



FH Bielefeld
University of
Applied Sciences

Campus Minden



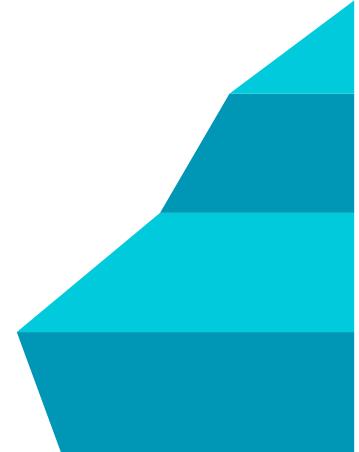
Studiengang Informatik Fachbereich Technik

Webbasierte Anwendungen

SS 2015

Grundlagen

Dozentin: Grit Behrens
<mailto:grit.behrens@fh-bielefeld.de>



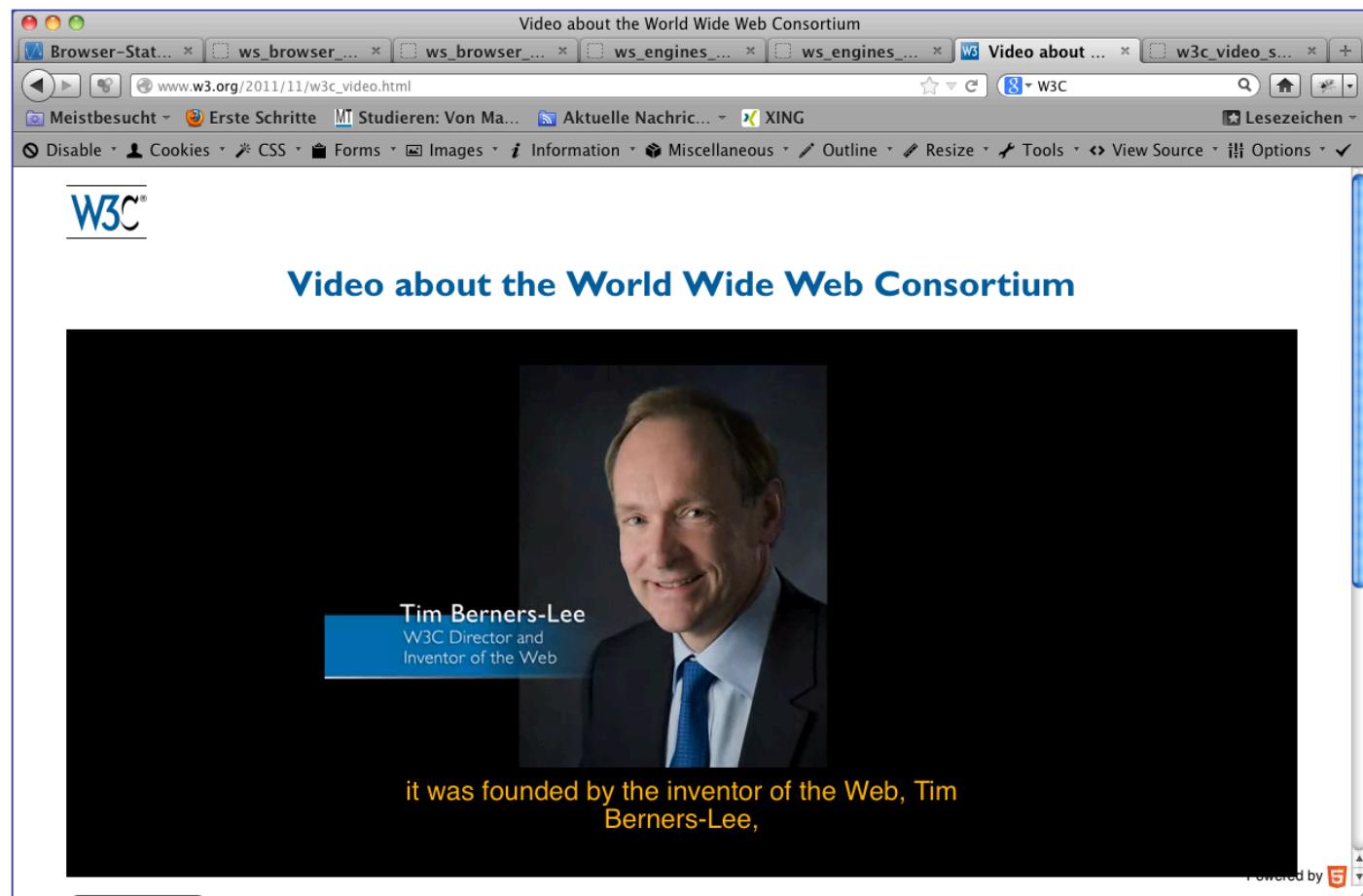
Studiengang Informatik Fachbereich Technik

Lehrinhaltsübersicht zu WBA

1. Einführung in Webbasierte Anwendungen,
2. **Web-Grundlagen, Historie, Struktur von Webseiten: CSS, XML, XHTML, HTML5**
3. Clientseitige Implementierungstechnologien:
Javascript,
4. Clientseitig: JavaScript, DOM, Ajax, (Java-Applet)
4. Serverseitige Implementierungstechnologien:
JSP, Java-Servlet, Webserver
5. WEB-Frameworks

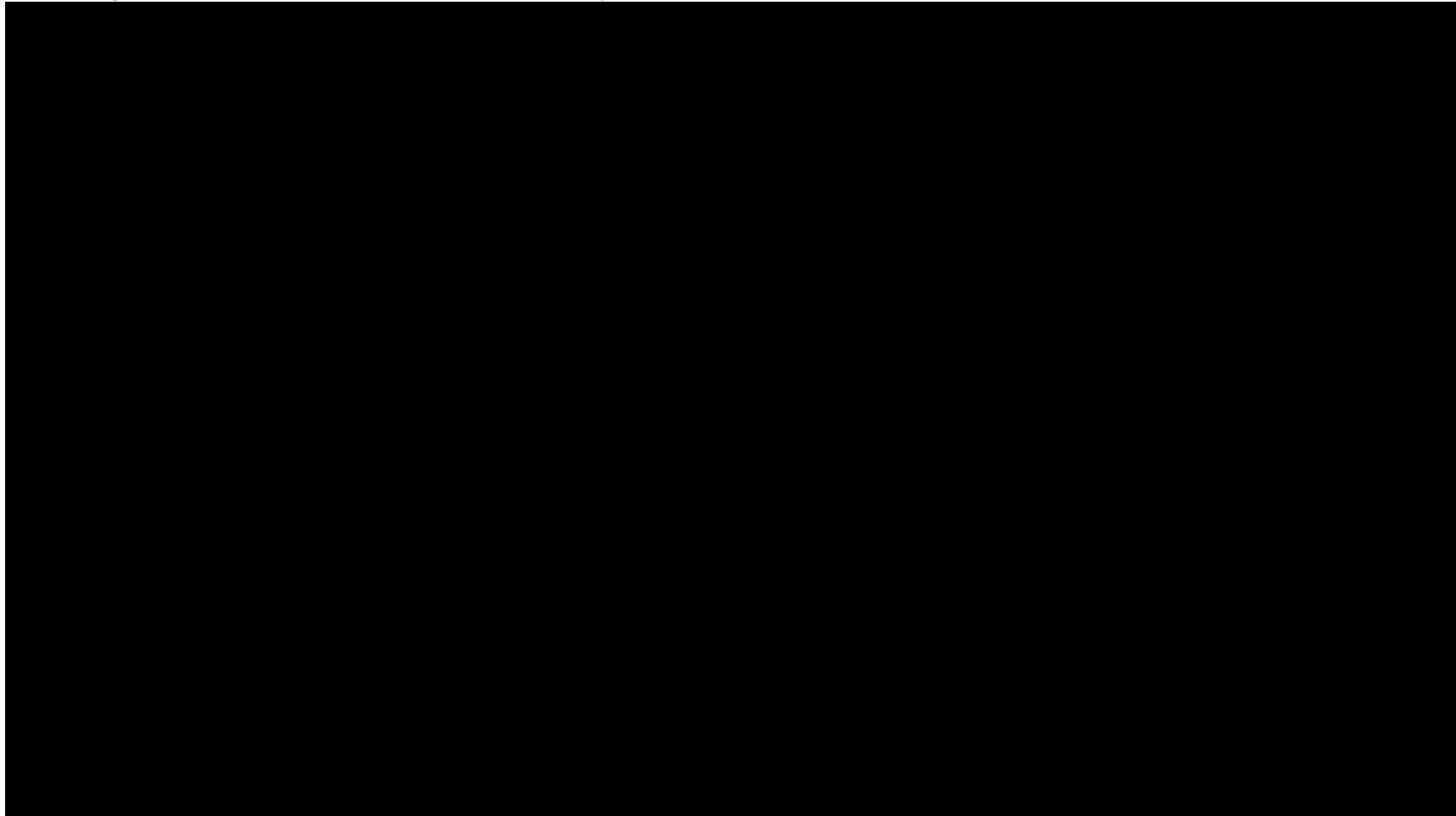
Historie und Bedeutung W3C

- Standards come from W3C.
- Imagefilm W3C http://www.w3.org/2011/11/w3c_video.html



Historie und Bedeutung W3C

- Standards come from W3C.
- Imagefilm W3C http://www.w3.org/2011/11/w3c_video.html



Grundlagen des WWW und die Struktur von Webseiten

1. Grundlagen WWW

1. Etwas zur Historie
2. Einfaches Client-Server-Modell
3. Browsen im WWW
4. Uniform Resource Locator (URL)
5. Mime-Type
6. HyperText Transfer Protocol (HTTP)
7. Statische und dynamische Webseiten

2. Scriptsprachen und HTML

3. Struktur von Webseiten

1. Grundlagen der CSS
2. Ausgewählte Eigenschaften der CSS
3. Validierung von CSS
4. Meta- Tags für Suchmaschinen

Grundlagen des WWW und die Struktur von Webseiten

1. Grundlagen WWW

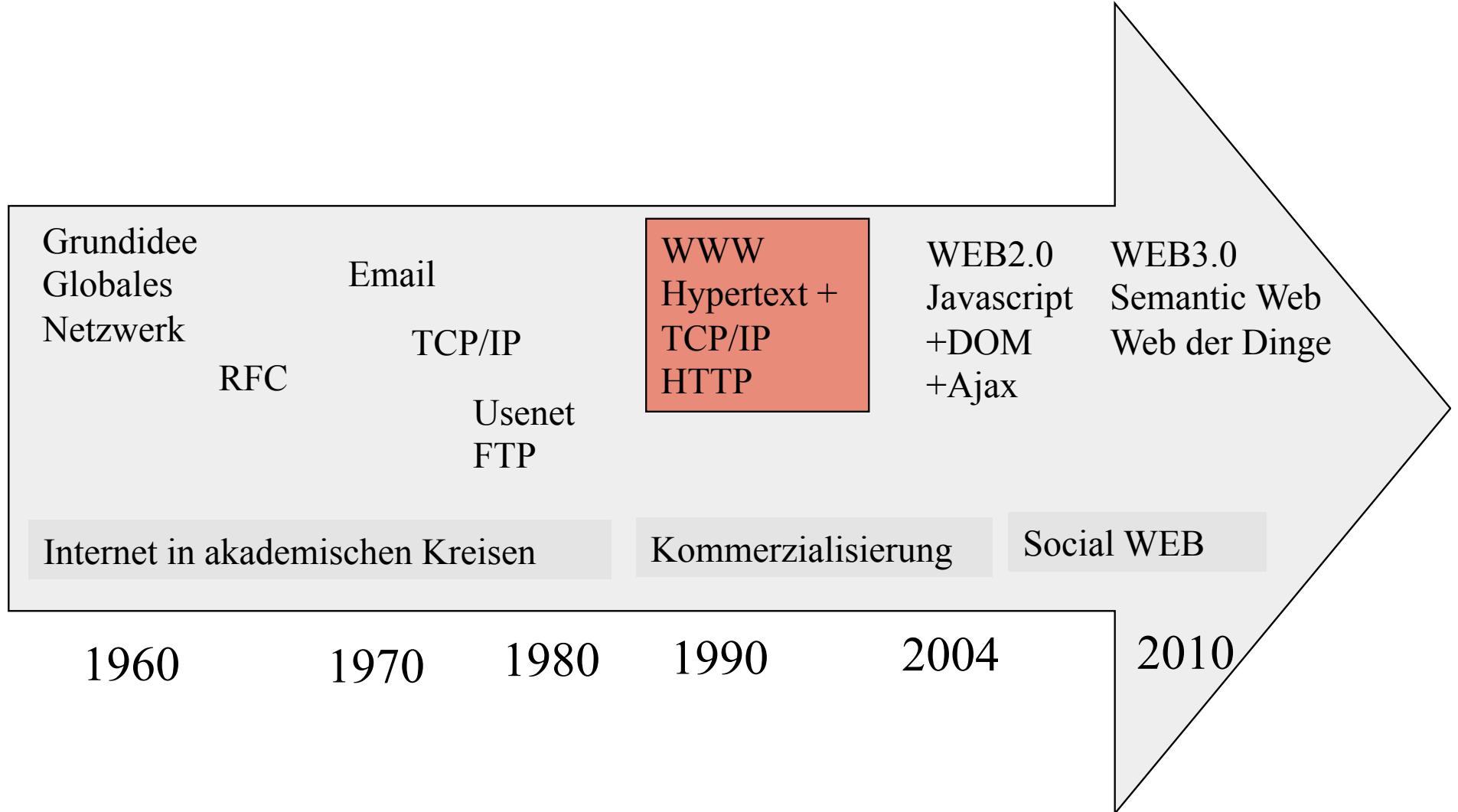
- 1. Etwas zur Historie**
2. Einfaches Client-Server-Modell
3. Browsen im WWW
4. Uniform Resource Locator (URL)
5. Mime-Type
6. HyperText Transfer Protocol (HTTP)
7. Statische und dynamische Webseiten

2. Scriptsprachen und HTML

3. Struktur von Webseiten

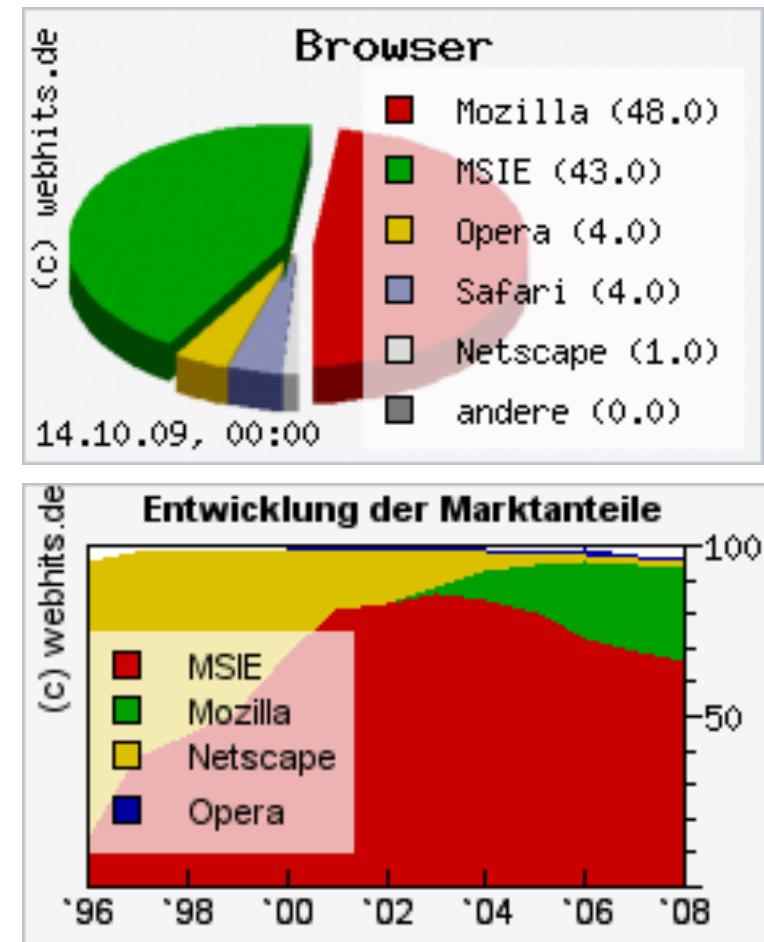
1. Grundlagen der CSS
2. Ausgewählte Eigenschaften der CSS
3. Validierung von CSS
4. Meta- Tags für Suchmaschinen

Historie (Wdh.)



