



# Webgrundtechniken



Hochschule  
Ravensburg-Weingarten

Technik | Wirtschaft | Sozialwesen

**Wintersemester 2011/2012**

**LV 4051**

**Bachelorstudiengang: Wirtschaftsinformatik / WI Plus**

**Karl Glatz, BSc.**

**[karl.glatz@hs-weingarten.de](mailto:karl.glatz@hs-weingarten.de)**

[PDF Version](#) | [HTML5 Version](#)

Teile der Präsentation basieren auf Arbeiten von Dr. Stefan Müller

# **INTRO:**

## **Lerziele und Organisation**

## **Lernziele**

- Vermittlung elementarer Techniken des WWW
- Beherrschung der grundlegenden Elemente einer statischen Webseite
- Grundsätze zur Erstellung einer modernen Web-Präsentation
- Vertiefung der Konzepte und Grundsätze im Rahmen einer praktischen Arbeit
- Erstellung einer Webseite auf Basis von (X)HTML und CSS
- Prinzip von modernen Webanwendungen verstehen

## Organisation

- Vorlesung Montags 14:15 in H061
- Übungen, Montags 14:15 in V206/V208 (zwei Räume parallel!)
  - Übungsaufgaben sind Pflicht! (mindestens 80%)
  - Testate durch Tutoren
- Prüfungsleistung: Praktische Arbeit
  - Durchführung eines umfangreicheren **Web-Projekts** (in 2-3er Gruppen)
    - Erstellung einer Web-Präsentation für eine Firma, ein Projekt, Verein etc. (fiktiv oder real)
    - Beginn: 28.11.2011 (spätestens)
    - Meilenstein 1: 18.12.2011 / Meilenstein 2: 8.01.2012
    - Finale Präsentation des Projekts (mündliche Prüfungsleistung):  
23.01.2012 / 30.01.2012

## Kalender Download

Kalender als iCal-Link



## E-Learning Plattform

Link: [elearning.hs-weingarten.de](https://elearning.hs-weingarten.de)

# **INTRO:**

## **Internet und WWW**

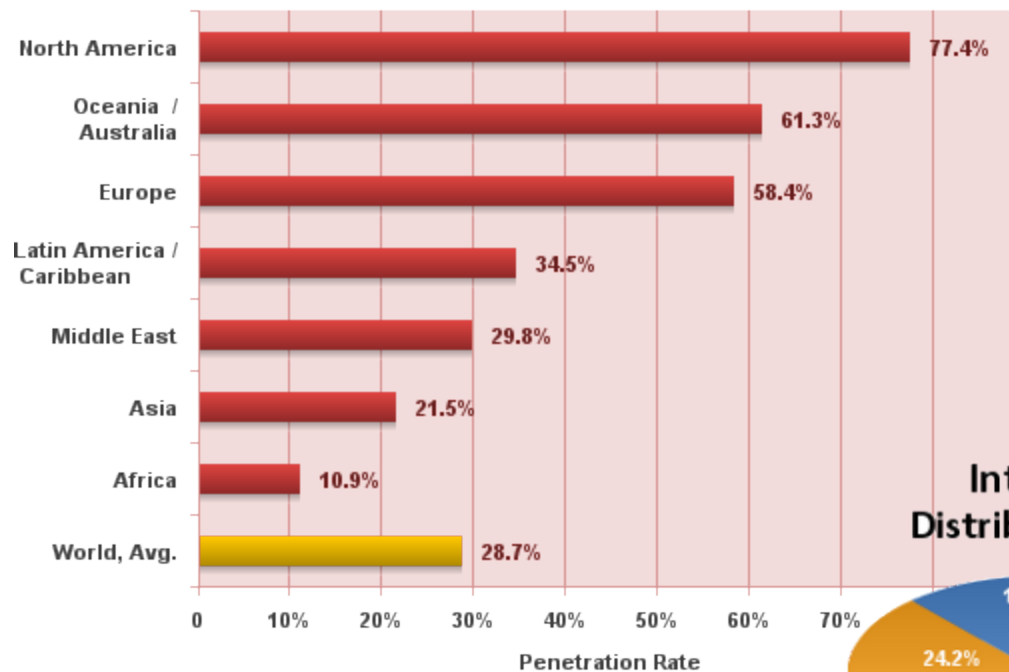
## Entstehung des Internet

- Vom ARPANET zum Internet
- 1969: ARPANET als erstes paketorientiertes Netzwerk
- 1977: Anschluss weiterer Netzwerke an das ARPANET via TCP/IP
  - TCP (Transmission Control Protocol)  
Paketorientiertes Datenübertragungsprotokoll (auf der 4. Schicht des OSI-Referenzmodells)
  - IP (Internet Protocol)  
Vermittlungsprotokoll (auf der 3. Schicht des OSI-Referenzmodells)  
Vermittlung zwischen Subnetzen mit unterschiedlicher Netzwerktechnologie (auf der 1. und 2. OSI-Schicht)
- ⇒ Geburt des Internet als Netz aus verschiedenen Teilnetzen (interconnected networks)



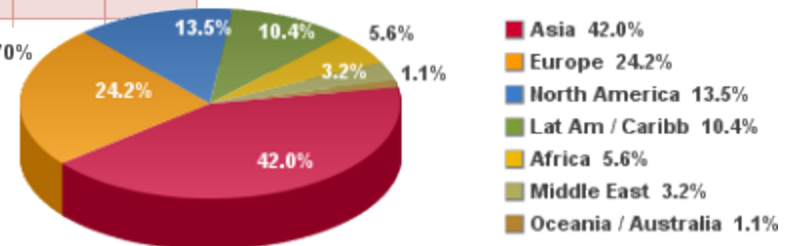
## Verbreitung des Internet

**World Internet Penetration Rates  
by Geographic Regions - 2010**



Source: Internet World Stats - [www.internetworldstats.com/stats.htm](http://www.internetworldstats.com/stats.htm)  
 Penetration Rates are based on a world population of 6,845,609,960  
 and 1,966,514,816 estimated Internet users on June 30, 2010.  
 Copyright © 2010, Miniwatts Marketing Group

**Internet Users in the World  
Distribution by World Regions - 2010**

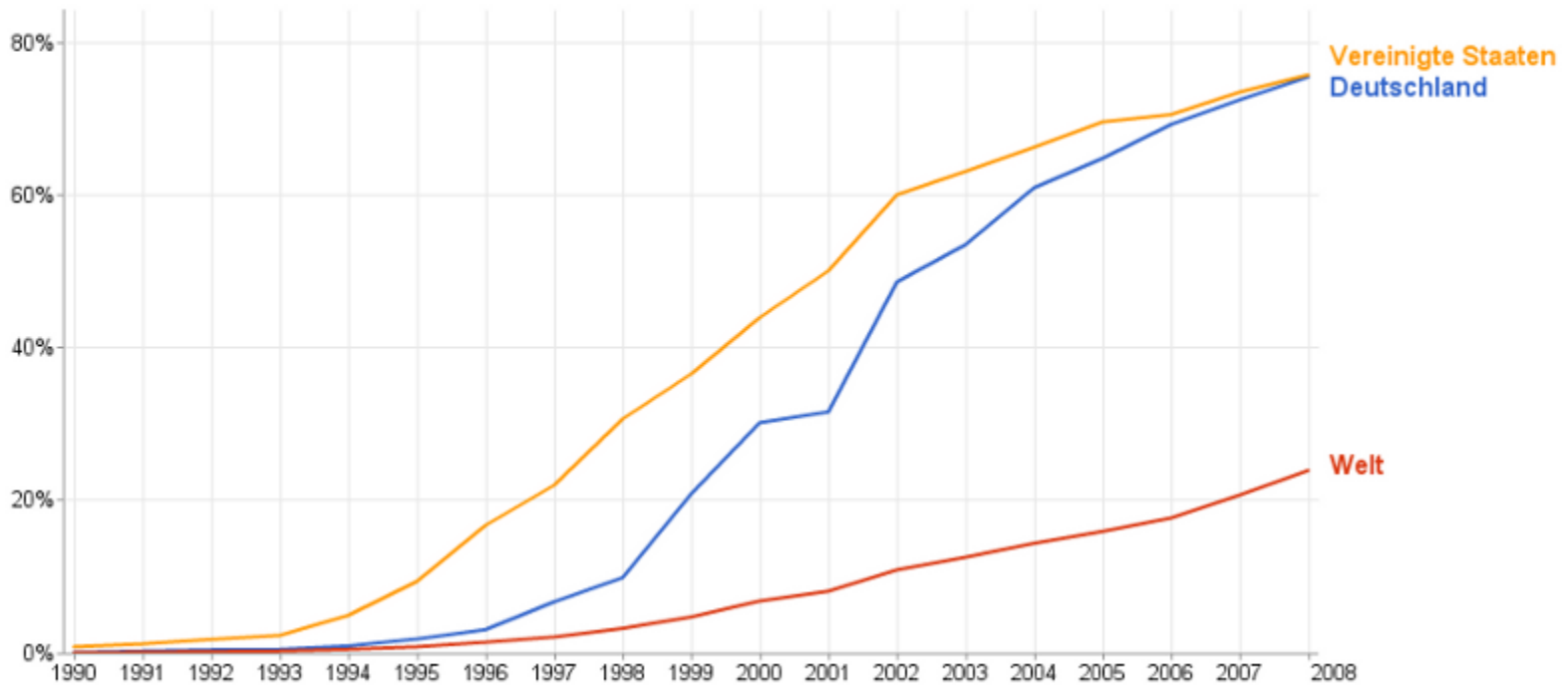


Source: Internet World Stats - [www.internetworldstats.com/stats.htm](http://www.internetworldstats.com/stats.htm)  
 Basis: 1,966,514,816 Internet users on June 30, 2010  
 Copyright © 2010, Miniwatts Marketing Group

## Verbreitung des Internet #2

### Internetnutzer in Prozent der Bevölkerung

Personen mit Internetzugang pro 100 Einwohner [Weitere Informationen »](#)



Datenquelle: [Weltbank, Weltentwicklungsindikatoren](#) - Last updated 8. Mär 2011

## Die Dienste des Internet

- World Wide Web (WWW)  
Standard zur Übermittlung von Multimedia-Dokumenten im HTML-Format.  
Protokoll: HTTP (Hypertext Transfer Protocol)
- File Transfer (Übertragung von Dateien)  
Standard zur Übertragung von Dateien
- Protokoll: FTP (File Transfer Protocol)
- E-Mail (elektronische Post)  
Standard für den Versand von E-Mails über das Internet  
Protokoll: SMTP (Simple Mail Transfer Protocol), POP3 (Post Office Protocol), IMAP
- Newsgroups (auch Usenet)  
Übertragung von Netzwerk-Nachrichten  
Protokoll: NNTP für Net News Transfer Protocol

## Die Dienste des Internet

- WAP (Wireless Applications)  
Internet-basierter Dienst zur Übertragung von WML-Dokumenten für Mobilfunkendgeräte  
Protokoll: WAP (Wireless Application Protocol)
- TELNET / SSH  
Terminal-Emulator zum Einloggen und Arbeiten auf entfernten Rechnern
- Voice over IP (VoIP)  
Telefonieren über das Internet  
Protokoll: SIP (Session Initiation Protocol)
- Chat / Instant Messaging  
IRC (Internet Relay Chat)  
XMPP (Jabber)

## Entstehung des World Wide Web

- Das WWW als weltweites Hypertextsystem
  - 1990: Entwicklung eines weltweiten Hypertextsystems auf Basis des Internets (Tim Berners-Lee)
- Die Säulen des WWW
  - **HTML:** Spezifikation einer Auszeichnungssprache für Web-Dokumente
  - **URIs** (Universal Resource Identifiers): Spezifikation für die Adressierung beliebiger Datenquellen im Internet
  - **HTTP-Protokoll:** Spezifikation für die Kommunikation zwischen Web-Clients und Web-Servern

## Geschichte der Web-Browser

- Mosaic - Erster graphischer Browser
- Netscape 4.x
- Microsoft Internet Explorer 6.0

⇒ Konkurrenzkampf gekennzeichnet durch die Entwicklung proprietärer Features



Initiative zur (kontrollierten) Weiterentwicklung des WWW Definition weltweiter Standards für das WWW HTML, CSS, XML, ...



