

Interaktive Medien



Prof. Dr. Frank Steinicke
Human-Computer Interaction
Fachbereich Informatik
Universität Hamburg



http://www



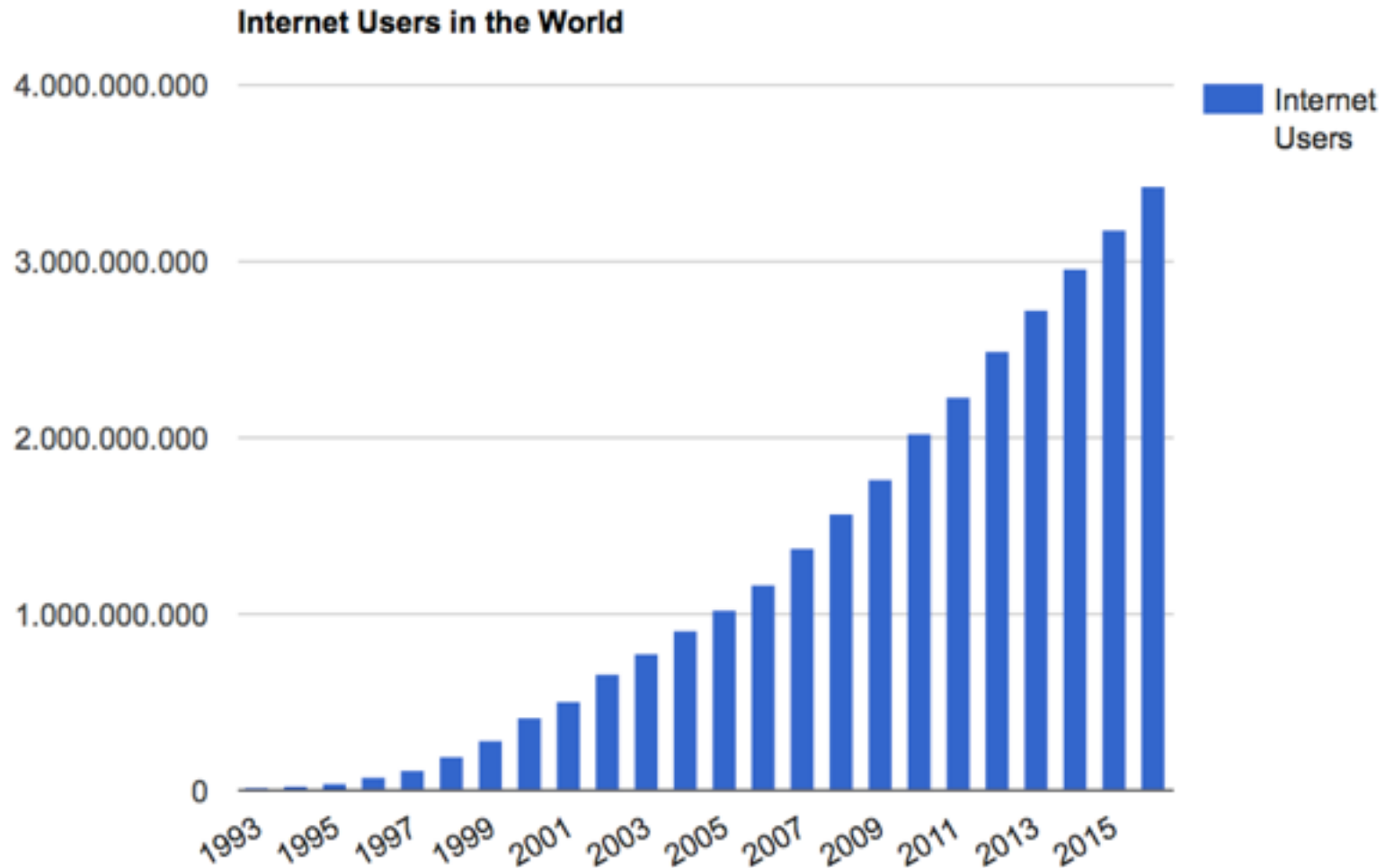
Interaktive Medien

Kapitel Mediensysteme & World Wide Web

Prof. Dr. Frank Steinicke

Human-Computer Interaction, Universität Hamburg

Internetnutzer



<http://www.internetlivestats.com/internet-users/>

Internetnutzer

Year	Internet Users**	Penetration (% of Pop)	World Population	Non-Users (Internetless)	1Y User Change	1Y User Change	World Pop. Change
2016*	3,424,971,237	46.1 %	7,432,663,275	4,007,692,038	7.5 %	238,975,082	1.13 %
2015*	3,185,996,155	43.4 %	7,349,472,099	4,163,475,944	7.8 %	229,610,586	1.15 %
2014	2,956,385,569	40.7 %	7,265,785,946	4,309,400,377	8.4 %	227,957,462	1.17 %
2013	2,728,428,107	38 %	7,181,715,139	4,453,287,032	9.4 %	233,691,859	1.19 %
2012	2,494,736,248	35.1 %	7,097,500,453	4,602,764,205	11.8 %	262,778,889	1.2 %
2011	2,231,957,359	31.8 %	7,013,427,052	4,781,469,693	10.3 %	208,754,385	1.21 %
2010	2,023,202,974	29.2 %	6,929,725,043	4,906,522,069	14.5 %	256,799,160	1.22 %
2009	1,766,403,814	25.8 %	6,846,479,521	5,080,075,707	12.1 %	191,336,294	1.22 %
2008	1,575,067,520	23.3 %	6,763,732,879	5,188,665,359	14.7 %	201,840,532	1.23 %
2007	1,373,226,988	20.6 %	6,681,607,320	5,308,380,332	18.1 %	210,310,170	1.23 %
2006	1,162,916,818	17.6 %	6,600,220,247	5,437,303,429	12.9 %	132,815,529	1.24 %
2005	1,030,101,289	15.8 %	6,519,635,850	5,489,534,561	12.8 %	116,773,518	1.24 %
2004	913,327,771	14.2 %	6,439,842,408	5,526,514,637	16.9 %	131,891,788	1.24 %
2003	781,435,983	12.3 %	6,360,764,684	5,579,328,701	17.5 %	116,370,969	1.25 %
2002	665,065,014	10.6 %	6,282,301,767	5,617,236,753	32.4 %	162,772,769	1.26 %
2001	502,292,245	8.1 %	6,204,310,739	5,702,018,494	21.1 %	87,497,288	1.27 %
2000	414,794,957	6.8 %	6,126,622,121	5,711,827,164	47.3 %	133,257,305	1.28 %

Inhalt

- Überblick
- Internet und WWW
- HTML
 - Medieneinbettung für Webseiten
 - Trennung von Inhalt und Darstellung
- Allgemeine Webdokumente: XML
- Dynamische Webseiten



Interaktive Medien

Kapitel Mediensysteme & World Wide Web

Wie funktioniert das Internet?

Diskussion

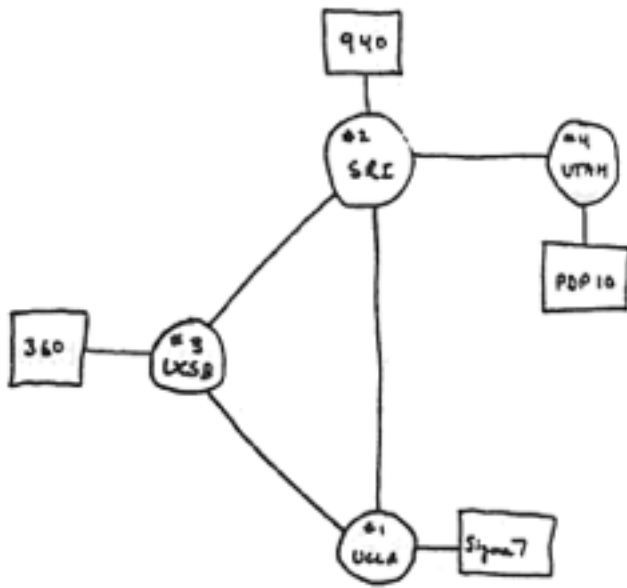


Wie hängen die Begriffe Internet und WWW zusammen?

Entstehung des Internet

- Internet \neq WWW
- Internet entstand in zwei Phasen
 - 1969: ARPANET (vernetztes Rechnersystem verschiedener Universitäten)
 - Anfang 1970er Jahre: Entwicklung von Protokollen zur Regelung und Vereinheitlichung von Sprachen
Regeln für die Kommunikation

ARPANET



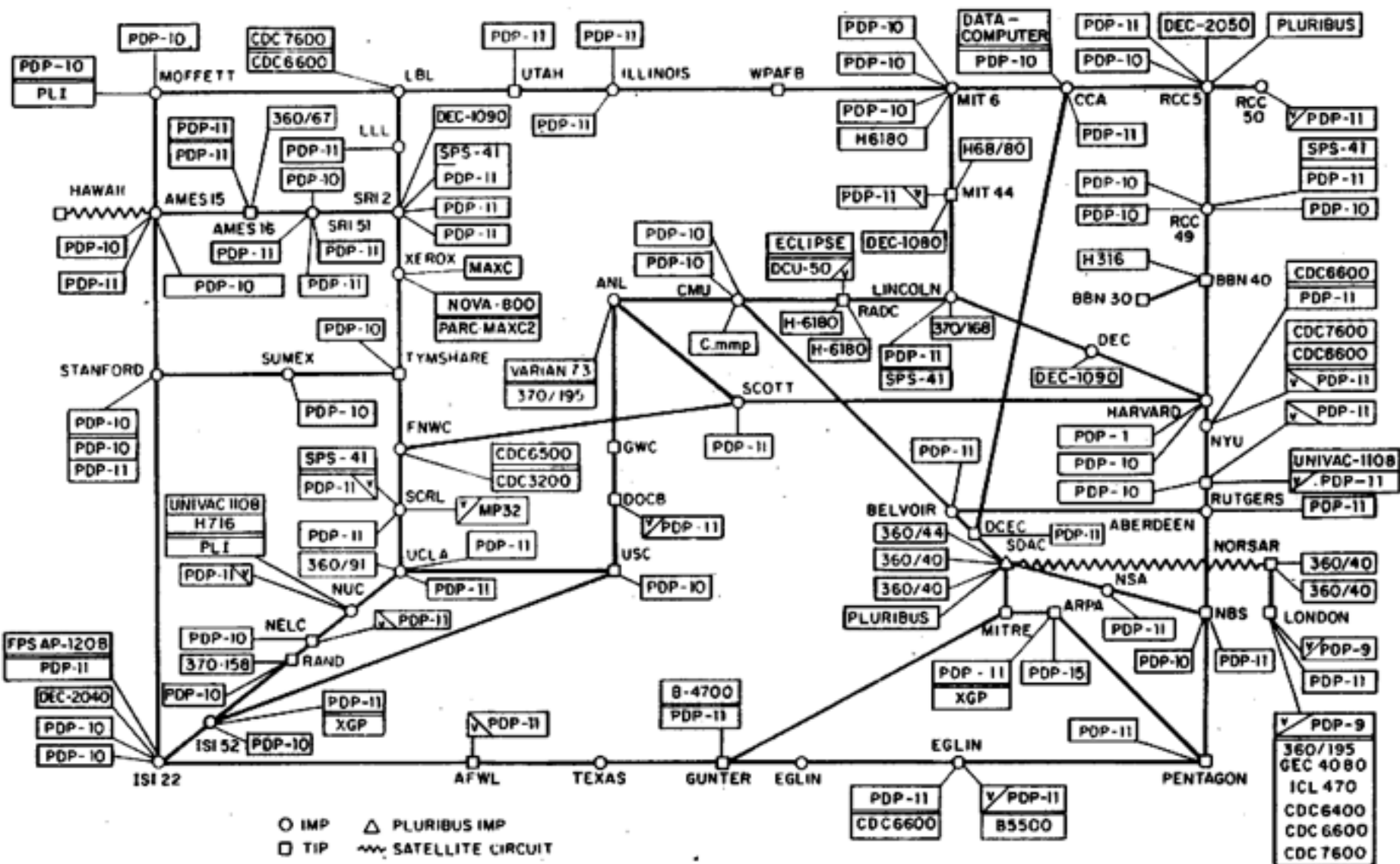
THE ARPA NETWORK

DEC 1969

4 NODES



ARPANET LOGICAL MAP, MARCH 1977



(PLEASE NOTE THAT WHILE THIS MAP SHOWS THE HOST POPULATION OF THE NETWORK ACCORDING TO THE BEST INFORMATION OBTAINABLE, NO CLAIM CAN BE MADE FOR ITS ACCURACY)

NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES

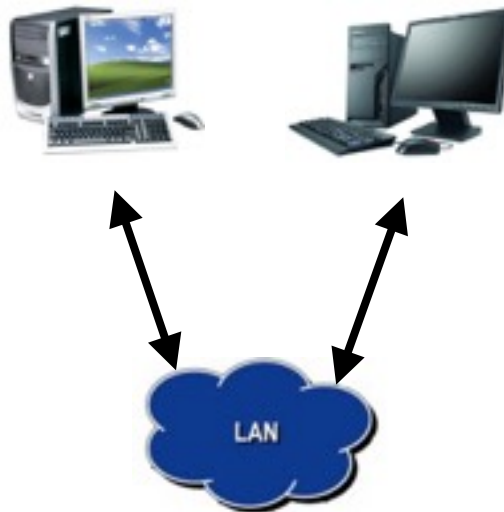
Rechnernetze

- **Server** bietet Ressourcen für **Client** an bzw. **Client** fordert Ressourcen vom **Server** an
- **Client** und **Server** müssen physikalische Verbindung besitzen
 - Beispiele für physikalische Verbindungen: Kupferkoaxialkabel, Glasfaserkabel, Satelliten (Mikrowellen) ...

Local Area Network

Beispiel

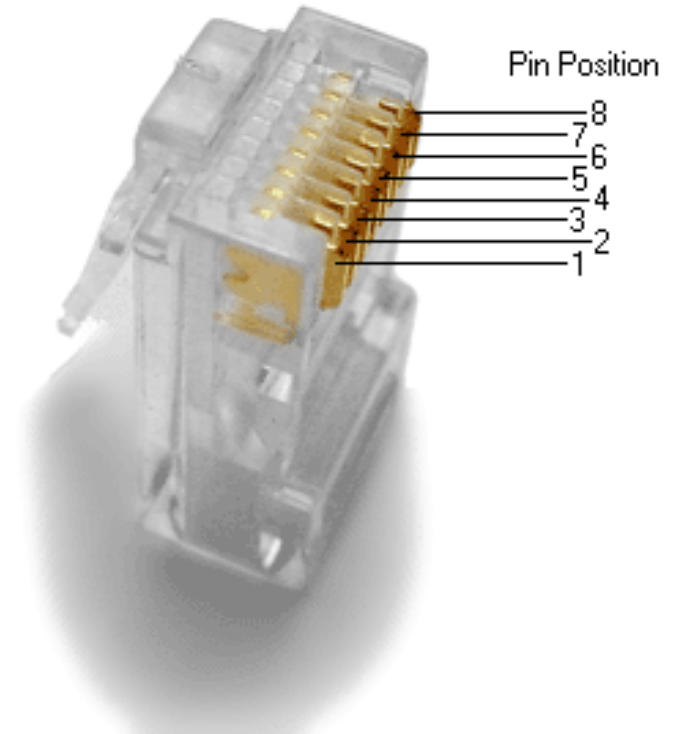
- **Local Area Network (LAN)** bezeichnet lokales Netz, welches Verbindungen von PCs, Workstations oder Servern in räumlich begrenzten Raum realisiert



