

# Vorlesung Werkzeuge der Informatik

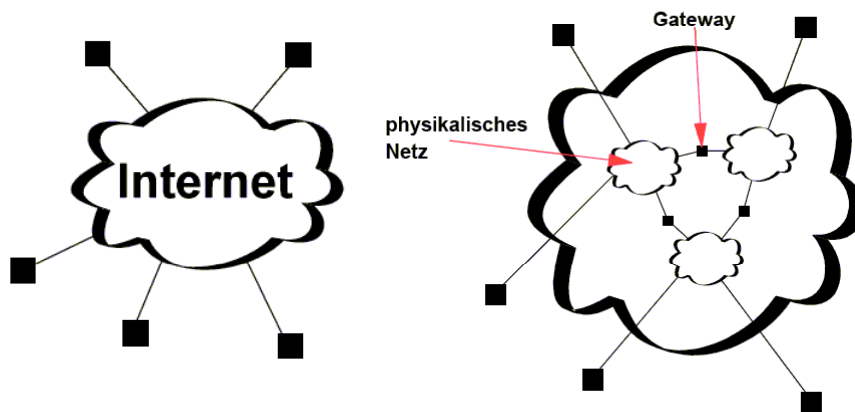
## Grundlagen und Werkzeuge des WWW

Jörg P. Müller

## Inhalt

- **Entwicklung von Internet und WWW**
- **Darstellung von WWW-Inhalten**
  - HTML als Web Content Sprache
  - Cascading Style Sheets (CSS)
  - XML Basics
  - Ausblick: XML Schema
- **WWW-Architektur und Protokolle**
  - WWW-Architektur (Client-Server)
  - Web Ressourcen (oder: Was ist eine URL)
  - Das HTTP-Protokoll

## Das Internet : Zusammenschluss vieler Teil-Netze



## Begriffe: Internet - Intranet - Extranet

- **Internet**
  - weltweites Netzwerk auf Basis der TCP/IP-Protokollfamilie (s.u.) mit mehreren Millionen Rechnern, für „jedermann“ offen
- **Intranet**
  - Kommunikationsnetz auf Basis von Internet-Technologien, das dem Informationsaustausch innerhalb einer begrenzten Interessengemeinschaft dient
  - z.B. Mitarbeiterportale großer Unternehmen
- **Extranet**
  - Variante des Intranet, bei dem Informationsflüsse aus dem Internet und dem Intranet verknüpft sind
  - z.B. Unternehmen erlaubt Geschäftspartnern Zugriff zu Teilen des eigenen Intranets

## Geschichte des Internet

- **Produkt des Kalten Krieges in den 60er Jahren**
- **Überlegungen US-Department of Defense (DoD): auch nach Atom-Angriffen soll Netz funktionieren**  
-> erhebliche finanzielle Unterstützung
- **„Advanced Research Project Agency“ ==> ARPANET**
- **1964 Vorstellung der Paketvermittlung durch Paul Barran**

## Paketorientierte Datenübertragung im Internet

- **Robuste Übertragung von Nachrichten zwischen Sender und Empfänger**
  - Aufteilen der Nachricht in Datenpakete
  - Senden der einzelnen Datenpakete durch das Internet
  - Rekonstruktion der Nachricht aus den einzelnen Datenpaketen
- **Verwendung der TCP/IP Protokollfamilie**
- **TCP: Transmissing Control Protocol**
  - Sender und Empfänger durch (IP-Adresse, Port) beschrieben
  - Verbindungsorientiert und zuverlässig (alle Daten kommen genau einmal und in richtiger Reihenfolge an)
  - Handhabt Zerlegung und Rekonstruktion der Nachrichten
- **Darunter liegend: Internet Protocol (IP)**
  - Definition und Übertragung (Routing) von Datenpaketen durch das Internet

## **Geschichte des Internet**

- **1969 Vernetzung von 4 Universitäten**
- **1971 Vernetzung von 13 Universitäten**
- **1972 Vernetzung von 37 Universitäten**
- **1973 Start des Internet durch Verbindung verschiedener paketorientierter Netze**
- **1978 Beschluss, bei staatlichen Datenübertragungen nur noch TCP/IP einzusetzen**
- **1983 Aufspaltung in Militär und Bildungsnetz**
  - **Militärischer Teil wird abgelöst: Defense ARPA (DARPA); später auch “Milnet”**
- **Kommerzialisierung ab 1989**

## **Internetdienste**

- **Internet bietet Infrastruktur, auf deren Basis für den Anwender nutzbare Dienste zur Verfügung stehen**
- **Beispiele:**
  - **World Wide Web: Zugriff auf Webseiten**
  - **File Transfer Service. Übertragung von Dateien (FTP, File Transfer Protocol)**
  - **Email Service (SMTP, Simple Mail Transfer Protocol)**
  - **Foren, Newsgroup (Usenet)**
  - **Internet Relay Chat**
  - **Instant Messaging**
  - **Internet-Telefonie**
- **Tendenz: Verschmelzen von Internet-Diensten mit dem WWW**

## Das WWW

- Ein Dienst basierend auf dem Internet
- **Globaler digitaler Informationsraum bestehend aus Millionen Clients und Servern, die auf verknüpfte Informationsobjekte zugreifen**
  - Server verwalten die Web Ressourcen
  - Clients geben Benutzern eine einfache Schnittstelle für Ressourcendarstellung und –zugriff (über Web Browser – Applikation)
- **Web-Ressourcen : z.B. Texte, Dokumente, Bilder, Multimediadaten, Datenbankinhalte, ausführbare Programme sein**
- **Informationsobjekte sind identifiziert durch kurze, eindeutige Schlüssel, sogenannte Uniform Resource Identifiers (URIs)**
- **Zugriff auf Web Ressourcen über Hyperlinks auf der Basis der URIs**
- **Das WWW unterstützt ein einheitliches Protokoll zur Kommunikation zwischen einem WWW Server und einem WWW Client (HTTP)**
- **Sprache zur Beschreibung von WWW-Inhalten:**
  - Hypertext Markup Language (HTML)