## Web-Browser Marktanteile

Datum	Firefox	IE	Chrome	Safari	Opera
Q4 2011	47,4 %	30,9 %	8,2 %	5,2 %	2,4 %
Q3 2011	48,5 %	31,2 %	7,4 %	5,1 %	2,4 %
Q2 2011	49,1 %	32,5 %	6,2 %	5,1 %	2,6 %
Q1 2011	49,6 %	34,6 %	5,1 %	4,5 %	2,6 %
Q4 2010	50,4 %	35,7 %	4,1 %	4,2 %	2,7 %
Q3 2010	50,7 %	37,1 %	3,3 %	3,9 %	2,7 %
Q2 2010	51,3 %	37,5 %	2,8 %	3,7 %	2,7 %
Q1 2010	50,0 %	40,1 %	2,1 %	3,5 %	2,7 %
Q4 2009	48,0 %	43,4 %	1,5 %	3,1 %	2,6 %
Q3 2009	46,1 %	46,1 %	1,1 %	2,9 %	2,5 %

# Hypertext

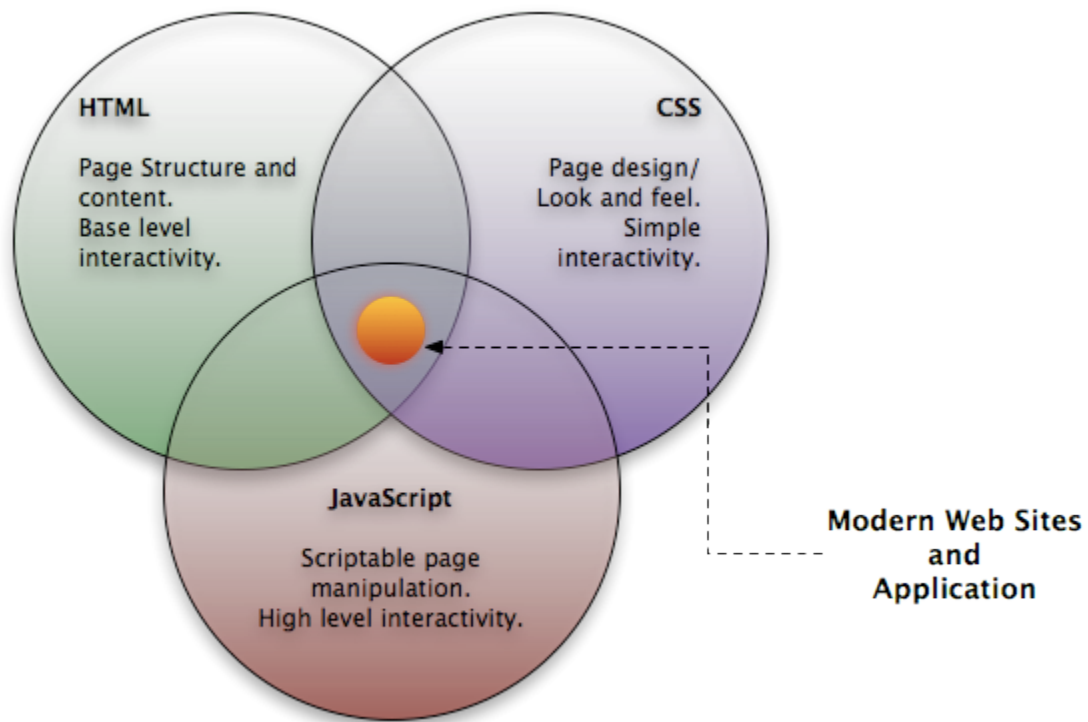
- Hypertext als nicht-lineares Medium
  - **Pro**
    - Flexibler Zugang zu Wissen (Nachschlagewerke)
    - Vernetztes Wissen kann leichter aufgenommen werden (wissenschaftlich nicht eindeutig belegt)
  - **Contra**
    - Steigende Komplexität
    - Gefahr des “Lost in Hyperspace”
  - Kohäsive Geschlossenheit
    - Verständlichkeit der Informationseinheiten unabhängig vom Verweiskontext
    - Einordnung von Informationseinheiten in ihren Kontext durch Verlinkung



## HTML und XHTML

- HTML (Hypertext Markup Language)
  - Sprache zur Auszeichnung von Texten bzw. deren Elementen (Überschriften, Absätze, Listen, Tabellen, usw.)
  - Definiert mittels SGML (Standard Generalized Markup Language)
  - Aktuell in der Version 4.01 (5.0 ist im „Working Draft“ State)
    - Rückbesinnung auf Kernaufgaben
    - Auslagerung von Layoutangaben (CSS)
    - Sprachvarianten Strict, Transitional und Frameset
- XHTML
  - HTML definiert mittels XML (XML Parser sind einfacher als SGML Parser)
  - XHTML 1.0 entspricht HTML 4.01
  - XHTML 1.1 reduziert auf Variante Strict

## Web-Technologien



# HTML5

- Neue Tags
  - Audio/Video
  - <article>, <section>, <header>
- SVG: Vektor Grafik
- CSS3
  - Schatten
  - Runde Ecken
  - Animation
  - Mehr-Spaltiger Text
- Javascript APIs -Canvas - WebGL - Offline Storage
- Test unter: [www.html5test.com](http://www.html5test.com)

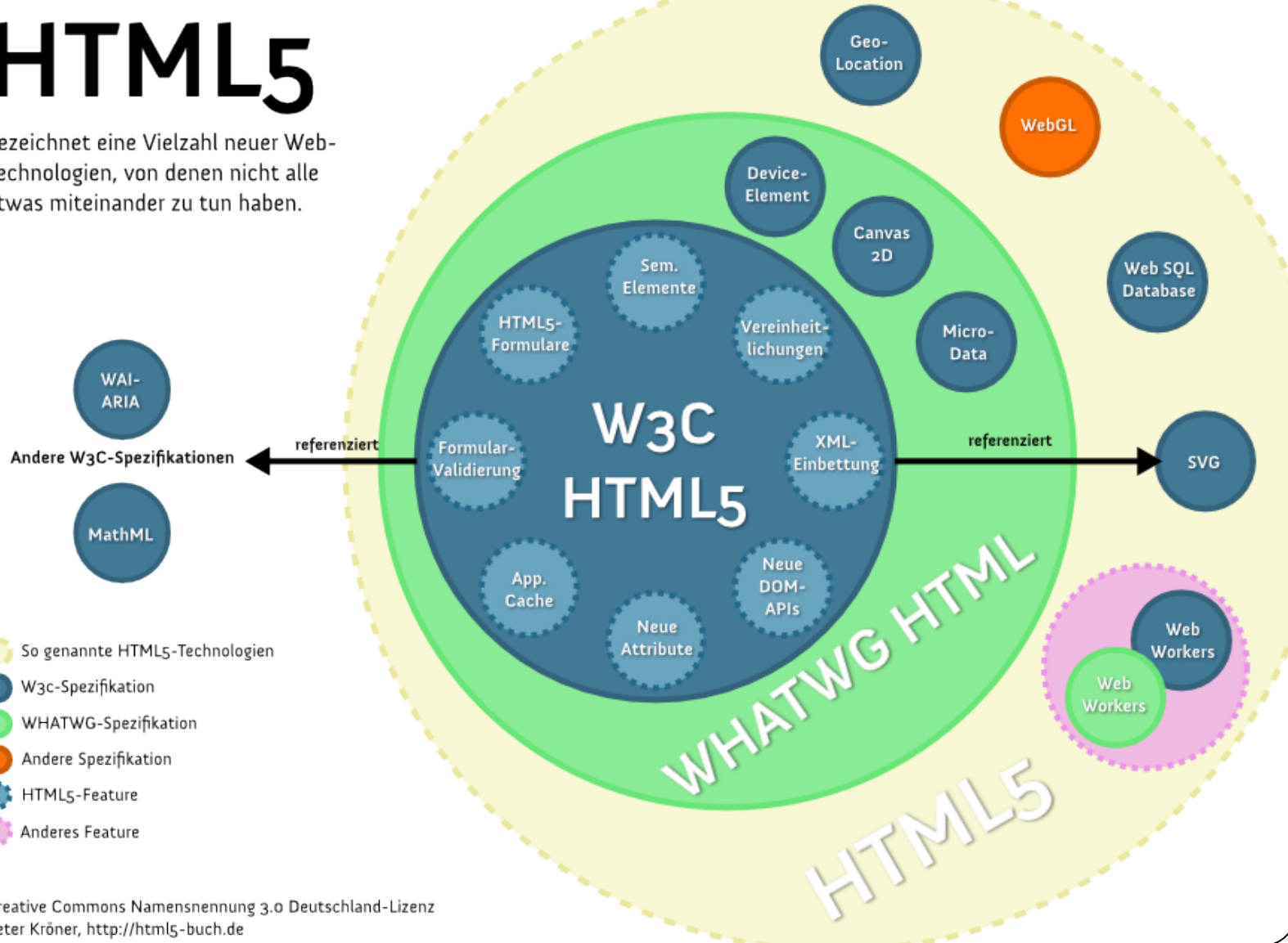


## HTML5 #2

Umgangssprachliches so genanntes

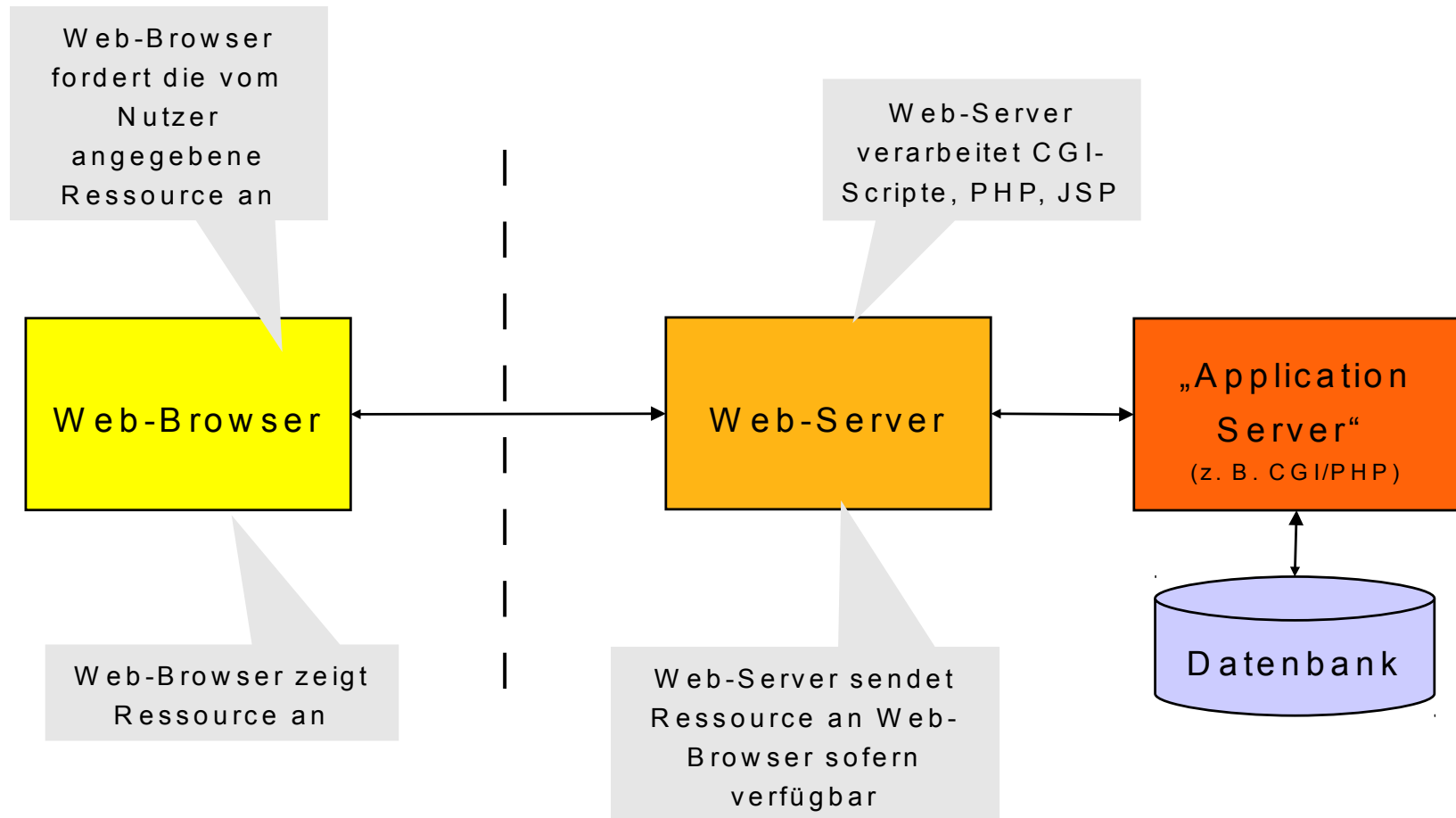
# HTML5

bezeichnet eine Vielzahl neuer Web-Technologien, von denen nicht alle etwas miteinander zu tun haben.



