform
- Anpassung ursprünglich inhaltsleerer Elemente wie <br> mittels einer speziellen Syntax
- Zwang zur Notation ursprünglich optionaler Abschlusstags
- Zwang zur Notation von Attributwerten (stets in Anführungszeichen)
- Einführung von Universalattributen
- Keine Überkreuzung von Elementen (<b><i>...</b></i>)

Historie
Entstehung der Markupsprachen
Versionsgeschichte
<b>XHTML</b>
Grundlegender Aufbau eines XHTML-Dokuments
XHTML als Nachfolger von HTML
Unterschiede zwischen HTML und XHTML
▶ Modularisierung von XHTML
Ausblick: XHTML 2.0
XHTML als Weg zum semantischen Markup
<b>CSS</b>
Grundlegendes
Praktisches Beispiel
Fortgeschrittene Selektoren
<div> und <span>
Positionierung von Elementen
Arten von Stilvorlagen
media und @page

# Modularisierung von XHTML

- Darstellung von (X)HTML auch durch kleinere Geräte (PDAs, Mobiltelefone)
- Problematik: Zu umfangreicher Funktionskatalog
- In der Vergangenheit: Entwicklung eigener HTML-Derivate für mobile Geräte
- Lösung: XML-konforme Modularisierung der HTML-Elemente durch XHTML
- Bsp. (Quelle: <http://www.teialehrbuch.de/Kostenlose-Kurse/Markup-Sprachen/16607-XHTML-Modularisierung.html>):

## Strukturmodul

Das Strukturmodul definiert die grundlegenden Elementtypen zur Auszeichnung der Struktur von XHTML-Dokumenten. Dieses Modul besteht aus den Elementtypen body, head, html und title.

## Textmodul

Das Textmodul definiert Elementtypen zur Auszeichnung von Textinformationen. Es beinhaltet sowohl die Elementtypen für Überschriften (h1 bis h6) als auch den Elementtyp p für Absätze und zahlreiche andere Elementtypen zur logischen Auszeichnung wie code und samp.

## Hypertext-Modul

Das Hypertext-Modul definiert den Elementtyp a, der für die Auszeichnung von Links und Link-Enden verwendet wird.

## Historie

Entstehung der  
Markupsprachen

Versionsgeschichte

## XHTML

Grundlegender Aufbau eines  
XHTML-Dokuments

XHTML als Nachfolger von  
HTML

Unterschiede zwischen HTML  
und XHTML

▶ Modularisierung von XHTML

Ausblick: XHTML 2.0

XHTML als Weg zum  
semantischen Markup

## CSS

Grundlegendes

Praktisches Beispiel

Fortgeschrittene Selektoren

<div> und <span>

Positionierung von Elementen

Arten von Stilvorlagen

media und @page

# Modularisierung von XHTML

---

- Zusammenfassung selten benötigter Elemente in eigenen Modulen
- Leichtere Aussagen über den Funktionsumfang von Browsersoftware
- XML-gemäße Erweiterbarkeit von XHTML
- Praktische Anwendung: XHTML Basic
  - Minimale XHTML-Variante für Mobiltelefone, Settopboxen etc.
  - Keine Unterstützung für Framesets und Skripttechnologien

# Ausblick: XHTML 2.0

- **Neue Sprachelemente und Strukturen, z. B.**
  - **Teilweise Ablösung des Anker-Elements `a` durch Listen-Elemente (`li`)**  
Bsp.: `<li href="http://www.bayern.de">Bayern</li>` statt  
`<li><a href="http://www.bayern.de">Bayern</a></li>`
  - **Ersetzung HTML-spezifischer Sprachkonstrukte durch XML-basierte Spezifikationen (Formulare über XForms, Eventhandling über XML Events, Frames über XFrames)**
  - **Generisches `src`-Attribut (Ersetzung beliebiger Objekte durch Bilder)**
  - **Ablösung veralteter Überschriftenelemente aus der HTML-Anfangszeit (`h1`, `h2`, ...) durch `<section>` und `<h>`-Tags (bessere Gliederung des XHTML-Dokuments)**
  - **Einführung übersichtlicher Navigationsstrukturen (`<nl>`) ohne Rückgriff auf JavaScript-Implementierungen (Barrierefreiheit)**
- **Unterstützung der Publikation von Texten im Internet**
  - **Automatische Einfügung von Zeilennummern**
  - **Abwechselnde farbige Markierung aufeinanderfolgender Zeilen**