## Geschichte des WWW

<http://www.w3.org/History.html>

- **Anfänge des WWW → Geschichte des Internet**
- **1980: Tim Berners-Lee (CERN) schreibt Programm "ENQUIRE", das es erlaubt, Knoten im Internet zu verlinken**
- **1989: Tim Berners-Lee: CERN-Internes Proposal "Hypertext and CERN"**
- **1990: TBL prägt Begriff "World Wide Web" – Beginn eines großen Hypertext-Projekts bei CERN**
- **April 1993: CERN kündigt freie Nutzbarkeit des WWW an**
- **September 1993: Mosaic Browser (NCSA) verfügbar für X, PC/Windows and Macintosh.**
- **Mai 1994: Erste internationale WWW Konferenz**
- **Oktober 1994: Gründung des World Wide Web Consortium**

## Geschichte des WWW (2)

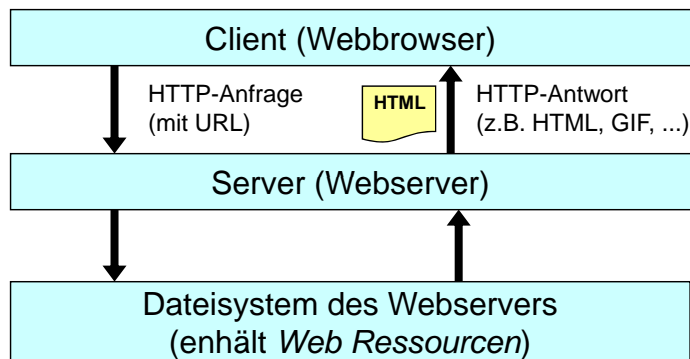
- **1993: Erste Web Search Engine (Wandex, MIT)**
- **1995: Sun bringt Java Programmiersprache heraus mit Unterstützung für WWW; wenig später kündigen Netscape und Microsoft an, dass ihre Browser Java unterstützen werden**
- **Google**
  - 1996 beginnt als Forschungsprojekt
  - 1998 Firmengründung
- **seit ca. 1998:**
  - Verfügbarkeit sicherer Kommunikations-protokolle für das WWW (https)
  - Entstehen der ersten webbasierten Electronic Commerce Systeme (eShops)
- **1999: Tim Berners-Lee prägt Vision des "Semantic Web"**
- **2004: Medienunternehmen O'Reilly Media prägt den Begriff des "Web 2.0"**

## Webbasierte Anwendungen

### Grundsätzlicher Aufbau webbasierter Anwendungen

Basiert auf einer Client-Server-Architektur

(Remote) Presentation



## Identifikation von Web-Ressourcen: Uniform Resource Identifier (URI)

- Zeichenfolge, die zur Identifikation einer Ressource dient
- Allgemeiner Aufbau:
  - <Schema>:<Schemaspezifischer Teil>
- <Schema> gibt Typ der URI an, z.B. http, ftp, mailto
- <Schema> legt Interpretation des Schemaspezifischen Teils fest
- URI-Schemata wie ftp und http sind hierarchisch aufgebaut:
 

```
<Schema>://[<Benutzer>[:<Passwort>]@]<Server>[:<Port>]/[<Pfad>
[?<Anfrage>][#<Fragment>]
```
- Zwei Formen von URIs
  - Uniform Resource Locators (URLs): Identifikation der Ressource durch Beschreibung des Zugriffs (positionsabhängige Referenz) auf die Ressource
  - Uniform Resource Names (URNs): Logische Identifikation i. S. einer global eindeutigen, positionsabhängigen und persistenten Referenz

## URL: Häufig verwendete Strukturen

- Verwendung nur eines DNS-Namens

Schema	Host-Name	Pfadname
--------	-----------	----------

http       ://    www.in.tu-clausthal.de    /home/mueller/mbox

- Kombination DNS-Name mit Portnummer

Schema	Host-Name	Schema	Pfadname
--------	-----------	--------	----------

http       ://    www.in.tu-clausthal.de    :    80        /home/mueller/mbox

- Kombination IP-Adresse mit Portnummer

Schema	IP-Adresse	Schema	Pfadname
--------	------------	--------	----------

http       ://    139. 174. 2. 135    :    80        /home/mueller/mbox

## Das Dokumentenmodell des WWW

- Dokumente kann Referenzen (Hyperlinks) auf ein anderes Dokument enthalten
- Hyperlinks werden durch Browser explizit angezeigt → Auswahl durch Anklicken
- Auswahl eines Hyperlinks erzeugt Anforderung an den Server, auf dem das entsprechende Dokument abgelegt ist (über URL)
- Von dort wird es auf die Maschine des Nutzers übertragen und im Browser angezeigt
- Sprache für Webdokumente:  
HTML: Hypertext Markup Language

## Hypertext Markup Language

- Markup-Sprache zur Annotation und Strukturierung von Dokumenten
- HTML-Dokumente bestehen aus Kopfteil `<HEAD>` und Rumpf `<BODY>`
- HTML-Sprachelemente (=Tags) zum
  - Markieren von Überschriften, Listen, Tabellen, Formularen
  - Einfügen von Bildern und Animationen
  - Gestaltung der Darstellung im Browser (z.B. Schriften, Schriftgrößen, kursive oder fettgedruckte Anzeige, Ausrichtung, Farben, Textabsätze)
- Notation von HTML-Tags in spitzen Klammern, z.B.
  - `<H1>Hello World</H1>`
- HTML-Tags schließen in der Regel Text ein
- Web Browser verwendet HTML-Parser: Software, die HTML-Tags erkennt und in strukturierten Text umsetzt.
- Erweiterungen erlauben die Darstellung von Teilen eines Dokumentes in Form eines im Browser ausführbaren Skriptes (z.B. Javascript)