## Web-Browser und Web-Server

### Client-Server-Kommunikation im WWW



## **Client Web-Technologien**

### **JavaScript & DOM**

Dynamisierung von HTML-Dokumenten im Web-Browser Verarbeitung von Maus- oder Tastatureingaben und Ausgaben am Bildschirm bzw. Veränderung des HTML-Dokuments DOM (Document Object Model): Schema für den Zugriff auf HTML-Dokumente Javascript Libraries (jQuery, MooTools, Dojo etc.) Erleichtern die Programmierung von dynamischen Seiten

### **GWT**

Google Web Toolkit, RIAs mit Java programmieren, ohne Browser Plugin ActiveX & Java Applets (Plugins)  
Veraltete Microsoft-Technologie für dynamische „Webseiten“ Veraltete Methode um Java Code im Browser auszuführen

### **Flash & Silverlight (Plugins)**

Modernere Technologien für RIA (Rich Internet Applications) Flash dient oft als „lückenschließer“ für fehlende HTML/JS Features

## **Server Web-Technologien**

### **CGI & Perl**

CGI (Common Gateway Interface): Web-Server-Schnittstelle für den Programmzugriff

### **Perl (Practical Extraction and Report Language)**

Script-Sprache zur Automatisierung von Datenzugriffen und -weiterverarbeitung

### **PHP (PHP Hypertext Preprocessor)**

Programmiersprache zur server-seitigen Dynamisierung von HTML Web-Server verarbeitet PHP-Code und modifiziert HTML-Dokument

### **JSP (Java Server Pages)**

Aufruf von Java-Applikationen durch den Web-Server

### **ASP/.NET**

ASP (Active Server Pages): Microsoft-Gegenstück zu JSP .NET: Web-Applikationsframework (incl. C#)

### **Python/Django, JSF uvm.**

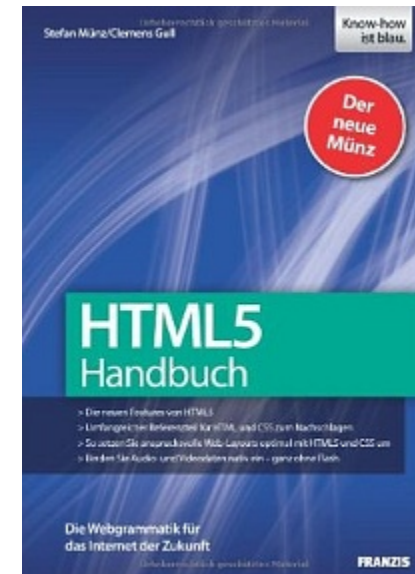


## (Online-)Literatur Buch

- HTML 5 Handbuch  
**Kostenlos Online verfügbar**  
<http://webkompetenz.wikidot.com/docs:html-handbuch>  
Vom Self-HTML Gründer Stefan Münz (2010)
- HTML5 Buch
  - <http://html5-buch.de/>
  - Viele Demos und Aufgaben

## Nachschlagewerke Online

- SELFHTML. [www.de.selfhtml.org](http://www.de.selfhtml.org)
- Online-Internet-Kurs. [www.kurs.de](http://www.kurs.de)



**INTRO:**

**Planung und Durchführung eines  
Web-Projekts**

## Vorgehensweise für Web-Design #1

# Strategische Positionierung

- Zielsetzung & Zielgruppe der Webseite
  - Kommunikation und Erreichbarkeit, Information, Interaktion, Verkauf, Service- und Kundenbindungsinstrument
  - Festlegung des beabsichtigten Mehrwerts (z.B. neue Kundensegmente, bessere Kundenbindung, etc.)
- **Wettbewerb:** Vergleich mit Wettbewerbern
- **Marketing:** Integration in Marketing-Strategie

## Vorgehensweise für Web-Design #2

# Realisierung einer Webseite

- Festlegung der Inhalte und Struktur (Navigation)
- Festlegung der Funktionen (Value-Added Services)
- Graphisches Design
- Technologie
- Design Guidelines, z.B. [www.useit.com](http://www.useit.com) (Jakob Nielsen)

## Vorgehensweise für Web-Design #3

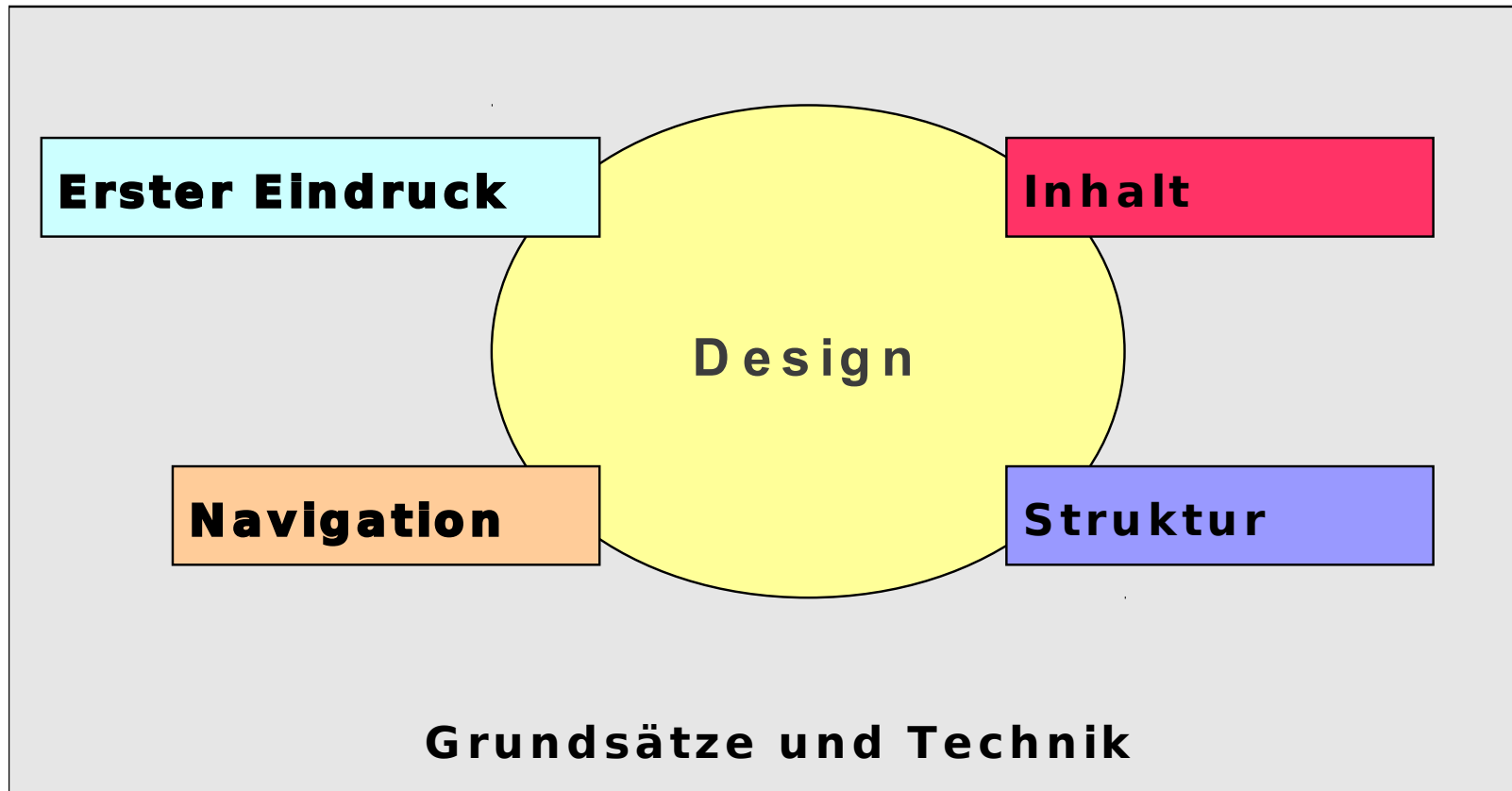
# Integration

- Organisatorische Integration
  - Pflege und Aktualisierung der Inhalte
  - Wartung und Betrieb
- Technische Integration
  - Einbindung in bestehende Infrastruktur

# Monitoring

- Kundenfeedback
- Kosten-/Nutzenanalyse
- Wettbewerbsanalyse (Benchmarking)
- Technologische Entwicklung

## Kriterienkatalog für Webseiten



## Beurteilungskriterien #1

# Erster Eindruck

Ist die Präsentation übersichtlich?

Ist das Grundthema klar erkennbar?

Wirkt ein einheitlicher angemessener Gesamteindruck?

## Inhalt

Erhält man auf der Startseite einen Überblick über den Inhalt?

Ist der Informationsgehalt ausreichend?

Ist die Zielsetzung klar ersichtlich?

Sind die für die Nutzung erforderlichen Funktionen vorhanden?

Ist der Inhalt nach Themen geordnet?

## Beurteilungskriterien #2

# Struktur

Ist die Struktur auf den ersten Blick durchschaubar und leicht nachvollziehbar?

Sind die einzelnen Themenblöcke gut strukturiert?

Sind Navigationselemente und Links erkennbar? Ist die Verlinkung sinnvoll und umfangreich?

# Navigation

Ist die Navigation bzw. Hierarchie nachvollziehbar?

Ist der Standort innerhalb der Web-Site jederzeit erkennbar?

Ist die Navigation durchgehend und prägnant?

Ist der Schritt zur Startseite jederzeit möglich?



## Beurteilungskriterien #3

# Design

Steht der Inhalt durch das Design weiterhin im Vordergrund?

Spiegelt das Design die Zielsetzung wider?

Spricht das Design die Zielgruppe an?

Ist genügend Farbkontrast vorhanden?

Sind die benutzten Farbtöne harmonisch?

Ist das Design durchgehend?

Ist der Text lesbar, die Schriftart und -größe angemessen?

Sind besuchte Links markiert?

## Beurteilungskriterien #4

# Grundsätze und Technik

Ist die Präsentation unabhängig vom Browser (IE, Firefox, Chrome, Safari)?

Passt sich die Darstellung den Anforderungen des Nutzers an (Auflösung, Schriftgrößen)?

Sind Inhalt und Design strikt getrennt (HTML und CSS)?

Ist die Präsentation bezüglich Inhalt, Struktur und Design ausreichend komplex?

Sind die verwendeten HTML- und CSS-Elemente sinnvoll eingesetzt?

Ist ein alternatives Design für den Ausdruck vorhanden?

# HTML Basics

## HTML Bascis: Inhalt / Ziel

- Was ist HTML?
- HTML Grundgerüst
- Textformatierung
- Links
- Listen
- Weitere Textauszeichnungen (physisch und logisch)
- Allgemeine HTML Elemente
- Bilder einbinden

=> Vertiefung der Kenntnisse in Übung 1 (nächste Woche).