## Historie

Entstehung der  
Markupsprachen

Versionsgeschichte

## XHTML

Grundlegender Aufbau eines  
XHTML-Dokuments

XHTML als Nachfolger von  
HTML

Unterschiede zwischen HTML  
und XHTML

Modularisierung von XHTML

Ausblick: XHTML 2.0

XHTML als Weg zum  
semantischen Markup

## CSS

Grundlegendes

Praktisches Beispiel

Fortgeschrittene Selektoren

<div> und <span>

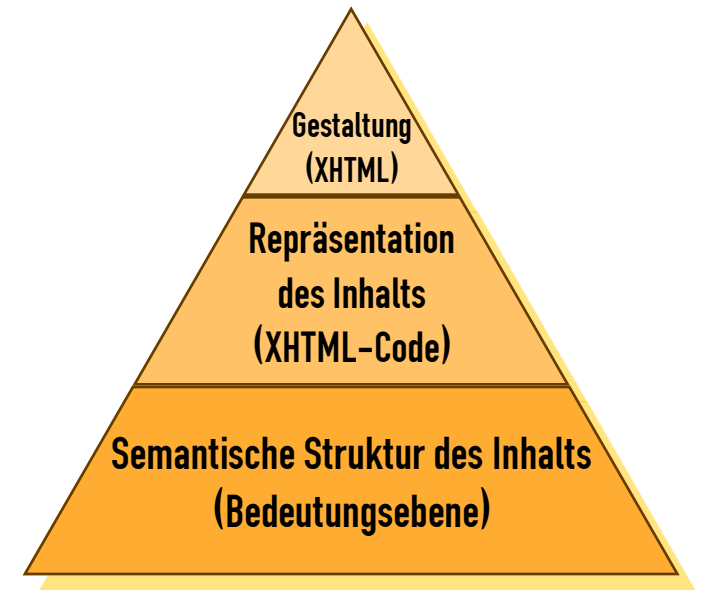
Positionierung von Elementen

Arten von Stilvorlagen

media und @page

# Fazit: XHTML als Weg zum semantischen Markup

- Rückkehr zu verbindlichen Standards durch Anbindung an XML und (strenge) DTDs
- Semantischer Aufbau von XHTML-Dokumenten
- Einführung semantischer Tags wie <section> durch XHTML 2.0 (derzeit in Entwicklung)
- Webmaster sind bei der konkreten Umsetzung gefordert



## Historie

Entstehung der  
Markupsprachen

Versionsgeschichte

## XHTML

Grundlegender Aufbau eines  
XHTML-Dokuments

XHTML als Nachfolger von  
HTML

Unterschiede zwischen HTML  
und XHTML

Modularisierung von XHTML

Ausblick: XHTML 2.0

XHTML als Weg zum  
semantischen Markup

## CSS

▶ Grundlegendes

Praktisches Beispiel

Fortgeschrittene Selektoren

<div> und <span>

Positionierung von Elementen

Arten von Stilvorlagen

media und @page

# Cascading Stylesheets (CSS)

- Stylesheets als „unmittelbare Ergänzung“ zu (X)HTML
- Anwendungsgebiet: Sprache zur Definition von Formateigenschaften einzelner HTML-Elemente
- Vorteile:
  - Einheitliche Gestaltung gleicher oder ähnlicher HTML-Elemente auf einer Seite
  - Einheitliche Gestaltung von Elementen in Hierarchien (z. B. alle Links in einer Tabelle grün markieren)
  - Deutlich umfangreichere Gestaltungsmöglichkeiten als mit klassischem HTML
  - Systematisierung der Gestaltungsregeln
  - Trennung von Design und Struktur
  - Anpassung der Anzeige an das Ausgabemedium