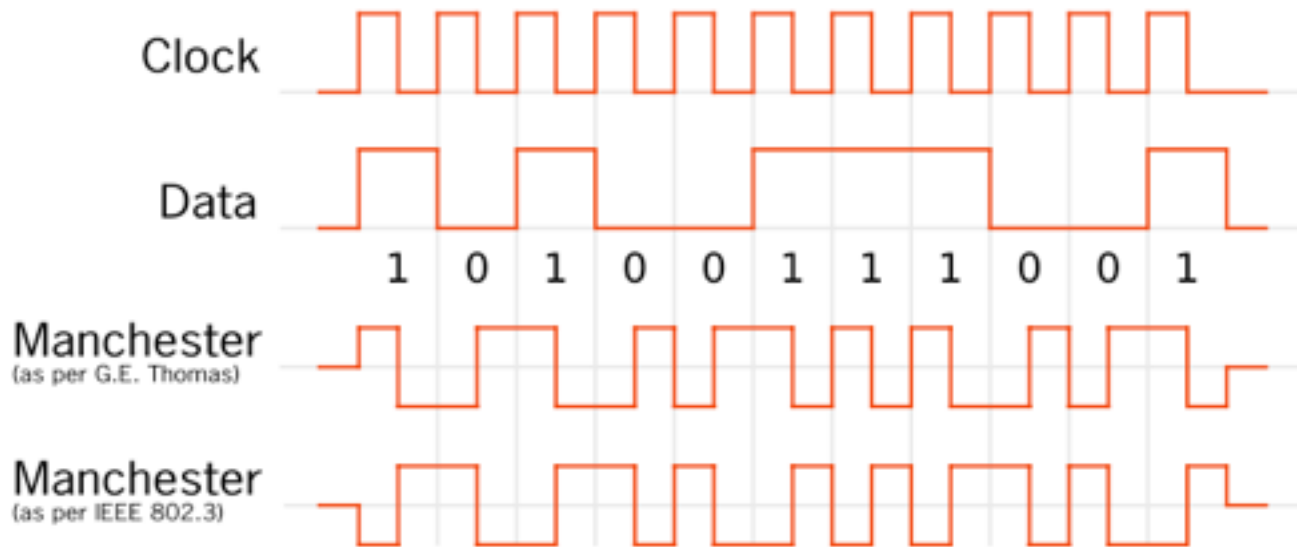
Patchkabel

Beispiel

Patchkabel



Manchester Codierung



Übertragungsrate

- **Übertragungsrate** bezeichnet digitale Datenmenge, die innerhalb einer Zeiteinheit über Übertragungskanal übertragen wird und hängt ab von
 - **Bandbreite:** Frequenzbereich, in dem elektrische Signale übertragen werden
 - **Signalstufen:** Anzahl der Zustände, die bei Codierung gleichzeitig verschickt werden

Übertragungsrate

Beispiele

- FireWire 400: ca. 400 Mbit/s,
FireWire 800: ca. 800 Mbit/s
- USB 3.0: 5 Gbit/s
- Bluetooth 2.0+EDR: 3 Mbit/s
- WLAN: 1 bis 1300 Mbit/s
- Ethernet: 10 Mbit/s, Fast Ethernet: 100 Mbit/s,
Gigabit Ethernet: 1 Gbit/s
- ...

Diskussion



Wie lange dauert es eine MP3-Datei (3 MB) bei einer Verbindungsgeschwindigkeit von 16 Mbit/s herunterzuladen?

Datenübertragung

Beispiele

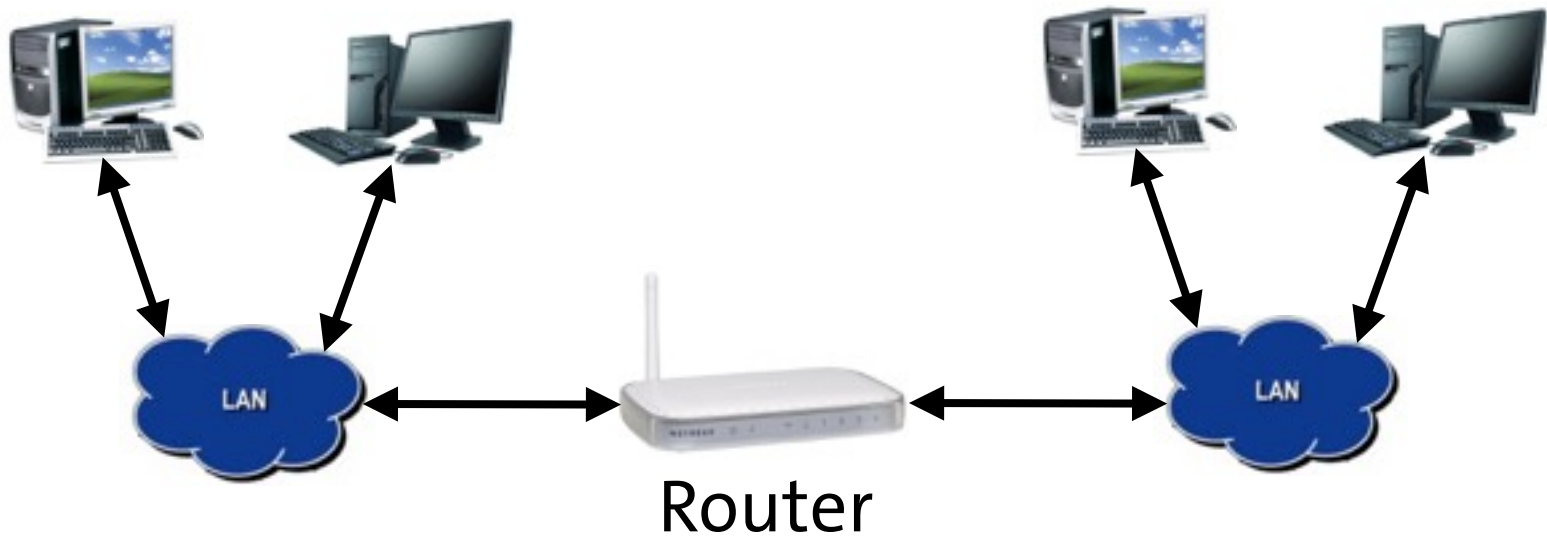
	Größe	Modem 33 600 bps	ISDN 64 000 bps	T-ADSL 768 000 bps
Brief	10 kB	2.4s	1.2s	0.1s
Musikstück WAV	35 MB	2:20h	1:12h	6min
Musikstück MP3	3 MB	12min	6min	30s
CD	650 MB	43h	22h	2h

Internetworking

- **Internetworking** beschreibt Verbindungen von verschiedenen Netzwerken zueinander
- **Interconnected Networks** (kurz *Internets*) ist Computernetzwerk aus mehreren kleinen Teilnetzwerken

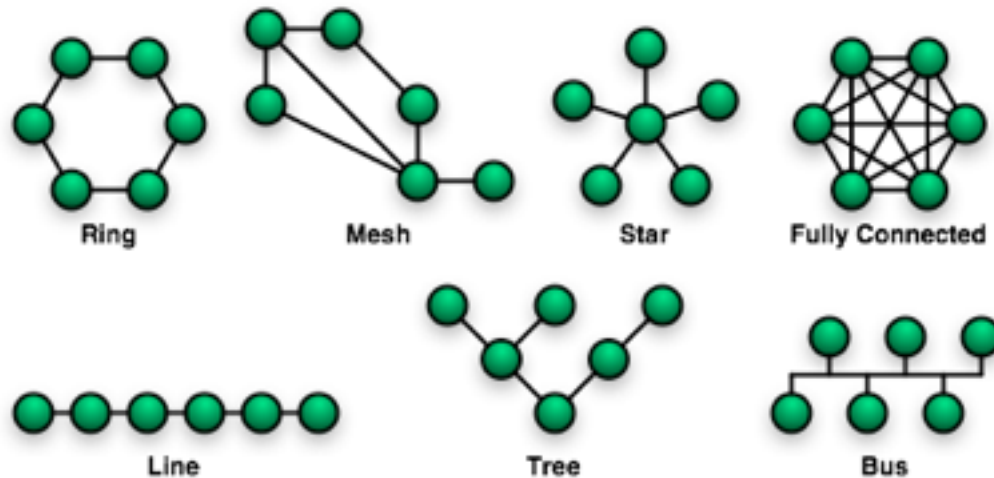
Internetworking

Beispiel

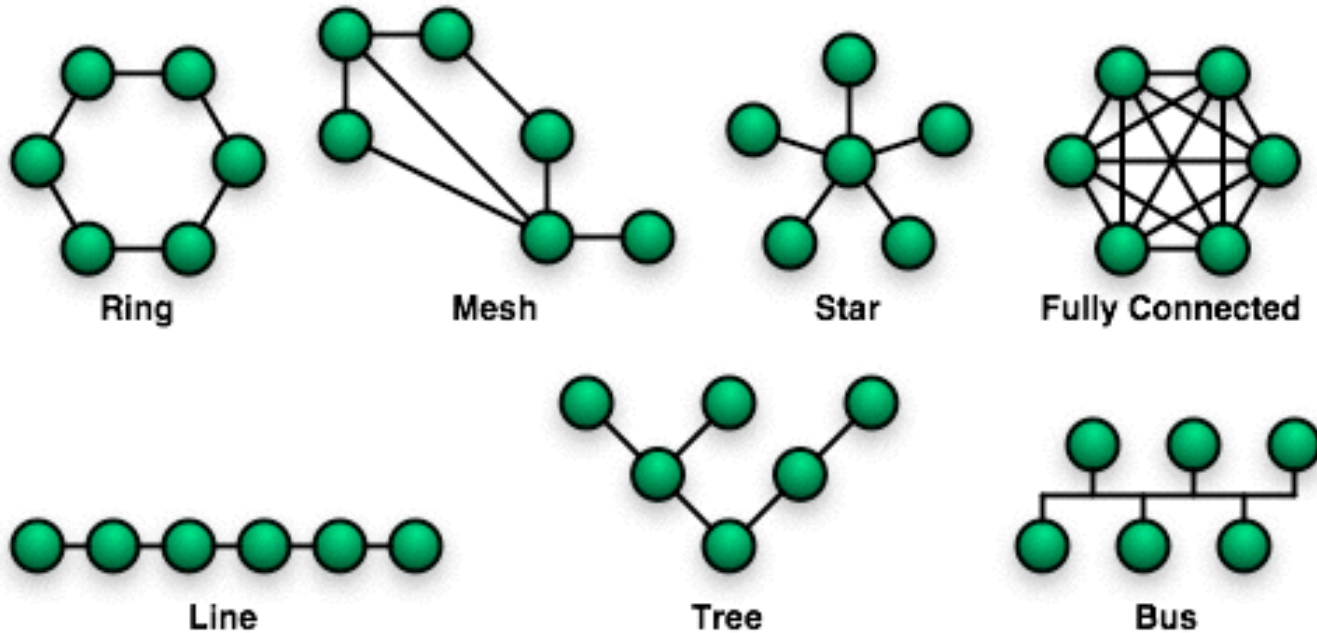


Netzwerk-Topologien

- **Topologie** eines Computernetzes bezeichnet Struktur der Verbindungen mehrerer Geräte untereinander, um gemeinsamen Datenaustausch zu gewährleisten



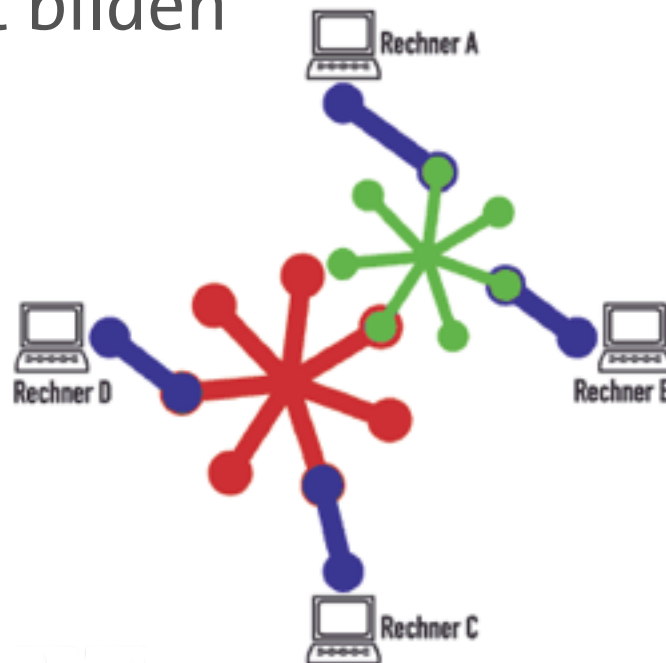
Diskussion



Diskutieren Sie Vor- und Nachteile der verschiedenen Topologien!

Topologie des Internet

- **Internet** ist grundsätzlich dezentral angeordnet und besteht aus Vielzahl einzelner Netze, die in Gemeinsamkeit Internet bilden



Protokolle

- vollständiges **Kommunikationsprotokoll** muss alle Aspekte von physikalischer Signalübertragung bis zu komplexen Diensten (gefordert durch Anwendungsprogramme) beschreiben



Protokolle

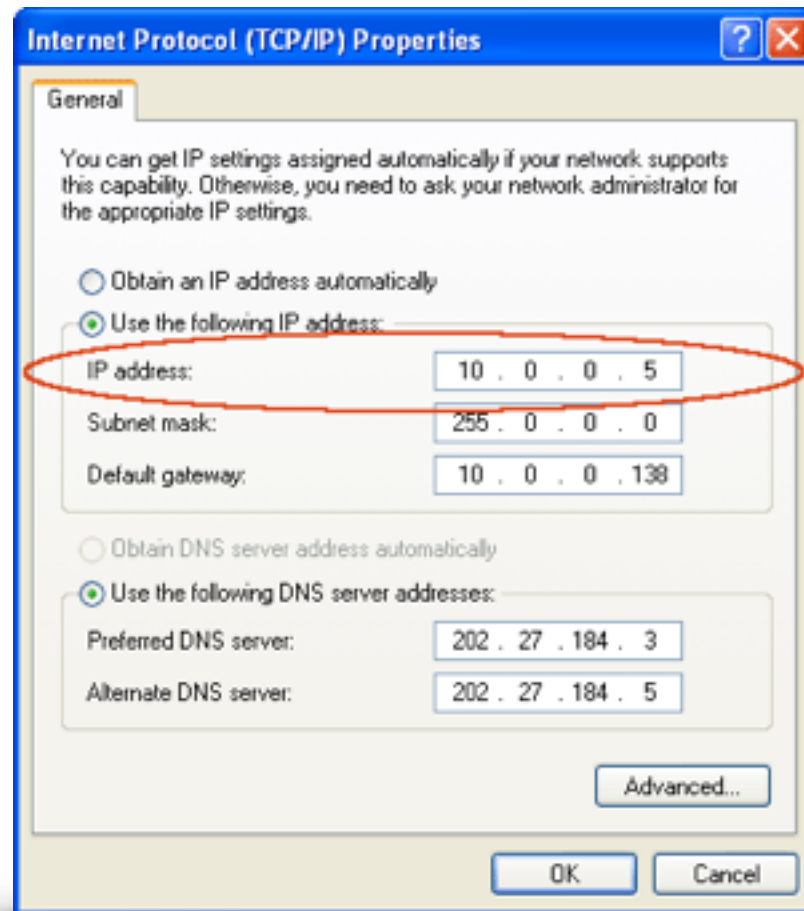
- wichtigster Schritt für Internet war Definition von Protokollen zur Kommunikation verschiedener Netzwerke miteinander
 - **Internet Protocol (IP)**
 - **Transmission Control Protocol (TCP)**

IP-Adresse

- Computer werden innerhalb eines Netzwerkes in logische Einheiten (**Subnetze**) gruppiert und können so adressiert werden
- Beispiel: **IP-Adresse** (engl. *IP-Address*) bei IPv4 mit allgemeinem Aufbau:
$$x1 . x2 . x3 . x4,$$
wobei jedes x_i mit 8-Bit repräsentiert wird, d.h. Wert zwischen 0 und 255
⇒ Datengröße: 4 Byte

Internet Protocol (IP)