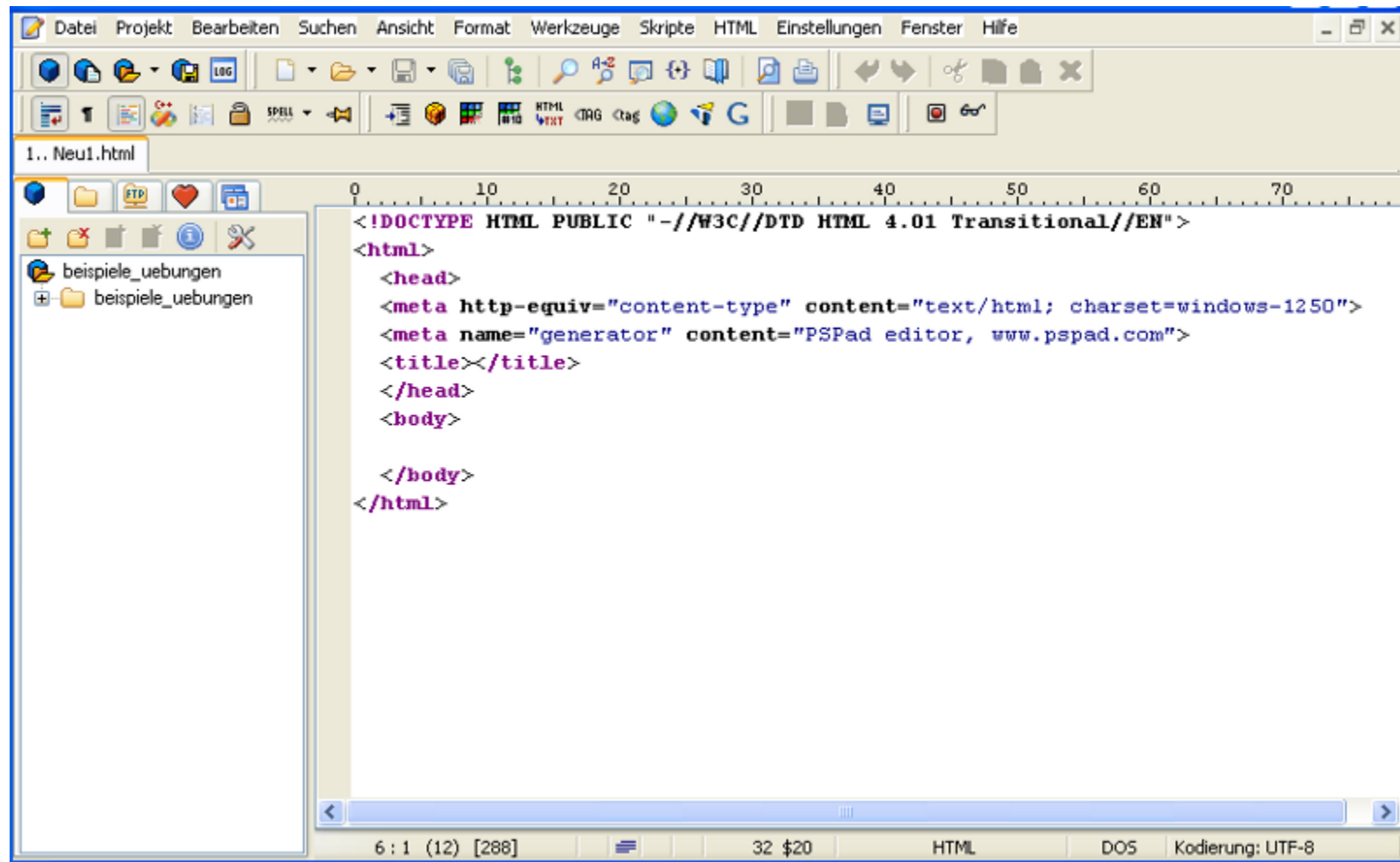
## Das HTML Format

- HTML Dateien sind Text-Dateien
- Bearbeitbar mit einfachem Texteditor (z. B. Notepad, gedit, TextMate)
- Spezialisierte HTML Editoren
  - Windows: PSPad / Notepad++
  - Linux: gEdit / Kate / Geany
  - OSX: TextMate
- Dateiendung *.html* oder *.htm*
- Programm zum Anzeigen einer HTML-Datei: Browser
  - Chrome, Firefox, Safari, Opera, IE

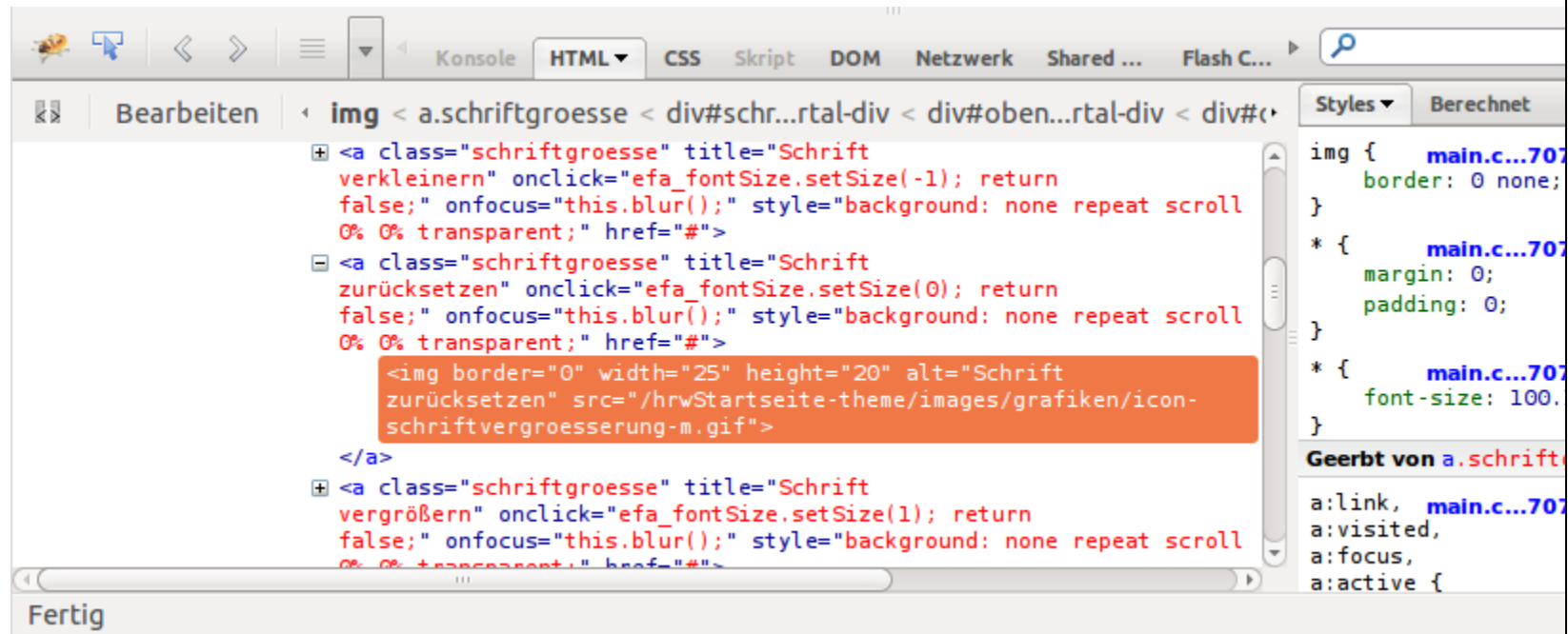
## Aufgabe von HTML

- Idealisiert:
  - Textauszeichnung
    - Bedeutung von Textelementen festlegen (Semantik)
  - NICHT Textpräsentation, Layout, Design
    - CSS
- Realität:
  - Bau von Webseiten im Zusammenspiel mit CSS und JavaScript

## Der HTML-Editor PSPad



## Entwicklertools im Browser



- Firefox: Addon Firebug
- Chrome: Schraubenschlüssel => Tools => Entwicklertools | UMSCHALT+STRG+I | F12
- Safari: ALT+CONTROL+I
- Opera: Tools => Advanced => Opera Dragonfly

Aktuelle Versionen vom Internet Explorer bieten ebenfalls Entwicklertools.



## HTML4-Datei Aufbau

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
2 "http://www.w3.org/TR/html4/loose.dtd">  
3 <html>  
4   <head>  
5     <title>Titel der Seite</title>  
6   </head>  
7   <body>  
8     <h1>Überschrift der Seite</h1>  
9     <p>Ein Absatz mit Etwas Text</p>  
10  </body>  
11 </html>
```

- Dokumenttyp-Deklaration: Deklariert die verwendete Auszeichnungssprache, d.h. das vereinbarte „Regelwerk“
- HTML 4.01 Sprachvarianten
  - Strict: Keine Verwendung unerwünschter Elemente
  - Transitional: Verwendung aller Elemente
  - Frameset: Definition von Framesets

## HTML5-Datei Aufbau

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Titel der Seite</title>
5   </head>
6   <body>
7   </body>
8 </html>
```

- HTML5 kennt keine verschiedenen Varianten mehr
  - Keine Frames mehr!
  - Kein DTD nötig, da HTML5 formell keine SMGL Sprache mehr ist.
  - Im nicht-HTML5 Browser: Standards Mode

## HTML Tags

- Tags sind immer in spitzen Klammern eingeschlosse

```
<tagname>
```

- Tags werden (fast) immer geschlossen

```
<tagname></tagname>
```

- Zwischen den Tags kann (meist) Text und/oder weitere Tags stehen

```
<b>Text</b>
```

- Tags können (beliebig viele) Attriube haben

```
<b class="test">Text</b>
```

## Text formatierung

```
1 <p>Das ist etwas Text</p>
```

Das ist etwas Text

P steht für Paragraph und bildet einen Absatz

```
1 <p>Das ist etwas Text<br>mit einem  
2 Umbruch im Text</p>
```

Das ist etwas Text  
mit einem Umbruch im Text

br erzeugt (weiche) Umbrüche im Text (in Word/Writer: STRG+ENTER)

- Überschriften

```
1 <h1>Überschrift 1</h1>  
2 <h2>Überschrift 2</h2>  
3 <h3>Überschrift 3</h3>  
4 <h4>Überschrift 4</h4>  
5 <h5>Überschrift 5</h5>  
6 <h6>Überschrift 6</h6>
```

# Überschrift 1

Überschrift 2

Überschrift 3

Überschrift 4

Überschrift 5

Überschrift 6

- Überschriften sind nicht nur ein visuelles Mittel!
  - Semantische Auszeichnung
  - Wichtig für maschinelle Verarbeitung, z. B. Suchmaschinen und Sehbehinderte

## Zeilenumbruch

- Zeilenumbruch erzwingen
  - `<br>` (Standalone-Element)
- Zeilenumbruch verhindern
  - `&nbsp;` erzeugt ein geschütztes Leerzeichen
  - An einer solchen Stelle erfolgt kein Umbruch
  - Alternative Schreibweise: `&#160;`
- Bedingter Zeilenumbruch
  - `&shy;` markiert eine Stelle an der getrennt werden darf Browserunterstützung mangelhaft, daher vermeiden

## Sonderzeichen

`&nbsp;` geschütztes leerzeichen

`&gt;` >

`&lt;` <

`&amp;` &

`&auml;` ä

`&uuml;` ü

`&Ouml;` Ö

`&szlig;` ß

Beispiel:

```
1 <b> 5 &gt; 3 </b>
```

**5 > 3**

# Hyperlinks

```
1 <a href="http://scooter.de" target="_blank">HYPER HYPER</a>  
2 <a href="[URI]" target="[ZIEL]">[SICHTBARER TEXT]</a>
```

## HYPER HYPER

- Herzstück des WWW: Ermöglichen Kontext von Informationen
- Prinzip: Weiterentwicklung von Literaturverzeichnissen aus Wissenschaftlichen Publikationen
- Vordefinierte Optionen für `target`
  - `_blank`: Anzeige des Verweisziels in neuem Fenster
  - `_self`: Anzeige des Verweisziels in aktuellem Fenster
  - `_parent`: Anzeige in Vaterfenster bei verschachteltem Frameset
  - `_top`: Anzeige in Hauptfenster bei verschachteltem Frameset





## URI: Uniform Resource Identifier

Ein Uniform Resource Identifier (URI) (engl. „einheitlicher Bezeichner für Ressourcen“) ist ein Identifikator und besteht aus einer Zeichenfolge, die zur Identifizierung einer abstrakten oder physischen Ressource dient. URIs werden zur Bezeichnung von Ressourcen (wie Webseiten, sonstigen Dateien, Aufruf von Webservices, aber auch z. B. E-Mail-Empfängern) im Internet und dort vor allem im WWW eingesetzt.



WIKIPEDIA

## Verweise innerhalb einer Datei

```
1 <a href="[URI]#[Ankername]">Verweistext</a>
```

- Regeln für Ankernamen (bzw. IDs):
  - sollten nur Buchstaben, Ziffern und die Sonderzeichen Unterstrich, Bindestrich, Punkt und Doppelpunkt enthalten
  - Ankernamen und ID-Attribute müssen dateiweit eindeutig sein

Beispiel: [Wikipedia Artikel HTML, Abschnitt Syntax](#)

## Verweise zu Email-Adressen

```
1 <a href="mailto:[email-Adresse]">Verweistext</a>  
2 <a href="mailto:karl.glatz@hs-weingarten.de">Karl</a>  
3 <a href="mailto:karl.glatz@hs-weingarten.de?cc=karl.glatz@gmail.com">  
4 Karl Glatz</a>
```

- Optionen bei Email-Verweisen
  - cc, bcc, subject und body
- Optionen werden als Parameter des href-Attributs übergeben, eingeleitet durch ein ?
- Problem: Spam-Bots durchsuchen internet nach mailto: Adressen!
  - Lösungen: oft wird die Adresse verunstaltet oder mit JS encodiert

## Listen: Aufzählungslisten (unordered list)

```
1 <h2>Einkaufsliste</h2>
2 <ul>
3     <li>Milch</li>
4     <li>Eier</li>
5     <ul>
6         <li>3x Bio</li>
7         <li>2x Freiland</li>
8     </ul>
9     <li>Spagetti</li>
10 </ul>
```

### Einkaufsliste

- Milch
- Eier
  - 3x Bio
  - 2x Freiland
- Spagetti

ul: unordered list | li: list item

## Nummerierte Listen (ordered list)

```
1 <h2>Anleitung</h1>
2 <ol>
3     <li>Download</li>
4     <li>Entpacken</li>
5     <li>Starten ...</li>
6 </ol>
```

### Anleitung

**1. Download**

**2. Entpacken**

**3. Starten ...**

**Achtung:** Verschachtelung nummerierter Listen bewirkt keine Nummerierungshierarchie (Mit CSS möglich)

## Präformatierter Text

```
1 <pre>präformatierter
2   <b>Text</b> mit ein paar
3 Umbrüchen und
4   Einrückungen!</pre>
```

```
präformatierter
   Text mit ein paar
Umbrüchen und
   Einrückungen!
```

Anzeige mit Formatierungen in dichtengleicher Schrift  
Aber: HTML-Zeichen werden interpretiert

- Verwendung
  - Anzeige von Quellcode
  - vordefinierten Tabellen, etc.