# Cascading Stylesheets (CSS)

□ Normierung durch das W3C-Konsortium (analog zu XHTML)

□ Versionsgeschichte:

- CSS 1.0 1996, Überarbeitung 1999
- CSS 2.0 1998
- CSS 2.0 Rev. 1 2007
- CSS 3.0 in Vorbereitung

□ Anderes Beispiel für eine Sprache zur Definition von  
Stylesheets: XSL

Abgrenzung:

- XSL zur visuellen Gestaltung von XML-Dokumenten aller Art
- CSS speziell auf XHTML ausgerichtet



# Cascading Stylesheets (CSS)

## □ Problematik:

- Unvollständige Browserunterstützung
- Unterschiedliche Ausgabemedien
- Barrierefreiheit

## □ Grundlegende Syntax:

**Selektor { Eigenschaft:Wert; }**

**Eigenschaft + Wert = Deklaration**

**Kommentare: /\* Kommentar \*/ (angelehnt an C/C++ u. Ä.)**

**Beachte: keine zeilenweisen Kommentare (// Kommentar)**

# Cascading Stylesheets (CSS)

## □ Einbindung in das (X)HTML-Dokument

### ■ Zentraler style-Bereich im Dokumentkopf

```
(...) <html><head>...<style type="text/css"> [... CSS ...]
</style></head></html>
```

↑ Angabe des MIME-Typs

### ■ Externe .css-Datei

Einbindung durch: `<link rel="stylesheet" type="text/css" href="formate.css" />`

Alternativ: `@import url("sheet.css");` (in style-Tags)

### ■ Inline-Definition (innerhalb einzelner XHTML-Elemente)

Bsp.: `<p href="..." style="color:red; font:10pt Arial bold italic">`

# CSS: Praktisches Beispiel

p, li { font-size:110%; ←

← Definition bezieht sich auf alle  
Absätze und Listenelemente

line-height:140%; ←

font-family:Helvetica, Arial, sans-serif;

Relative Schriftgrößenangabe  
(Bezug auf die Schriftgröße des  
jeweiligen Elternelements)

Relative Angabe der Zeilenhöhe  
(Bezug auf die Schriftgröße des  
Elements, das beschrieben wird)

← Festlegung der Schriftart und  
eventueller Ersatzschriftarten bei  
Nichtverfügbarkeit

letter-spacing:0.1em;

word-spacing:0.3em; }

↑ Angabe des Wortabstands (nur in em)

← Angabe des Zeichenabstands  
(nur absolut)  
em = bezogen auf die  
Schriftgröße

# Fortgeschrittene Selektoren

## □ Individualformate

- Kennzeichnung eines HTML-Tags mit einem eindeutigen ID-Attribut
- Bezugnahme auf die eindeutige ID im CSS-Code
- Syntax:
  - `<a href="..." id="link_ausnahme">...</a>`
  - `... #link_ausnahme { font-weight: bold; }`

## □ Klassenformate

- Unterschied: mehrere HTML-Tags können derselben Klasse angehören
- Syntax:
  - `<a href="..." class="link_ausnahmen">...</a>`
  - `... .link_ausnahme { font-weight: bold; }`

# Fortgeschrittene Selektoren

## □ Verschachtelte Formate

### ■ Für HTML-Elemente

- Bsp.: `div table td a { ... }`
- Bedeutung: Auswahl aller Links in Tabellenzellen einer Tabelle, die in div-Tags eingeschlossen ist

### ■ Gemischte Selektoren

- Bsp.: `div#header div.wichtig ol li > a { ... }`
- Bedeutung: Auswahl aller Links, die
  - direkt in einem Listenelement enthalten sind, das
  - in einer nummerierten Liste enthalten ist, die
  - in einem div-Element der Klasse wichtig enthalten ist, das
  - im div-Element mit der eindeutigen ID header enthalten ist