Beispiel



Vergabe von IP-Adressen

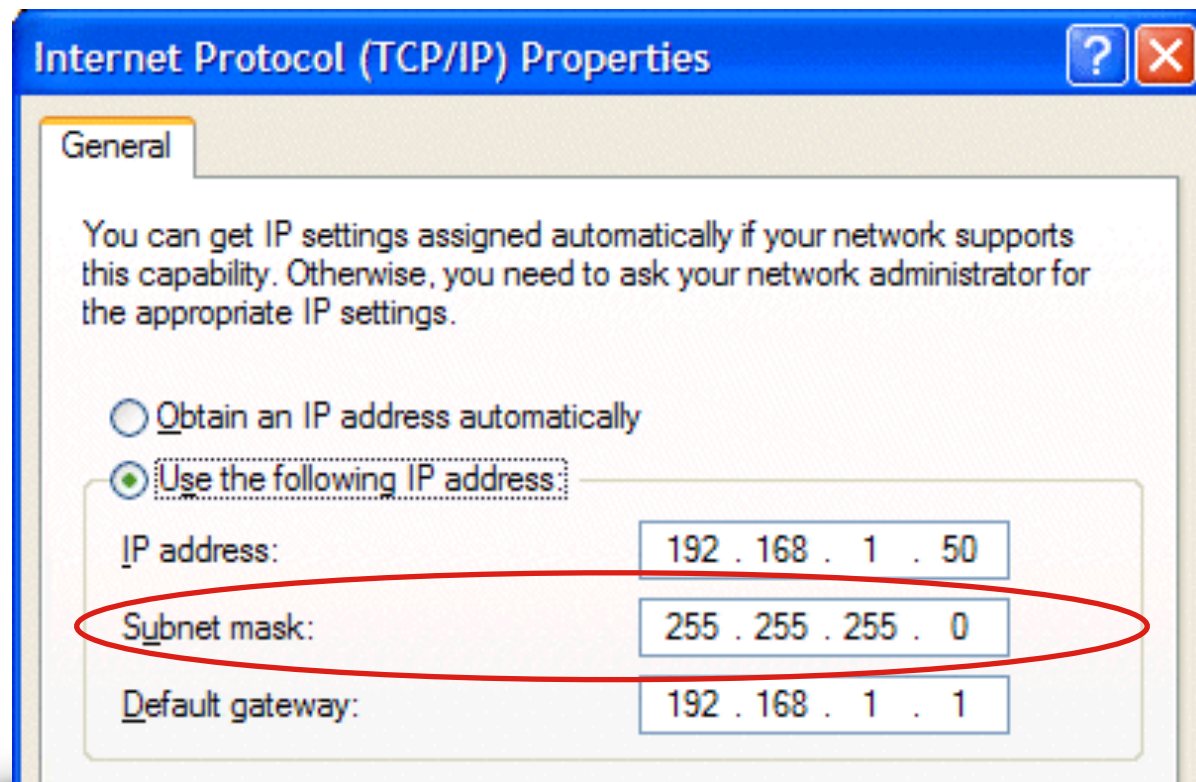
- öffentliche IP-Adressen müssen weltweit eindeutig zugeordnet werden können
 - Beispiel: 216 . 58 . 213 . 36
 - Vergabe ist durch **Internet Assigned Numbers Authority (IANA)** geregelt
- **Subnetze** können z.B. an Internetprovider vergeben werden

Lokale IP-Adressen

- Unterscheidung zwischen **internen** und **externen** IP-Adressen
 - im Internet wird mit vom Internet-Provider vergebenen **externen** IP-Adresse kommuniziert
 - im internen Netzwerk wird über **interne** IP-Adresse kommuniziert
- **Subnetzmaske** gibt Bereich der internen IP-Adressen an

Lokale IP-Adressen

Beispiele



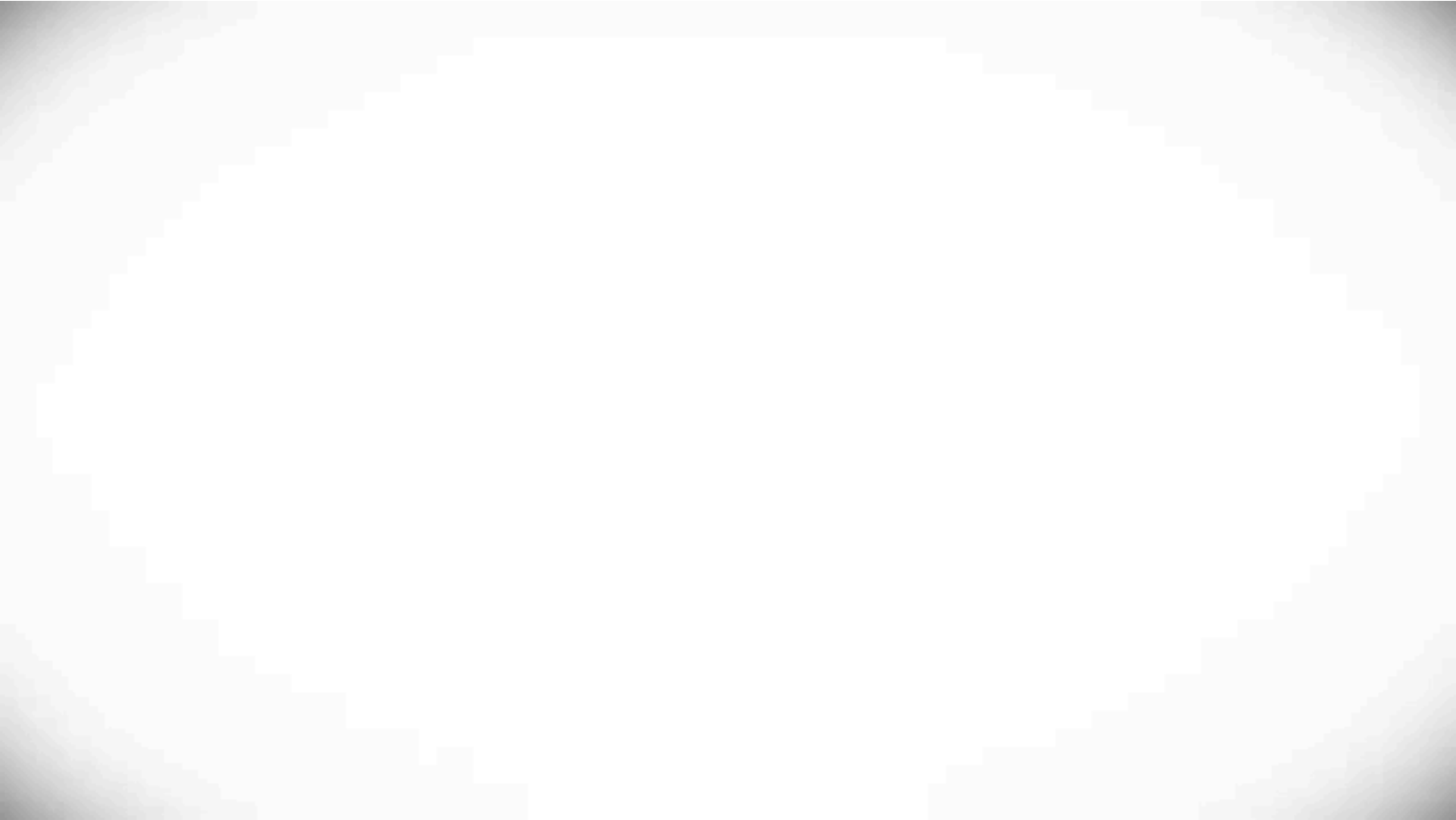
IP-Adressen: IPv6

- Zur Lösung der Knappheit von IP-Adressen vervierfacht IPv6 die Länge der IP-Adressen auf 128 Bit statt 32 Bit
- insgesamt stehen daher bei IPv6
340 000 000 000 000 000 000 000 000 000 000 000 000 000
IP-Adressen zur Verfügung

Diskussion



Wieviele im Internet erreichbare IP-Adressen
kann es bei IPv4 maximal geben?



IP-Pakete

- Informationsübertragung erfolgt durch **IP-Pakete**
- **IP-Paket** (max. 64kB) ist Grundelement der Internet-Kommunikation und besteht aus
 1. **Kopfdaten** (ca. 60 Bytes) beinhalten Informationen über Quelle, Ziel, Status, Fragmentierung, Länge etc.
 2. **Nutzdaten**

Diagnose-Werkzeug

Beispiel: Ping

- **Ping** ist Diagnose-Werkzeug, mit dem überprüft werden kann, ob bestimmter **Host** im IP-Netzwerk erreichbar ist
- **Ping** misst Zeitspanne zwischen Senden und Empfangen eines IP-Pakets (= **Paket-umlaufzeit** (engl. *Round trip time, RTT*))

```
code — bash — 77x11
mcimac11:code steinicke$ ping www.uni-hamburg.de
PING rzlinw1.rz.uni-hamburg.de (134.100.29.217): 56 data bytes
64 bytes from 134.100.29.217: icmp_seq=0 ttl=62 time=0.901 ms
64 bytes from 134.100.29.217: icmp_seq=1 ttl=62 time=0.675 ms
64 bytes from 134.100.29.217: icmp_seq=2 ttl=62 time=0.716 ms
^C
--- rzlinw1.rz.uni-hamburg.de ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.675/0.764/0.901/0.098 ms
mcimac11:code steinicke$
```

IP-Router

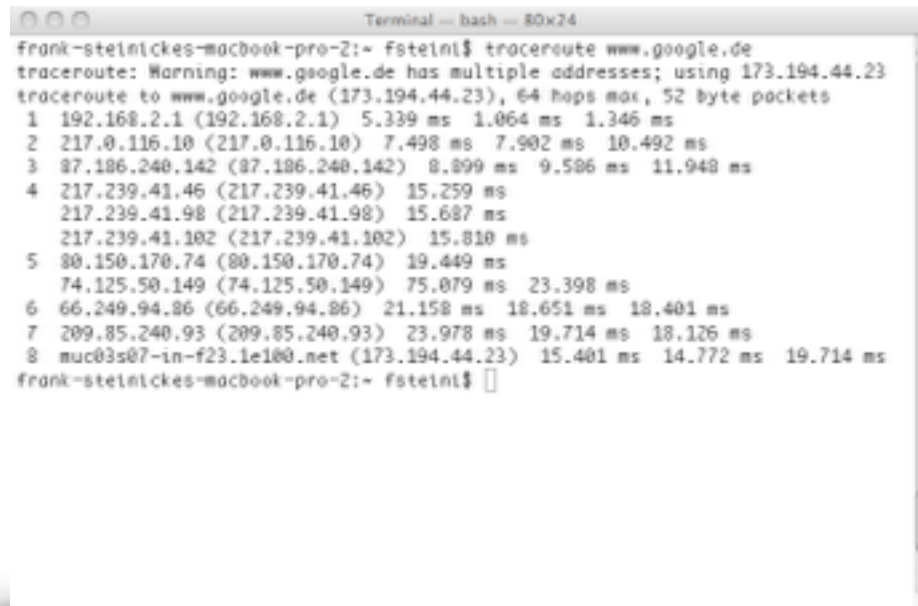
- Versand der IP-Pakete erfolgt über Netzwerkgeräte, die IP-Pakete zwischen mehreren Rechnernetzen weiterleiten, sogenannte **Verteilerknoten** (engl. *IP-Router*)
- **Absender** und **Empfänger** werden durch IP-Adressen spezifiziert



Diagnose-Werkzeug

Beispiel: Traceroute

- **Traceroute** ist Diagnose-Werkzeug, mit dem IP-Router ermittelt werden können, über die IP-Pakete vermittelt werden



```
Terminal — bash — 80x24
frank-steinickes-macbook-pro-2:~ fsteint$ traceroute www.google.de
traceroute: Warning: www.google.de has multiple addresses; using 173.194.44.23
traceroute to www.google.de (173.194.44.23), 64 hops max, 52 byte packets
 1  192.168.2.1 (192.168.2.1)  5.339 ms  1.064 ms  1.346 ms
 2  217.0.116.10 (217.0.116.10)  7.498 ms  7.902 ms  10.492 ms
 3  87.186.240.142 (87.186.240.142)  8.899 ms  9.586 ms  11.948 ms
 4  217.239.41.46 (217.239.41.46)  15.259 ms
    217.239.41.98 (217.239.41.98)  15.687 ms
    217.239.41.102 (217.239.41.102)  15.810 ms
 5  80.150.170.74 (80.150.170.74)  19.449 ms
    74.125.50.149 (74.125.50.149)  75.079 ms  23.398 ms
 6  66.249.94.86 (66.249.94.86)  21.158 ms  18.651 ms  18.401 ms
 7  209.85.240.93 (209.85.240.93)  23.978 ms  19.714 ms  18.126 ms
 8  muc03s07-in-f23.1e100.net (173.194.44.23)  15.401 ms  14.772 ms  19.714 ms
frank-steinickes-macbook-pro-2:~ fsteint$
```

Netzverbindung

- Verbindungen werden zwischen zwei Software-Modulen einer Netzverbindung (**Sockets**) hergestellt
 - **Stream Sockets** (meist über *Transmission Control Protocol, TCP*)
 - **Datagram Sockets** (meist über *User Datagram Protocol, UDP*)

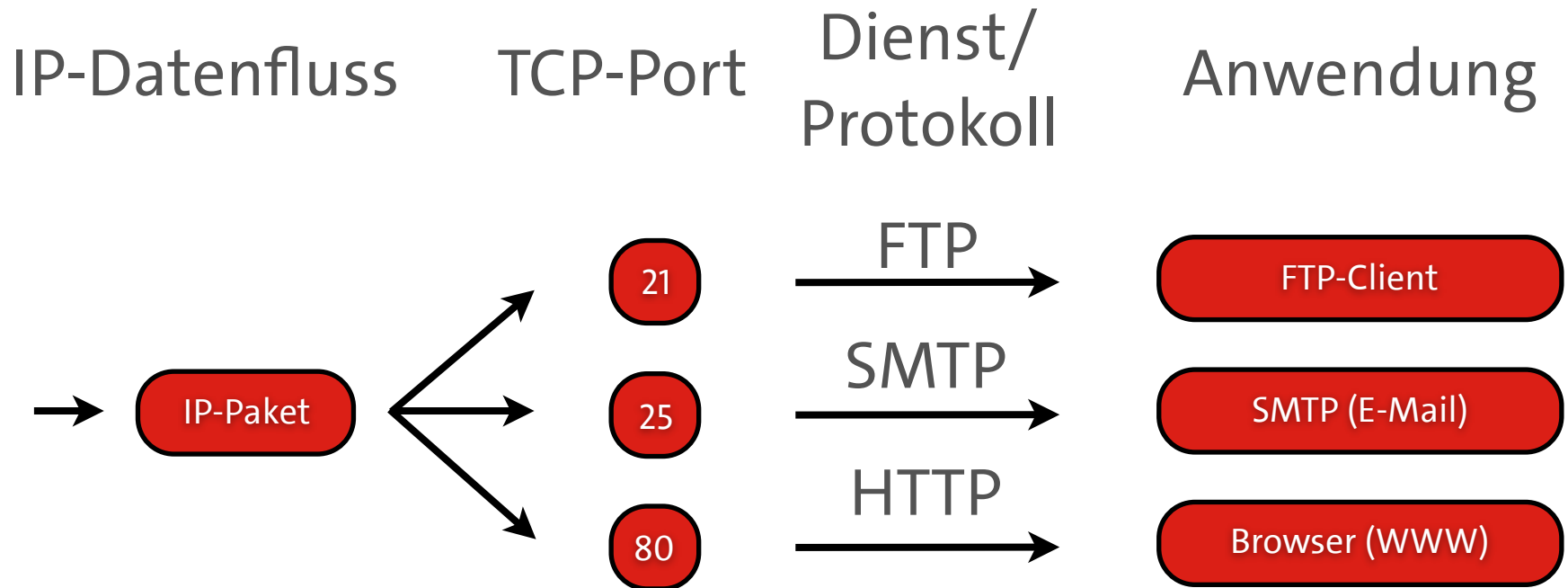
Protokolle

Beispiel: TCP

- **Transmission Control Protocol (TCP)**
vereinbart, wie Daten ausgetauscht werden
- **TCP** ist Ende-zu-Ende-Verbindung in **voll-duplex**, welche Übertragung der Informationen in beide Richtungen zur selben Zeit zulässt
- Endpunkt stellt Netzwerk-Adresse, d.h. **IP-Adresse** und **Portnummer**

TCP

Beispiel



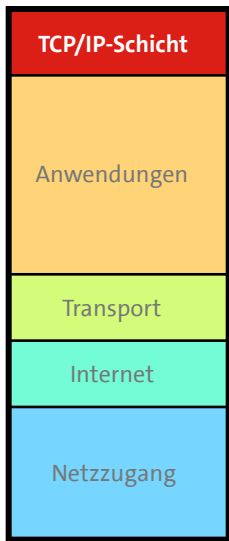
OSI-Referenzmodell

- **Open Systems Interconnection (OSI-Referenzmodell)** zerlegt Protokoll in sieben Schichten, wobei jede Schicht Funktionalität für nächsthöhere Schicht bereitstellt