## HTML-Beispieldokument

```
<HTML>                                <!-- Start des HTML-
<BODY>                                Dokuments -->
<H1>Hello World</H1>                 <!-- Start des
<P>                                    Haupttrumpfs -->
<SCRIPT type = "text/javascript">    <!--Grundlegender
    document.writeln("<H1>Hello World</H1>"); // Eine Zeile Text schreiben
</SCRIPT>                             <!--Neuer Absatz -->
</P>                                  <!--Skriptingsprache
</BODY>                               identifizieren-->
</HTML>                               <!--Ende Skripting-
                                       abschnitt -->
                                       <!--Ende Absatz -->
                                       <!--Ende Haupttrumpf -->
                                       <!--Ende HTML-Abschnitt -->
```

## Zeichendarstellung in HTML

- **HTML-Texte werden in Dateien in binärer Form (bytes) abgespeichert**
- **Web-Browser muss Bytes wieder in Zeichen (Buchstaben etc.) umwandeln**
  - Zeichenkodierung, erfolgt mithilfe einer Codetabelle, die jedem erlaubten Zeichen aus dem Zeichenvorrat eine fortlaufende Nummer zuweist
  - Verschiedene Zeichenkodierungsschemata
  - z.B. UNICODE: "a" → 97, "b" → 98, "c" → 99, etc.
- **Browser muss bei der Anzeige die gleiche Kodierung angeben, die der Editor beim Abspeichern verwendet**
  - Spezifikation der Zeichenkodierung im Kopf des HTML-Dokumentes

## Zeichendarstellung in HTML (2)

- **Beispiel:**

```
<head>
<meta http-equiv="content-type"
      content="text/html; charset=ISO-8859-1">
<!-- ... andere Angaben im Dateikopf ... -->
</head>
```

- **Darstellung von Sonderzeichen mittels benannter HTML-Elemente, z.B.**

- `&uuml;` für "ü", `&Uuml;` für "Ü"
- `&amp;` für "&", `&szlig;` für "ß"

- **Beispiel:**

```
In M&uuml;nchen steht ein Hofbr&auml;uhaus.
Dort gibt es Bier aus Ma&szlig;kr&uuml;gen.
```

## Elemente zur Textstrukturierung

- **Bereits kennengelernt: Überschriften, Textabsätze**
- **Darstellung von Listen und Aufzählungen**

```
<ul> <!-- dies ist eine unsortierte Liste -->
<li>Element 1</li>
<li>Element 2</li>
<li>Element 3</li>
</ul>
```

```
<ol> <!-- dies ist eine nummerierte Liste -->
<li>Element 1</li>
<li>Element 2</li>
<li>Element 3</li>
</ol>
```

- **Listen können geschachtelt sein**

## HTML Tabellen

- **Aufbau einer Tabelle in HTML**

- **Tabelle** <table>
- **Zeile** <tr>
- **Tabellenkopfelement** <th>
- **Tabellenelement** <td>
- **Weitere Attribute für Form** (z.B. border)

```
<table>
<tr>
  <th>
</th>
  <th>
</th>
  <th>
</th>
</tr>
<tr>
  <td>
</td>
  <td>
</td>
  <td>
</td>
</tr>
<tr>
  <td>
</td>
  <td>
</td>
  <td>
</td>
</tr>
</table>
```

Wirtschaftsinformatik  
© Prof. Dr. Jörg P. Müller, 2007

Quelle: de.selfhtml.org

- **Beispiel: Tabelle mit Gitternetzlinien**

```
<table border="1">
<tr>
  <th>Berlin</th>
  <th>Hamburg</th>
  <th>M&uuml;nchen</th> </tr>
<tr>
  <td>Berliner Weisse</td>
  <td>Flensburger</td>
  <td>Augustiner</td>
</tr>
<tr>
  <td>Buletten</td>
  <td>Frikadellen</td>
  <td>Fleischpflanzerl</td>
</tr>
</table>
```

21

## HTML Formulare

- **Zweck: Eingabe (über Eingabefelder o. Auswahllisten) und Senden (über Button) von Daten vom Client zum Server**
- **Erstellen eines Formulars in HTML, dabei wird spezifiziert, was mit den eingegebenen Daten passieren soll**
- **Anwendungen:**
  - Suchmaschinen
  - Benutzeranmeldung bei einer Web-Applikation
  - Erfassen von Nutzer- und Bestelldaten in E-Shops
- **Formular definiert mit <form> Tag:**

```
<form action="http://www.tu-clausthal.de/cgi-bin/feedback.pl" method="get">
<!-- hier folgen die Formularelemente -->
</form>
```

Wirtschaftsinformatik  
© Prof. Dr. Jörg P. Müller, 2007

22

## Attribute des <form> Tags

- **action:** Gibt an, was mit den Eingabedaten passieren soll, wenn das Formular abgesendet wird, z.B.
  - **Email senden:**  
`<form action="mailto:mueller@tu-clausthal.de" ...>`
  - **Programm auf dem Server aufrufen, das die Daten weiterverarbeitet**  
`<form action="http://www.tu-clausthal.de/cgi-bin/feedback.pl" ...>`
- **method:** Auswahl der http-Übertragungsmethode für die Formulardaten (s.o.)
  - **get:** Daten des ausgefüllten Formulars als Parameter an die Aufrufadresse angehängt
  - **post:** Daten des ausgefüllten Formulars werden vom Web-Server über den Standardeingabekanal (d.h. im Body des HTTP-Requests) zur Verfügung gestellt
  - **put:** Verwendet für das Upload einer Datei vom Client zum Server