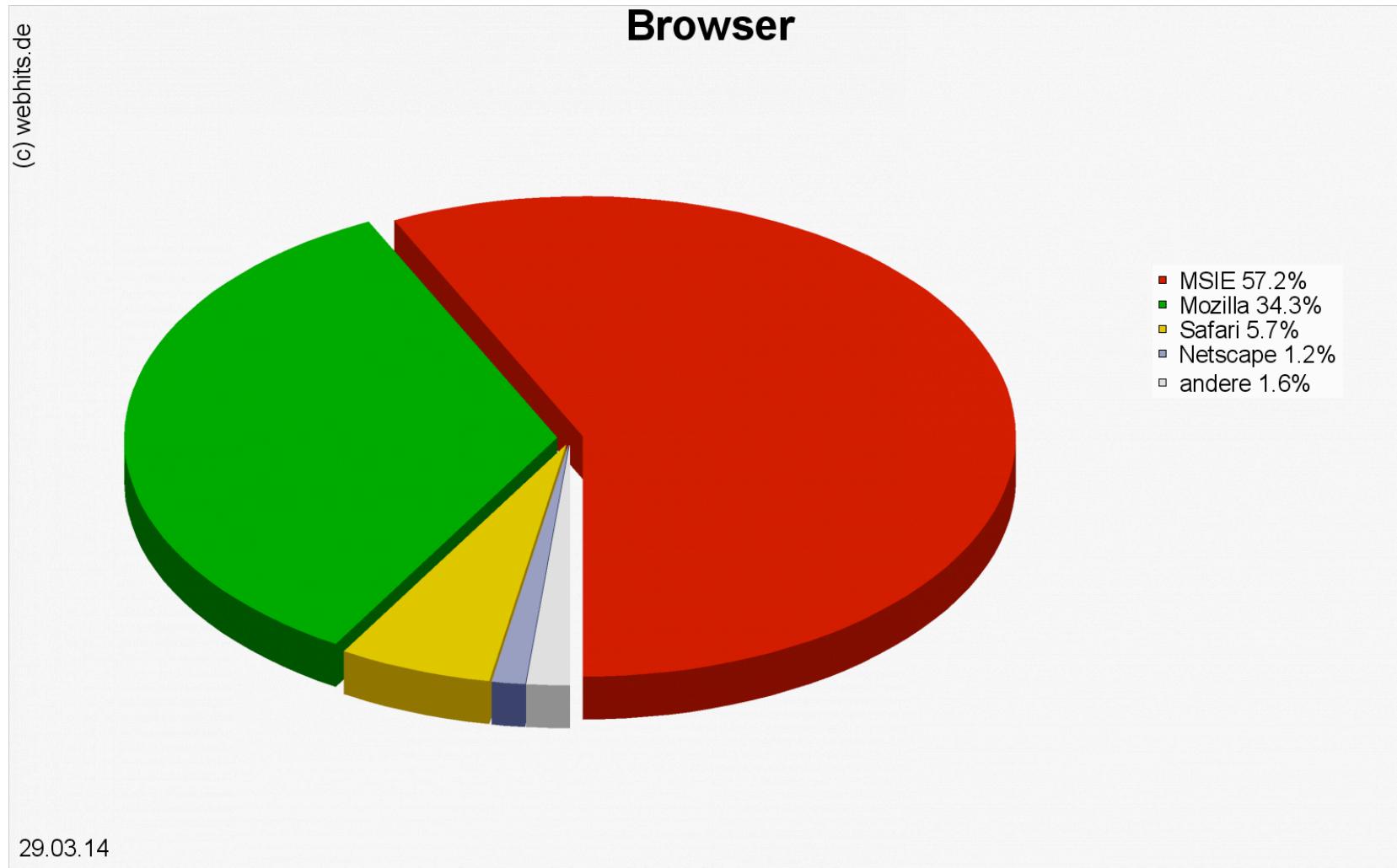
Browserkriege und WWW

- Für Microsoft ist WWW noch kein Thema. **(1994)**
 - Bill Gates: "... an Internet Browser is a trivial piece of software. There are at least 30 companies that have written creditable Internet browsers, so that's nothing..."
- Microsoft greift ein ... **(1995)**
 - ... der Browser-Krieg Microsoft-Netscape beginnt
- Verquickung von Betriebssystem und Browser durch Microsoft wird durch US-Justizministerium verfolgt.
Microsoft gibt nach. **(1998)**



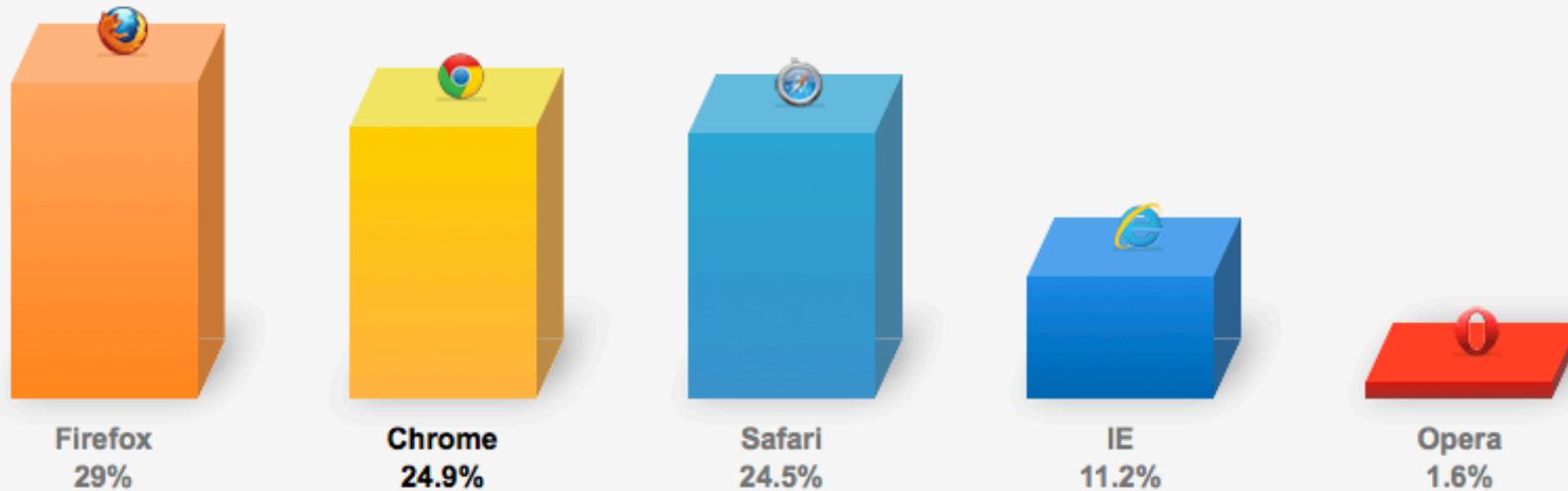
<http://www.webhits.de>

Marktanalyse Browernutzung I



<http://www.webhits.de>

Marktanalyse Browsernutzung III



15.02.-17.03.2015
<http://www.browser-statistik.de>

World-Wide Web (WWW)

- Konsequenzen (*seit 1996*)
 - Sicherung der Interoperabilität durch Entwicklung firmenunabhängiger Standards ("Empfehlungen" bzw. "recommendations"), beginnend mit HTML und CSS
 - Gründung des Gremiums:
World-Wide Web Consortium W3C
<http://www.w3.org>
- Geschichte des W3C unter <http://www.w3.org/Consortium/history>



Tim Berners-Lee,
Erfinder des WWW
Gründungsdirektor des W3C

Web - Meilensteine

Aug. 1993	NCSA-Browser Mosaic : erster Webbrowser, der Grafiken mit anzeigen konnte
Okt. 1994	Gründung des W3C
Dez. 1994	Netscape Navigator 1.0
Sommer 1995	Microsoft Internet Explorer
Dez. 1996	Cascading Stylesheets : deklarative Stylesheet -Sprache für strukturierte Dokumente
Jan. 1997	HTML 3.2 : Tabellen, Textfluss um Bilder, Einbindung von Applets
Dez. 1997	HTML 4.0 : neu sind Stylesheets, Skripte und Frames
Feb. 1998	XML : Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdateien
Mai 1998	CSS, Level 2
Oct. 1998	Document Object Model, Level 1
Jan. 2000	XHTML 1.0 : Eine Neuformulierung von HTML 4.01 mit Hilfe von XML
Okt. 2004	Web2.0
Feb. 2005	Ajax
2007-vorr.2015	Standardisierung von HTML5 im W3C

Was ist das „Web 2.0“ ? (Wdh.)



Was ist das „Web 2.0“ ?

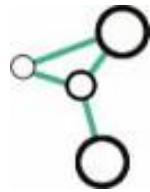
- Begriffseinführung durch O ‘Reilly über Konferenznamen (Oktober 2004)
 - Brainstorming zu diesem Begriff
- Artikel „What is Web 2.0“ von Tim O ‘Reilly (September 2005) (<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>)
 - Artikel allgemein stark kritisiert , da keine neuen Technologien
- Begriff ***klingt gut und suggeriert etwas Neues*** -> Marketingbegriff
- massenhafte Verwendung des Begriffes
- Interaktivität im Mittelpunkt:
 - User generated Content in Blogs, Wikis und Foren
 - Flickr (Fotoportal)
 - YouTube (Videoportal)

Was ist das „Web 2.0“ ?

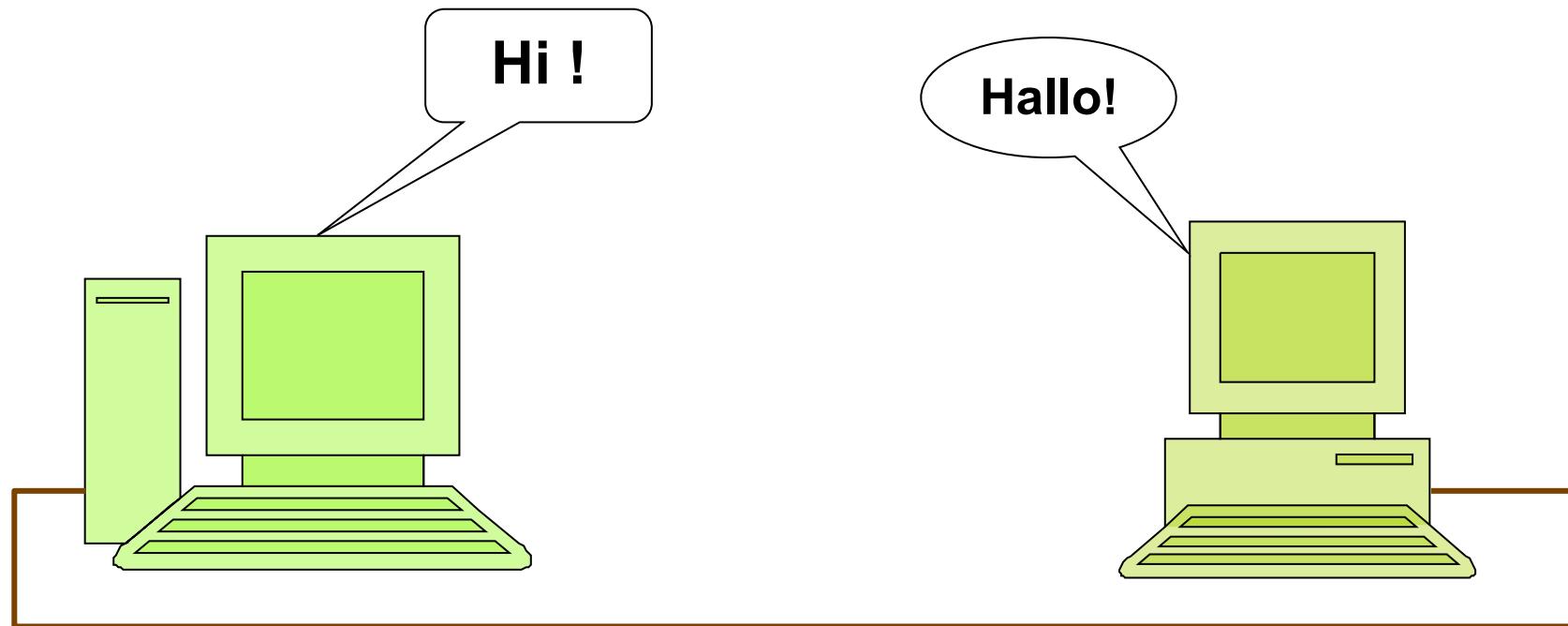
In our initial brainstorming, we formulated our sense of Web 2.0 by example:

- | | |
|------------------------------|----------------------------|
| • Web 1.0 | Web 2.0 |
| • DoubleClick | Google AdSense |
| • Ofoto | Flickr |
| • mp3.com | Napster |
| • Britannica Online | Wikipedia |
| • personal websites | blogging |
| • domain name speculation | search engine optimization |
| • page views | cost per click |
| • screen scraping | web services |
| • Publishing | participation |
| • content management systems | wikisdirectories |

Auszug aus dem Artikel. „What is web 2.0“

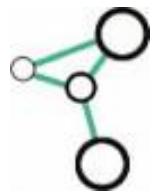


Was ist Web 3.0? (Wdh.)