## Zitate und Adressen

```
1 <blockquote><p>there are only two hard problems in computing: caching,  
2 concurrency and off-by-one errors</p></blockquote>
```

there are only two hard problems in computing: caching, concurrency and off-by-one errors

### Zitate

- Attribut cite: URI der zitierten Quelle (ohne Visualisierung)
- `<blockquote cite="http://www.hs-weingarten.de/"> ...`

### Adressen

```
1 <address>Hochschule Ravensburg-Weingarten<br>  
2 Doggenried Str.<br>88250 Weingarten</address>
```

## Logische Textauszeichnung

- Logische Auszeichnungen im Text
  - Elemente definieren logische Bedeutung unabhängig von einer konkreten Darstellung
  - Logische Auszeichnungen sind Inline-Elemente
- Elemente zur logischen Textauszeichnung
  - em - empathisch, betont
  - strong - stark betont
  - code - Quelltext
  - samp - Beispiel
  - kbd - Benutzereingaben
  - var - Variable
  - cite - Quelle oder Autor
  - dfn - Definition
  - abbr - Abkürzung
  - acronym - Akronym
  - q - Zitat



- del - gelöschter Text
- ins - eingefügter Text

- Attribut datetime: Zeitpunkt der Änderung
- Attribut cite: URI als Grund für Änderung

## Physische Textauszeichnung

- HTML4: Elemente definieren direkt eine gewünschte Darstellung
- HTML5: Jeweils "schwache" semantische Bedeutung zugeordnet => Styling CSS
- Elemente zur physischen Textauszeichnung
  - b - fett (bold)
  - i - kursiv (italic)
  - ~~tt~~ - diktengleich (teletyper)\*
  - ~~big~~ - größer als normal\*
  - ~~center~~ - zentriert\*
  - ~~strike~~ - durchgestrichen\*
  - small - kleiner als normal
  - sup - hochgestellt (superior)
  - sub - tiefgestellt (subordinate)
- Sonstige Elemente: hr - trennlinie

\* nicht in HTML5: W3C: HTML5 vs HTML4

## Allgemeine Elemente für Textbereiche

- Allgemeines Block-Element
  - div
  - Kind-Elemente: Block-Elemente und Inline-Elemente
- Allgemeines Inline-Element
  - span
  - Kind-Elemente: Inline-Elemente
- Formatierung allgemeiner Elemente mit CSS
  - Allgemeine Elemente ermöglichen die logische Auszeichnung von Abschnitten oder Blöcken
  - Formatierung mit CSS

## Allgemeine Elemente HTML5

- HTML5 bietet neue allgemeine Elemente
  - article - Artikel z. B. in einem Blog
  - section - Abschnitt eines Textes
  - nav - Navigation, Menü
  - header - Kopf einer Seite
  - footer - Fuß einer Seite
  - aside - z.B. Sidebar bei einem Blog
- Alle Elemente verhalten sich wie das div, bieten jedoch die Möglichkeit das HTML-Dokument besser zu strukturieren.

## Grafikformate: Vektor und Pixel

- Pixel:
  - **JPEG:** Joint Photographic Experts Group
    - Verlustbehaftetes Format
    - Einsatzzweck: Fotos von Personen, Landschaften etc.
  - **PNG:** Portable Network Graphics
    - Verlustfreies Format
    - Echte Transparenz: Alpha Kanal
    - Animierte Varianten: APNG, MNG (fehlender Browser Support)
    - Einsatz: Alle Grafiken wie Verläufe, Comics, Zeichnungen, Buttons etc.
  - **GIF:** Graphics Interchange Format
    - Verlustfreies Format
    - Limitierter Farbraum
    - Einfache Transparenz (1-Bit)
    - Einsatzzweck: Animation
    - **Veraltetes Format!**

- Vektor:
  - **SVG:** Scalable Vector Graphics
    - Einsatzzweck: Grafiken wie Karten, Logos, Wappen, Zeichnungen, usw.
  - **Canvas:** Generierte Vektorgrafik
    - Einsatzzweck: Interaktive Spiele, Animationen etc.
    - Grafik muss programmiert werden in JavaScript



## Einbinden von Bildern

```
1 
```

- Referenz auf eine Graphik
  - img-Element ist Standalone-Element
- Attribut `src` bestimmt die Graphikdatei Beachten Sie die Möglichkeiten zur Referenzierung von Dateien in HTML
- Attribut `alt` definiert alternativen Text
  - IE: Tooltip
    - Tooltips normalerweise via Universalattribute `title`
  - Alle: Falls Bild nicht angezeigt werden kann; Screenreader

## (Veraltete) Attribute für Bilder

- Größe der Grafik: `width`, `height`
  - Bei langsamen Verbindungen führt Angabe zur schnelleren Seitendarstellung
  - Abweichende Größen zum Original sind nicht empfehlenswert:
    - Schlechte Skalierungsalgorithmen der Browser
    - Höhere Ladezeiten bei verkleinerung der Grafiken
- Ausrichtung:
  - `align`, `vspace`, `hspace`
- Für beides sollte heute CSS verwendet werden! (HTML4 Strict/HTML5)
- Image Maps: Verweissensitive Grafiken



# CSS Basics

# Stylesheets

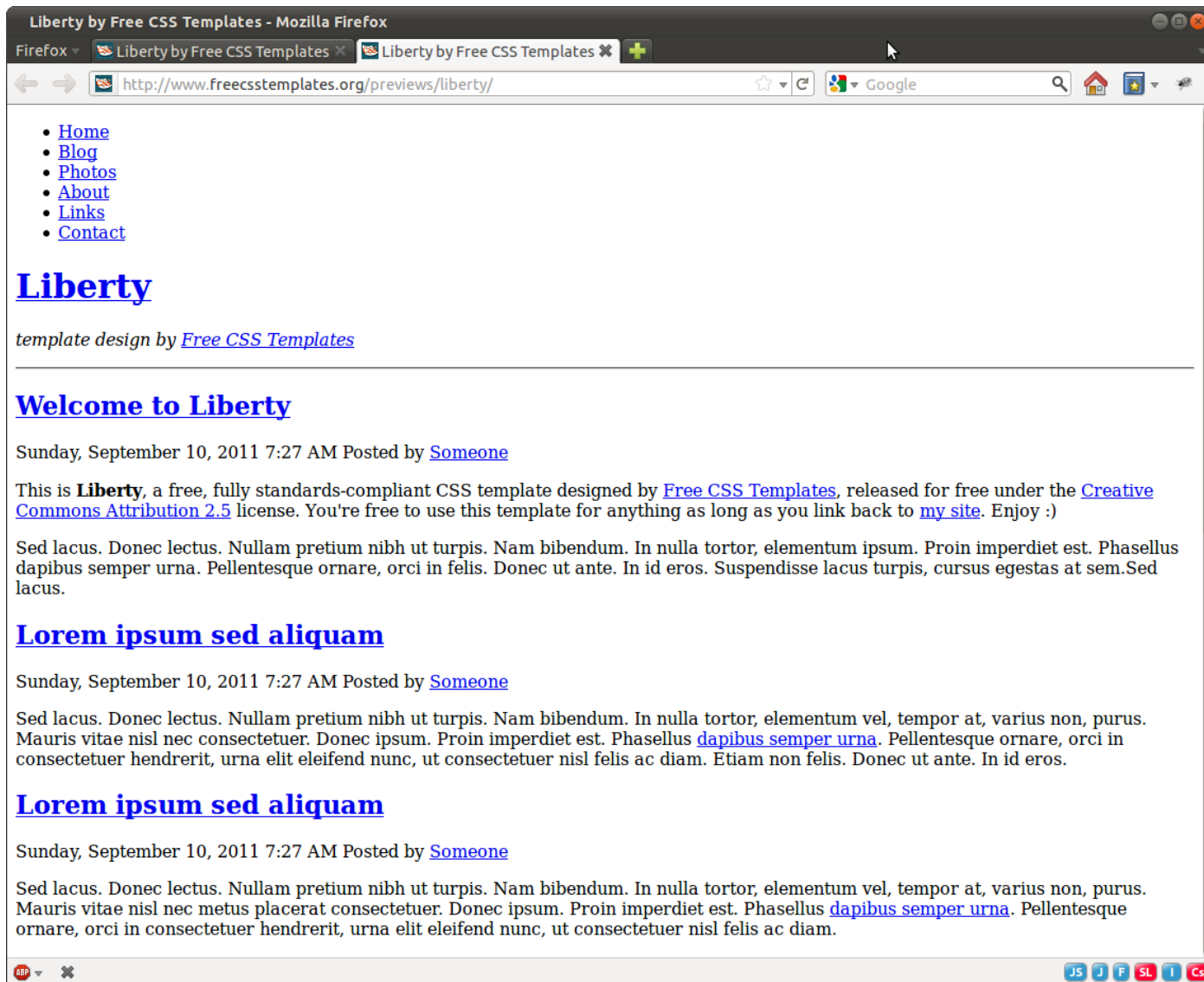
## Grundidee von Stylesheets

- Definition von *Formateigenschaften* für HTML-Elemente
- Strikte Trennung von Inhalt und Layout/Design
- Zentralisierung/Externalisierung von Formatdefinitionen
- Bereitstellung unterschiedlicher Layouts für ein Web-Projekt (adaptive Webseiten)

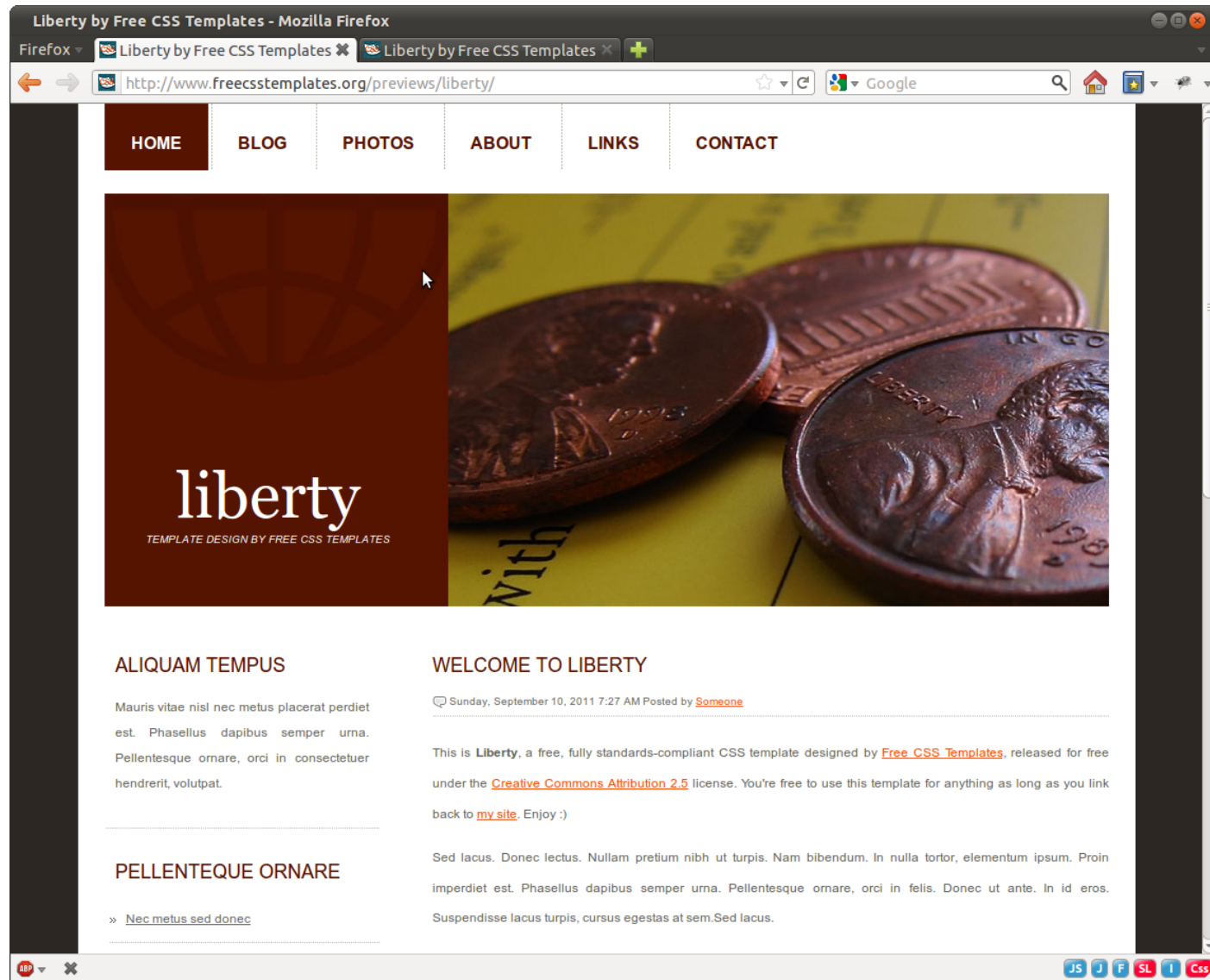
## Stylesheet-Sprachen

- CSS (Cascading Style Sheets): Formatierung von HTML-Dokumenten (W3C-Standard)
- XSL (Extensible Stylesheet Language): Formatierung von XML-Dokumenten (W3C-Standard)