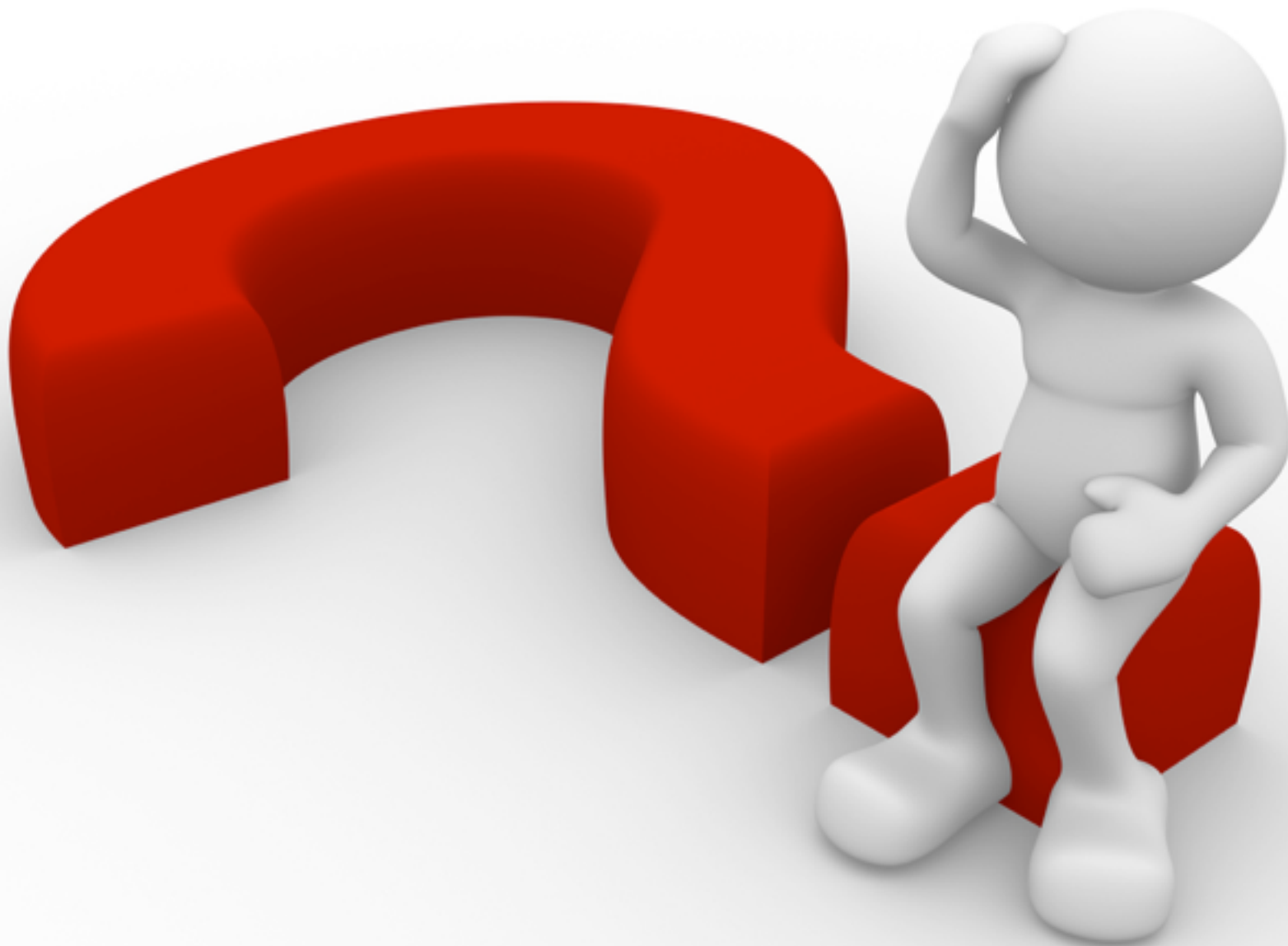
OSI-Schicht	TCP/IP-Schicht	Beispiel
Anwendungen (7)	Anwendungen	Email, Browser,...
Darstellung (6)		HTTP, FTP, ...
Sitzung (5)		RPC
Transport (4)	Transport	TCP, UDP
Vermittlung (3)	Internet	IP (IPv4, IPv6)
Sicherung (2)	Netzzugang	Ethernet, 802.11 (WLAN)
Bitübertragung (1)		

OSI-Referenzmodell

TCP/IP-Schicht



- **Anwendungsschicht** umfasst Protokolle der Anwendungsprogramme
- **Transportschicht** stellt Ende-zu-Ende-Verbindung her
- **Internetschicht** ist für Weitervermittlung von Paketen und Wegwahl (***Routing***) zuständig
- **Netzzugangsschicht** ist Platzhalter für verschiedene Datenübertragungstechniken





Interaktive Medien

Kapitel Mediensysteme & World Wide Web

World Wide Web

World Wide Web (WWW)

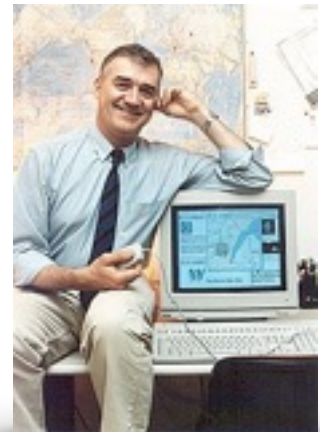
- **World Wide Web** ist über Internet abrufbares System von elektronischen Hypertext-Dokumenten



- Internet ist Basistechnologie für das WWW

Gründer des WWW

- **Sir Timothy John Berners-Lee**
 - britischer Physiker und Informatiker
- **Robert Cailliau**
 - belgischer Informatiker



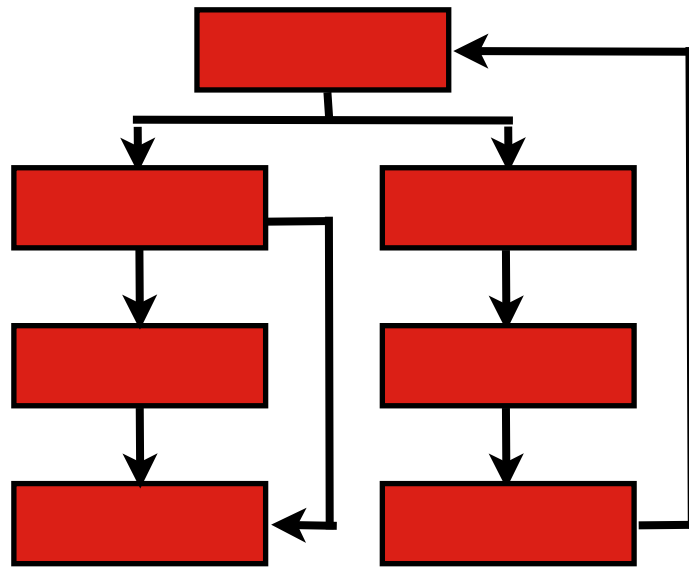
Hypertext

- **Hypertext** ist Text, der mit netzartiger Struktur von Informationsblöcken durch Querverweise (**Hyperlinks**) verknüpft
- **Hypertext** wird zu nichtlinearem Text, der in beliebiger Reihenfolge abrufbar ist

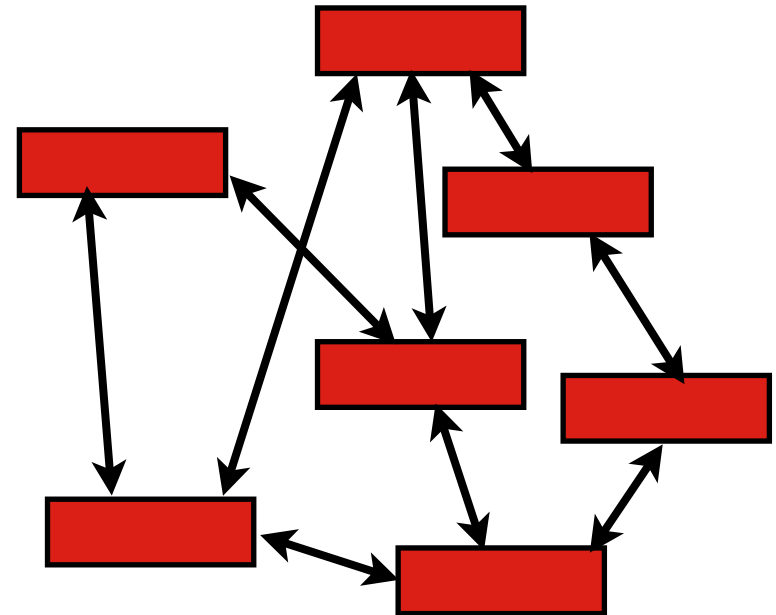
Hypertext

Beispiel

Verzweigte lineare Struktur



Hypertext-Struktur

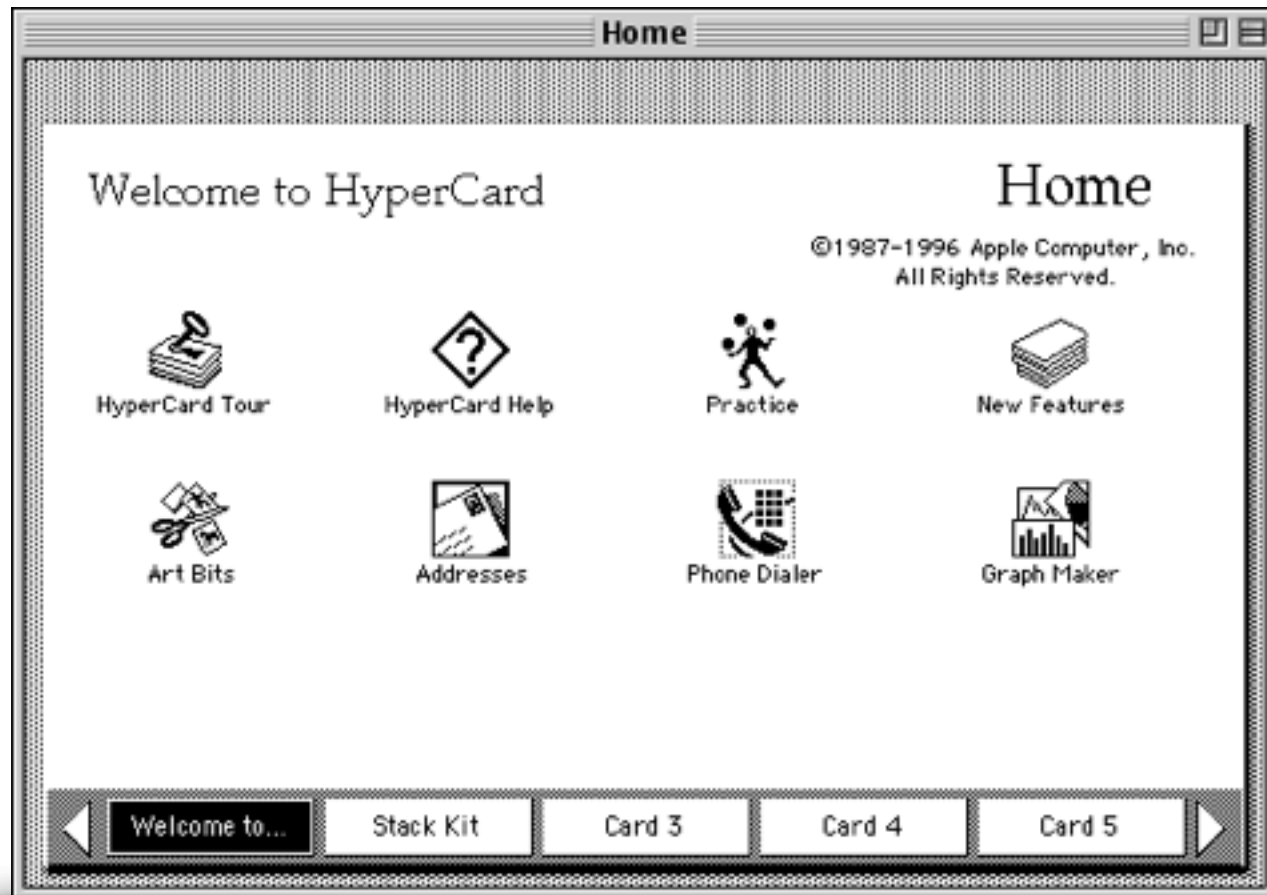


Hypermedia

- Knoten können neben Text auch **Multimedia-Dokumente** beinhalten
- Bsp: **HyperCard** (Karteikartensystem, Karten enthalten verschiedene Medientypen)
 - Autorensystem zur visuellen Programmierung
 - Vorgänger von aktuellen Multimedia-Autorensystemen (z.B. Adobe Flash)

Hypermedia

Bsp: HyperCard



WWW

Standardisierungsgremien

- **International Organization for Standardization (ISO):**
grundlegende Standards, z.B.
Zeichensätze
- **International Telecommunication Union (ITU):** grundlegende
Standards zur Datenübertragung
und Multimedia-Standards



WWW

Standardisierungsgremien

- **Internet Engineering Task Force (IETF):** Technologie des Internets, z.B. TCP/IP und HTTP
- **World Wide Web Consortium (W3C):** Standardisierung von Technologien für das WWW, z.B. XML, SVG, X3D, HTML

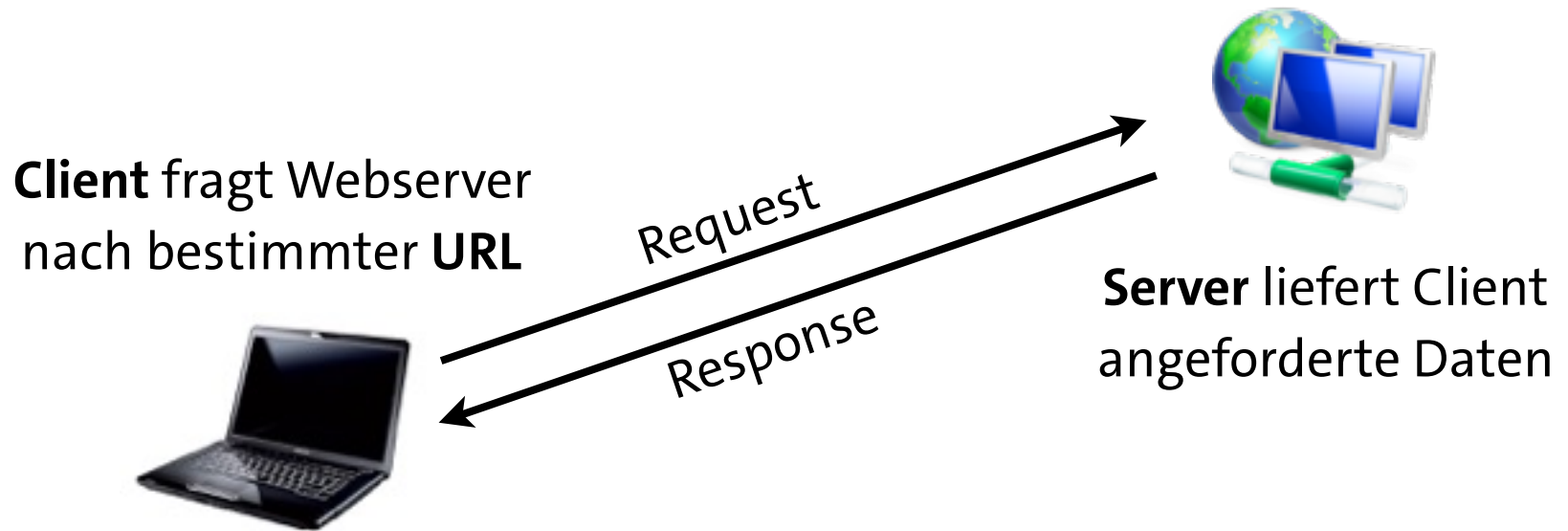


HTTP

- **Hypertext Transfer Protocol (HTTP)** beschreibt Art und Weise der Übertragung von Webseiten
- Grundprinzip:
 - Identifizierung über **Domain Name Service (DNS)**
 - Datenverbindungsaufbau über **TCP**
 - Datenaustausch (IP-Pakete) über **Requests** und **Responses**