## Darstellung von Web-Layouts durch Cascading Stylesheets (CSS)

- **Idee:** Trennung der Spezifikation von Inhalt und Form
- **CSS = Sprache zur Definition von Formateigenschaften** (z.B. Rahmen, Farben, Hintergrund, Zeichengröße, Font) einzelner HTML-Elemente
- **Stylesheet-Format wird vom Browser angewendet, um Darstellung der Elemente zu bestimmen**
- **Für einen Inhalt (HTML) kann durch Auswahl des Stylesheets festgelegt werden, wie der Inhalt angezeigt wird**
- **Anwendungen:**
  - Festlegung von Richtlinien für die Webdarstellung eines Unternehmens ("Corporate Look & Feel")
  - Optimierung der Darstellung von Inhalten für unterschiedliche Bildschirmformate und -größen (dafür besser geeignet: XSLT!)

## CSS Sprache

- **Stylesheet besteht aus einer Menge von Formaten (Regeln)**
- **Regel hat die Form**  
`Selektor { Eigenschaft:Wert; }`
  - Selektor wählt gewünschte Elemente aus
  - Eigenschaft: zu spezifizierende Eigenschaft des Elements sowie der ihr zugewiesene Wert
- **Beispiele:**
  - `h1 { border-bottom:4px double blue; padding-bottom:4px; margin:0px; }`
  - `h2 { border-top:1px solid blue; margin:0px; }`
  - `p {color:blue;font-size:200%;}`

## Definition eines CSS in einem HTML-Dokument

### 1. Definition im Header einer Datei durch `<style>` Tag

- **Beispiel:**

```
<html> <head>
<title>Titel der Datei</title>
<style type="text/css">
<!-- ... Hier werden die Formate definiert ... -->
</style>
</head> <body> </body> </html>
```

### 2. Definition in einer eigenen Stylesheet-Datei

```
<html> <head>
<title>Titel der Datei</title>
<link rel="stylesheet" type="text/css" href="formate.css">
<style type="text/css">
<!-- ... hier sind dateispezifische Formate erlaubt ... */
</style>
</head> <body> </body> </html>
```

### 3. Definition in einem HTML-Element

## HTML-Editoren

- **In der Praxis werden zur Erstellung von HTML Editoren verwendet (z.B. Adobe GoLive, Microsoft Frontpage)**
- **Content-Management-Systeme wie Typo3 erlauben das Erstellen von HTML-Dokumenten in WYSIWYG-Form (z.B. als RTF)**
- **Für die meisten größeren Anwendungen gilt, dass HTML von einem Serverprogramm dynamisch erstellt wird (z.B. aus**
  - **Active Server Pages (Microsoft)**
  - **Java Server Pages**
  - **XML Dokumenten durch XML Stylesheet Transformation (XSLT)**

## HTML: Zusammenfassung

- **Hypertext Markup Language (HTML)**
  - **einheitliches, fixes Markup-System**
- **Vorteile:**
  - **einfach**
  - **portabel**
  - **einheitliche Handhabung von Verweisen (Links)**
- **HTML erlaubt erstmals den breiten Einsatz einheitlicher Schnittstellen zu Information und Software-Applikationen innerhalb einer Organisation**

## Nachteile von HTML

- **HTML ist nicht erweiterbar:**
  - Markup (Elemente und Attribute)
  - Verweise
- **HTML unterstützt nicht die Modellierung strukturierter Informationen (Objektmodelle, relationale Datenbanken)**
- **Integration von Software-Applikationen**
  - JavaScript, PlugIns
  - Proprietäre Formate und Sprachen
- **Handhabbarkeit: unzureichende Entkopplung von Inhalt und Präsentation / Layout**
- **Ständiger Wandel der HTML Sprache**

## XML

- **XML = eXtended Markup Language**
- **Ursprung: HTML4.0  $\in$  XML  $\subset$  SGML (Standard Generalized Markup Language, ISO 8879)**
- **Web-Standard (W3C) für semi-strukturierte Dokumente**
- **Zwei Arten von Dokumenten:**
  - Document-Type-Definitions (DTD's) definieren über eine kontextfreie Grammatik, was ein legales XML Dokument ist
  - XML Dokumente: sind die Datendokumente, strukturiert bzgl. einer DTD
- **HTML & XML:**
  - HTML beschreibt die Präsentation eines Dokumentes durch feste Tags
  - XML beschreibt den Inhalt durch benutzerdefinierte Tags
- **Grundidee von XML: Trennung von Inhalt, Struktur und Präsentation**

## HTML und XML: Ein kleines Beispiel



## HTML Beispiel

```
<HTML>
<HEAD>
  <META HTTP-EQUIV="Content-Type"
    CONTENT="text/html; charset=iso-8859-1">
  <META NAME="Author" CONTENT="Joerg Mueller">
  <TITLE>Ein alter Witz</TITLE>
  <H1><FONT SIZE=+4>Ein alter Witz</FONT></H1>
</HEAD>

<BODY>
<FONT SIZE=+2>Tümmes: "Sach ma Bescheid,
Schäl!"</FONT>
<P>
<FONT SIZE=+2>Schäl: "Bescheid!"</FONT>
</BODY>
</HTML>
```



## Das gleiche in XML

```
<?XML version="1.0"?>
<!DOCTYPE alterwitz PUBLIC "-//Katalog//DTD WITZ//DE-KOELSCH">
<alterwitz>
<wtitel>Ein alter Witz</wtitel>
<tuennes> T&uuml;nnes:
<zitat language="ge:koelsch">Sach ma Bescheid</zitat>
<anrede> , Sch&auml;l! </anrede>"</tuennes>
<schael> Sch&auml;l: <zitat language="ge:koelsch"> Bescheid,
Sch&auml;l!</zitat></schael>
<applaus/>
</alterwitz>
```