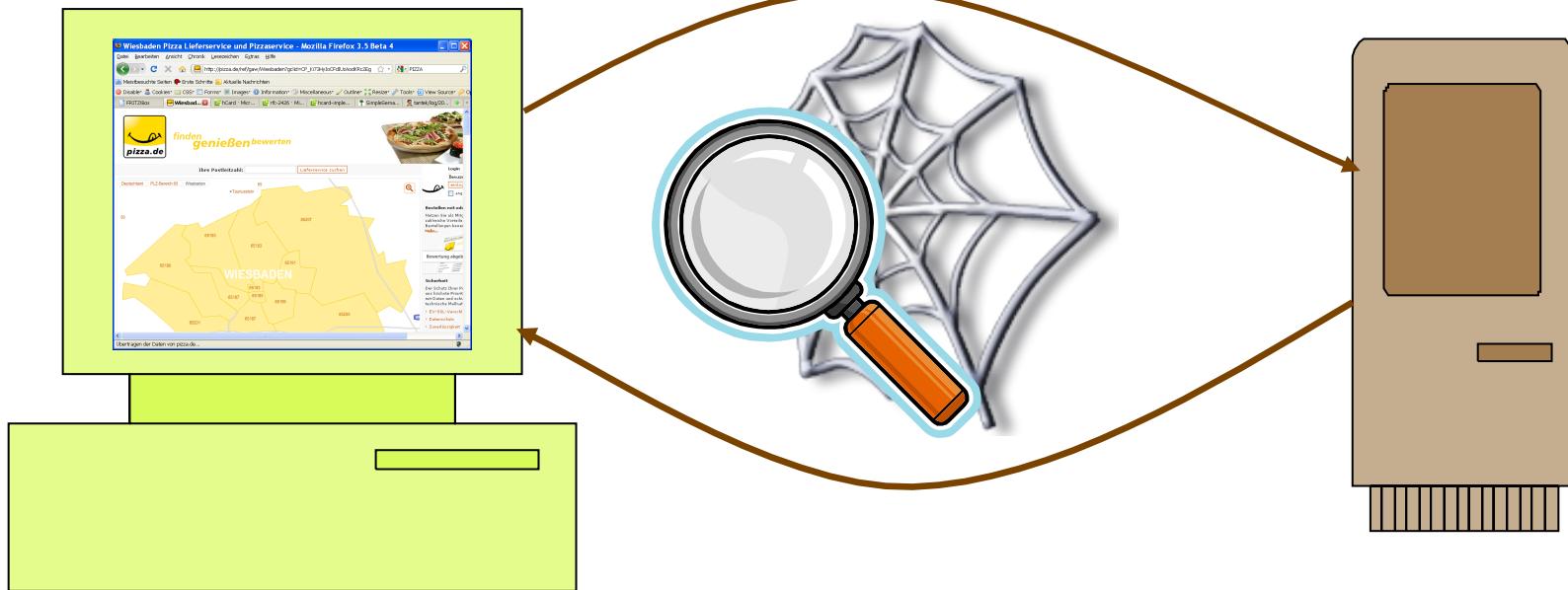
Syntax und Semantik spielen eine Rolle in der Kommunikation.
Das Internet ist für die Kommunikation zwischen Computern, zum Datenaustausch zwischen Computern entworfen worden.

„Hi“, „Hello“ - ohne Verständnis, nur Informationsaustausch, aber kein Lernen von Bedeutungen im reinen Entwurf des Internets.

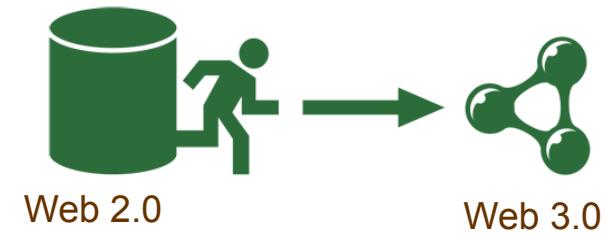


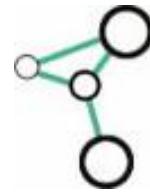
Was ist Web 3.0

Problem !

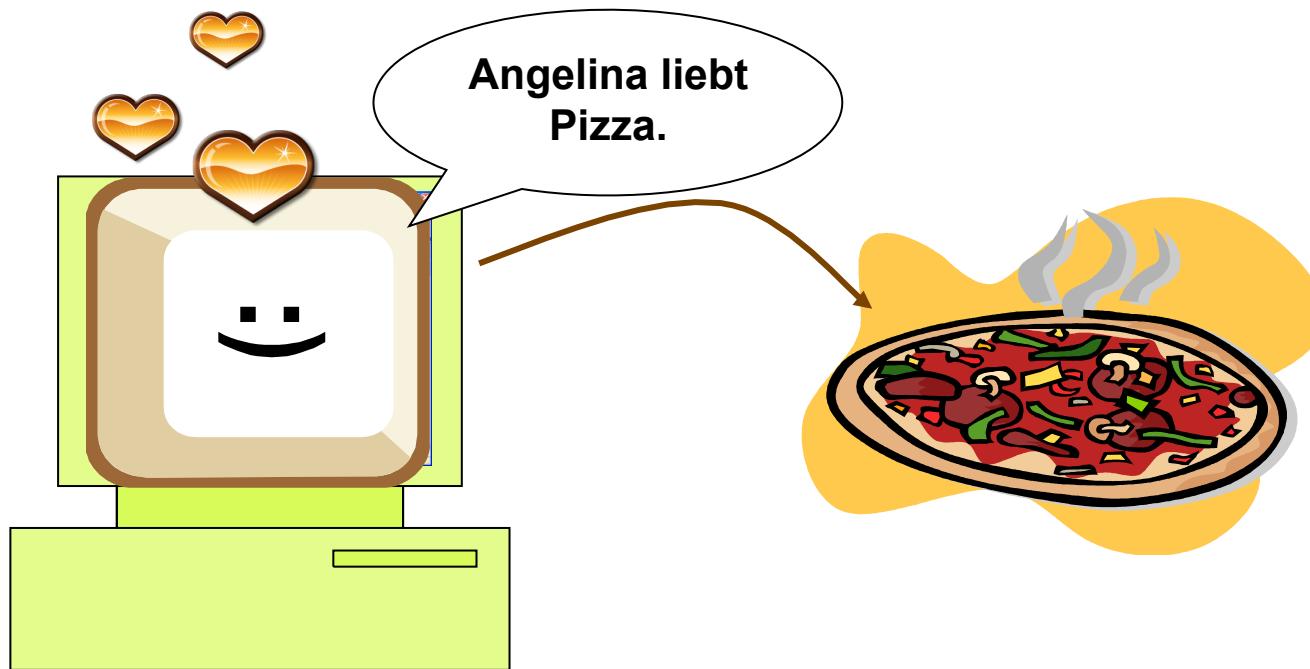


PROBLEM Immer mehr Dokumente (WEB 2.0) ->
Suchmaschinen finden zu viele Antworten, Auswahl
der korrektesten oft nicht mehr möglich
Überflutung: -Antwort Semantics!

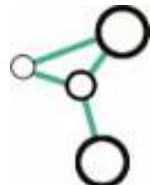




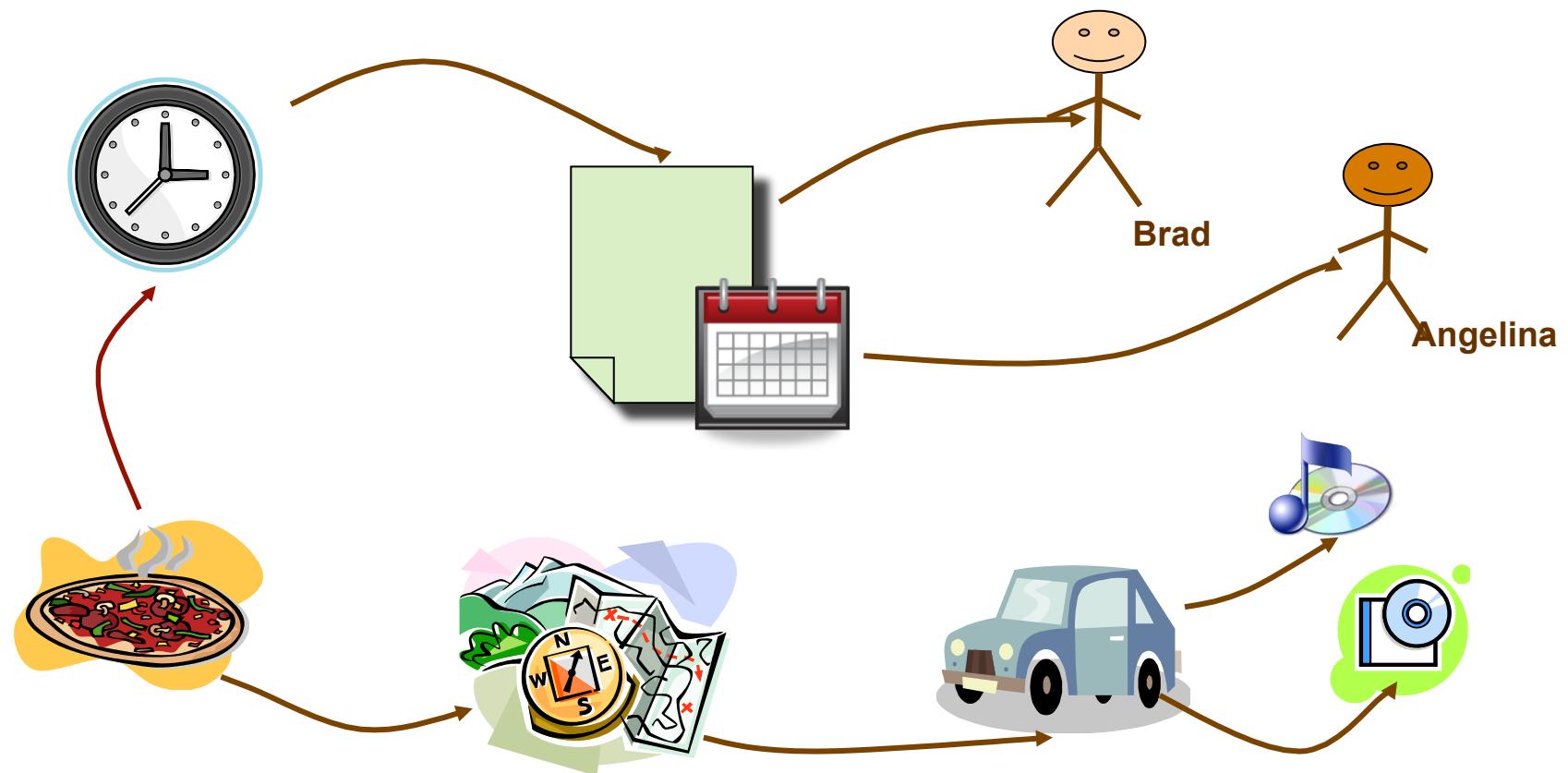
Web 3.0 - Das Semantic Web



***Erkennungsalgorithmen für Sematische Bedeutungen innerhalb der Webseiten. -> Lernen möglich.
-> Computer beginnen aktiv zu helfen
Semantic web hilft den Computern die Bedeutung von Webpages zu verstehen.***



Semantic Web = „Web der Dinge“



Das Sematic Web besteht aus Dingen, Begriffen, Konzepten.
„Web der Dinge“ : über Menschen, Ereignisse, Lokalisationen, Organisationen, Musik, Filme und über Relationen zwischen den Dingen.

Grundlagen des WWW und die Struktur von Webseiten

1. Grundlagen WWW

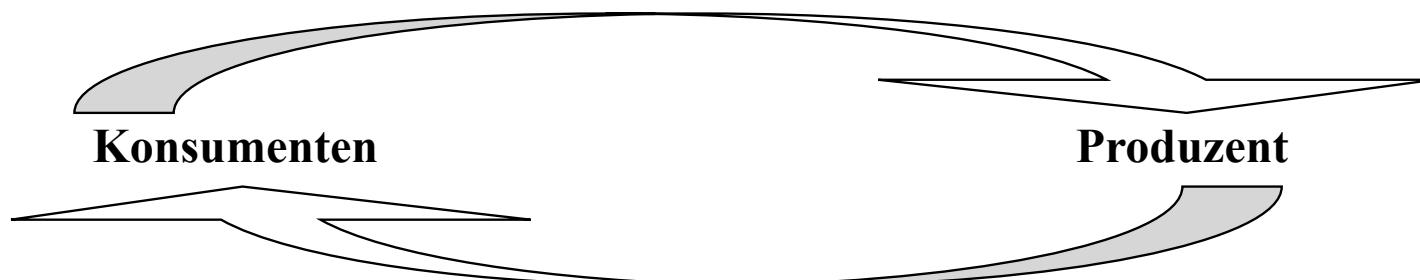
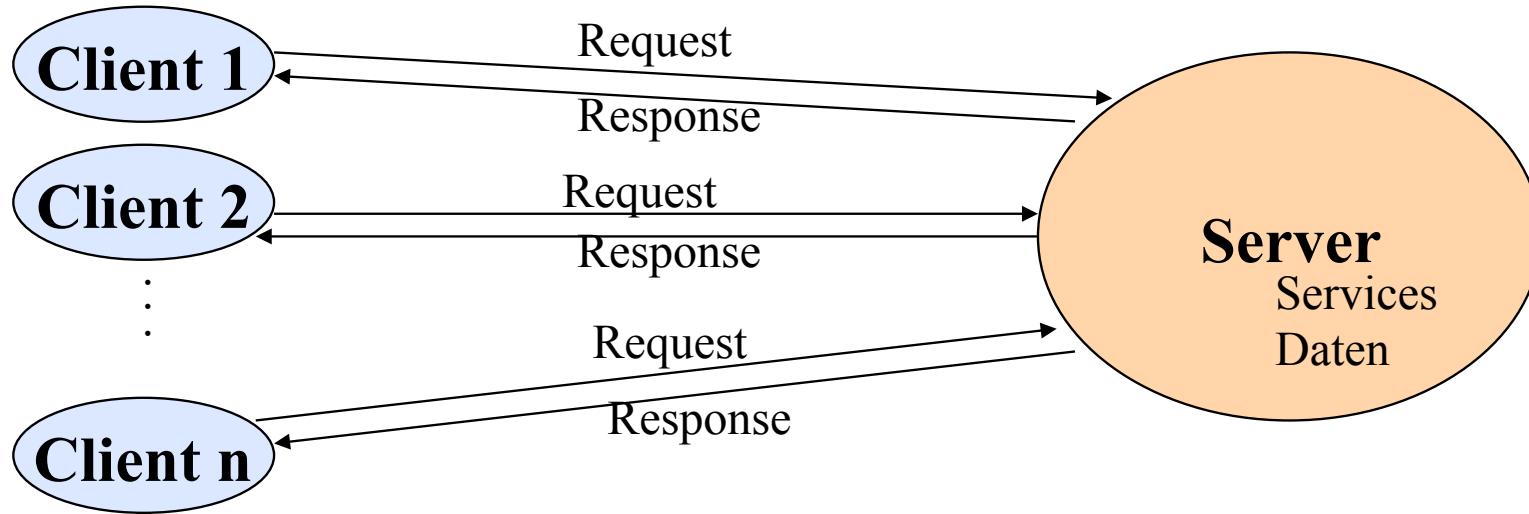
1. Etwas zur Historie
- 2. Einfaches Client-Server-Modell**
3. Browsen im WWW
4. Uniform Resource Locator (URL)
5. Mime-Type
6. HyperText Transfer Protocol (HTTP)
7. Statische und dynamische Webseiten

2. Scriptsprachen und HTML

3. Struktur von Webseiten

1. Grundlagen der CSS
2. Ausgewählte Eigenschaften der CSS
3. Validierung von CSS
4. Meta- Tags für Suchmaschinen

Einfaches Client-Server-Modell:



Grundlagen des WWW und die Struktur von Webseiten

1. Grundlagen WWW

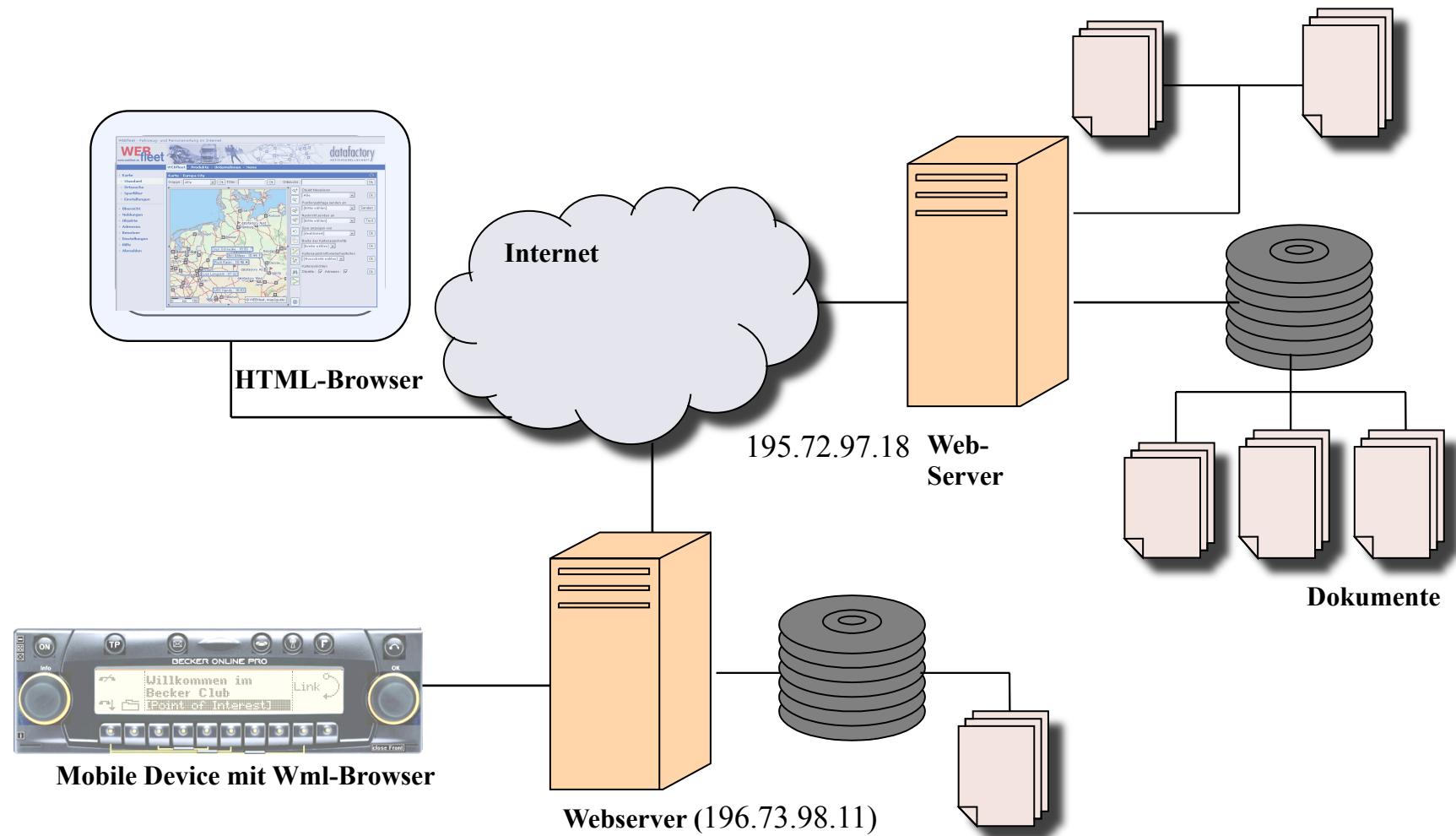
1. Etwas zur Historie
2. Einfaches Client-Server-Modell
- 3. Browsen im WWW**
4. Uniform Resource Locator (URL)
5. Mime-Type
6. HyperText Transfer Protocol (HTTP)
7. Statische und dynamische Webseiten

2. Scriptsprachen und HTML

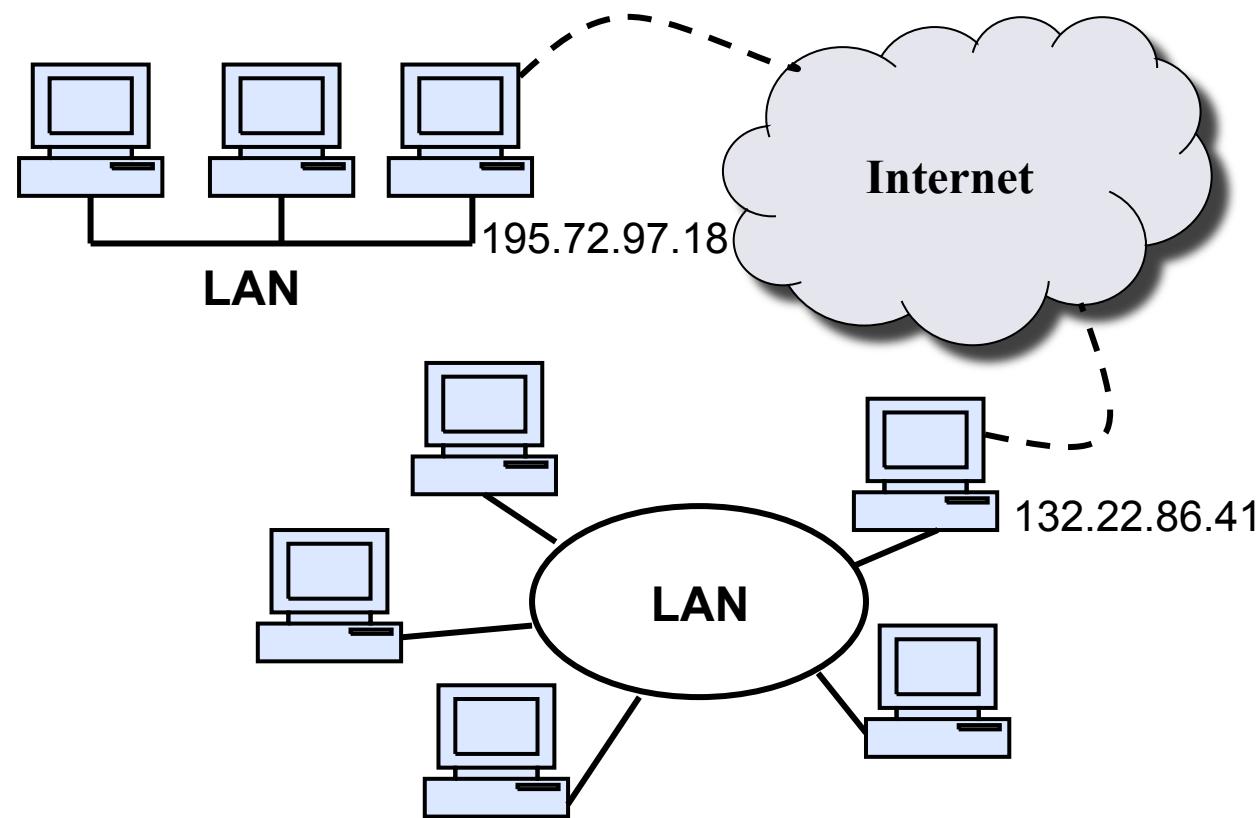
3. Struktur von Webseiten

1. Grundlagen der CSS
2. Ausgewählte Eigenschaften der CSS
3. Validierung von CSS
4. Meta- Tags für Suchmaschinen

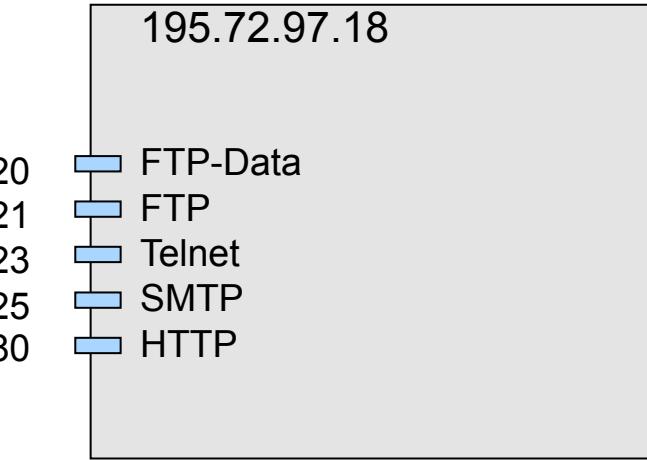
Browsen im WWW:



Infrastruktur des Internets



TCP/IP Protokollfamilie:



- ***IP (Internet Protocol)***

- paketorientiert, verbindungslos, unzuverlässig
- Weiterleiten von Daten an eindeutige IP-Adressen
(Rechner od. Subnetze)
- legt Paketformat fest

- ***TCP (Transmission Control Protocol)***

- Auswahl des Ports (Programms) auf dem Zielrechner

Grundlagen des WWW und die Struktur von Webseiten

1. Grundlagen WWW

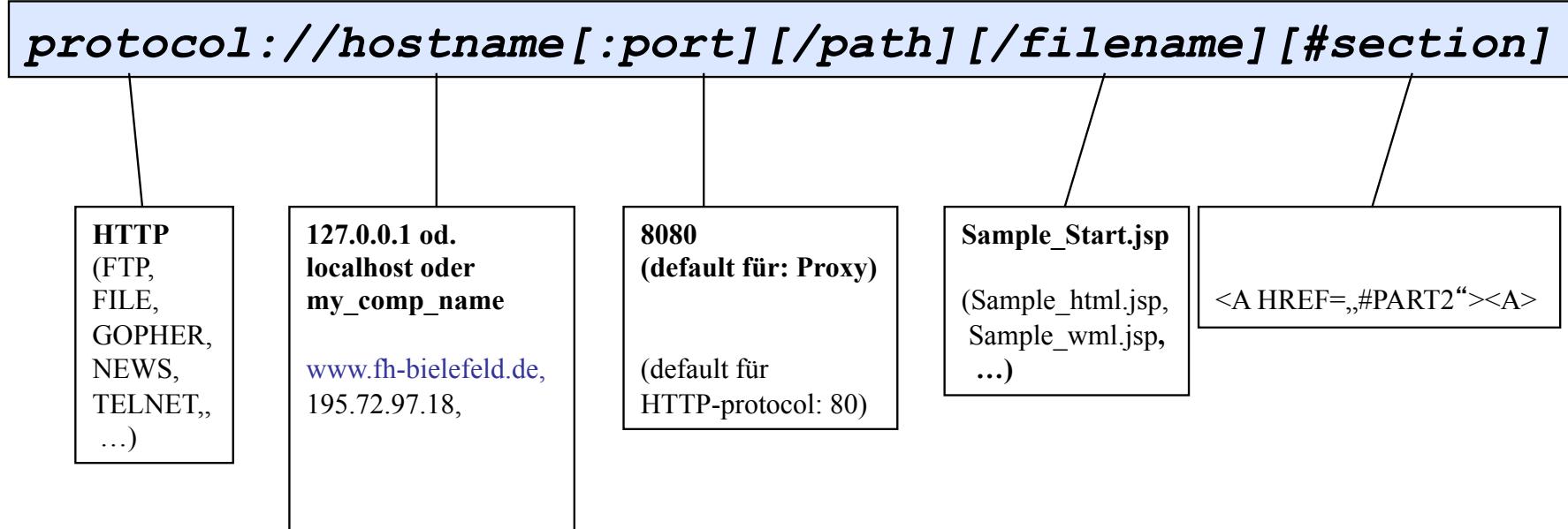
1. Etwas zur Historie
2. Einfaches Client-Server-Modell
3. Browsen im WWW
- 4. Uniform Resource Locator (URL)**
5. Mime-Type
6. HyperText Transfer Protocol (HTTP)
7. Statische und dynamische Webseiten

2. Scriptsprachen und HTML

3. Struktur von Webseiten

1. Grundlagen der CSS
2. Ausgewählte Eigenschaften der CSS
3. Validierung von CSS
4. Meta- Tags für Suchmaschinen

Der Uniform Resource Locator (URL)



Anwendungsbeispiel: `http://127.0.0.1:8084/source/Sample_Start.jsp`

Der Uniform Resource Locator (URL)

Vorgang im Browser:

1. Client aktiviert im Browser die URL
2. Browser bestimmt ggf. durch Domain Name Systems (DNS) die IP-Adresse
3. Browser baut mit IP-Adresse eine TCP-Verbindung auf und schickt dem Webserver eine Seitenanforderung (Request)
4. WebServer schickt dem Browser die gewünschte Seite zurück (Response)
5. TCP-Verbindung wird wieder gelöst
6. Browser bringt Webseite zur Anzeige

Grundlagen des WWW und die Struktur von Webseiten

1. Grundlagen WWW

1. Etwas zur Historie
2. Einfaches Client-Server-Modell
3. Browsen im WWW
4. Uniform Resource Locator (URL)
- 5. Mime-Type**
6. HyperText Transfer Protocol (HTTP)
7. Statische und dynamische Webseiten

2. Scriptsprachen und HTML

3. Struktur von Webseiten

1. Grundlagen der CSS
2. Ausgewählte Eigenschaften der CSS
3. Validierung von CSS
4. Meta- Tags für Suchmaschinen

MIME - Typ

- ursprünglich: *Multipurpose Internet Mail Extension*
- heute: *Multimedia Internet Message Extension*
 - auch: **Medientyp**

Syntax: Haupttyp/Untertyp

Beispiele:

- *text/plain, text/html*
- *image/gif, image/jpeg*
- *application/pdf, application/msword*

Grundlagen des WWW und die Struktur von Webseiten

1. Grundlagen WWW

1. Etwas zur Historie
2. Einfaches Client-Server-Modell
3. Browsen im WWW
4. Uniform Resource Locator (URL)
5. Mime-Type
- 6. HyperText Transfer Protocol (HTTP)**
7. Statische und dynamische Webseiten

2. Scriptsprachen und HTML

3. Struktur von Webseiten

1. Grundlagen der CSS
2. Ausgewählte Eigenschaften der CSS
3. Validierung von CSS
4. Meta- Tags für Suchmaschinen

Hypertext Transfer Protocol (HTTP)

1. Herstellung der Verbindung - URL

2. Senden der Anforderung

z.B. **GET /Sample_Start.jsp HTTP/1.1**

Accept: text/html, image/gif, image/jpeg ; q=0.9 (Qualitätsfaktor 0.0 - 1)

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)

Accept-Language: en; q=0.5, de; q=0.9...