HTTP

Beispiel



URL

- **Uniform Resource Locator (URL)** ist Adresse der zu ladenden Daten auf Webserver
- Mischung verschiedener Informationen in Adressleiste
 - Allgemeine Zusammensetzung:
`protokoll://hostname/dateiname`

URL

Beispiel



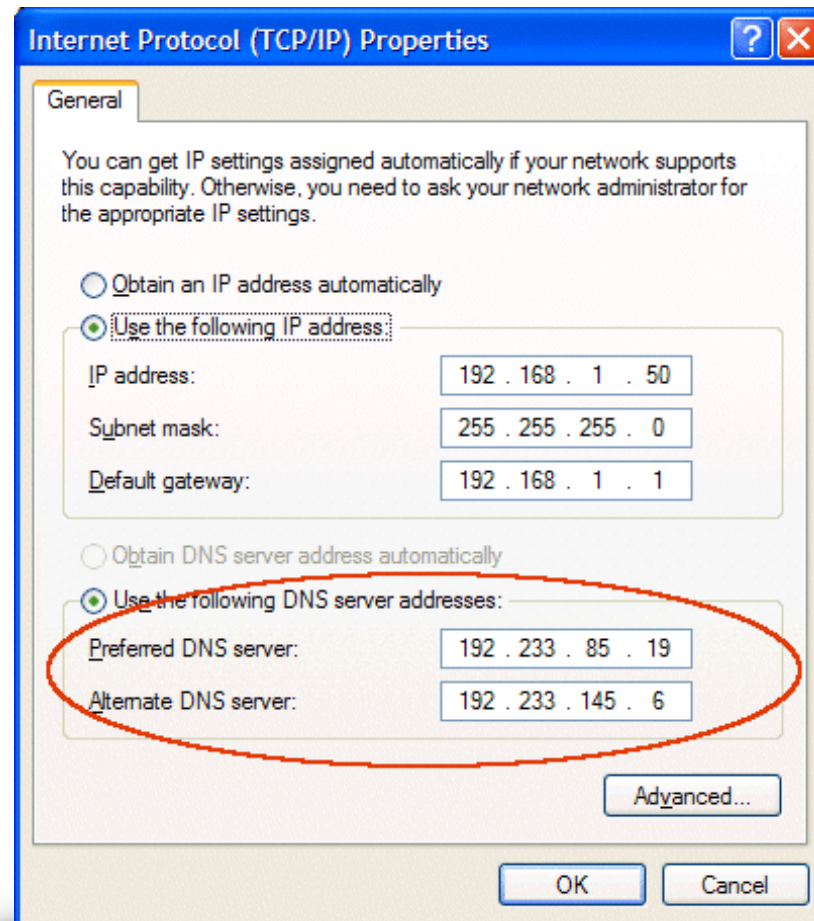
- **Protokollname** (z.B. `http://`, `ftp://`)
- **Hostname/domain** (hier: `de.wikipedia.org`)
- **Pfadname** (hier: `/wiki/`)
- **Dateiname** (hier: `Wikipedia:Hauptseite`)

Domain Name Service

- **Domain Name Service (DNS)** ist verantwortlich für Auflösung von Hostnamen, die sich leichter merken lassen als IP-Adressen
- Beispiel: DNS liefert auf Anfrage mit Hostnamen `www.example.org` als Antwort zugehörige IP-Adresse: `192.0.43.10`

Domain Name Service

Beispiel

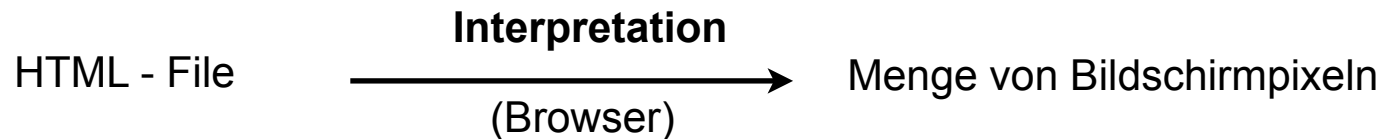


(Web-)Browser

- **(Web-)Browser** sind spezielle Computerprogramme zur Darstellung von Webseiten im World Wide Web oder allgemein von Dokumenten und Daten



(Web-)Browser



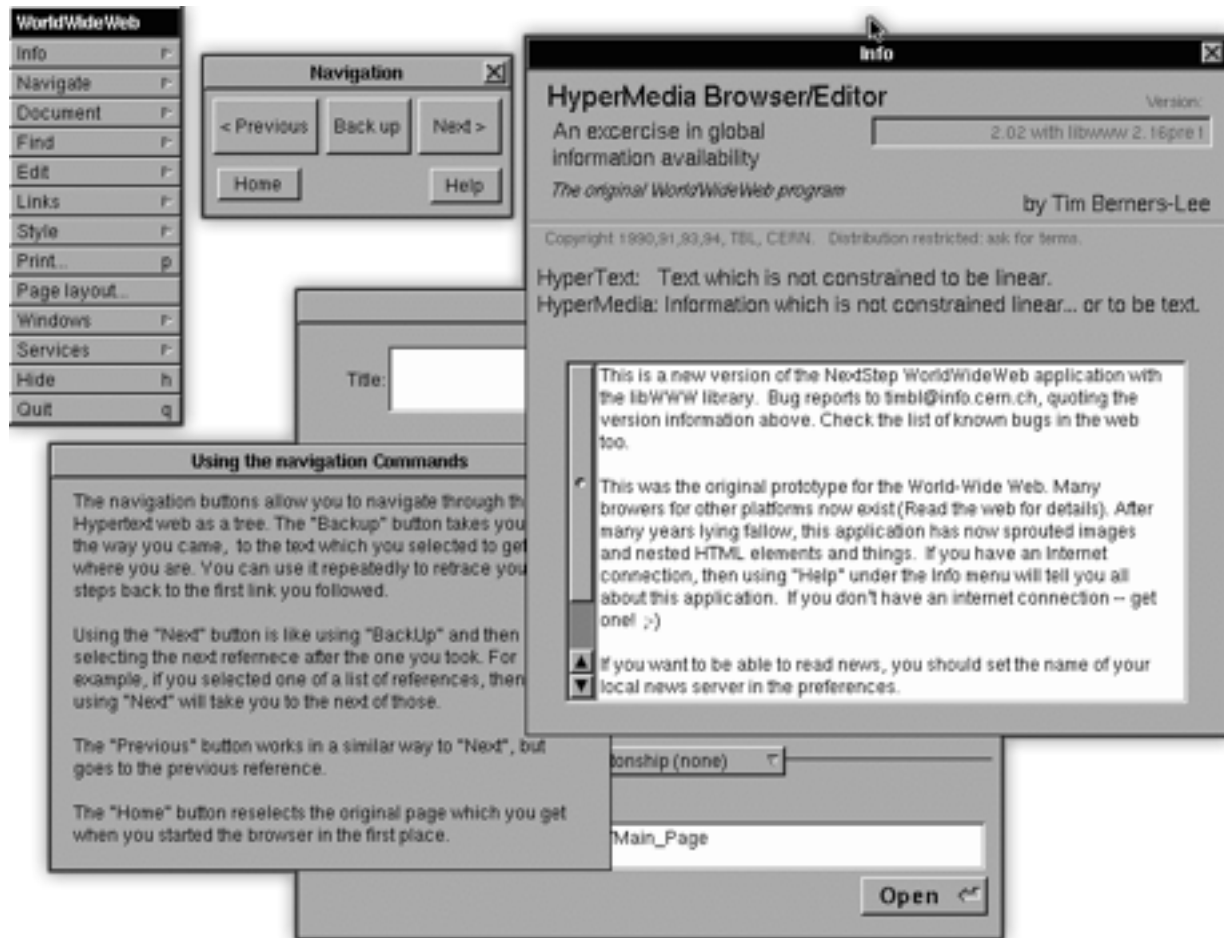
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<meta name="Copyright" content="Copyright Statement" />
<title>Title of the document</title>
<meta name="Description" content="Your description" />
<meta name="Keywords" content="first, second, third" />
<link rel="stylesheet" type="text/css" href="stylesheet.css" />
<script type="text/javascript">
    <!--Your script-->
</script>
</head>
<body>

</body>
</html>
```

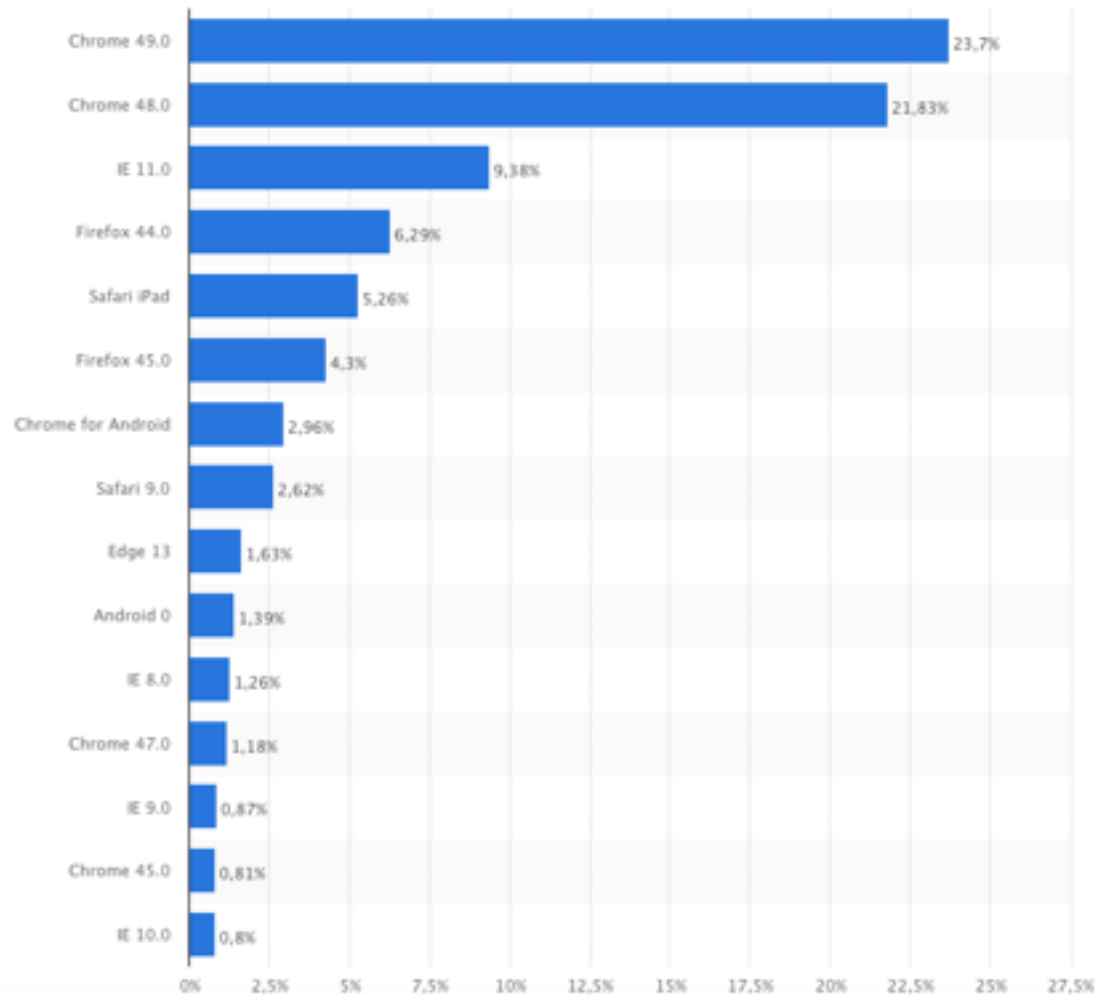


(Web-)Browser

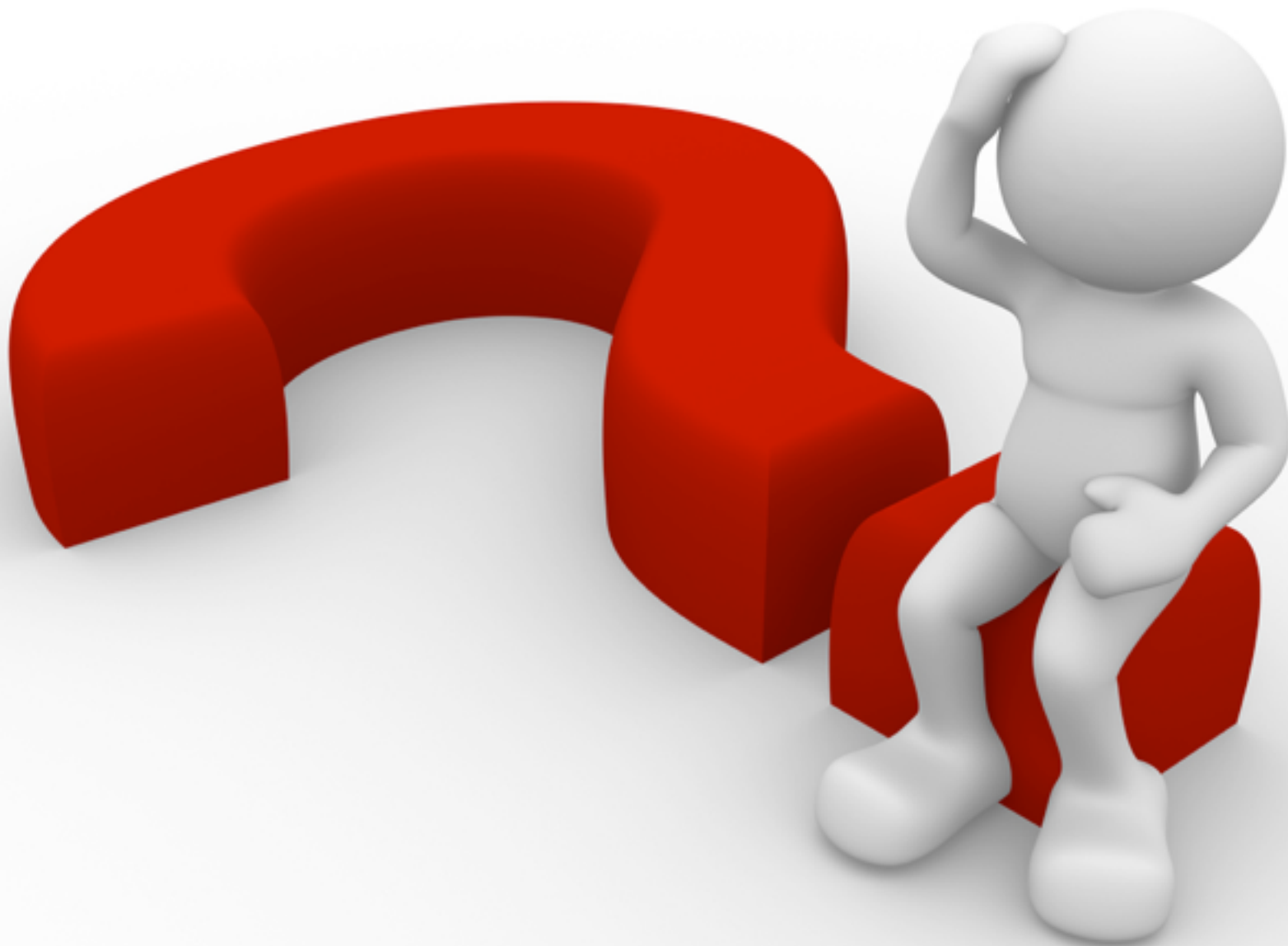
Bsp: WorldWideWeb (1990)



(Web-)Browser



<http://de.statista.com>





Interaktive Medien

Kapitel Mediensysteme & World Wide Web

Allgemeine Webdokumente

Markup-Sprache

- **Markup**-Sprache (engl. *markup* = Auszeichnung)
 - **Auszeichnungen** sind Ergänzungen im Text, die Eigenschaften, Zugehörigkeiten und Darstellungsformen eines Bereichs beschreiben
- Beispiel: `Hallo` im HTML-Code ergibt **Hallo** in Ausgabe

XML

- **Extensible Markup Language (XML)** ist Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdateien
- Einsatz für vielfältige Zwecke, z.B. Betriebssysteme, Formulierung von Wissen in diversen Fachdisziplinen...

XML

- **Familie von Sprachen** umfasst mehrere konkrete Sprachen
- konkrete Sprachen legen jeweils **zusätzlich** fest, welche Elemente und Attribute es gibt und wie diese **kombiniert** werden können (für HTML z.B. `html`, `head`, `body`, `p`, ...)

XML

Syntax

- Generische Auszeichnungssprachen verwenden **ähnlichen Syntax**
- Beispiel: Syntax von HTML ähnlich zu SVG (spitze Klammern, Element-Attribut-System ...)

XML

Syntax

- ähnlicher Syntax ist sinnvoll, weil
 - Sprachkonstrukte in andere Sprachen **übernommen** werden können (z.B. Links werden in vielen Sprachen benötigt)
 - **Editoren** für Sprachfamilie entworfen werden (und nicht nur für eine Sprache)
 - **Arbeitersparnis** bei Entwicklung neuer Sprachen in dieser Familie möglich ist

XML

Beispiel: Postadresse