## XML Markups

- **Elemente**

```
<alterwitz
  language="ge:koelsch"
  status 'lustig'>
</alterwitz>
<wtitel>...</wtitel>
<applaus/>
```

- **Objekt-Referenzen**

```
&lt; &gt; &amp; &apos; &quot;
&uuml;
&zunologo;
```

- **Kommentare**

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

## XML Markups

- **Anweisungen**
  - dienen zur Übergabe von Information an Applikationen, z.B.  
`<?XML VERSION=1.0 RMD="NONE" ENCODING="UTF-8"?>`
  - allgemeine Form:  
`<?name parameterliste?>`
- **CDATA Sektionen**
  - Zeichen in CDATA Sektionen werden vom XML-Prozessor ignoriert; z.B. Programm-Listings:  
`<! [CDATA[ *p = &q;  
          b = ( I <= 3 ); ] ]>`
- **Dokument-Typ-Definitionen**
  - Metainformation über die Dokumentstruktur
  - Optional
  - Definiert: Elemente, Attributlisten, Objekte, Notationen

## Beispiel XML DTD

```
<!--DTD fuer alterwitz -->
<!ELEMENT alterwitz - - (wtitel, tuennes+, schael, applaus?)>
<!ELEMENT wtitel - - (#PCDATA)+>
<!ELEMENT tuennes - - (zitat, anrede)>
<!ELEMENT schael - - (zitat)>
<!ELEMENT zitat - - (#PCDATA)+>
<!ELEMENT anrede - - (#PCDATA)+>
<!ELEMENT applaus empty>
<!ATTLIST alterwitz
    LANGUAGE CDATA #IMPLIED
    STATUS ( lustig | nicht_lustig ) 'lustig'>
<!ENTITY uuml #x00FC>
<!ENTITY auml #x00E4>
<!NOTATION GIF87A SYSTEM "GIF"> <!--info fuer Applikationen -->
...

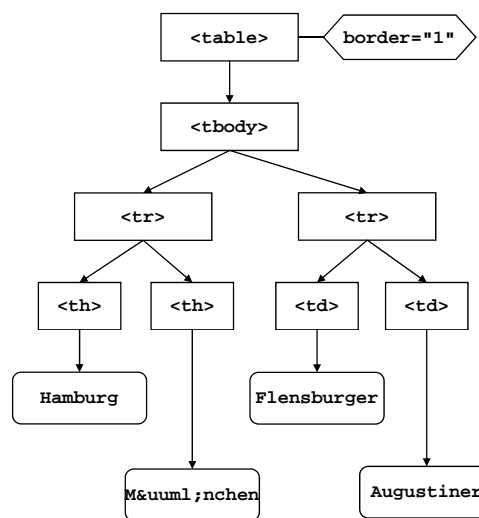
```

## Interne Darstellung eines XML-Dokuments

- Darstellung als Baumstruktur
- Knoten sind Elemente
- Kanten: Beziehungen zwischen Elementen
  - Nachfolger ("Kind")
  - Vorgänger ("Elternknoten")
- Existenz eines "Wurzelknoten"
  - <?xml ...>
- Nachfolger ("Kinder") eines Knotens sind die in ihm enthaltenen Elemente
- Attribute sind dem ihr Element entsprechenden Knoten zugeordnet
- XML-Anwendungen (Parser, XSLT) arbeiten auf dieser Baumstruktur
- Standard-Modell: ([www.w3.org/DOM](http://www.w3.org/DOM))  
DOM (Document Object Modell) mit Implementierungen z.B. für Java

## Beispiel: DOM Darstellung eines XML-Dokuments

```
<table border="1">
<tbody>
<tr>
<th>Hamburg</th>
<th>M&uuml;nchen</th> </tr>
<tr>
<td>Flensburger</td>
<td>Augustiner</td>
</tr>
</tr>
</tbody>
</table>
```



## XML und HTML: Hauptunterschiede

- XML erlaubt flexible Definition von Markups
- Strikte Sprachdefinition von XML erlaubt Test auf
  - Wohlgeformtheit (bzgl. der Syntax von XML)
  - Gültigkeit (bezüglich einer DTD)
- XML enthält strukturelle Information
- Diese kann verwendet werden zur
  - Suche von Dokumenten
  - Präsentation von Dokumenten
  - Steuerung des Informationsflusses
- Präsentationsaspekte von XML sind in sogenannten "Stylesheets" definiert (eigene Sprache XML Stylesheet Language, XSL)
- Eigene Sprachen fuer Verweisstrukturen (XPath, Xpointer, XLL)

## Nutzung von XML als Datenformat

```
<ORDER>
  <SOLD-TO>
    <PERSON><LASTNAME>Layman</LASTNAME>
      <FIRSTNAME>Andrew</FIRSTNAME>
    </PERSON>
  </SOLD-TO>
  <SOLD-ON>19970317</SOLD-ON>
  <ITEM>
    <PRICE>5.95</PRICE>
    <BOOK>
      <TITLE>Number, the Language of Science</TITLE>
      <AUTHOR>Dantzig, Tobias</AUTHOR>
    </BOOK>
  </ITEM>
  <ITEM>
    <PRICE>12.95</PRICE>
    <BOOK>
      <TITLE>Introduction to Objectivist
        Epistemology</TITLE>
      <AUTHOR>Rand, Ayn</AUTHOR>
      <ISBN>0-452-01030-6</ISBN>
    </BOOK>
  </ITEM>
</ORDER>
```

## XML Schema

- **Schwäche von XML:** Es gibt keine Datentypen, es gibt nur STRINGS

```
<ITEM>
  <PRICE>5.95</PRICE>
</ITEM>
```