3. Die Rückantwort

z.B. **HTTP/1.1 OK 200** (200 -299 Erfolg,

400-499 Fehler beim Client

z.B. 404 Not Found

500-599 Fehler beim Server

z.B. 500 Server Error

Date: Mo, 18 Oct 2004 23:20:55 GMT

Server: Apache/1.3.24 (Win32) PHP/4.3.0

Content-Type: text/html (MIME)

<html> ... </html> (Dokument)

4. Schließen der Verbindung

- Bis hierher

Request-Methoden im HTTP

DELETE

GET

HEAD

OPTIONS

POST

PUT

TRACE

Server-Antwortcodes (Auswahl) I

100-199	Information während die Anfrage auf dem Server bearbeitet wird		
100	Continue	Anfang der Frage empfangen	
101	Switching Protocols	Server stimmt Protokollwechsel zu	
200-299	Erfolgreiche Anfrage, Aktion wird ausgeführt		
200	OK	angeforderte Daten werden geliefert	
201	Created	Datei wurde auf dem Server angelegt	
202	Accepted	Anfrage akzeptiert, wird später ausgeführt	
300-399	Umleitung der Anfrage, weitere Bearbeitung notwendig		
300	Multiple Choices	Liste von mehreren Dateien zur Auswahl wird gesendet	
301	Moved Permanently	Dokument verschoben, neue Adresse ggf. im Header enthalten	
302	Moved Temporarily	temporär verschoben	
304	Not modified	wertet Header-Anfrage „if-modified-since“ aus falls Dokument seit ... nicht aktualisiert wird eine Kopie aus dem Browsecache verwendet	

Server-Antwortcodes (Auswahl) II

400-499

Anfrage unvollständig oder fehlerhaft, Abbruch

400	Bad Request	Syntaxfehler im Request
401	Unauthorized	keine Berechtigung für Webbereich
403	Forbidden	Aufruf von geschützten Verzeichnissen
404	Not Found	Dokument existiert nicht auf dem Server

500-599

Fehler auf dem Server aufgetreten

500	Internal Server Error	Anfrage kann nicht beantwortet werden, da Fehler auf dem Server aufgetreten
501	Not Implemented	Aktion kann vom Server nicht ausgeführt werden
502	Bad Gateway	Server konnte anderen Server oder Proxy nicht erreichen, der zur Anfragebearbei- tung notwendig gewesen war
503	Service Unavailable	Server ist vorübergehend nicht verfügbar Im header „retry after“ gibt Zeitdauer an

Grundlagen des WWW und die Struktur von Webseiten

1. Grundlagen WWW

1. Etwas zur Historie
2. Einfaches Client-Server-Modell
3. Browsen im WWW
4. Uniform Resource Locator (URL)
5. Mime-Type
6. HyperText Transfer Protocol (HTTP)
- 7. Statische und dynamische Webseiten**

2. Scriptsprachen und HTML

3. Struktur von Webseiten

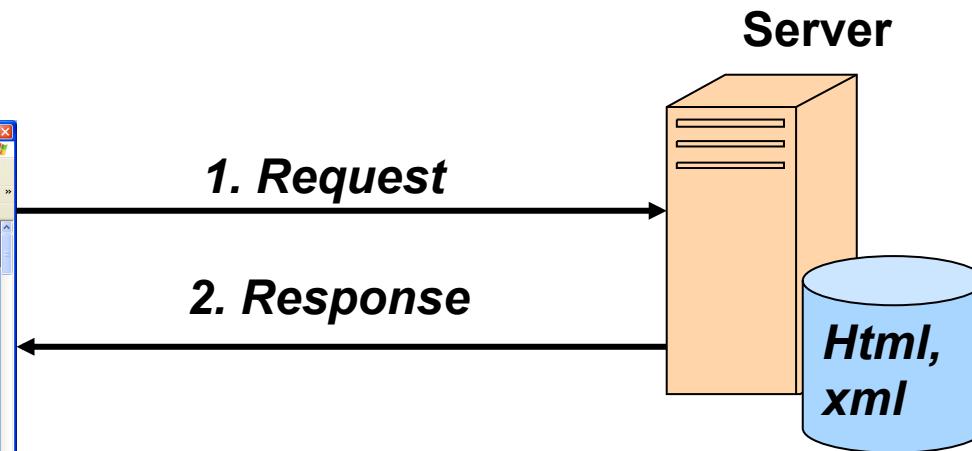
1. Grundlagen der CSS
2. Ausgewählte Eigenschaften der CSS
3. Validierung von CSS
4. Meta- Tags für Suchmaschinen

Statische Webseiten

Client



Server



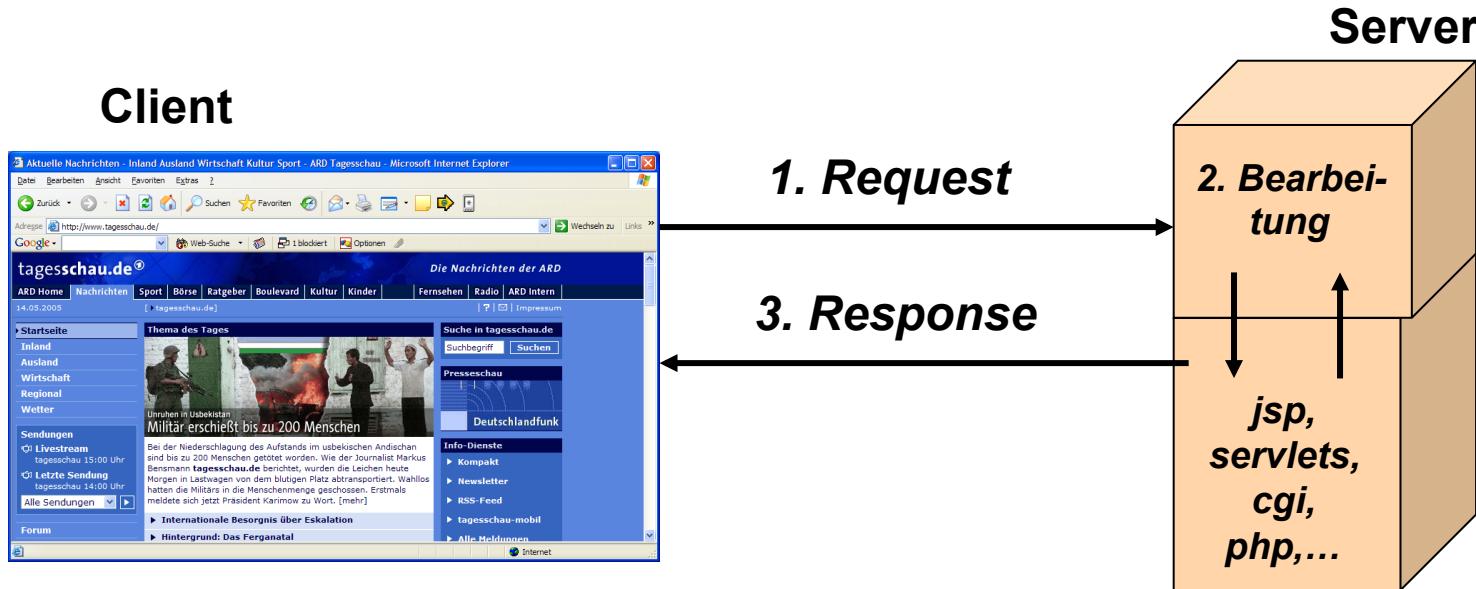
Statische Web-Seiten:

- einfache Vorläufer der Webanwendungen
- einfach
- stabil
- kurze Antwortzeiten
- Verlinkung als grundlegende Eigenschaft
 (` Home `)
- Links repräsentieren Verzeichnisstruktur
- einfache Firmendarstellungen, statische Homepages

Nachteile:

- Unflexibel (nur Vorgedachtes), keine Zusammenstellung
- Schwer änderbar (neues Layout)
- häufig inaktiv
- anfällig für Inkonsistenzen
- Keine Interaktion (E-Commerce Einkaufswagen)

Dynamische Webseiten



Dynamische Web-Seiten:

- Lesende Interaktivität: Eingabebereiche, Radio-Buttons, Auswahllisten
- Dynamische Generierung von Web-Seiten und Verlinkungen
z.B. virtuelle Ausstellungen, etwas komplexere Firmendarstellungen, News-Seiten, Fahrplanauskünfte
- z.T. Generieren des HTML auf **Anfrage mit Skriptsprachen**
 - Entweder im Server oder im Client
 - Mischformen statische und dynamische Seiten (HTML-Einbettung)

Grundlagen des WWW und die Struktur von Webseiten

1. Grundlagen WWW

1. Etwas zur Historie
2. Einfaches Client-Server-Modell
3. Browsen im WWW
4. Uniform Resource Locator (URL)
5. Mime-Type
6. HyperText Transfer Protocol (HTTP)
7. Statische und dynamische Webseiten

2. Scriptsprachen und HTML

3. Struktur von Webseiten

1. Grundlagen der CSS
2. Ausgewählte Eigenschaften der CSS
3. Validierung von CSS
4. Meta- Tags für Suchmaschinen

Scriptsprachen und HTML

- html-codierte Dokumente sind im WWW uneingeschränkt anzeigenfähig
- html-codierte Dokumente sind das Endprodukt der meisten Client/Server
 - Interaktionen
- Dynamisierung des WWW: Einbettung der Scriptsprachen in html oder umgekehrt Einbettung von html in die Scriptsprachen

Scriptsprache	Clientseite	Serverseite
JavaScript	<pre><script type = JavaScript> ...Code... </script> document.write("HTML ");</pre>	
PHP		<pre><? Code ?> <?php Code ?> XML-Stil <% Code %> ASP-Stil Echo "HTML ";</pre>

Clientseitige Einbettungen

- enthalten Programmcode, der im Browser des Clients abläuft

Möglichkeiten der clientseitigen Einbettung:

1. meist in einem oder mehreren `<script type> ... </script>` - Abschnitten eingebettet;
Attribut: `script type = verwendete Scriptsprache` (meist JavaScript, auch VBScript)
2. durch **Browser-Plugins** auch andere Scriptsprachen als JavaScript und VBScript möglich
3. an fast jeder Stelle des html-Codes möglich

Einbettung von html-Code in Strings, welche per Ausgabeanweisung zurück an den Anforderer gegeben werden:

```
document.write("Hello World!")      # JavaScript clientseitig
```

Serverseitige Einbettungen

- enthalten *Programmcode, der vom Webserver zur Ausführung in Auftrag gegeben wird*

Serverseitige Einbettung verschiedener Scriptsprachen:

- ASP : <% ...code... %>
- PHP : <? ...code... ?> oder <?php ...code...?>

Einbettung von html-Code in Strings, welche per Ausgabeanweisung zurück an den Anforderer gegeben werden:

- *print "Hello World!";* # Perl
- *response.write("Hello World!")* # JScript
- *echo "Hello World!"* # Php
- *Response.Write "Hello World!"* # VBScript

Grundlagen des WWW und die Struktur von Webseiten

1. Grundlagen WWW

1. Etwas zur Historie
2. Einfaches Client-Server-Modell
3. Browsen im WWW
4. Uniform Resource Locator (URL)
5. Mime-Type
6. HyperText Transfer Protocol (HTTP)
7. Statische und dynamische Webseiten

2. Scriptsprachen und HTML

3. Struktur von Webseiten

1. **Grundlagen der CSS**
2. Ausgewählte Eigenschaften der CSS
3. Validierung von CSS
4. Meta- Tags für Suchmaschinen

Cascading Stylesheets (CSS) : Einführung

**CSS ist die Standard –Stylesheetsprache für Webseiten.
CSS formatieren und positionieren HTML-Elemente
in zentralen Definitionen und können auch in Skripten
dynamisch verändert werden.**

Vorteile der CSS:

- Trennen Struktur und Inhalt von Webseiten
- Redundanzfreie Formatierung an einer Stelle
- Änderungen der Formatdefinition an nur einer Stelle
- erlauben einheitlichen (unternehmensspezifischen) Stil für umfangreiche Webauftritte
- Bedienen verschiedene Ausgabegeräte mit der gleichen Webseite durch unterschiedliche Stilvorlagen

Cascading Stylesheets (CSS) : Historie und aktuelle Entwicklungen I/II

- Um 1994 von Hakon Wium Lie (geb. in Norwegen) und Bert Bos ab 1995 entwickelt
- Neuerung: Definition von Regeln, die über mehrere Stylesheets hinweg vererbt werden können.
- Übername der Standardisierung durch W3C
- 1996 CSS Level 1 publiziert – Umsetzung in aktuellen Browsern nahezu durchgängig
- 1998 CSS Level 2 Positionierung und unterschiedliche Ausgabemedien
- danach Revision 1 (CSS 2.1) wegen Unzulänglichkeiten in den verschiedenen Browserimplementierungen
- CSS 3 –Entwicklung (u.a. zusätzlich Sprachausgabe)
- Interview mit Hakon Wium Lie (März 2007):
http://www.youtube.com/watch?v=_bmQ7IZiaYo
 - Webstandards sind wichtig über Konkurrenzdenken hinaus
 - Eigene Entwicklungen: CSS1, CSS2, CSS3
 - Unterstützung schon von verschiedenen Modulen aus CSS3 im Opera-Browser
 - Andere Browser wie MSIE, Firefox sind z.T. stärker an technischen Standardisierungen und Implementierungen beteiligt

, „We are competitors, but we want that the WEB will succeed“



Hakon Wium Lie
• heute technischer Leiter des Browserherstellers Opera

Cascading Stylesheets (CSS) : Historie und aktuelle Entwicklungen II/II

- Interview mit Hakon Wium Lie (März 2007):
http://www.youtube.com/watch?v=_bmQ7IZiaYo



CSS Validator des W3C

The screenshot shows a web browser window displaying the results of a CSS validation. The title bar reads "Ergebnisse des CSS-Valid...". The address bar shows the URL "jigsaw.w3.org/css-validator/validator?url=www.fh-bielefeld.de&...". The page header is "Der W3C CSS Validierungsdiest" with the subtitle "Ergebnisse des CSS-Validators http://www.fh-bielefeld.de (CSS level 3)". Below the header, there are links for "Zu: Fehlern (7) Warnungen (4) Ihrem validierten Cascading Style Sheet". The main content area is titled "Ergebnisse des CSS-Validators http://www.fh-bielefeld.de (CSS level 3)" and contains a red box with the message "Entschuldigung! Es wurden die folgenden Fehler gefunden (7)". It lists seven errors from three different CSS files:

URI	Fehler
http://www.fh-bielefeld.de/skin/default/style.css	241 #scroll_left Einlese-Fehler opacity=0
http://www.fh-bielefeld.de/skin/default/style.css	253 #scroll_right Einlese-Fehler opacity=0
http://www.fh-bielefeld.de/skin/default/style.css	470 .selectbox_personal Einlese-Fehler opacity=0
http://www.fh-bielefeld.de/skin/default/style_s.css	115 #scroll_right Einlese-Fehler opacity=0
http://www.fh-bielefeld.de/skin/default/lightbox.css	15 #gallery #g_background Die Eigenschaft zoom existiert nicht : 1
http://www.fh-bielefeld.de/skin/default/lightbox.css	18 #gallery #g_background Einlese-Fehler opacity=70
http://www.fh-bielefeld.de/skin/default/lightbox.css	62 #gallery #gallery_wrapper Ungültige Nummer : height Zu viele Werte oder die Werte werden nicht erkannt : 75% \000009