```
<adresse>
  <namensangabe>
    <vorname>Max</vorname>
    <nachname>Mustermann</nachname>
  </namensangabe>
  <strassenangabe>
    <strasse>Musterstrasse</strasse>
    <hausnr>1</hausnr>
  </strassenangabe>
  <ortsangabe>
    <plz>12345</plz>
    <ort>Musterstadt</ort>
  </ortsangabe>
</adresse>
```

Sprachebenen

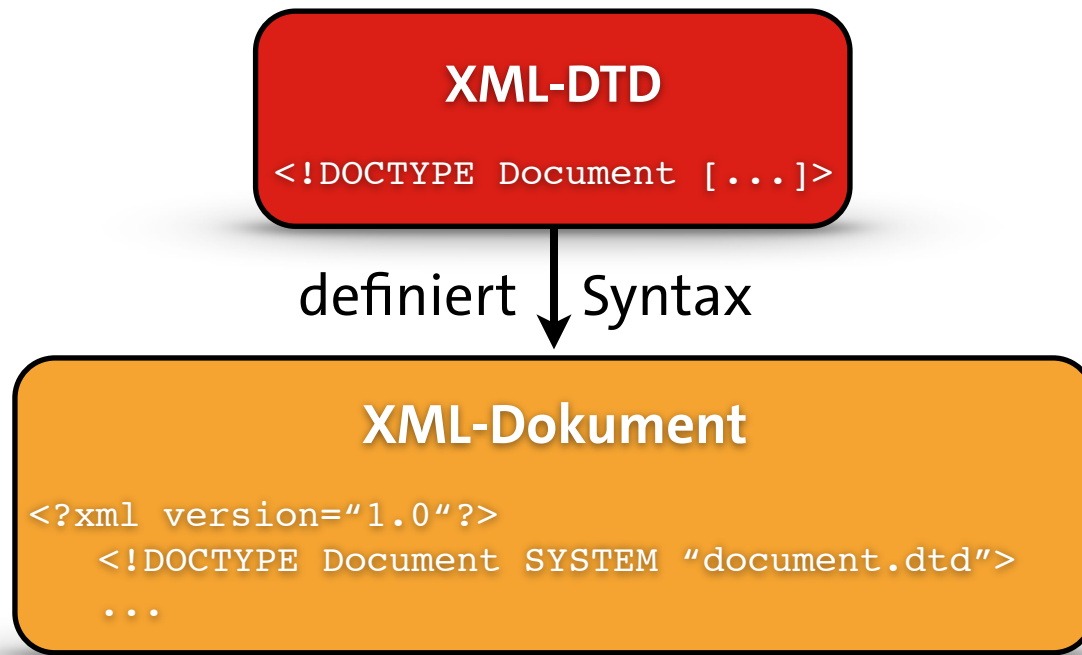
Sprachdefinition (Metaebene)	Sprach- verwendung
<ul style="list-style-type: none">• Festlegung der Elemente (z.B. adresse, vorname, strasse, plz, ...)• Reihenfolge und Position der Elemente	<ul style="list-style-type: none">• konkrete Angaben der Elemente• Beispiel: Musterstadt für ort

Dokumenttypdefinition

- **Dokumenttypdefinition** ist Satz an Regeln, der benutzt wird, um Dokumente eines bestimmten Typs zu deklarieren
- **Document Type Definition (DTD)** ist eigene Sprache zur Definition von **Dokumenttypen** auf Metaebene in XML

Dokumenttypen

Beispiel



DTD

Beispiel

- Element-Deklaration von „ort“:

`<!Element ort (#PCDATA)>`

- `#PCDATA` (*parsed character data*) besagt, dass beliebige Zeichenkette folgen darf
- beachten: `#PCDATA` beinhaltet **keine** Sonderzeichen der XML-Syntax, z.B. “<“, “>”

DTD

Beispiel: Postadresse

```
<!Element adresse (namensangabe, strassenangabe,  
                    ortsangabe)>  
<!Element namensangabe (vorname?, nachname)>  
<!Element vorname (#PCDATA)>  
<!Element nachname (#PCDATA)>  
<!Element strassenangabe (strasse, hausnr)>  
<!Element strasse (#PCDATA)>  
<!Element hausnr (#PCDATA)>  
<!Element ortsangabe (plz, ort)>  
<!Element plz (#PCDATA)>  
<!Element ort (#PCDATA)>
```

XML

Beispiel: Postadresse

```
<adresse>
  <namensangabe>
    <vorname>Max</vorname>
    <nachname>Mustermann</nachname>
  </namensangabe>
  <strassenangabe>
    <strasse>Musterstrasse</strasse>
    <hausnr>1</hausnr>
  </strassenangabe>
  <ortsangabe>
    <plz>12345</plz>
    <ort>Musterstadt</ort>
  </ortsangabe>
</adresse>
```

DTD

Vorschriften

- **Optionales Element:**
gekennzeichnet durch „?“
- **Alternative** zwischen zwei Möglichkeiten:
gekennzeichnet durch „|“
- **Wiederholung** eines Unterelements:
gekennzeichnet durch „+“ oder „*“, d.h.
“+” = Element ist 1 bis n mal vorhanden
“*” = Element ist 0 bis n mal vorhanden

DTD

Alternative & Wiederholung

```
<!Element adresse (namensangabe,  
                    addresszeile+,  
                    ortsangabe)>  
<!Element addresszeile (einrichtung |  
                          strassenangabe)>  
<!Element einrichtung (#PCDATA)>
```

Attribute

- Beschreibung von Attributen für Elemente

optional



```
<!--ATTLIST ort bundesland CDATA #IMPLIED-->
```

```
<!--ATTLIST ort bundesland CDATA #REQUIRED-->
```



verpflichtend

- Beispiel:

```
<ort bundesland="Hamburg">Hamburg</ort>
```


Datentypen

- Beliebige Zeichenkette (CDATA)
(*character data*)
- Eindeutige Bezeichner (ID)
<!ATTLIST eintrag name ID #REQUIRED>
- Verweise auf Bezeichner (IDREF)
<!ATTLIST eintrag elterneintrag
IDREF #REQUIRED)

Datentypen

ID und IDREF

```
<eintrag  
  name="wurzel">Inhalt  
</eintrag>  
<eintrag  
  name="einleitung"  
  elterneintrag="wurzel">Einleitung  
</eintrag>
```

XML-Syntax

- DTD beschreibt nur **Struktur** und **Grammatik** von XML-Dokumenten
- XML-Dokumente müssen sich an weitere **Regeln** halten

XML-Syntax

Regeln

- XML-Dokument beginnt mit **Präambel**, z.B.
`<?xml version="1.0"
encoding="ISO-8859-1" ?>`
- XML-Dokument enthält nur XML-Elemente
- XML-Dokument hat genau ein XML-Element als Beginn („**Wurzelement**“)

XML-Syntax

Regeln

- öffnende Element müssen **geschlossen** werden (inhaltlich **leere** Elemente werden als `<element/>` dargestellt)
- Elemente müssen direkt geschachtelt sein
 - d.h. `<a>.........` ist **nicht** erlaubt
- Attributwerte müssen in doppelten Anführungszeichen angegeben werden

XML-Syntax

Regeln

- jedes Attribut darf beim selben Element nur **einmal** angegeben werden
- Groß- und Kleinschreibung wird unterschieden

XML-Syntax

Wohlgeformtheit

- XML-Dokument, dass vorherige Regeln befolgt, wird als **wohlgeformt** (engl. ***well-formed***) bezeichnet
- **Parser-Programme** können Wohlgeformtheit überprüfen

XML-Syntax

Gültigkeit

- hat XML-Dokument zusätzlich eine DTD und entspricht dieser, wird es als **gültig** (engl. *valid*) bezeichnet
- Parser sind meist schon in XML-Editoren integriert

Namensräume

- XML-Sprachen können auch in anderen XML-Dokumenten verwendet werden
- XML-Dokument kann Elemente verschiedener XML-Sprachen beinhalten, z.B. Links

Namensräume

- Elemente gehört einem **Namensraum** (engl. *name space*) an
- Basis ist immer der **Standardnamensraum**
- Elemente anderer Namensräume müssen mit **Präfix** gekennzeichnet werden
- Allgemein:
`<nr:element>`

Namensräume

Beispiel: Links

- **SVG**-Dokument soll **Link** nutzen
(*xlink*-Standard)
- Standardnamensraum ist Namensraum von SVG
- Einbinden von *xlink* an entsprechenden Stellen
`<use xlink:href...>`

Namensräume

Beispiel: Links

- Code:

```
<svg xmlns="http://www.w3.org/2000/svg"  
      xmlns:xlink="http://www.w3.org/1999/xlink">
```

 - erste Zeile:
Standardnamensraum SVG wird gesetzt
 - zweite Zeile:
Verweis auf Namensraum von `xlink`
- `xlink`-Definitionen können nun im Dokument verwendet werden

XML-Sprachen

Beispiele

- Vektorgrafikformat **SVG** wird von meisten Browsern unterstützt
- 3D-Vektorgrafikformat **X3D** zum Austausch von 3D-Szenenmodellen
- Multimedia-Dokumentformat **SMIL** (Synchronized Multimedia Integration Language)



XML-Sprachen

Beispiele

- **VoiceXML** zur Dialogbeschreibung von Sprachdialogsystemen (Telefonnavigation)
- Nachrichtenquellen, die als **RSS Feeds** abgelegt werden
- moderne **Podcasts** sind i.d.R. als RSS Feeds realisiert

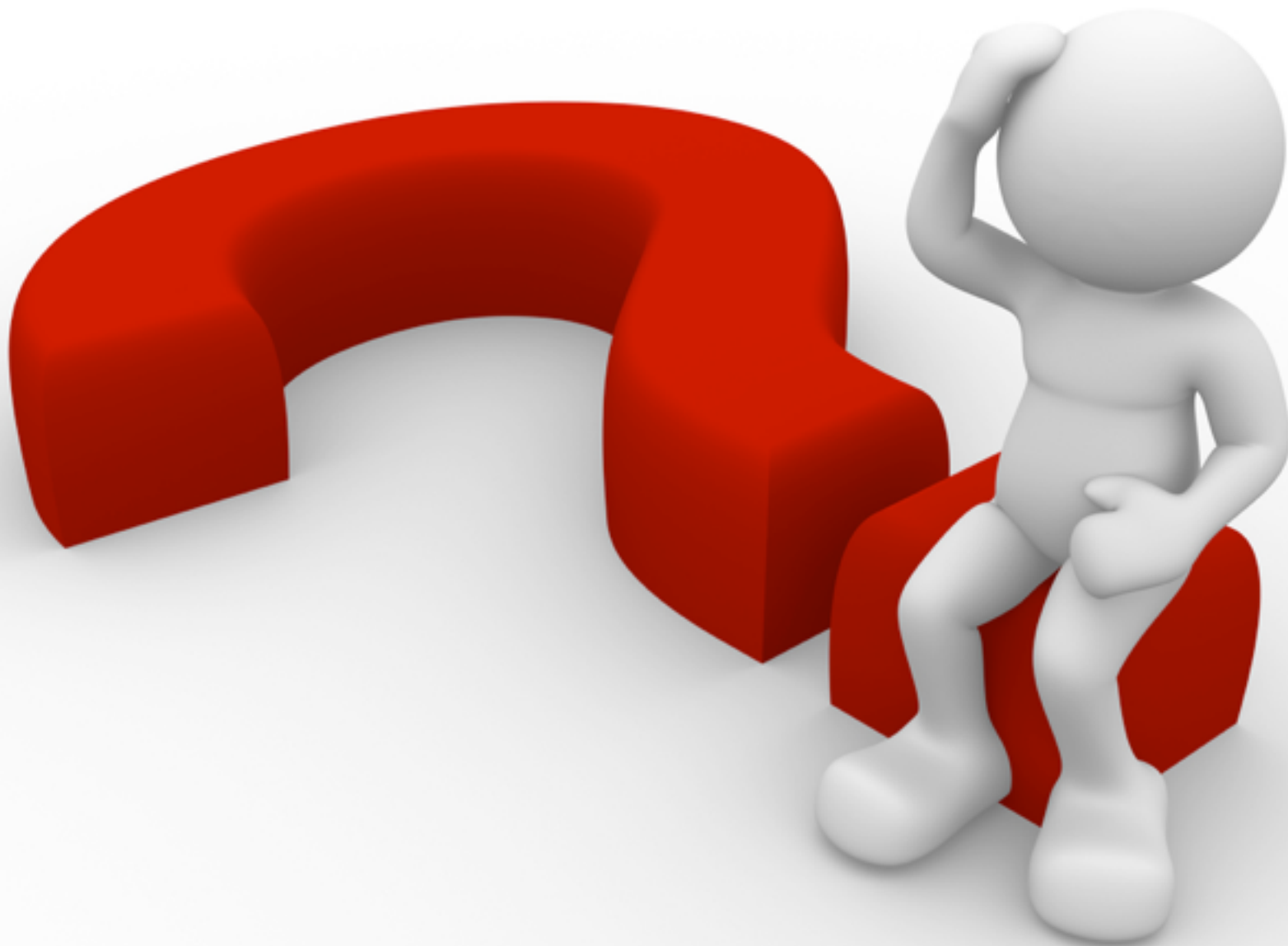


XML-Sprachen

Beispiele

- Konfigurationsdateien in Programmsystemen, z.B. Mac OS X
- Dokumentablage von Office-Programmpaketen, z.B. Microsoft Office
- XML-Dateien können als Baumansicht in Browser-Programmen angezeigt werden







Interaktive Medien

Kapitel Mediensysteme & World Wide Web

Erstellung von Webseiten

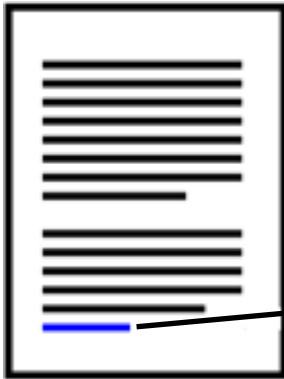
HTML

- **HTML (Hypertext Markup Language)** ist Kerntechnologie für das WWW
- **HTML** ist textbasierte Auszeichnungssprache zur semantischen Auszeichnung von Inhalten wie Texten, Bildern und Hyperlinks
- **HTML** erlaubt geräteunabhängige Mediendateien zu erstellen, die über einheitliches **Verweissystem** (engl. *Links*) verbunden werden können

HTML

Beispiel

Textdokument 1



Textdokument 3



Verweis (Link)

Textdokument 2



Mediendatei



Verweis (Link)

HTML

- Stetiges Wachstum der Dokumente im **Indexable Web** (Teil des Web, der von Suchmaschinen erfasst wird)
- Tatsächlich verfügbare Informationsmenge aufgrund von dynamischen Webseiten ist viel größer (wird teilweise als **Deep Web** bezeichnet)

Geschichte von HTML

- **Generic Markup Language (GML)** war erster Vorläufer von HTML (1969) (IBM: Goldfarb, Mosher, Lorie)
 - Beinhaltet bereits Möglichkeit, **Dokumenttypen** zu definieren und zu verfassen
- Auszeichnungen wurden in spitzen Klammern (“<“ und “>”) definiert

Geschichte von HTML

- 1978: GML wurde als ISO-Standard verabschiedet → Ergebnis: **Standard Generic Markup Language (SGML)**
- 1989: Berners-Lee und Cailleau entwickelten darauf basierend HTML
- 1990: Anzeigeprogramm für HTML namens „**WorldWideWeb**“ (erster Browser)
- Stufenweise Weiterentwicklung von HTML (aktuell Version 5)

HTML

Beispiel

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<meta name="Copyright" content="Copyright Statement" />
<title>Title of the document</title>
<meta name="Description" content="Your description" />
<meta name="Keywords" content="first, second, third" />
<link rel="stylesheet" type="text/css" href="stylesht.css" />
<script type="text/javascript">
    <!--Your script-->
</script>
</head>
<body>

</body>
</html>
```