# Fortgeschrittene Selektoren

---

## ■ Nachbar-Selektoren

- Bsp.: `ul + li { ... }`
- Bedeutung: Selektion nur aller **ersten** Listenelemente unsortierter Listen
- Auswahl nur **direkt** auf `ul`-Elemente folgender `li`-Elemente

## ■ Kind-Selektoren

- Bsp.: `p > code { ... }`
- Bedeutung: Auswahl aller Quellcode-Elemente, die direkt innerhalb eines Absatzes notiert sind
- Elemente, die in der Selektion `p table code { ... }` enthalten sind, werden nicht erfasst

# Fortgeschrittene Selektoren

## □ Attributbedingte Formate

- Bsp.: `input[type=submit] { ... }`
- Bedeutung: Auswahl aller Absendebutons
- Bsp.: `img[alt ~ =Mecklenburg-Vorpommern Bayern Berlin] { ... }`
- Bedeutung: Auswahl aller Bilder mit bestimmten Wörtern im Alternativtext
- Ermöglichung inhaltsabhängiger Formatierungen

## □ Pseudoelemente/-klassen

- Bsp.: `a:hover { color: red; }`
- Bedeutung: Links beim Überfahren mit der Maus rot markieren (JavaScript-Pendant: `onmouseover`)

# <div> und <span>

---

- Gruppierung mehrerer *aufeinanderfolgender* Elemente
- Formatierung als Gesamtheit
- <div> und <span> als neutrale Gruppierungselemente
- Unterschied:
  - Blockelement <div>  
erzeugt neue Zeile zu Beginn des div-Elements
  - Inlineelement <span>  
Verwendung innerhalb einer Zeile



## Historie

Entstehung der  
Markupsprachen

Versionsgeschichte

## XHTML

Grundlegender Aufbau eines  
XHTML-Dokuments

XHTML als Nachfolger von  
HTML

Unterschiede zwischen HTML  
und XHTML

Modularisierung von XHTML

Ausblick: XHTML 2.0

XHTML als Weg zum  
semantischen Markup

## CSS

Grundlegendes

Praktisches Beispiel

Fortgeschrittene Selektoren

<div> und <span>

▶ Positionierung von Elementen

Arten von Stilvorlagen

media und @page

# Positionierung von Elementen

---

- **HTML: Positionierung von Elementen im Textfluss**
- **Neue Möglichkeiten der Positionierung durch CSS**
  - Absolute Positionierung (mitscrollend, [Bsp.](#))
  - Absolute Positionierung (fix)
  - Relative Positionierung)
  - (Statische Positionierung)
  - Positionierung mittels float und clear
- **Kombinierter Einsatz mit XML und XSLT möglich**

# Historie

Entstehung der  
Markupsprachen

Versionsgeschichte

## XHTML

Grundlegender Aufbau eines  
XHTML-Dokuments

XHTML als Nachfolger von  
HTML

Unterschiede zwischen HTML  
und XHTML

Modularisierung von XHTML

Ausblick: XHTML 2.0

XHTML als Weg zum  
semantischen Markup

## CSS

Grundlegendes

Praktisches Beispiel

Fortgeschrittene Selektoren

<div> und <span>

▶ Positionierung von Elementen

Arten von Stilvorlagen

media und @page

```
<html>
```

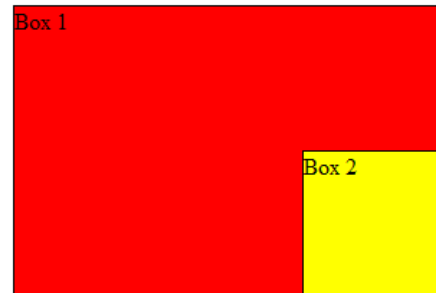
```
<head>
```

```
<style type="text/css">
```