<http://jigsaw.w3.org/css-validator/>

CSS: Einige Gestaltungshinweise

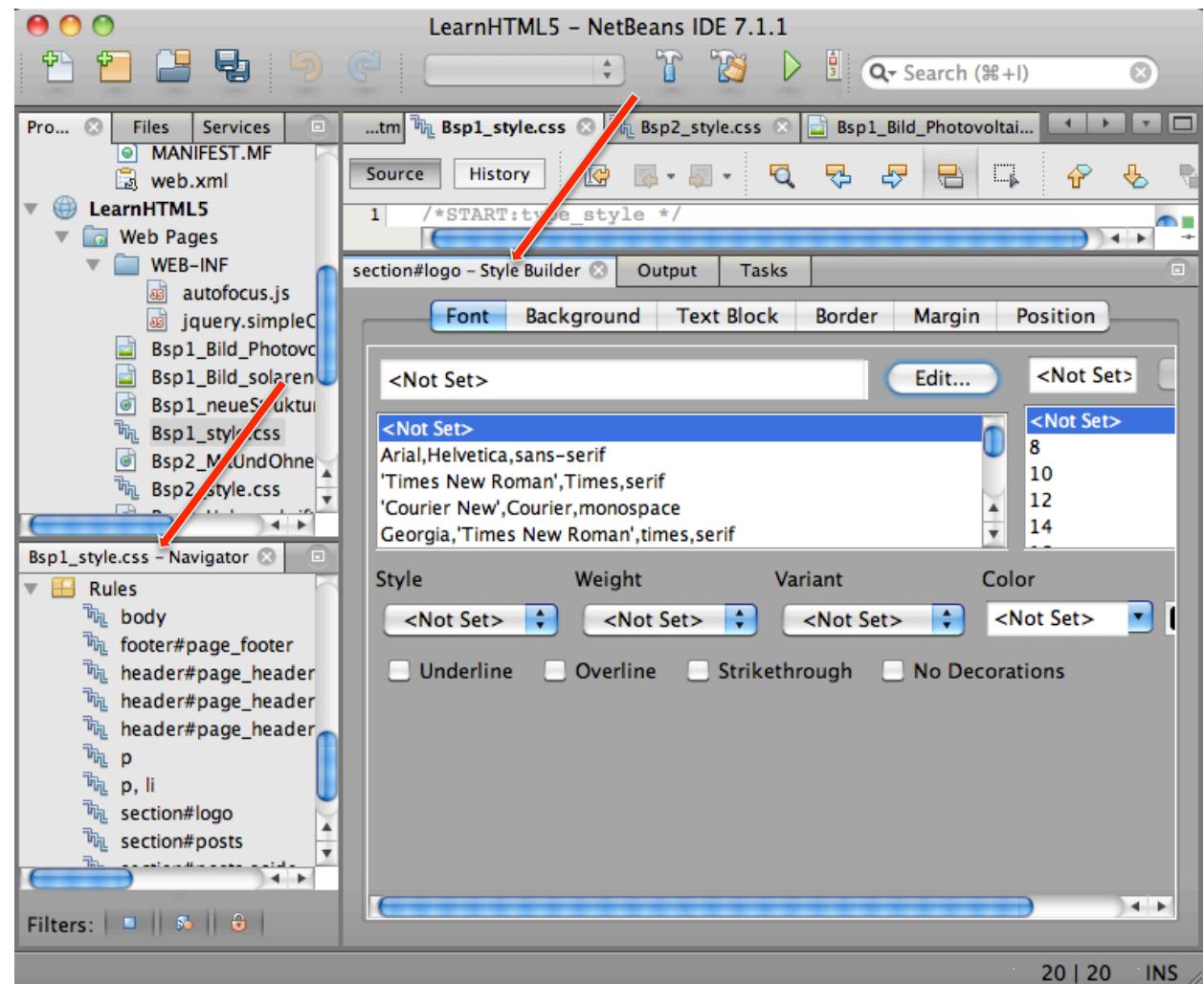
- **Usabilityanforderungen** sollten im Vordergrund stehen
- Die Seiten müssen leserlich sein z.B. in Farbauswahl, Kontrast, Schriftgröße
- Angleichung an spezielle Vorlagen in Farben oder komplettes Design einer Firma
- Browernutzung im Kundenkreis
- Welche Hardware ? (evtl. kleinere Bildschirme auf MiniPC 's beachten)
- Design ist nicht im Mittelpunkt dieser Veranstaltung; sondern die **Befähigung zum Umsetzen eines vorgegebenen Designs** sowie Verständnis für Anforderungen von Gestaltern

CSS3 und HTML5 : Mobile Designprinzipien

- **Vorteile von HTML5 für Design Mobiler Anwendungen**
 - Website auf Mobilem Gerät immer mit dabei (Großer Kommunikationsanteil des Besitzers) -> **Schnittstellen zur Kommunikation** anbieten wie z.B.
 - Telefonanrufe initiieren
 - E-Mails generieren
 - Inhalte in Soziale Netzwerke posten
 - HTML5 nutzt Location-Info über Benutzer – lokationsbasierte passgenaue Dienste anbieten
 - Offlinekapazitäten von HTML5 , toleriert Verbindungsausfälle
 - HTML5 (+ javascript) nutzen Nachladen von Teilen von Seiten
- **Besonderheiten Mobiler Anwendungen**
 - kleinerer Bildschirm
 - Bedienung mit Finger
 - Texteingabe unkomfortabel
 - Unterwegs andere Anforderungen
 - Verbindungsausfälle

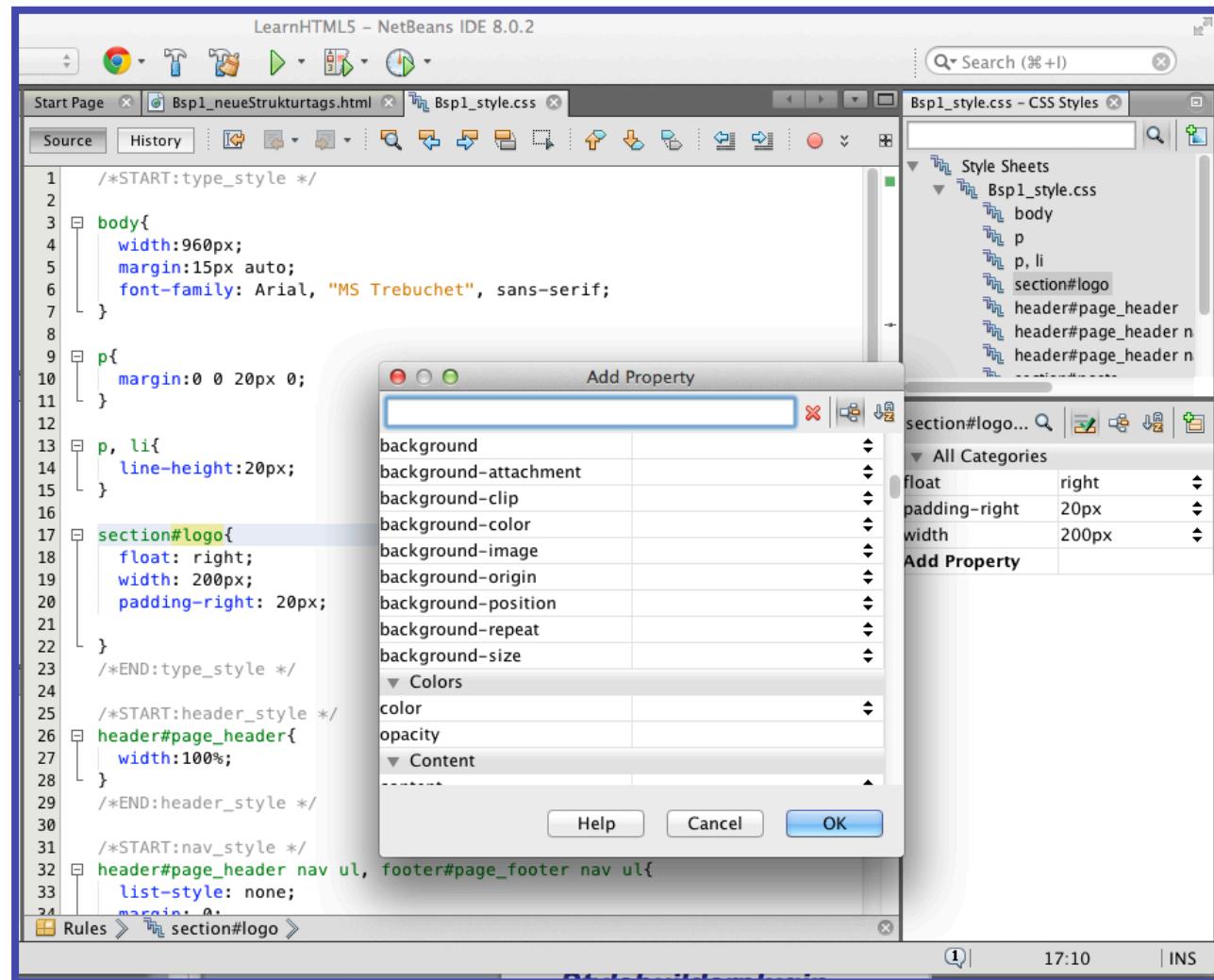
CSS - Erstellungshilfen: z.B. Netbeans IDE 7.1.1

- Entwicklung mit **Stylebuilderplugin, CSS-Navigator** möglich
- Werkzeug für CSS
- Man braucht nur die Struktur und Regeln der Stylesheetsprache CSS zu lernen
- Hilfsmittel für die Attribut-Wertepaare(Font, Background, Text Block, Border, Margin, Position)
- Anderes kostenpflichtiges Werkzeug z.B.
 - Topstyle



CSS - Erstellungshilfen: z.B. Netbeans IDE 8.0.2

- Entwicklung mit **Plugin, CSS-Styles** möglich
- Aufruf über **>Window >Web >CSS Styles**
- Werkzeug für CSS
- Man braucht nur die Struktur und Regeln der Stylesheetsprache CSS zu lernen
- Hilfsmittel für die Attribut-Wertepaare(Font, Background, Text Block, Border, Margin, Position)
- z.B. **>add property**, dann kann man die Werteberechnungen jeweils aussuchen, incl. farbiger Color-Icons, styles....



CSS: Grundbegriffe

- **CSS-Formate** sind die Bestandteile der Stylesheets,
 - können definiert werden:
 - a) in einem zentralen Style-Bereich,
 - b) in einer externen CSS-Datei oder
 - c) direkt im einleitenden Tag eines HTML-Elements
 - bestehen aus **Attributen** mit ihren **Wertzuweisungen**
- **Selektoren** dienen der Auswahl von Gruppen von HTML-Elementen, für die ein Format gelten soll mit folgender Syntax:

Selektor {Attribut:Wert; }

CSS: Einbinden in HTML

1. *Direkt als style-Attribute von HTML-Elementen:*

```
<h1 style="font-style:italic;  
          "text-align:center;">  
    Headline italic centered  
</h1>
```

2. *Im Header einer HTML-Datei* mit dem Style-Tag von dokumentenweiter Gültigkeit:

```
<style type = "text/css">      //Style-Tag mit Mimetype css  
  H1{  
    font-style: italic;        // font-Style  
    text-align: center;       // Textausrichtung  
  }  
</style>
```

3. *mit zentraler .css-Datei*, die mit dem Link-Tag im Header der HTML-Datei eingebunden wird

```
<link rel="stylesheet" type="text/css" //rel - Bezug  
      href="../css/stilvorlage.css"> //href -Hypertext  
                                         //referenz
```

CSS: unterschiedliche Ausgabemedien

Unterschiedliche Ausgabemedien können mit separaten Stylesheets angesprochen werden.

```
<link rel="stylesheet" media="medientyp"  
      href="stylesheet.css">
```

Beispiel:

```
<link rel="stylesheet" media="screen" href="website.css">           //Monitor  
<link rel="stylesheet" media="print, embossed" href="druck.css"> //Drucker +  
                                         //
```

Blindenschrift

```
<link rel="stylesheet" media="aural" href="speaker.css">           // Sprachausgabe
```

Zwischen unterschiedlichen Ausgabemedien kann innerhalb eines Stylebereiches unterschieden werden.

```
@media medientyp { selektor {Eigenschaft.wert; } }
```

Beispiel:

```
<style type="text/css"> @media print {selektor { eigenschaft:wert; } }  
                           @media screen, handheld { selektor { eigenschaft:wert; } } </  
style>
```

CSS: Warum „Cascading“?

Mehrere Stylesheets können zu einem gemeinsamen Stylesheet zusammengefaßt werden. Zuletzt importierte Regeln überschreiben die zuvor gültigen.

Beispiel:

```
<style type="text/css">
@import http://www.beispielwebseite.de./css/stilvorlage.css;
@import "meinestilvorlage.css";
</style>
```

Frage:

Welchen praktischen Vorteil kann das Einbinden mehrerer Stylesheets bringen?

Antwort:

Es kann auf vorhandene Bausteine zurückgegriffen werden.

Selektiv können nachträglich einige Regeln geändert werden.

Grundlagen des WWW und die Struktur von Webseiten

1. Grundlagen WWW

1. Etwas zur Historie
2. Einfaches Client-Server-Modell
3. Browsen im WWW
4. Uniform Resource Locator (URL)
5. Mime-Type
6. HyperText Transfer Protocol (HTTP)
7. Statische und dynamische Webseiten

2. Scriptsprachen und HTML

3. Struktur von Webseiten

1. Grundlagen der CSS
- 2. Ausgewählte Eigenschaften der CSS**
3. Validierung von CSS
4. Meta- Tags für Suchmaschinen

Ausgewählte Eigenschaften der CSS

- Schriftgestaltung
- Farbangaben
- Boxmodell
- Listen
- Hyperlinks

Komplette Angaben fürs Weiterlernen z. B. unter:
<http://de.selfhtml.org/css/index.htm>

Schriftgestaltung mit CSS

Die Schrift wird über die **Font-Attribute** gestaltet, zum Beispiel:
font-family (Schriftart), **font-style** (Schriftstil), **font-size** (Schriftgröße) oder **font-weight** (Schriftgewicht).

Einige Angabemöglichkeiten der Schriftgröße

(Masseinheiten gelten auch für Rahmendicke, Ränder...):

- Punkt z.B. `font-size: 12pt;`
- Pixel z.B. `font-size: 15px;`
- relativ zur Schriftgröße des Elternelements z.B. `font-size: 1.5em;`
- relativ zur Schriftgröße des Anzeigeelements in Prozent `font-size: 105%;`

Beispiele zum Schriftgewicht:

- fett: `font-weight: bold;`
- extrafett: `bolder;`
- dünner: `lighter;`
- normal: `normal;`

Beispiele zum Schriftstil:

- kursiv: `font-style: italic;`
- normal: `normal;`

Beispiele zur Schriftart:

- mehrere: `font-family: 'Times New Roman', Times, serif;`
//Browser prüft der Reihe nach Verfügbarkeit
// letzte Angabe häufig generische Schriftart
- eine: `font-family: Arial;`

Farbangaben mit CSS

Farbe und Hintergrundfarbe von Elementen wird in den Attributen

- **color** bzw. **background** festgelegt. Die Eigenschaften können durch festgelegte **Namen** oder **Werte für die RGB-Komponenten in dezimaler oder hexadezimaler Schreibweise angegeben werden.**

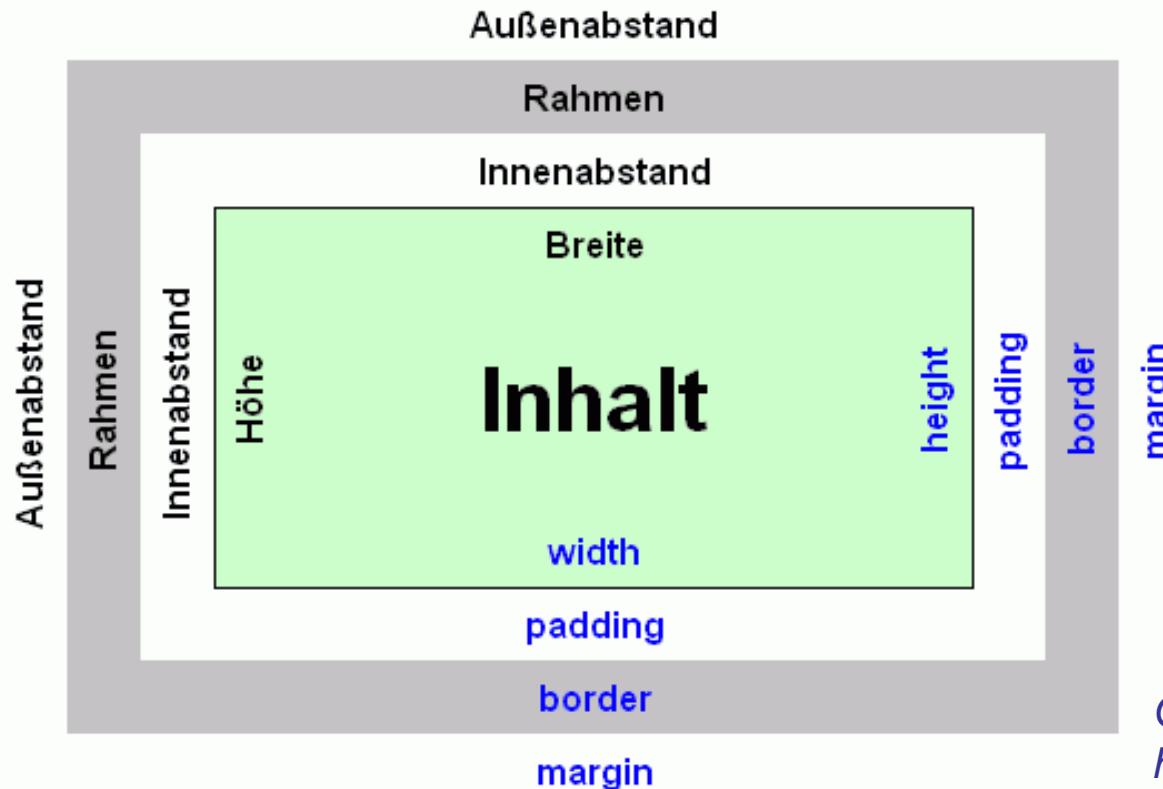
Beispiel für Farangaben:

```
background: #eeeeee;  
color: rgb (64,64,64);
```



Das Boxmodell in CSS

HTML-Auszeichnungen, die einen Block oder Absatz bilden (z.B. <h1>, <body>, <div>, <table> oder <p>), werden in eine Box eingebettet. Die Box wird umgeben von einer Polsterung (padding), dem Rahmen (border) und dem Rand (margin). Die Gesamtausmaße der Box ergeben sich immer aus einer Addition dieser Größen.