Keine Möglichkeit, auszudrücken, dass hier nur Zahlen erlaubt sind

- **XML Schema:**
  - Standard entwickelt vom W3C (siehe : [www.w3.org/XML/Schema](http://www.w3.org/XML/Schema))
- **Standardisierte Beschreibung komplexerer (auch zusammengesetzter) Datenformate**
  - Vordefinierte Datentypen
  - Untertypenbildung
  - Namensräume
- **XML Schema beschreibt ein XML Dokument**
  - ähnliche Funktion wie die DTD, aber mehr Struktur
  - ist selbst ein XML Dokument (Meta-Beschreibung) und keine DTD

## Einfache Datentypen

- **Standard-Datentypen sind vordefiniert z.B.:**
  - string, integer, positiveInteger, boolean,
  - time, date, duration,
  - ID und IDREF (aus XML)
- **Datentypen können eingeschränkt werden, z.B.:**
  - length, minLength, maxLength (z.B. Länge von Strings)
  - minInclusive, maxInclusive: Wertebereiche bei numerischen Typen
  - enumeration: Definition von Aufzählungstypen (basierend auf string)
- **Definiert durch Schema-Elemente:**  
`<xsd:simpleType>` und `<xsd:restriction>`

## Syntaxbeispiele

```
<xsd:simpleType name="sizeType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="S" />
    <xsd:enumeration value="M" />
    <xsd:enumeration value="L" />
  </xsd:restriction>
</xsd:simpleType>
```

Aufzählungstyp  
{S, M, L}

```
<xsd:simpleType name="ageType">
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="0" />
    <xsd:maxInclusive value="150" />
  </xsd:restriction>
</xsd:simpleType>
```

Range [0...150]

## Komplexe Datentypen

- Definiert durch Schema-Elemente: `<xsd:complexType>`
- Definition von verschachtelten XML-Elementen:
  - `<xsd:sequence>` : Folge von Elementen
  - `<xsd:choice>` : Auswahl aus einer Menge von Elementen
  - `<xsd:all>` : Menge von Elementen, Reihenfolge egal
  - `<xsd:group>` : Definition und Referenz auf wiederverwendbare Gruppe von Elementen
- Innerhalb dieser Schemaelemente werden lokale XML-Elemente definiert durch:
  - `<xsd:element name="...Elementname..." type="...Typname..." minOccurs="...Zahl..." maxOccurs="...Zahl..." />`

Kardinalität: untere  
Schranke

Kardinalität: obere  
Schranke

## Syntaxbeispiele

```
<xsd:complexType name="personType">  
  <xsd:sequence>  
    <xsd:element name="name" type="xsd:string" />  
    <xsd:element name="age" type="ageType" minOccurs="0"/>  
    <xsd:element name="size" type="sizeType" minOccurs="0"/>  
  </xsd:sequence>  
</xsd:complexType>
```

Komplexer  
Typ

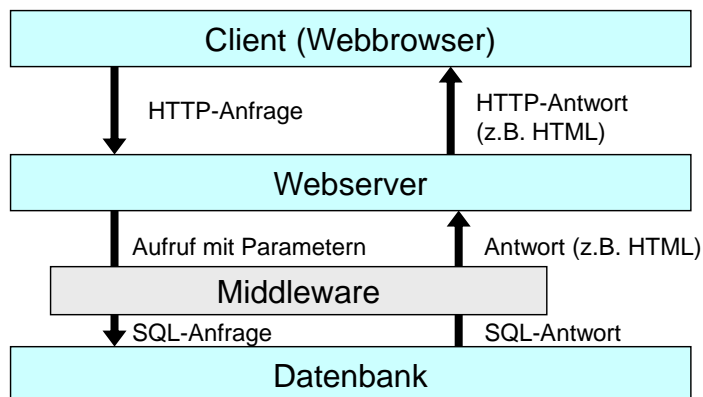
```
<xsd:element name="person" type="personType" />
```

Top-Level  
Element für  
Instanz

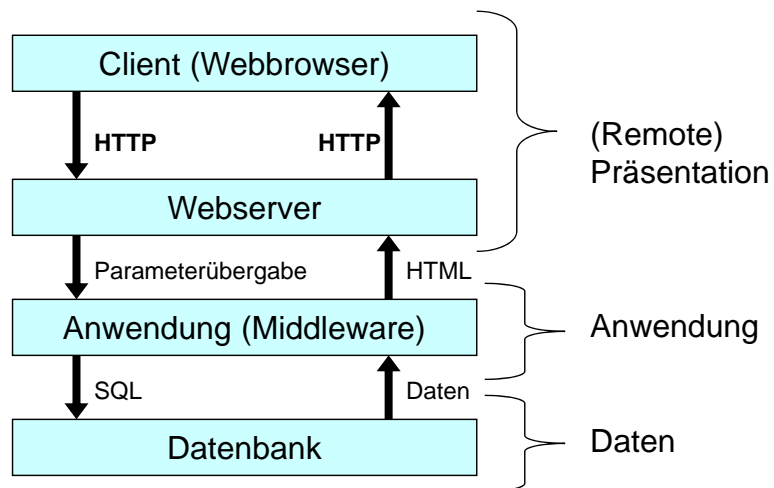
### Instanz des Schemas:

```
<person>  
  <name> Fritz </name>  
  <age> 24 </age>  
  <size> M </size>  
</person>
```