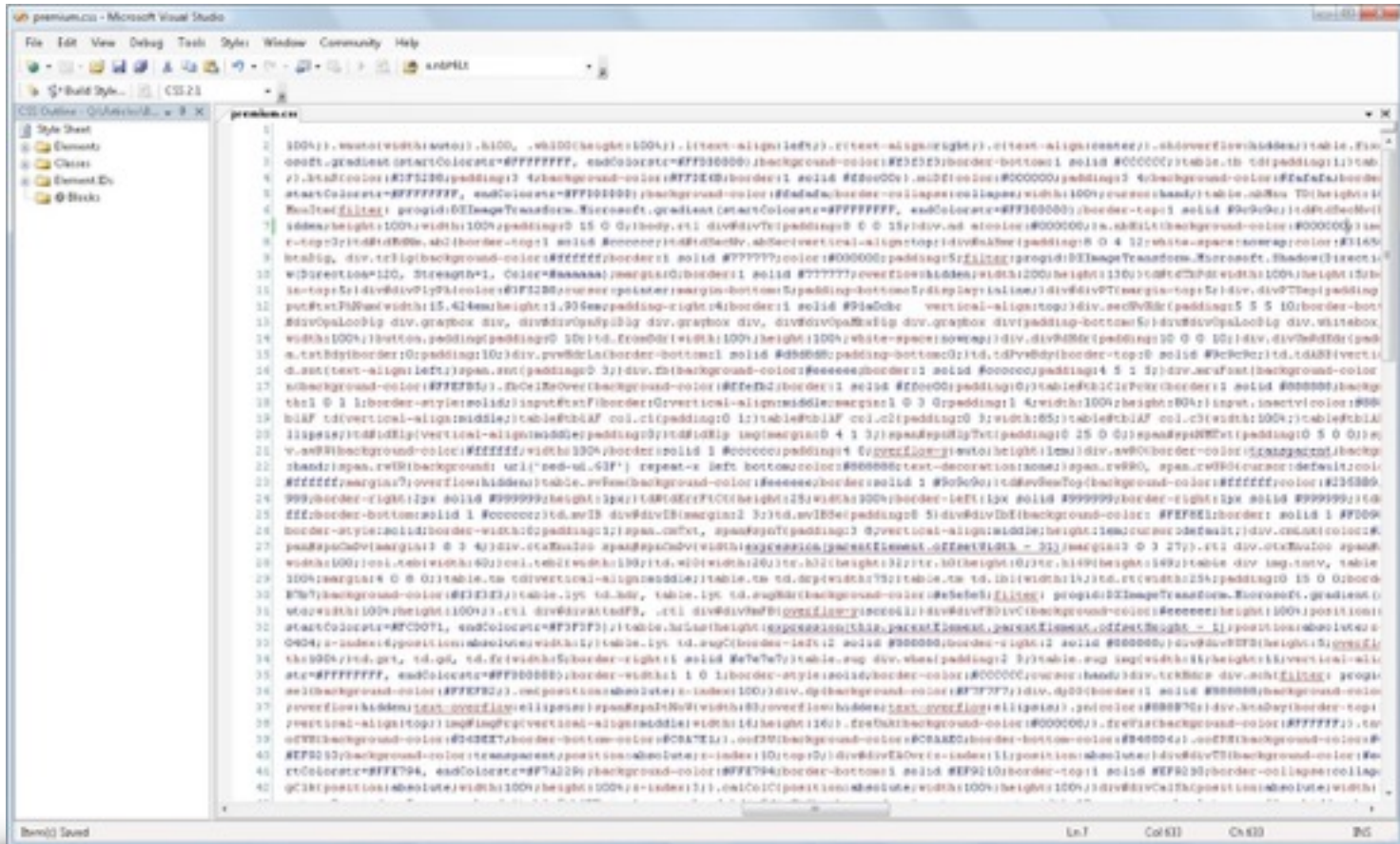
Warum HTML lernen?

- HTML-Seiten können mit Hilfe von Autorenwerkzeugen automatisch erzeugt werden
- dennoch sorgen HTML-Kenntnisse für:
 - grundlegendes Verständnis von Mediensystemen im Internet
 - einfachere Fehlerquellenfindung
 - Verständnis des von fortgeschrittenen Programmen erzeugten HTML-Codes



Autorenwerkzeuge

Beispiel: Word



HTML

- **Markup**-Sprache (engl. *markup* = Auszeichnung)
 - Auszeichnungen sind Ergänzungen im Text, die Formatierung und Struktur eines Bereichs genauer beschreiben

- Beispiel:

`<p>Hallo</p>`

im HTML-Code ergibt “Hallo” in Ausgabe

Grundlagen

- **Elemente** (engl. *Tags*) werden in spitzen Klammern und klein geschrieben:

`<elementname>`

- Beispiele:
 - Elementname `br` (= *break*, beginnt neue Zeile)
 - Elementname `b` (= *bold*, Fettdruck)
 - ...

Grundlagen

- Unterscheidung **alleinstehende** und **paarweise** auftretende Elemente, die Beginn und Ende markieren
 - Alleinstehende Elemente:
`
`
 - Paarweise auftretende Elemente:
`fett gedruckter Text`

Attribute

- Zusätzliche Informationen für Elemente, z.B. die Definition des Elements als Link mit a für *anchor* (dt. **Anker**)
- **Attributwert** in Anführungszeichen
- Beispiel:

```
<a href="http://  
www.example.de">Link</a>
```

Besonderheiten

- Formatierung des HTML-Dokuments hat keinen Einfluss auf Ausgabe, sondern Auszeichnungselemente sind entscheidend
- Text, der nicht in Ausgabe erscheinen soll, ist **Kommentar**

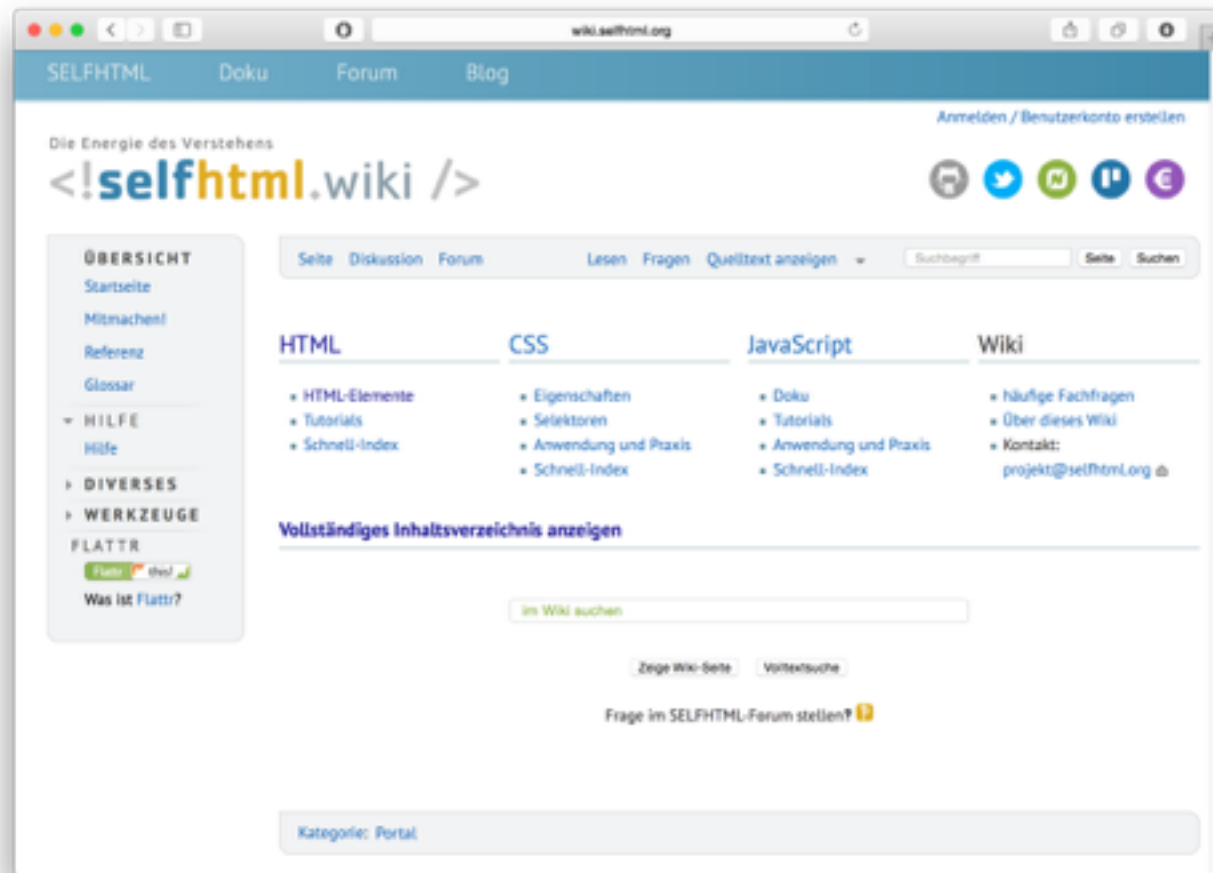
```
<!-- HTML-Kommentar -->
```

Besonderheiten

- **Sonderzeichen** werden bei ASCII/ANSI-Kodierung nicht ausgeschrieben, sondern sollten codiert werden
- Beispiele:
 - „ä“ = `ä` „Ä“ = `Ä`
 - „ö“ = `ö` „Ö“ = `Ö`
 - „ß“ = `ß`
- Weitere Informationen und Beispiele:
SELFHTML, W3Schools, MDN

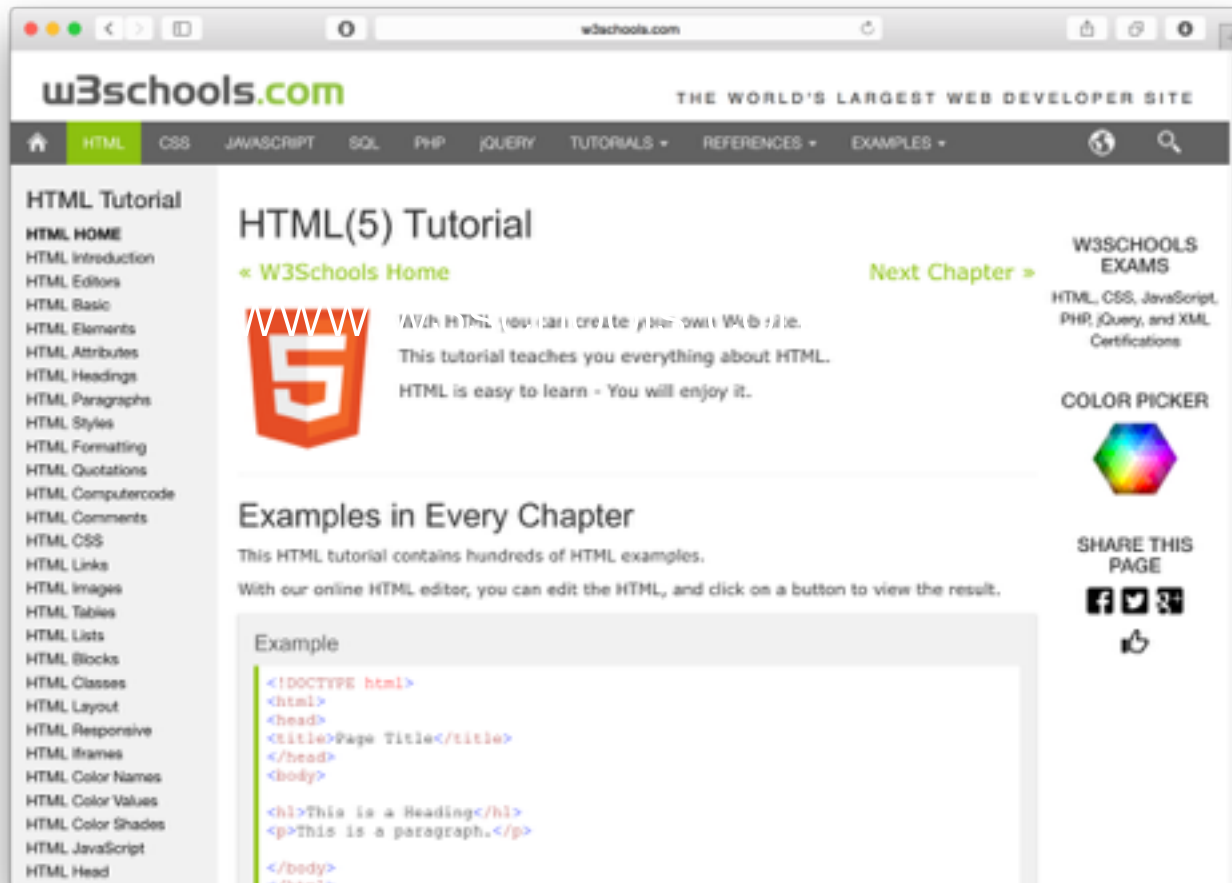
Dokumentation

SELFHTML



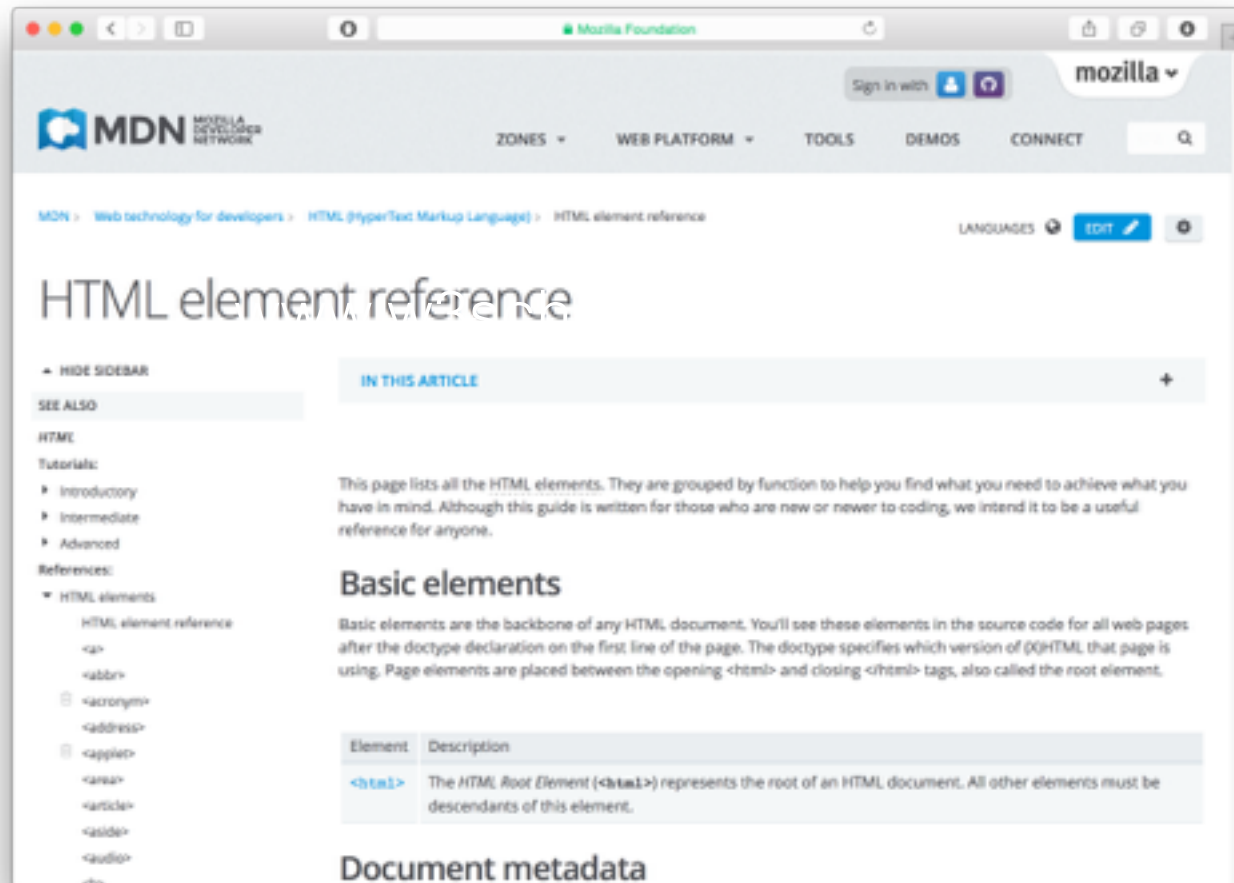
Dokumentation

W3Schools



Dokumentation

MDN



Struktur

- Beginnt i.d.R. mit Kopfzeile, dem **DOCTYPE**, der HTML-Version festlegt
- Für HTML5:

```
<!DOCTYPE html>
```

Struktur

- Hauptteil wird in HTML-Element eingeschlossen
- Dokument wird in zwei Teile geteilt:
Kopf- und **Rumpfbereich**
- Kopfbereich enthält immer Titel, der in Kopfzeile des Browser-Fensters angezeigt wird:

```
<title>Der Titel</title>
```

Struktur

`<html>`

allgemeine Definition
der HTML-Datei

`<head>`

Kopf der HTML-Datei beinhaltet Titel sowie
Beschreibung des Inhalts

`<body>`

beinhaltet Inhalt, d.h. Struktur und
Aussehen der HTML-Datei

Metainformationen

- **Metainformation** beschreiben Dokument genauer
 - z.B. Autor, letztes Änderungsdatum, Suchbegriffe ...
- **Metainformation** werden im Header spezifiziert und somit nicht dargestellt, aber von Suchmaschinen ausgewertet

Metainformationen

- Allgemeiner Aufbau:
`<meta name="Bezeichnung"
content="Inhalt">`
- Beispiel für Autor:
`<meta name="author"
content="Frank Steinicke">`
- Analog für andere Informationen, z.B.
Inhaltsbeschreibung (*description*),
Schlüsselwörter (*keywords*) ...