**Was kann CSS?**



**Was kann CSS?**



// TODO:

Image Zoom

# CSS3

## CSS3 Beispiele





## **CSS: Gefahren/Fallstricke**

- Mangelnde Browserunterstützung (IE < 9.0)
- Farbdarstellung oder Schriftgrößen nicht zwingend einheitlich (Rendering Bugs)
- Umfassende Formatierungsmöglichkeiten verführen oft zu übertriebenen Layouts

### **Grundregeln**

- Ordentliche Farbkontraste
- Keine kritischen Farbkombinationen (rot/grün)
- Normale Schriftarten, Größen, Zeilenhöhen, usw.
- Durchgängiges Layout/Design (CI/CD)

## CSS-Definition: HTML Kopf

### Definition zentral für eine HTML-Datei

```
1 <head>
2 <style type="text/css">
3 [selector] { [Eigenschaft]:[Wert]; ...}
4 </style>
5 </head>
```

### Beispiel

```
1 <head>
2 <style type="text/css">
3     h1 { color:green; font-size:24px; }
4 </style>
5 </head>
```

## CSS-Definition: Datei

### Definition in separater CSS-Datei

```
1 <head>
2 <link rel="stylesheet" type="text/css" href="formatierung.css"></link>
3 </head>
```

### CSS-Datei

- Textdatei mit der Endung `.css`
- Zeichenkodierung `@charset "UTF-8";`
- Kommentare: `/* ... */`
- Stylesheet-Definitionen
  - `[selector] { [Eigenschaft]:[Wert]; ... }`

## CSS-Definition: HTML-Element

### Definition innerhalb eines HTML-Elements

```
1 <h1 style="color:blue;">...</h1>
```

- Universalattribut `style="[Eigenschaft]:[Wert];"`
- Kein Selektor (bezieht sich nur auf aktuelles Element)
- Verschachtelte Definitionen
  - Definitionen in externen CSS-Dateien, im HTML-Kopf sowie innerhalb eines HTML-Elements können kombiniert werden
  - Die inneren Definitionen überschreiben die äußeren

`CSSDateien -> HTMLKopf -> HTML-Element`

## Ausgabemedien-spezifisches Design

- Separate Stylesheet-Dateien für unterschiedliche Ausgabemedien
- Einbindung unterschiedlicher CSS-Dateien für unterschiedliche
- Ausgabemedien
  - Mögliche Medien: `all, aural, handheld, print, screen, ...`
  - Begrenzte Unterstützung durch Browser

```
1 <link rel="stylesheet" type="text/css" href="[CSS-Datei]" media="[Medium, ...]">
2 </link>
```

## Ausgabemedien-spezifisches Design

Spezifische Stylesheet-Definitionen für unterschiedliche Ausgabemedien

```
1 <style type="text/css">
2 @media [medium], ...
3   { [CSS-Definitionen] }
4 </style>
```

Einbindung mittels CSS-@import

```
1 <style type="text/css">
2 @import url("[URL]") [Medium], ...;
3 </style>
```

In einer CSS-Datei müssen `@import`-Anweisungen am Anfang der Datei stehen

## CSS-Regeln festlegen

- Aufbau zentraler CSS-Definitionen

```
[Selector] { [CSS-Eigenschaft] : [Wert]; }
```

- **Selektor** wählt aus, für welche Elemente die folgenden Definitionen gelten
- CSS-Definitionen für HTML-Elemente

Definition der Formatierung von HTML-Elementtypen (h1, b, etc.)

```
1 h1 { color:green; font-size:24px }  
2 h1, h2 { color:green; font-size:24px }  
3 * { color:green; font-size:24px }
```

Universalselektor: \*

## CSS-Regeln: Verschachtelte Elemente

CSS-Definitionen für verschachtelte Elemente

Nachfahren-Selektoren

```
1 h1 b { font-style:italic }
```

Mindestens eine Ebene zwischen h1 und b

```
1 h1 * b { font-style:italic }
```

b direktes Kindelement von h1 Nachbar-Selektor

```
1 h1 > b { font-style:italic }
```

p unmittelbarer Nachfolger von h1

```
1 h1 + p { font-style:italic }
```



## CSS-Regeln: Attribute und Klassen

### Formatdefinitionen für Elemente mit bestimmten Attributen oder Attributzuweisungen

```
1 h1[align] { font-style:italic }  
2 h1[align=center] { font-style:italic }  
3 *[align=center] { font-style:italic }
```

### CSS-Definitionen für Klassen

#### Formatdefinitionen für Elemente einer bestimmten Klasse

```
1 h2.neu { color:red }  
2 *.neu { color:red }  
3 /* oder */  
4 .neu { }
```

## CSS-Regeln: class + span/div

- CSS-Definitionen für Klassen mit div und span
- Die Elemente div und span ermöglichen die Zuweisung (semantischer) Klassennamen an beliebige Bereiche oder Gruppen von Elementen
- Diese Bereiche können dann mit CSS formatiert werden

### CSS Code

```
1 .neu { color:red; }  
2 .anmerkung { font-style:italic; font-size:10px }  
3 .quelle { font-style:italic; font-size:8px }
```

### HTML Code

```
1 <span class="neu">...</span>
```

## CSS-Regeln: id / Pseudoklassen

### CSS-Definitionen für einzelne Elemente

Formatdefinitionen für Elemente mit einer bestimmten ID

```
1 #titel { color:red; }  
2 h1#titel { color:red; }
```

### Pseudoelemente und Pseudoklassen

Formatdefinitionen für HTML-Bestandteile, die sich nicht durch ein HTML-Element ausdrücken lassen

```
1 a:link { } /* noch nicht besuchter Verweis */  
2 a:visited { } /* besuchter Verweis */  
3 a:hover { } /* Verweis auf den die Maus zeigt */
```

## CSS-Regeln: Pseudoklassen Before / After

### CSS

```
1 h4:before { content: ">> " }  
2 h4:after { content: url('http://up.frubar.net/1170/css_icon.png') }
```

### HTML

```
1 <h4>Testueberschrift</h4>
```

**>> Testueberschrift**

Beispiel: Linkadressen werden beim Drucken mit ausgedruckt

## Verhalten von HTML-Elementen: Block

```
1 <div>
2   <h1>Überschrift</h1>
3   <p>Etwas Text</p>
4   Nur <b>Text</b> ohne p...
5 </div>
```

# Überschrift

Etwas Text

Nur Text ohne p...

- Block-Elemente
  - Erzeugen einen eigenen Absatz im Textfluss
  - Können i.d.R. enthalten
    - Text (#PCDATA)
    - Block-Elemente **und** Inline-Elemente
    - Beispiele: <h1>, <div>, <table>

## HTML: Inline / CSS: display

- Inline-Elemente
  - Erzeugen keinen Absatz im Textfluss
  - Können i.d.R. enthalten -Text ( `#PCDATA` )
    - Inline-Elemente
  - Beispiele: `<br>`, `<i>`, `<span>`, ...
- Unsichtbare-Elemente
  - Beispiele: `<meta>`, `<style>`, `<script>`

- Steuerbar mit CSS

```
1 display: none | block | inline | ...
```

- Demo
- W3C: Default Style HTML4

# Kaskadierung von Stylesheets

Unterschiedliche Stylesheets (aufst. Priorität)

- **Browser-Stylesheet** (Vorgegeben vom W3C)
  - Grundlegende Formatierung für alle HTML-Elemente
- **Benutzer-Stylesheet** (Im Browser einstellbar)
  - Spezifische Formatierungen des Benutzers
- **Autoren-Stylesheet** (Mit HTML Datei zusammen ausgeliefert)
  - Formatierungen in HTML-Seite oder importierter CSS-Datei
- **Autoren-Stylesheet mit `!important`**
  - Als wichtig gekennzeichnete Formatierungen im Autoren-Stylesheet
- **Benutzer-Stylesheet mit `!important`**
  - Als wichtig gekennzeichnete Formatierungen im Autoren-Stylesheet

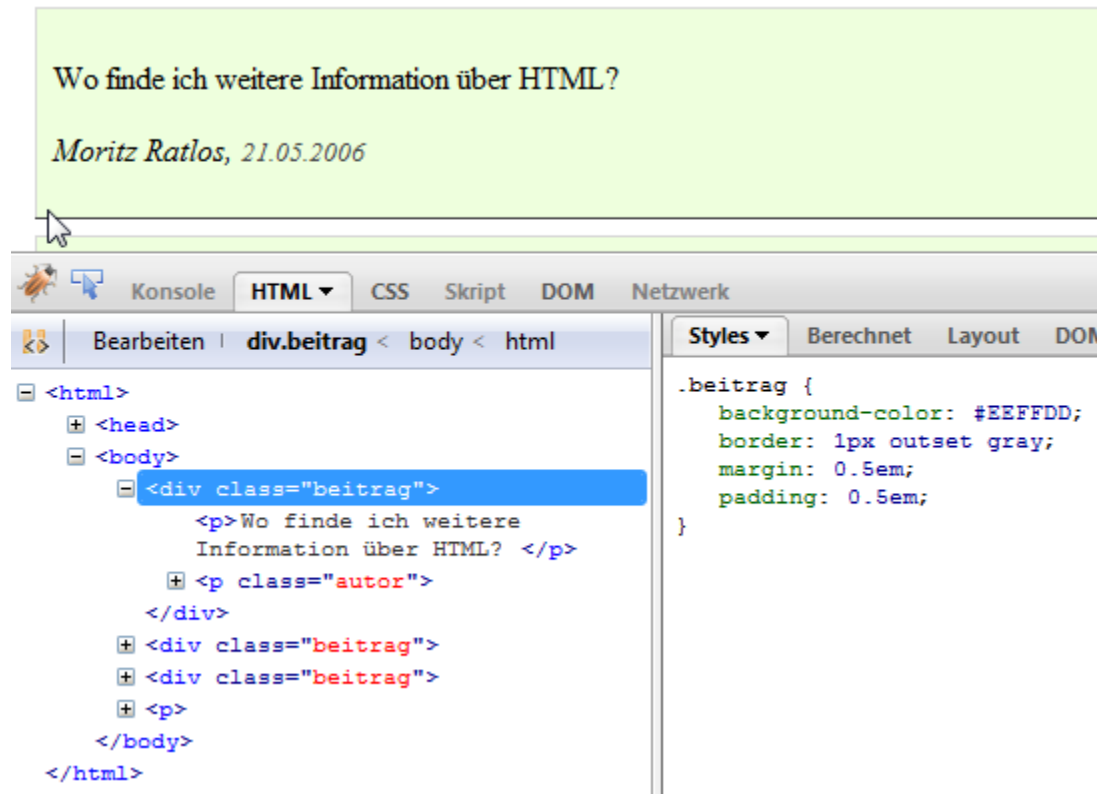
## Kaskadierung von Stylesheets

- Spezifität der Selektoren
  - Grundidee: *spezifischere* CSS-Definitionen *haben Vorrang* vor unspezifischeren
  - Beispiel: `h2.myheading` überschreibt `h2`
- Sortierung nach Spezifität
  - Nach dem *Ort* der Definition: CSS-Definitionen per `style`-Attribute haben Vorrang
  - Nach der *Zahl* der selektieren *ID-Attribute*
  - Nach der *Zahl* der selektieren sonstigen Attribute (z.B. Klassen oder Pseudoklassen)
  - Nach der *Zahl* der selektierten *Elementnamen* und *Pseudoelemente*
  - *Reihenfolge* des Vorkommens (spätere Selektoren überschreiben vorherige)