## Architektur datenbankbasierter Web-Anwendungen



## Drei-Schichten-Architektur Modell



Wirtschaftsinformatik  
© Prof. Dr. Jörg P. Müller, 2007

47

## HTTP – das Hypertext-Transfer-Protokoll

- **Kommunikation zwischen Web Server und Web Clients**
- **Einfaches Client-Server-Protokoll**
  - Client schickt Anforderungsnachricht an Server (sog. HTTP-Request)
  - Server verarbeitet Anforderung und sendet Antwort (sog. HTTP-Response)
- **HTTP ist ein zustandsloses Protokoll, d.h.**
  - Server verwaltet keine Information über Clients
  - Im Prinzip: Für jeden Request-Response-Vorgang wird eine neue TCP-Verbindung zwischen Client und Server aufgebaut


Wirtschaftsinformatik  
© Prof. Dr. Jörg P. Müller, 2007

48



## Parameterübergabe an den Webserver

- Typischer Anwendungsfall: Formulardaten verarbeiten



Serverseitiges  
Programm  
z.B. Email versenden

Zwei Methoden:

- Get
- Post

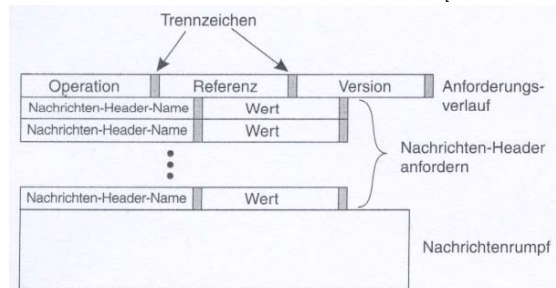
## Parameterübergabe GET vs. POST

- GET
  - Anhängen an URL
  - für Benutzer sichtbar (auch Passwort-Felder)
  - begrenzte Länge
  - in Formularen und in Links verwendbar
- POST
  - Übertragung im Request
  - unbegrenzte Länge
  - vor allem für Formulare verwendet



## Aufbau eines HTTP-Request

[Quelle: Tanenbaum (2003), p.737]



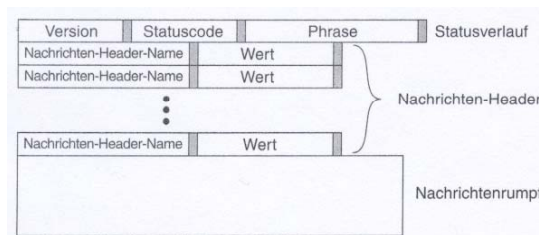
- Enthält IP-Adresse des Servers und die Bezeichnung der angeforderten Ressource (Seite)

```
GET /verzeichnis/seite.pl HTTP/1.0
Host: 100.101.102.103
```

- Möglichkeiten zur Übermittlung weiterer Informationen:
  - QueryStrings: an die URL angehängte Informationen
  - im Request-Body (bei Methode POST)
  - Cookies

## Aufbau einer HTTP-Response

[Quelle: Tanenbaum (2003), p.737]



- **Dreistelliger Status-Code mit textueller Beschreibung**
  - Z.B. 200 = "OK", 405 = "Method not allowed"
- **Weitere Information im Response-Header,**
  - z.B. "Allow head, get", "LastModified 11.11.2005"
- **Nachrichtenrumpf enthält in der Regel das HTML-Dokument**

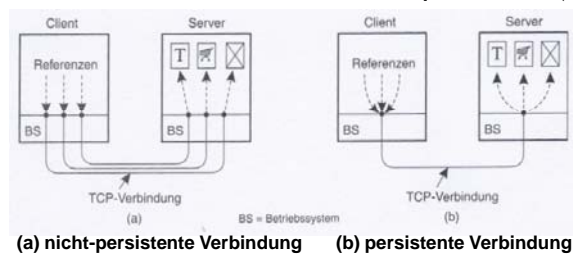
## Beispiel HTTP Post Request-Nachricht

```
POST /send.php HTTP/1.1
Host: meinserver.de
User-Agent: Mozilla/4.0
Accept: image/gif, image/jpeg, */*
Content-type: application/x-www-form-urlencoded
Content-length: 51
Connection: close

Vorname=Max&name=Mustermann&mail=max%40muster%2Ede
```

## TCP-Verbindungsarten in HTTP

[Quelle: Tanenbaum (2003), p.735]



- **Problem:** Der Zugriff auf ein „logisches Webdokument“ erfordert mehr als einen physischen HTTP-Request (z.B. im Web-Dokument referenzierte Bilder)
- **Abhilfe:** Ab HTTP Version 1.1: Unterstützung für persistente TCP-Verbindungen, d.h. mehr als ein Request-Response-Paar pro aufgebauter TCP-Verbindung
  - Kostspieliger Aufbau der TCP-Verbindung bei jedem Request entfällt.
- **Pipelining:** Client kann mehrere Requests absetzen, ohne auf die Antwort auf die erste zu warten

## Weltweite Nutzung – Das Netz für alle?

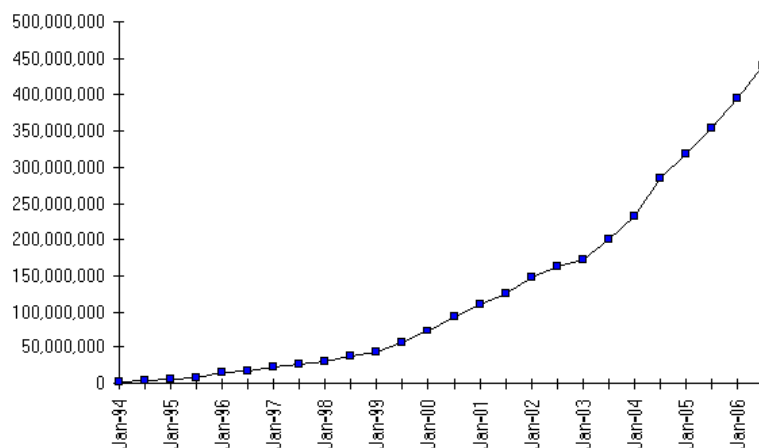
Region	Anz. Nutzer in Mio.	% der Bevölkerung	Wachstum in % 2000-2006
Afrika	32,8 (13,5)*	3,6	626
Asien	394,9 (302,3)	10,8	246
Europa	308,7 (259,7)	38,2	194
Naher Osten	19,0 (19,4)	10,0	470
Nordamerika	229,1 (221,4)	69,1	112
Mittel- u. Südamerika	83,4 (56,2)	15,1	361
Ozeanien / Australien	18,4 (16,3)	54,1	141
<b>GESAMT</b>	<b>1.086,3</b>	<b>16,7</b>	<b>200</b>