Quelle:
<http://de.selfhtml.org>

Das Boxmodell : komplettes Beispiel

```
<html>
<head>
<style type="text/css">
body { /*Body-box*/
    color: blue; background-color: white;
    font-size: 105%; font-family: Arial; font-weight: bold;
    margin: 1em; padding: 1em;
    min-width: 31em; /*Mindestbreite */
    border-width: 2px; border-style: solid; border-color: blue;
}
h1 { /*Überschriftenbox */
    color: rgb(64, 64, 64); background-color: #eeeeee;
    font-size: 1.5em;
    margin: 0 0 0.7em 0; padding: 0.3em;
    text-align: center;
    border-width: 2px; border-style: solid; border-color: rgb(64, 64, 64);
}
</style>
</head>
<body>
<h1>Box der Überschrift</h1>
Box des Bodies
</body>
</html>
```



Listen gestalten mit CSS

Stylesheet-Angaben für Listen beziehen sich auf die HTML-Elemente **ul** und **ol**, sie ermöglichen die Gestaltung des Darstellungsstyps (**list-style-type**), der Listeneinrückung. (**list-style-position**) und der verwendeten Bullet-Grafik (**list-style-image**).
(die in HTML möglichen Attribute sind ab HTML 4.0 deprecated – unerwünscht.)

list-style-type z.B. für **ol** *decimal* (1,2,3,...); *lower roman* (i, ii, iii, iv, ...);
für **ul** *disc*, *circle*, *square*, *none*

list-style-position *inside* = eingerückt;
 outside = ausgerückt,

list-style-image mit *list-style-image:url([Dateiname])*

Listen gestalten mit CSS – ein komplettes Beispiel

```
<html>
<head>
<title>Listendarstellung in CSS</title>
<style type="text/css">
ol#Listenbeispiel li {
    list-style: lower-alpha;
}
</style>
</head>
<body>
    <ol id=Listenbeispiel>
        <li>ein Listeneintrag</li>
        <li> noch ein Listeneintrag </li>
    </ol>
</body>
</html>
```



Hyperlinks mit CSS

Die `<a>`-Tags können über die Selektoren der **Pseudoelemente** abhängig vom jeweiligen Zustand formatiert werden.

<code>a: link</code>	–	Zustand eines Links bei noch nicht besuchter Seite
<code>a: visited</code>	–	bereits besuchte Seite
<code>a: hover</code>	–	Zustand eines Links bei Mausberührung
<code>a: active</code>	–	gerade besuchter Link
<code>a: focus</code>	–	Link erhält Focus

```
<html>
<head>
<title>Hyperlinks mit Pseudoelementen im CSS</title>
<style type="text/css">
  #Linkbeispiel a {display: block;padding: 0.2em;font-weight:
bold;}
  #Linkbeispiel a:link { color: black;}
  #Linkbeispiel a:visited {color: #666;}
  #Linkbeispiel a:hover {color: black; background-color:
white;}
  #Linkbeispiel a:active {color: white; background-color:
gray;}
</style>
</head>
<body>
  <div id="Linkbeispiel">
    <a href="http://www.ard.de">Link 1</a>
    <a href="http://www.link2.de">Link 2</a>
    <a href="http://www.link3.de">Link 3</a>
  </div>
</body>
</html>
```



Grundlagen des WWW und die Struktur von Webseiten

1. Grundlagen WWW

1. Etwas zur Historie
2. Einfaches Client-Server-Modell
3. Browsen im WWW
4. Uniform Resource Locator (URL)
5. Mime-Type
6. HyperText Transfer Protocol (HTTP)
7. Statische und dynamische Webseiten

2. Scriptsprachen und HTML

3. Struktur von Webseiten

1. Grundlagen der CSS
2. Ausgewählte Eigenschaften der CSS
3. **Validierung von CSS**
4. Meta- Tags für Suchmaschinen

Validierung von CSS

Zur Verbesserung der Konformität und Zuverlässigkeit des Internets bietet das W3C Validierungsservices für Stylesheets an:
<http://validator.w3c.org>.

- Herunterladen der Tools Markup-Validator oder CSS-Validator oder Online – Nutzung.
- Checken aller Standardvorschriften des W3C
- Fehlerliste als negatives Ergebnis oder **Zertifikat** als positives Ergebnis
- **Auszeichnung der Seite als fehlerfrei und W3C-konform**
- **aber : kein Schutz vor Browserinkompatibilitäten**

Validierung von CSS (2012)

Ergebnisse des CSS-Validators <http://www.fh-bielefeld.de> (CSS level 2.1)

Entschuldigung! Es wurden die folgenden Fehler gefunden (21)

Zu:	Fehler
104 #archiv_search	Die Eigenschaft -moz-border-radius existiert nicht : 5px
105 #archiv_search	Die Eigenschaft border-radius existiert nicht in CSS level 2.1, aber in [css3] : 5px
184 #navigation_bg	Ungültige Nummer : background-color rgba(255,255,255,0.65) ist kein background-color-Wert : rgba(255,255,255,0.65)
207 #nav_scroll	Ungültige Nummer : background-color rgba(255,255,255,0.65) ist kein background-color-Wert : rgba(255,255,255,0.65)

Onlinenutzung:
<http://jigsaw.w3.org/css-validator/>

Validierung von CSS (2014)

The screenshot shows a web browser window displaying the W3C Markup Validation Service. The address bar shows the URL <http://validator.w3.org/check?uri=http%3A%2F%2Fwww.fh-bielefeld.de&charset=%28detect+automatically%29&doctype=Inline&group=0#jumpbar>. The main content area shows the validation interface with the following details:

Result:	1 Error, 1 warning(s)
Address :	<input type="text" value="http://www.fh-bielefeld.de/"/>
Encoding :	utf-8 <input type="button" value="(detect automatically)"/>
Doctype :	HTML5 <input type="button" value="(detect automatically)"/>
Root Element:	html

The page also features a banner from HP and links for "Options" and "4040 Flattr".

Onlinenutzung:<http://validator.w3.org/check?uri=http%3A%2F%2Fwww.fh-bielefeld.de&charset=%28detect+automatically%29&doctype=Inline&group=0#jumpbar>

Grundlagen des WWW und die Struktur von Webseiten

1. Grundlagen WWW

1. Etwas zur Historie
2. Einfaches Client-Server-Modell
3. Browsen im WWW
4. Uniform Resource Locator (URL)
5. Mime-Type
6. HyperText Transfer Protocol (HTTP)
7. Statische und dynamische Webseiten

2. Scriptsprachen und HTML

3. Struktur von Webseiten

1. Grundlagen der CSS
2. Ausgewählte Eigenschaften der CSS
3. Validierung von CSS
- 4. Meta- Tags für Suchmaschinen**

Meta – Tags für Suchmaschinen

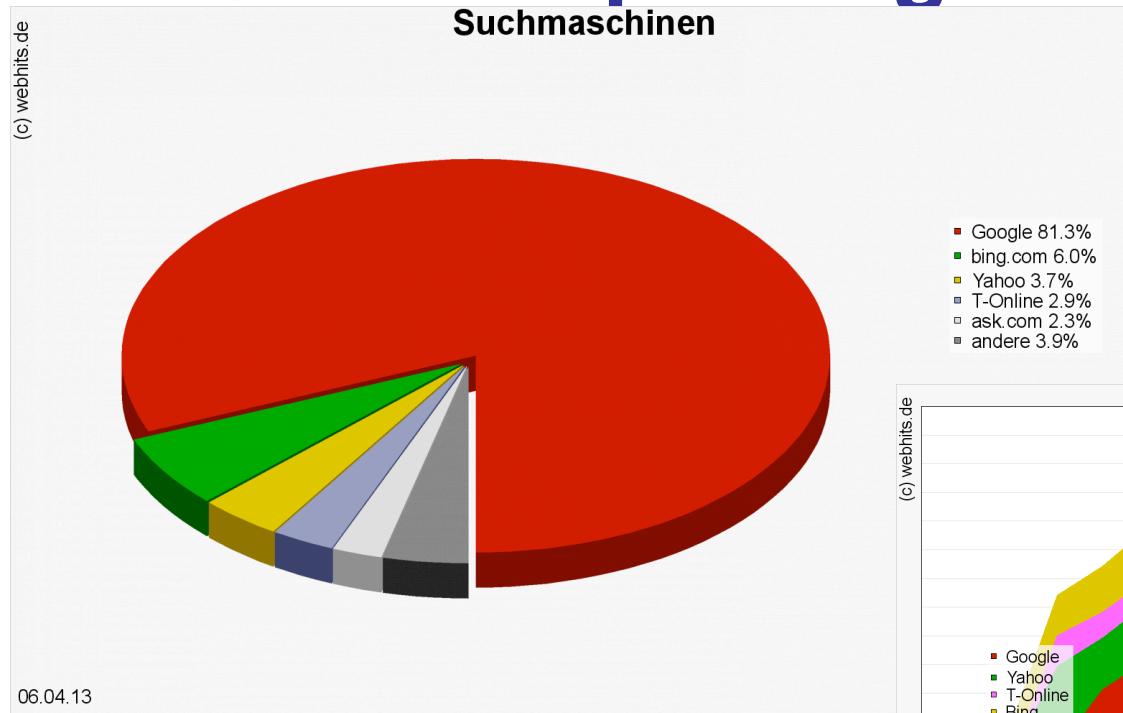
*Das INTERNET bietet eine Fülle nichtstrukturierter Daten.
Zum Auffinden der gewünschten Informationen werden
Suchmaschinen genutzt.*

Suchmaschinen:

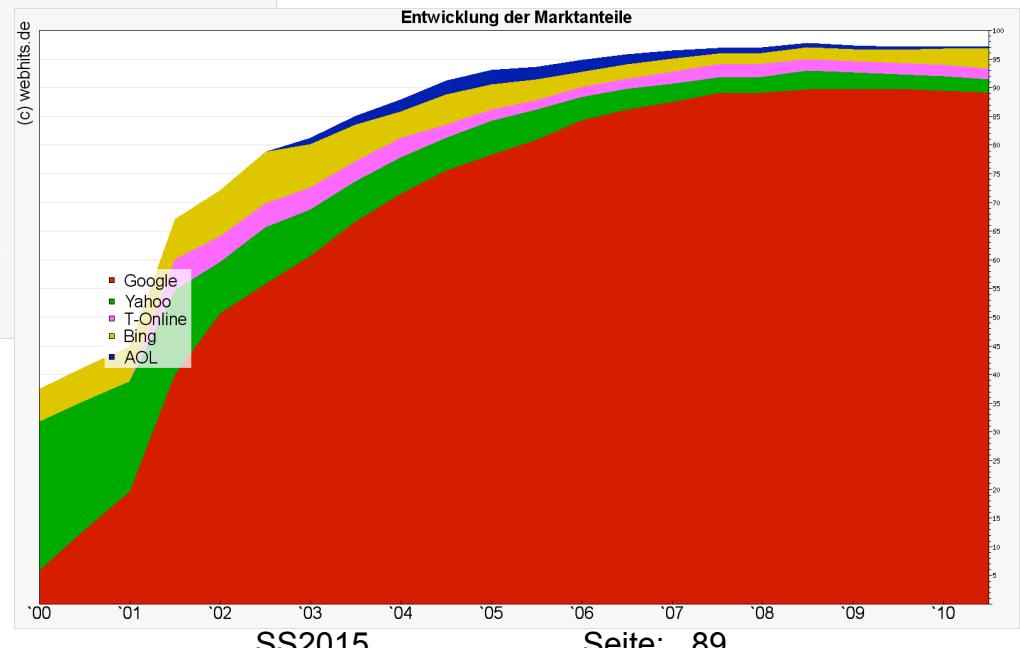
- halten in sehr großen Datenbanken Teile des Internets vor
- Robots (auch Web-Spider, Web-Crawler genannt) liefern die Informationen
- werten Textinhalte, Title und Metatags im Header aus
- bekannte Beispiele : www.google.de, www.webcrawler.com, www.altavista.com

allg. Syntax für Meta-Tags: <**meta name=“Name“ content=“Wert“**>

Nutzungsverteilung der Suchmaschinen im deutschsprachigen Raum (2014)



Quelle: www.webhits.de



Meta – Tags für Suchmaschinen

Beispiele für häufig genutzte Meta-Tags:

- Suchmaschinen werten meist die Angaben zu den Attributen **keyword** und **description** aus:
z.B: `<meta name="keywords" content="HTML, XML, CSS" >`
`<meta name="description" content="Grundlagen des WWW und die Struktur von Webseiten" >`
- Verbot für die Suchmaschine, Inhalte aus der Seite zu entnehmen:
`<meta name="robots" content="noindex" >`
- Suchmaschine sollte nach 14 Tagen die Informationen über den Inhalt der Seite neu auffrischen:
`<meta name="revisit-after" content="14" >`
- Lokationsinformationen für Geosuchmaschinen:
`<meta name="geo.position" content="52.184878;10.107132"/>`
`<meta name="geo.placename" content="„Neuer Weg 1a, 34421 Webhausen"/>`

Meta – Tags für Suchmaschinen

Meta-Tags kann man sich mit Hilfe von Tools (oft frei verfügbar) automatisch erstellen lassen.