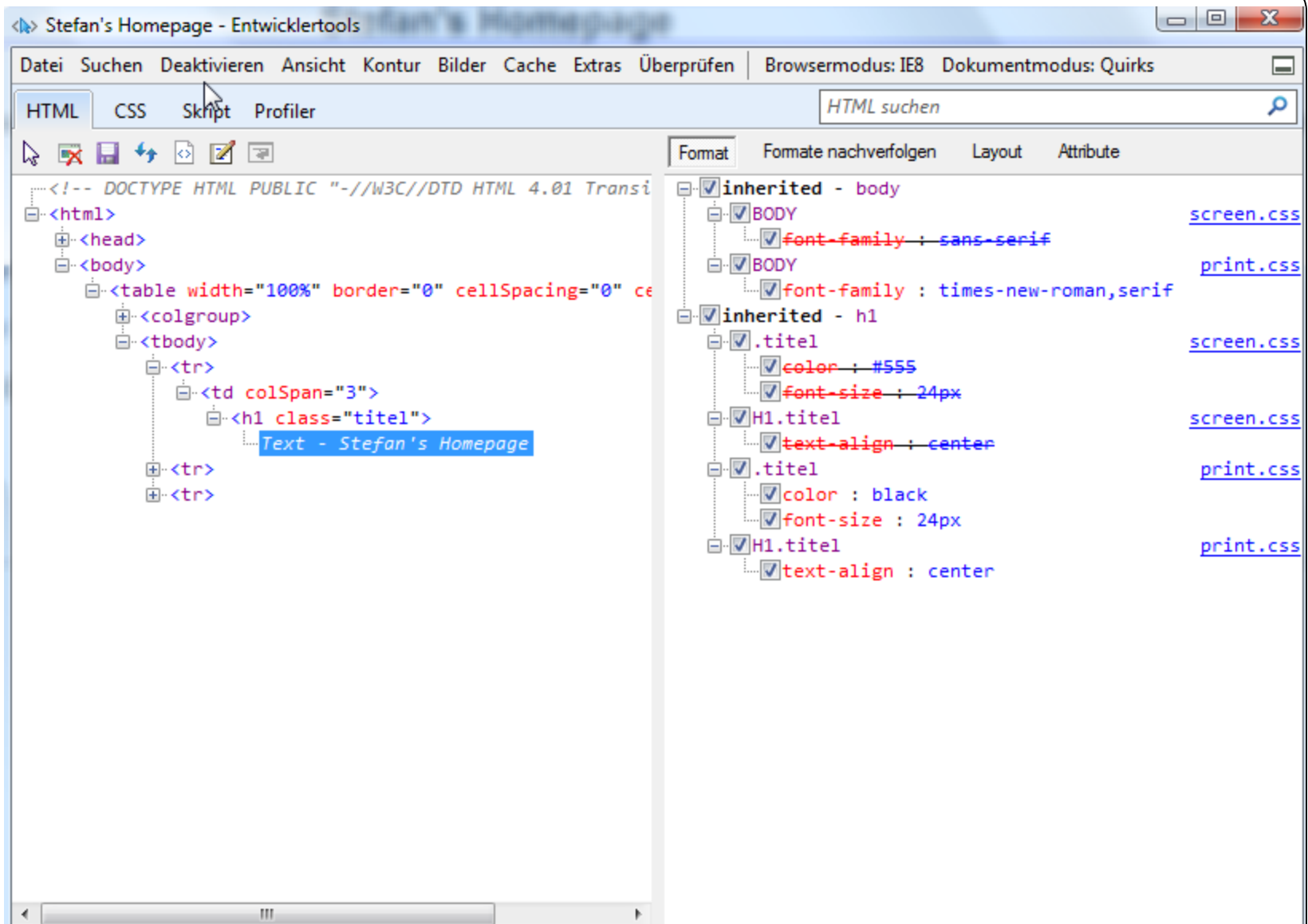
## Analyse von Styles: FF

Analyse der Darstellung von CSS-formatierten Seiten im Firefox mit dem Add-on Firebug



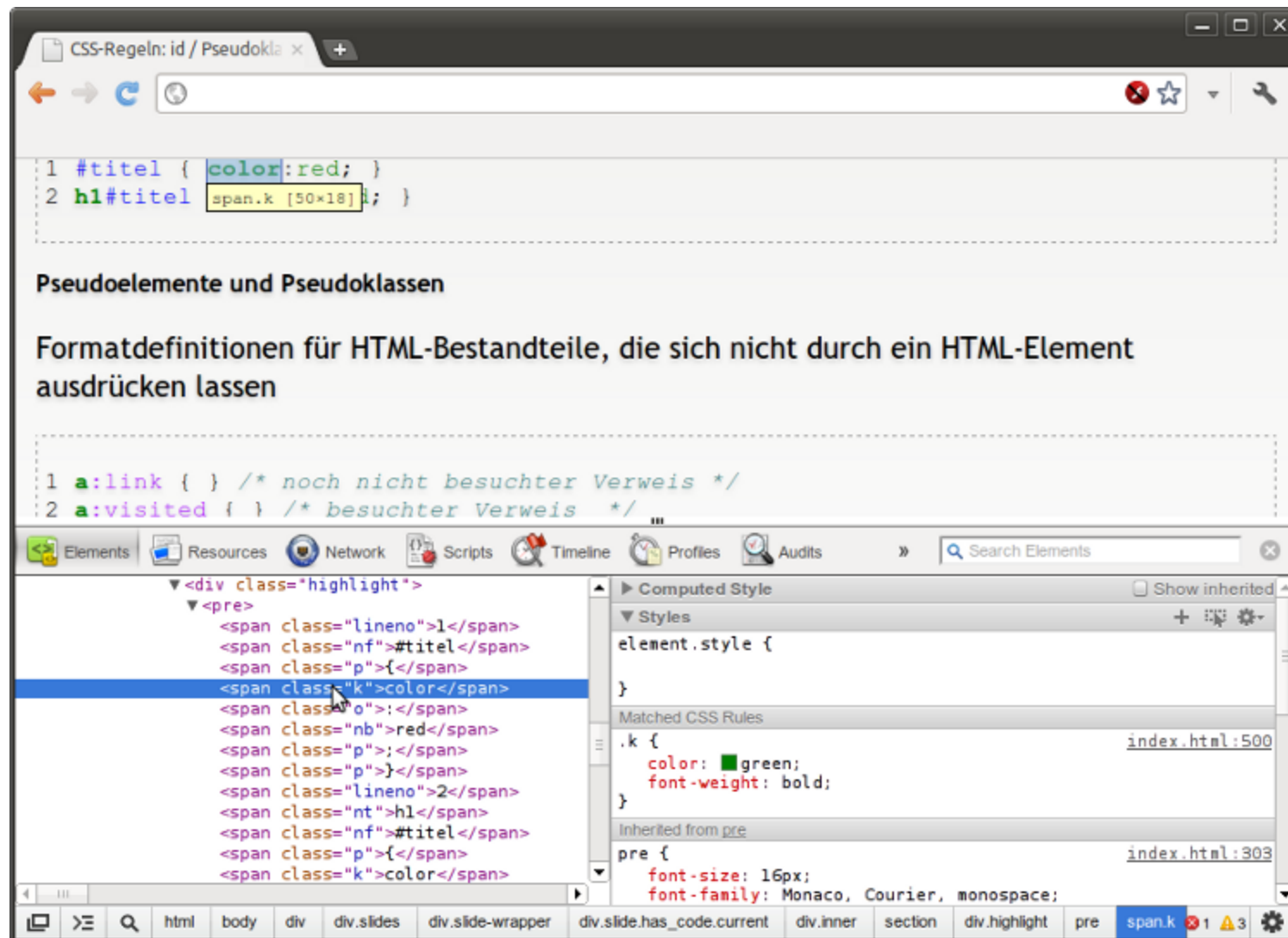
## Analyse von Styles: IE

Analyse der Darstellung von CSS-formatierten Seiten im IE mit den eingebauten Entwicklertools (F12)



## Analyse von Styles: Chrome

Analyse der Darstellung von CSS-formatierten Seiten im Chrome mit den eingebauten Entwicklertools (F12)



## CSS-Eigenschaften: Schriftformat

### Schriftart

```
1 font-family:[Schriftart/Schriftfamilie],[Schriftart/Schriftfamilie],...;
```

- Schriftarten werden in angegebener Reihenfolge verwendet
- Schriftarten mit Leerzeichen werden in Hochkomma gesetzt
- Schriftfamilien: serif, sans-serif, cursive, fantasy, monospace (oft als letzte Angabe)

Beispiel:

```
1 font-family:Verdana,sans-serif
```

### Schriftstil

```
1 font-style:[italic|oblique|normal]
```

## CSS-Eigenschaften: Schriftformat #2

Schriftvariante (1) / Schriftgröße (2)

```
1 font-variant:[small-caps|normal]
2 font-size:[Schriftgroesse]
```

- Numerische Angabe

- Absolut: `px` (pixel), `pt` (point), `mm`, `cm`
- Relativ: `em` (Elementgröße), `ex` (Höhe x), `%`

- Schlüsselwörter

- Absolut: `xx/x-small`, `small`, `medium`, `large`, `x/xx-large`
- Relativ: `smaller`, `larger`

Absolute numerische Angaben für Drucklayouts

Relative Angaben oder Schlüsselwörter für Screen-Layouts

## CSS-Eigenschaften: Schriftformat #3

### Schriftgewicht

```
1 font-weight: [bold|bolder|lighter|normal|100..900]
```

### Schrift gesamt

```
1 font: [font-style  
2 font-variant  
3 font-weight  
4 font-size/line-height  
5 font-family]
```

Reihenfolge ist einzuhalten `font-size` und `font-family` sind obligatorisch `font` setzt zunächst alle Angaben auf Standardwerte

```
1 font: italic bold 1.2em/1.4em Verdana
```



## CSS: Farben

```
1 p {  
2   color: red;  
3   background-color: blue;  
4 }
```

- Vordergrundfarbe: `color`
- Hintergrundfarbe: `background-color`
- Farbangaben:
  - `#RGB` bzw. `#RRGGBB` (Hexadezimal)
  - `rgb(0-255, 0-255, 0-255)`
  - Keywords: `black`, `white`, `red`, `blue`, `green`, `lightblue`
- Beispiel **ROT** `#FF0000` oder `rgb(255,0,0)` oder `red`

# CSS: Hintergrund

## Hintergrundbilder

```
1 p { background: red url(images/bg.png) fixed repeat-y; }
```

**Werte:** background-color, background-image, background-attachment, background-repeat, background-position

**Bild-Adressen mit** url('pfad/zum/bild/Datei.png')

## CSS: Absatzformatierung

### Texteinrückung

```
1 text-indent:[Einrueckung f. erste Zeile]
2 text-indent:1em;
```

### Zeilenhöhe

```
1 line-height:[Zeilenhoehe]
```

### Vertikale Ausrichtung

```
1 vertical-align:[top|middle|bottom|
2 baseline|sub|super|text-top|text-bottom]
```

## CSS: Absatzformatierung

### Horizontale Ausrichtung

```
1 text-align: [left|center|right|justify]
```

### Zeilenumbruch

```
1 white-space: [normal|pre|nowrap]
```

# Listenformatierung

## Darstellungstyp

```
1 list-style-type:[ decimal | lower-roman | upper-roman | lower-alpha |  
2 upper-alpha | disc | circle | square | none ]
```