Quelle: <http://www.internetworldstats.com/stats.htm> von Sept 18, 2006  
\* zum Vergleich in Klammern: Werte vom 24. März 2005)

Wirtschaftsinformatik  
© Prof. Dr. Jörg P. Müller, 2007

55

## Statistiken: Anzahl der Server im Internet

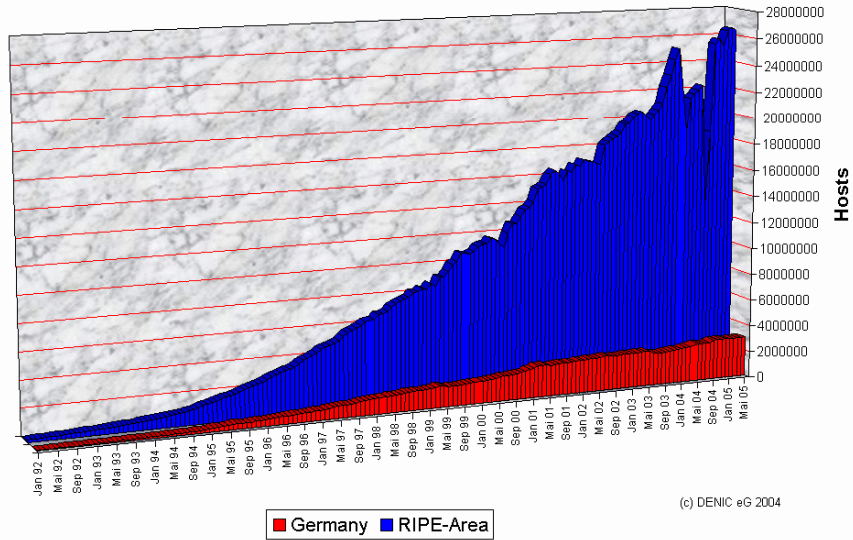


Quelle: Internet Systems Consortium, <http://www.isc.org/ds/> (3.11.2006)

Wirtschaftsinformatik  
© Prof. Dr. Jörg P. Müller, 2007

56

## Hosts in Deutschland und Europa



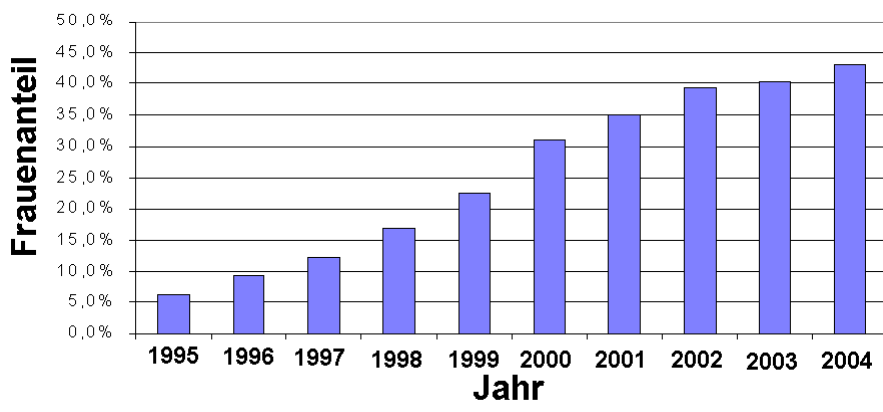
Wirtschaftsinformatik  
© Prof. Dr. Jörg P. Müller, 2007

Quelle: <http://www.denic.de/de/domains/statistiken/hostentwicklung/hostcount.html>

57

## Internet-Nutzung in Deutschland

### Anteil der weiblichen Internetnutzer



Wirtschaftsinformatik  
© Prof. Dr. Jörg P. Müller, 2007

58

## Organisationen

- **Internet Activities Board (IAB)**
  - betreut den Standardisierungsprozess
  - Verwaltung der RFCs (Request for Comments)
  - Unterorganisationen:
    - IETF (Internet Engineering Task Force): Funktion des Internet sowie für die Lösung aller Protokoll- und Architekturfragen
    - IRTF (Internet Research Task Force): Entwicklung neuer Technologien
- **World Wide Web Consortium (W3C)**
  - Zusammenschluss der Industrie

## Lerninhalte dieses Kapitels

- **Internet und WWW: Definition, Geschichte, Bedeutung**
- **Dokumentenmodell des WWW: HTML**
- **Web-Layout mit Cascading Stylesheets beschreiben**
- **Grundlagen von XML und XML Schema**
- **Architekturen und Protokolle des Internet / WWW**

## Ressourcen

- **Internet / WWW:**
  - <http://www.w3.org/WWW>
  - Internet Society: <http://www.isoc.org>
  - History of the Internet: <http://www.isoc.org/internet/history/>
  - History of the WWW: [www.w3history.org/](http://www.w3history.org/)
- **HTML / CSS:**
  - <http://selfhtml.org>
- **XML:**
  - <http://www.w3.org/XML/>
  - Elliotte R. Harold und W. Scott Means. *XML in a Nutshell*, O'Reilly, 2005.
- **XML Schema:**
  - <http://www.w3.org/XML/Schema>