Beispiel eines freien Online-Tools zur Generierung der Meta-Tags zur Geolokation:

The screenshot shows a web browser window titled "Geo Meta-Tags generieren | KartenGenerator". The address bar displays the URL "www.kartengenerator.com/geo-met...". The main content area is titled "Karten Generator" and contains text about the importance of geo-information for search engines. It provides instructions on how to embed geo-meta-tags into an HTML page by entering an address and generating the tags. A large blue button labeled "Meta-Tags generieren" is prominent. Below it, a section titled "Kopieren Sie die folgenden Zeilen in den <head> Ihrer HTML-Seite:" contains the generated meta tags:

```
<meta name="ICBM" content="52.2970416, 8.903795700000046" />
<meta name="geo.position" content="52.2970416;8.903795700000046" />
<meta name="geo.placename" content="Ringstraße 94, 32427 Minden" />
```

<http://www.kartengenerator.com/geo-meta.html>

Zusammenfassung

- Grundlegende Mechanismen, Protokolle und Standards des WWW kennengelernt
- Historie (W3 Consortium, Browserkriege, WEB2.0 WEB3.0)
- Unterschied zwischen dynamischen und statischen Webseiten
- Möglichkeiten der Einbindung von Scripten in Webseiten
- Trennung von Inhalt und Struktur durch CSS-Technologie
- Grundlegende Kenntnisse in der Anwendung von CSS
- Unterstützung von Suchmaschinen im WWW

Literatur für WWW, HTTP, HTML, CSS

- Wolfgang Dehnhardt: „Scriptssprachen für dynamische Webauftritte“, Carl Hanser Verlag München Wien, 2001
- <http://selfhtml.org> (HTML-Tutorial, CSS-Tutorial)
- <http://www.w3c.org>
- JavaScript – Eine Einführung, RRZN Hannover, Skript
- Tim O'Reilly: „What ist Web 2.0“ – Design Patterns and Business Models for the Next Generation of Software
<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>
- IXSpecial: „Web 2.0 – das Kompendium“ Januar 2007, Heise Zeitschriften Verlag GmbH & Co.KG, Hannover
- http://de.wikipedia.org/wiki/Cascading_Style_Sheets

Ausblick

- Clientseitige Implementierungstechnologien:
Javascript

Appendix : HTML/HTML5-Grundlagen

Inhalt:

- 1. Motivation**
- 2. CSS (Einbinden, Vorteile)**
- 3. Deklaration (Doctype)**
- 4. Überschriften und Abschnitte**
- 5. Listen und Hyperlinks**
- 6. Bilder**
- 7. Strukturtags von HTML5**
- 8. Tabellen**
- 9. Formulare**
- 10. Definitionslisten**
- 11. Hyperlinks auf andere Dokumenttypen**

Tabellen

- Tag `<table>...</table>` - leitet Tabelle ein
- Tag `<tr>...</tr>` - legt Zeilen innerhalb der Tabelle fest
- Tag `<td>...</td>` - legt Zellen innerhalb der Zeile fest
- Attribut `border="integer"` – legt Rahmenbreite in Pixeln fest
- Attribut `colspan/rowspan = "integer"` – legt Anzahl der verbundenen Zellen fest entlang der Spalte/Zeile.

The screenshot shows a web browser window titled "Tabelle". The address bar displays the URL `http://localhost:8080/LearnHT`. The page content on the left shows a table with a single row containing two cells. The first cell is labeled "Kopfzeile" and the second cell is divided into two smaller cells, each labeled "zelle links" and "zelle rechts". On the right side of the browser, the source code is displayed:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Tabelle</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    <table border="1">
      <tr> <td colspan="2"> Kopfzeile </td> </tr>
      <tr> <td> Zelle links </td> <td> Zelle rechts </td> </tr>
    </table>
  </body>
</html>
```

Formulare

Formulardefinition:

```
<form name="Formularname" action=mailto:irgendwer@webmail.de  
method="POST|GET" enctype="text/plain" target="Ziel">  
...Formularelemente...  
</form>
```

- Tag <form> #engl. Formular – leitet ein Formular ein. In ihm können Schaltflächen, Eingabefelder und Auswahllisten definiert werden; Verschachtelung des Tags <form> ist nicht möglich.
- Attribut name – legt Formularname fest; kann z.B. von Javascript aufgerufen werden
- Attribut target – legt das Zielfenster für Rückmeldung des Skriptes fest
- Attribut method bestimmt Versendemethode (Post/Get)
- Attribut action – gibt Programm an, das nach Abschicken der Daten ausgeführt wird und zu dem die Daten gesendet werden
- Attribut enctype – MimETYPE, z.B. "text/plain"

Formularelemente

- Tag <input> #engl. Eingabe – leitet alle Formulareingaben ein
- Attribut type –legt den Attributtyp fest, mögliche Werte sind:
 - text – einzeiliges Eingabefeld
 - password - einzeiliges Eingabefeld mit nicht sichtbaren Zeichen
 - checkbox - Kontrollfeld, das markiert werden kann
 - radio - Optionsfelder; nur eins pro Gruppe kann markiert werden
 - image - Schaltfläche mit hinterlegter Grafik
 - reset - Schaltfläche zum Zurücksetzen der Formulardaten
 - submit - Schaltfläche zum Versenden der Daten

```
<form>
  <table border=1>
    <tr><td>Anzahl</td>
    <td><input type="text" name="Anzahl" Value=1kg></td>
    <tr> <td>Produkt</td>
    <td><select multiple Name="Item" Size = "3">
      <option>&Auml;pfel</option>
      <option>Birnen</option>
      <option>Quitten</option>
    </select>
    </td></tr>
    <tr>
      <td><input type="submit" value="Bestellen"></td>
      <td><input type="reset" value="L&ouml;schen"></td>
    </tr>
  </table>
</form>
```

Formularelemente

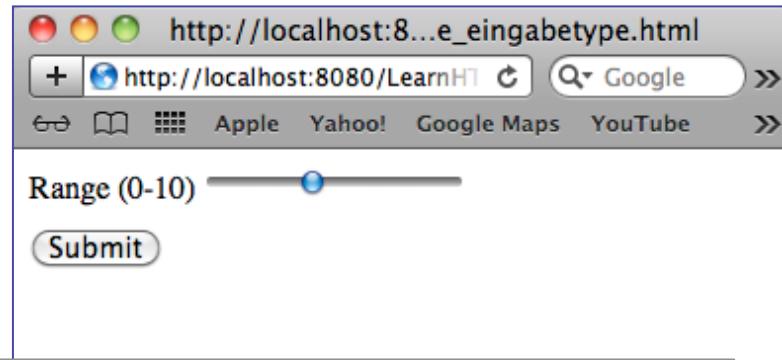
The screenshot shows a web browser window with the URL http://localhost:8080/LearnHTML5/Bsp8_formular.html?Anzahl=1kg&Item=Birnen. The page displays a form with a table structure. The first row has two cells: 'Anzahl' with value '1kg' and 'Produkt' with a dropdown menu containing 'Äpfel', 'Birnen', and 'Quitten'. The second row contains two buttons: 'Bestellen' and 'Löschen'.

Anzahl	1kg
Produkt	<ul style="list-style-type: none">ÄpfelBirnenQuitten
<input type="button" value="Bestellen"/>	<input type="button" value="Löschen"/>

```
<form>
  <table border=1>
    <tr><td>Anzahl</td>
    <td><input type="text" name="Anzahl" Value=1kg></td>
    <tr> <td>Produkt</td>
    <td><select multiple Name="Item" Size = "3">
      <option>Äpfel</option>
      <option>Birnen</option>
      <option>Quitten</option>
    </select>
    </td></tr>
    <tr>
      <td><input type="submit" value="Bestellen"></td>
      <td><input type="reset" value="Löschen"></td>
    </tr>
  </table>
</form>
```

(HTML5)Slider mithilfe von input-type : range

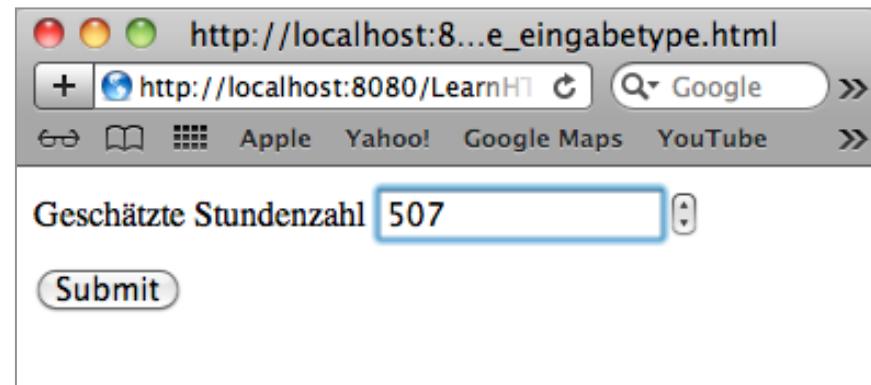
- Tag <range> #engl. Wertebereich – dient der Eingabe mithilfe eines *Schiebereglers*.
- Input type –dient der bequemen Eingabe von numerischen Werten



```
<form method="post" action="">
<ol>
    <li>
        <label for="slider">Range (0-10) </label>
        <input type="range" min="0" max="10" name="slider" value="0"
               id="slider">
    </li>
    <li>
        <input type="submit" value="Submit">
    </li>
</ol>
</form>
```

(HTML5)Spinboxen für Zahlen mithilfe von input-type : number

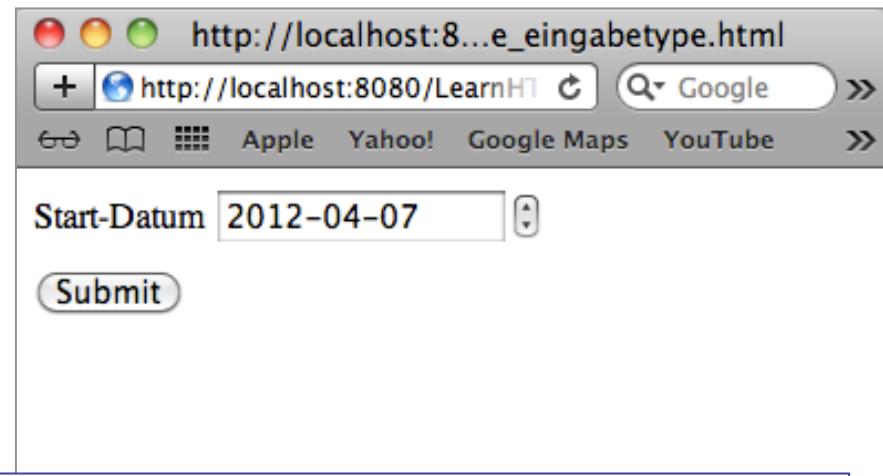
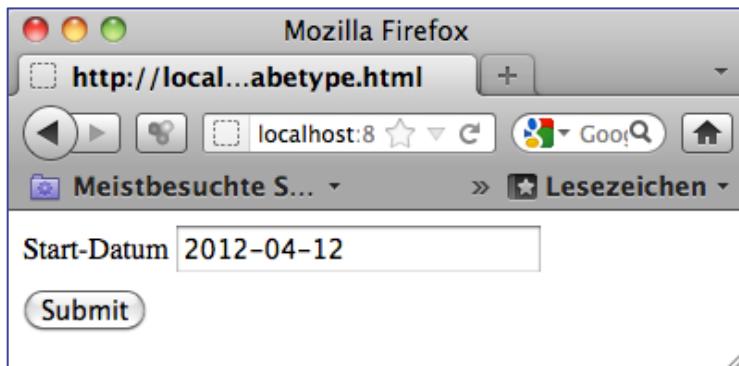
- Tag <number> #engl. Zahl – dient der Eingabe mithilfe einer *Spinbox* mit Pfeilen zur Einstellung der einzugebenden Zahl.
- Input type „number“ –dient der bequemen Eingabe von numerischen Werten
- es können auch direkte Eingaben in das Feld geschrieben werden, dann gelten min- und max-Werte nicht mehr.



```
<li>
    <label for="Stundenzahl">Geschätzte Stundenzahl </label>
    <input type="number" min="0" max="1000" name="Stundenzahl"
           value="500" id="Stundenzahl">
</li>
<li>
    <input type="submit" value="Submit">
</li>
```

(HTML5) Datumswerte eingeben mit: **date**

- Input Type = „**date**“ #engl. Datum – dient der Erstellung eines Kalenderpickers
- bequeme Eingabe von Datumswerten
- weder Mozilla Firefox, noch Safari schaffen aktuell den Kalenderpicker. ;-(, von Opera wird behauptet, dass er konform ist



```
<li>
    <label for=„start-datum“>Start-Datum </label>
    <input type=“date” name=“start-datum” id=“start-datum
value=2015-04-12“>
</li>
<li>
    <input type=“submit” value=“Submit”>
</li>
```

HTML5: weitere neue Formularelemente

Input Typs:

<input type="url"> - für bequeme Eingabe Tastaturlayout für URL (z.B. beim Iphone)
<input type="email"> - Email-Eingabe
<input type ="color"> - Eingabe von Farbcodes

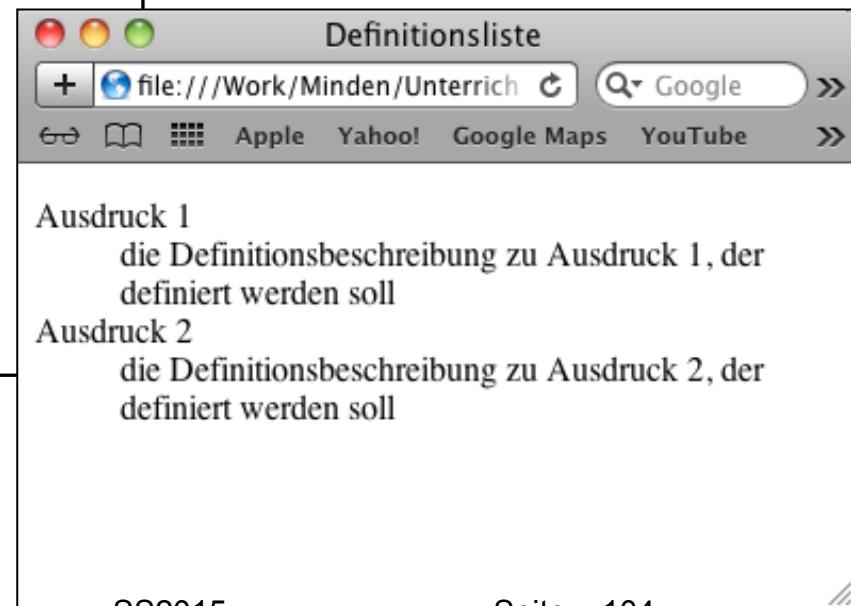
Attribut:

autofocus – Platzieren des Mauszeigers in einem Formularfeld

Definitionslisten

- Tag `<dl> ...</dl>` # definition list - Definitionsliste
- Paare der Tags:
 - `<dt> ...</dt>` # definition term – zu definierender Ausdruck
 - `<dd> ...</dd>` # definition description – erläuternder Text

```
<html>
<body>
    <dl>
        <dt>Ausdruck</dt>
        <dd>die Definitionsbeschreibung zu
            dem Ausdruck, der definiert
            werden soll</dd>
        <dt>Ausdruck 2</dt>
        <dd>die Definitionsbeschreibung
            zu Ausdruck 2, der definiert
            werden soll</dd>
    </dl>
</body>
</html>
```



Hyperlinks auf andere Dokumenttypen

Dokumenttyp	Erläuterung
Textdatei (*.txt) <code></code>	Textdatei wird im Browser als einfacher Text angezeigt
Grafik (*.jpg, *.gif) <code></code>	Grafik wird in den Browser geladen und angezeigt
Video (*.mpg, *.avi) <code></code>	Ist ein Programm zum Abspielen von Videodateien vorhanden, wird es zum Abspielen genutzt
Musik (*.mp3, *.wav) <code></code>	Falls Programm vorhanden, wird Musikdatei damit abgespielt
Programm, Datendatei (*.zip, *.gz, *.exe) <code></code>	Download von Programmen und zip-Dateien aus dem Internet
Andere Formate (z.B. *.pdf, *.doc, *.docx, *.xls, *.xlsx) <code></code>	Öffnen der Datei direkt im Browserfenster oder mit entsprechendem Programm auf dem Rechner