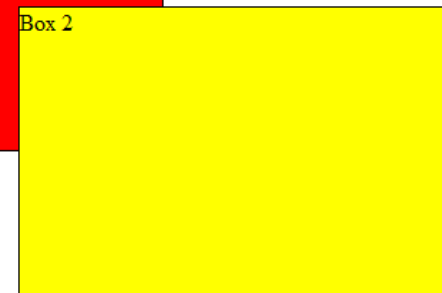
```
  #box1
```

```
  {  
    background-color: red;  
    border: 1px black solid;  
    position: absolute;  
    left: 200px;  
    top: 100px;  
    width: 300px;  
    height: 200px;  
  }
```

Box 1



Box 2



```
  #box2 {  
    background-color: yellow;  
    border: 1px black solid;  
    position: absolute;  
    left: 200px;  
    top: 100px;  
    width: 300px;  
    height: 200px; }  
  }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div id="box1">Box 1
```

```
<div id="box2">Box 2
```

```
</div>
```

```
</div>...
```

# Arten von Stilvorlagen

---

## □ Browserspezifische Stylesheets

- Voreinstellungen einzelner Webbrowser
- Anwendung, sofern kein autoren- oder benutzerspezifisches Stylesheet vorliegt

## □ Autorenspezifische Stylesheets

- Codierung in einer Stylesheetsprache (i. d. R. CSS)
- Benutzer kann Anwendung seitenspezifischer Stile unterdrücken (z. B. Firefox)

## □ Benutzerspezifische Stylesheets

- Festlegung im Browser
- Bsp.: Besonders große Schriftarten für Sehbehinderte

# Das media-Attribut

- **Verwendungszweck: Definition verschiedener Stylesheets für unterschiedliche Ausgabemedien**
- **Syntax z.B.: <link rel="stylesheet" media="..." href="..." />**
- **Attributwerte:**
  - **screen** Ausgabe auf dem Bildschirm
  - **print** Ausgabe beim Drucken
  - **braille, embossed** Braille-Drucker (für Sehbehinderte)
  - **handheld** PDAs
  - **projection** Beamer etc.
  - **aural** Sprachausgabe
  - **tv** Fernsehgeräte

# Das media-Attribut

---

## □ Vorteile:

- Höhere Benutzerfreundlichkeit
- Gesteigerte Barrierefreiheit
- Plattformunabhängigkeit von XHTML-Dokumenten

## □ Codebeispiel (Quelle: [de.selfhtml.org](http://de.selfhtml.org)):

```
@media screen { body { font-size:80%; } }
```

```
@media print { body { font-size:100%; } }
```

```
@media screen, handheld { body { color:black; } }
```

➔ Zusätzliche Schachtelung der CSS-Definitionen in  
generisches media-Attribut

# Seitengestaltung mit @page

- **Ziel: Erstellung von Drucklayouts mit CSS 2.0**
- **Anwendungsgebiete:**
  - **Druck auf Papier**
  - **Bildschirmseitenorientierte Präsentation**
  - **Denkanstöße zur zukünftigen Verwendung**
    - Transformation in andere Formate (PDF, PS, ...)
    - Anbindung an e-book-Software
    - Anbindung an DTP-Software
    - Anbindung an Textverarbeitungssoftware
    - ...

## Historie

Entstehung der  
Markupsprachen

Versionsgeschichte

## XHTML

Grundlegender Aufbau eines  
XHTML-Dokuments

XHTML als Nachfolger von  
HTML

Unterschiede zwischen HTML  
und XHTML

Modularisierung von XHTML

Ausblick: XHTML 2.0

XHTML als Weg zum  
semantischen Markup

## CSS

Grundlegendes

Praktisches Beispiel

Fortgeschrittene Selektoren

`<div>` und `<span>`

Positionierung von Elementen

Arten von Stilvorlagen

media und @page

**Vielen Dank für die Aufmerksamkeit!**