## Listeneinrückung

```
1 list-style-position:[inside|outside]
```

## Bullet-Graphik

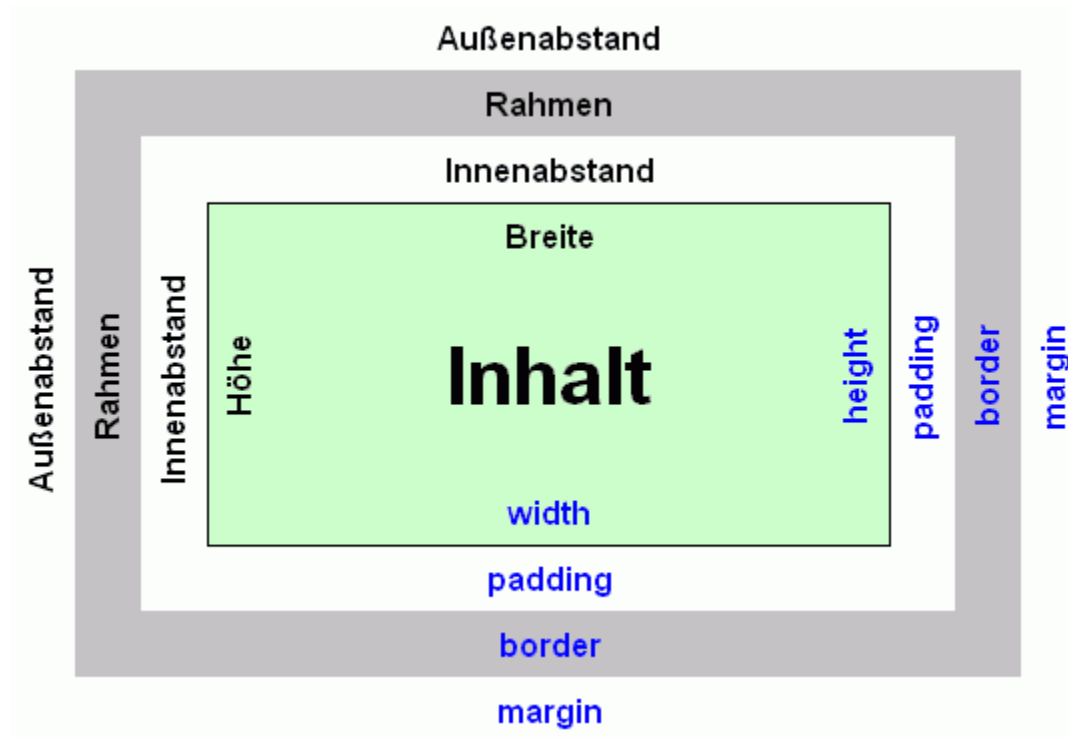
```
1 list-style-image:url([URL|none])
```

## Listenformatierung gesamt

```
1 list-style:[type position image]
```

**Layout**

## Das Box Modell



Quelle: [SelfHTML](#)

## Aussen-Abstand: Margin

Aussen-Abstand oben/rechts/unten/links

```
margin-[top|right|bottom|left]: [Abstandsangabe|auto|inherit]
```

Abstände rechts/links aneinandergrenzender Elemente werden addiert  
Abstände oben/unten aneinandergrenzender Elemente werden nicht addiert sondern der größere Abstand gilt

`auto` oben/unten bewirkt Abstand `0` `auto` links bewirkt rechtsbündige Position `auto` rechts und links bewirkt Zentrierung

```
margin-top:10px; margin-bottom:10px
```



## Aussen-Abstand: Margin

```
1 margin:10px  
2 margin:10px 5px 10px 5px
```

Außenrand/Abstand oben/rechts/unten/links

```
margin:[Angabe 1] ... [Angabe 4]
```

- Eine Angabe für alle 4 Ränder
- Angabe für oben/unten und rechts/links
- Angabe für oben, rechts/links und unten
- Angabe für oben, rechts, unten und links

## Innenabstand: Padding

Innenabstand oben/rechts/unten/links

```
1 padding-[top|right|bottom|left]:[Abstand]
2 padding-top:10px; padding-bottom:10px
3 padding:[Abstand 1] ... [Abstand 4]
```

- Eine Angabe für alle 4 Ränder
- Angabe für oben/unten und rechts/links
- Angabe für oben, rechts/links und unten
- Angabe für oben, rechts, unten und links

## Rahmen: Border

### Rahmentyp

```
1 border-[top|right|bottom|left]-style: [none|hidden|dotted|dashed|solid|double| groove|ridge]
2 border-style:[Angabe 1] ... [Angabe 4]
3 border-style:solid
4 border-top-style:solid
```

### Rahmenbreite

```
1 border-[top|right|bottom|left]-width: [Rahmenbreite|thin|medium|thick]
2 border-width:5px; border-top-width:5px
```

## Rahmen: Border

### Rahmenfarbe

```
1 border-[top|right|bottom|left]-color: [Rahmenfarbe]
2 border-color:blue; border-top-color:blue
```

### Rahmen gesamt

```
1 border-[top|right|bottom|left]: [border-width border-style border-color]
```

Reihenfolge der Angaben ist egal

```
1 border:5px solid red
2 border-top:5px solid red
```

## Rahmen: Border

### Outline (Kontur)

Rahmen für nicht rechteckige Bereiche (Textfluss) Unterschied zu `border`: Konturen werden über dem Element gezeichnet, daher keinen eigenen Platz Outline-Rahmen können nur vollständig geschlossen gezeichnet werden

```
1 outline-width: [Rahmenbreite]
2 outline-style: [Rahmentyp]
3 outline-color: [Rahmenfarbe]
4 outline: [width style color]
```

# Hintergrundfarben und -bilder

## Hintergrundfarbe

```
1 background-color:[Farbe]
```

## Hintergrundbild

```
1 background-image:url([URI])
```

## Wiederholung Hintergrundbild

```
1 background-repeat:[repeat|repeat-x|repeat-y|no-repeat]
```

## Wasserzeichen-Effekt

```
1 background-attachment:[scroll|fixed]
```

# Hintergrundfarben und -bilder

## Hintergrundposition

```
1 background-position:[left|center|right] [top|center|bottom]
2 background-position: left top;
```

## Hintergrund gesamt

```
1 background:[background-color    background-image    background-repeat
2    background-attachment    background-position]
3
4 background:url(wi-team-1.jpg) no-repeat fixed center;
```

# Größe von Elementen

Angabe von Breite und Höhe

`width: [Breite|auto]` Breite des Elements

`min-width: [Breite]` Mindestbreite des Elements

`max-width: [Breite]` Maximalbreite des Elements

`height: [Höhe|auto]` Höhe des Elements

`min-height: [Breite]` Mindesthöhe des Elements

`max-height: [Breite]` Maximalhöhe des Elements



# Position von Elementen

## Art der Positionierung

```
1 position: [static|relative|absolute|fixed]
```

`static` normaler Elementfluss (Normalposition)

`relative` relativ zur Normalposition (Verschieb.)

`absolute` absolute Position (zum nächsthöheren *nicht static* positionierten Element)

`fixed` absolute Position (zum Browserfenster) bleibt beim Scrollen stehen

Mit `absolute` oder `fixed` positionierte Elemente sind nicht Teil des Elementflusses

## Angabe der Position

`top: [Abstand|auto]` Position von oben

`left: [Abstand|auto]` Position von links

`bottom: [Abstand|auto]` Position von unten

`right: [Abstand|auto]` Position von rechts

## Anzeige von Elementen

### Anzeige von übergroßem Inhalt

```
1 overflow:[visible|hidden|scroll|auto]
```

`visible` Inhalt ragt aus Element heraus

`hidden` Inhalt wird abgeschnitten

`scroll` abschneiden + scrollen

`auto` Web-Browser entscheidet

### Textfluss

```
1 float:[left|right|none]
```

Element steht links/rechts und wird von nachfolgendem Text umflossen

```
1 clear:[left|right|both|none]
```

Textumfluss wird abgebrochen und nachfolgender Text beginnt unterhalb des Elements

# Anzeige von Elementen

## Ebene bei Überlappung

```
1 z-index:[Ebene];  
2 z-index:3;
```

## Art der Anzeige

```
1 display:[block|inline|inline-block|list-item|run-in|none]
```

`block` Anzeige als Block

`inline` Anzeige im Textfluss

`inline-block` Block im Textfluss

`list-item` Block mit Bullet

`run-in` kontextabhängig

`none` Keine Anzeige

## Anzeige von Elementen

```
1 visibility:[visible|hidden|collapse]
```

`visible` sichtbar

`hidden` versteckt (Platzhalter)

`collapse` Tabelle gibt Platz frei

Unterschied zwischen `visibility: hidden` und `display: none`: Berechnung der Position anderer Elemente!

## Positionierung Beispiel

TODO

## **Layout Methoden**

### **Veraltete Methoden des Seitenlayouts**

- Frames
  - „Deprecated“: kein gültiges HTML in der Variante Strict
- Tabellenbasiertes Seitenlayout
  - Verletzt die Trennung zwischen Inhalt und Layout

Beide Varianten ermöglichen keine Anpassung an Smartphones/Tablets!!

### **CSS-basiertes Seitenlayout**

- ➔ Flexible Definition der Anordnung der Elemente einer Seite
- ➔ Erleichtert die Pflege der Seiten
- ➔ Ermöglicht alternative Layouts
- ➔ Reduziert den Umfang des HTML-Codes

## Layout Best-Practise

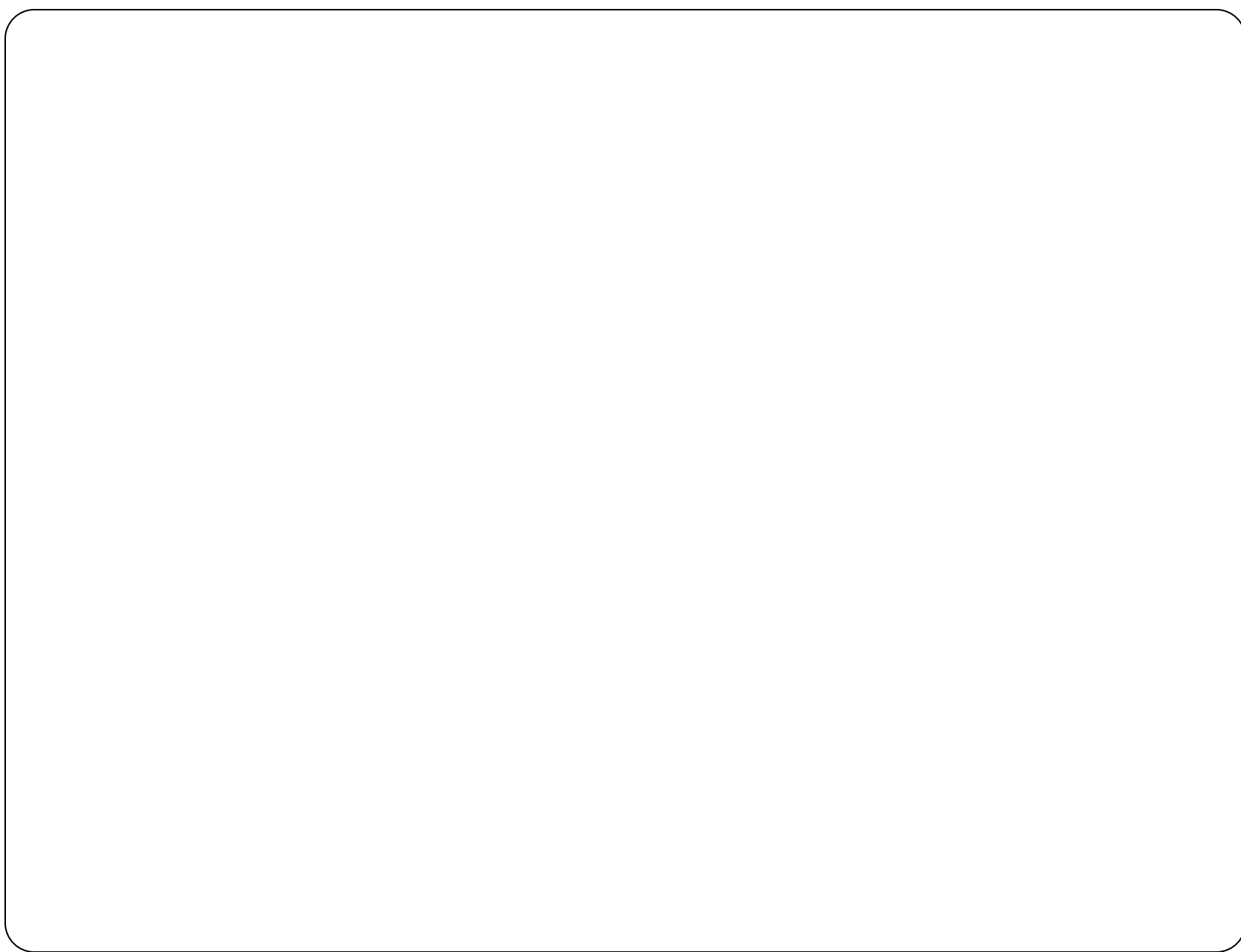
TODO

# HTML Advanced



## Inhalt

- Tabellen
- Formulare



# Java Script

# **Semantic Web**