Übersicht

DOCTYPE

```
<!DOCTYPE html>
```

```
<html>
```

HEAD

```
<head>
```

```
  <title>Titel</title>
```

```
  <meta charset=UTF-8" />
```

```
  <meta name="description" content="Homogene Koordinaten" />
```

```
  <meta name="author" content="Frank Steinicke" />
```

```
  <meta name="keywords" content="Vektorgrafiken,2D" />
```

```
</head>
```

BODY

```
<body>
```

```
  ...
```

```
</body>
```

```
</html>
```


Rumpfbereich (Body)

- **Body** enthält eigentlichen Inhalt der Seite
- **Body** benötigt keine besondere Struktur
- Unterschiedliche Auszeichnungen:
 - **Logische Auszeichnungen** beschreiben Strukturelemente
 - **Physische Auszeichnungen** beschreiben Formatierung

Auszeichnungen

Logische

- **Logische Auszeichnungen** beschreiben Strukturelemente
- Wichtigste logische Auszeichnungen:
 - **Normaler Textabsatz (*paragraph*):**
<p> . . . </p>
 - **Überschrift (*heading*):**
<h1> . . . </h1> für die erste Ebene
<h2> . . . </h2> für die zweite Ebene
...

Auszeichnungen

Logische

- Wichtigste logische Auszeichnungen:
 - **Unnummerierte Liste** (*unordered list* mit *list items*):

```
<ul>  
    <li>...</li>  
    <li>...</li>  
</ul>
```

Auszeichnungen

Logische

- Wichtigste logische Auszeichnungen:
 - **Nummerierte Liste (*ordered list*)**:

```
<ol>  
    <li>...</li>  
    <li>...</li>  
</ol>
```

Auszeichnungen

Physische

- **Physische Auszeichnungen** beschreiben Formatierung
- wichtigste physische Auszeichnungen:
 - Fett (*bold*): `...`
 - Kursiv (*italic*): `<i>...</i>`
 - Betonter Text (*emphasis*): `...`
 - Erzwungenes Leerzeichen (*no-break space*): ` `

Hyperlinks

- **Hyperlinks** werden per Auszeichnung `<a>` (*anchor*) definiert, **href** (*hyper reference*) gibt Ziel des Links an
- Beispiel (hier ist nur das Wort „Link“ ein Hyperlink):

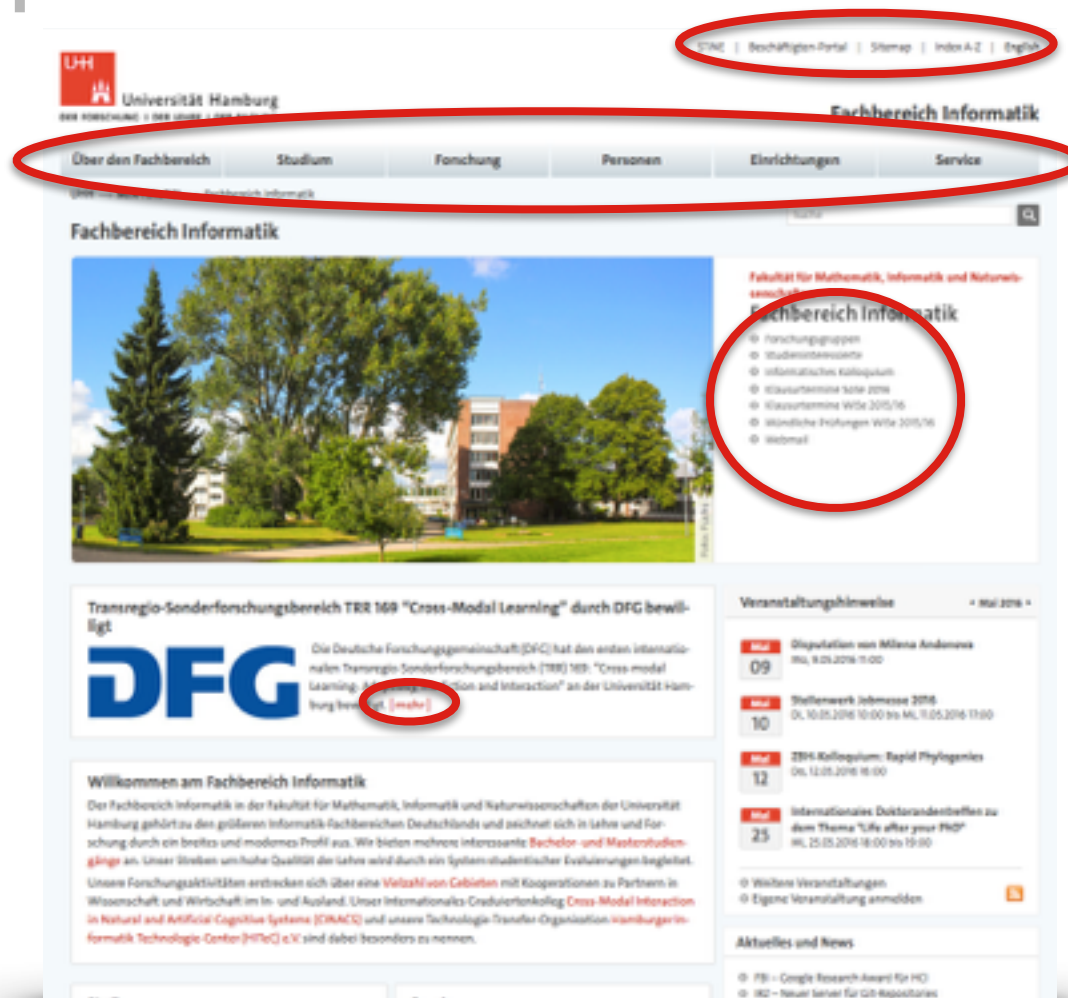
```
<p>Dies ist ein  
<a href="www.example.de">Link</a>.  
</p>
```

Hyperlinks

- Falls Ziel des Hyperlinks sich auf gleichem Server befindet, genügt Angabe des Pfad- und/oder Dateinamens
- Hyperlinks auf Stellen innerhalb des gleichen Dokuments werden mit Zeichen “#” gekennzeichnet

Hyperlinks

Beispiel



Tabellen

- **Tabellen** dienen der tabellarischen Darstellung von Text und Zahlen, aber auch Links und Grafiken
- universell einsetzbar
(z.B. für Layout, obwohl nicht empfohlen)
- **Tabellen** werden von Tag `<table>` umschlossen und können Kopfzeile besitzen

Tabellen

Auszeichnungen

- Tabelle (***table***):
`<table>...</table>`
- Tabellenzeilen (***table row***):
`<tr>...</tr>`
- Tabelleneinträge Kopfzeile (***table head***):
`<th>...</th>`
- restliche Tabelleneinträge (***table data***):
`<td>...</td>`

Tabellen

Übersicht

`<table>`

<code><tr></code>	<code><th></code> <code></th></code>	<code><th></code> <code></th></code>	<code><th></code> <code></th></code>	<code></tr></code>
<code><tr></code>	<code><td></code> <code></td></code>	<code><td></code> <code></td></code>	<code><td></code> <code></td></code>	<code></tr></code>
<code><tr></code>	<code><td></code> <code></td></code>	<code><td></code> <code></td></code>	<code><td></code> <code></td></code>	<code></tr></code>

`</table>`

Tabellen

- Weitere Formatierungsmöglichkeiten:
 - **Tabellenfußzeile (*table footer*):**
`<tfoot>...</tfoot>`
 - **Spaltenbreite** (absolut oder prozentual):
`<colgroup>`
`<col width=...>...<col width=...>`
`</colgroup>`
 - **Mehrspaltige/-reihige** Tabelleneinträge:
`<th colspan=„2“>...</th>`
`<th rowspan=„2“>...</th>`

Tabellen

Beispiel

Europameisterschaft: interaktiver Tabellenrechner zur Saison 2012 - kicker online

http://www.kicker.de/news/fussball/em/startseite/europameisterschaft/2012/4/tabellenrechner.html

RSS

Google

sacc OpenSceneGraph Apple iso LED Yahoo! Google Maps YouTube Wikipedia News Popular

Europameisterschaft - Tabellenrechner

Spiehtag/Tabelle **Tabellenrechner** Teams Scorer Torjäger Topspieler Elfmeter Karten Elf des Tages Mann des Tages Zuschauer

Gruppe A

Pl.	Team	Tore	Diff.	Pkt.
1	Tschechien	4:5	-1	6
2	Griechenland	3:3	0	4
3	Russland	5:3	2	4
4	Polen	2:3	-1	2
Polen 1 : 1 Griechenland				
Russland 4 : 1 Tschechien				
Griechenland 1 : 2 Tschechien				
Polen 1 : 1 Russland				
Tschechien 1 : 0 Polen				
Griechenland 1 : 0 Russland				

Gruppe B

Pl.	Team	Tore	Diff.	Pkt.
1	Deutschland	5:2	3	9
2	Portugal	5:4	1	6
3	Dänemark	4:5	-1	3
4	Niederlande	2:5	-3	0
Niederlande 0 : 1 Dänemark				
Deutschland 1 : 0 Portugal				
Dänemark 0 : 3 Portugal				
Niederlande 1 : 2 Deutschland				
Portugal 2 : 1 Niederlande				
Dänemark 1 : 2 Deutschland				

Gruppe C

Pl.	Team	Tore	Diff.	Pkt.
1	Spanien	6:1	5	7
2	Italien	4:2	2	5
3	Kroatien	4:3	1	4
4	Irland	1:9	-8	0
Spanien 1 : 1 Italien				
Irland 1 : 3 Kroatien				
Italien 1 : 1 Kroatien				
Spanien 4 : 0 Irland				
Kroatien 0 : 1 Spanien				
Italien 2 : 0 Irland				

Gruppe D

Pl.	Team	Tore	Diff.	Pkt.
1	England	5:3	2	7
2	Frankreich	3:3	0	4
3	Ukraine	2:4	-2	3
4	Schweden	5:5	0	3
Frankreich 1 : 1 England				
Ukraine 2 : 1 Schweden				
Ukraine 0 : 2 Frankreich				
Schweden 2 : 3 England				
England 1 : 0 Ukraine				
Schweden 2 : 0 Frankreich				

Viertelfinale

Fr 23.06. 20:45	Cardiff	Deutschland	4 : 2	Griechenland
Do 21.06. 20:45	Warschau	Tschechien	0 : 1	Portugal

Viertelfinale

Sa 23.06. 20:45	Donostia	Spanien	2 : 0	Frankreich
So 24.06. 20:45	Kiew	England	2 : 4	Italien

Halbfinale

Mi 27.06. 20:45	Donostia	Portugal	1 : 1	Spanien
-----------------	----------	----------	-------	---------

Halbfinale

Do 28.06. 20:45	Warschau	Deutschland	4 : 2	Italien
-----------------	----------	-------------	-------	---------

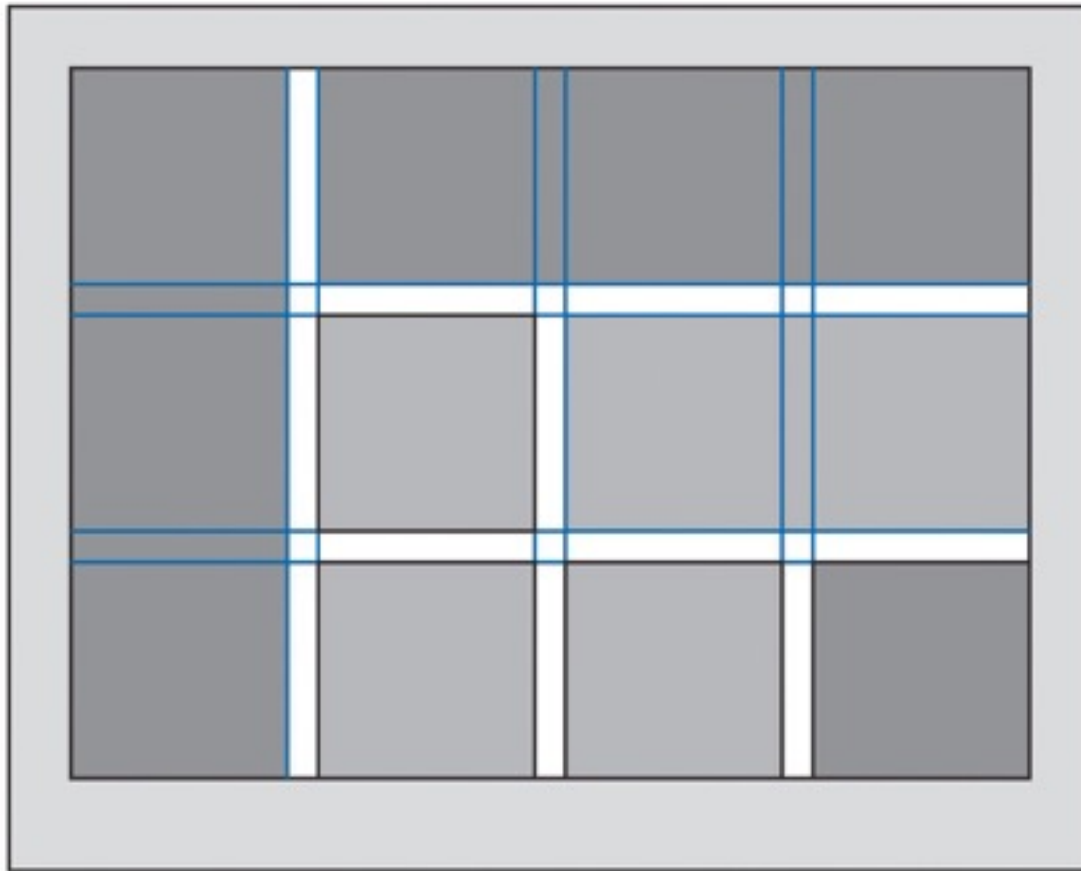
Finale

So 01.07. 20:45 Kiew

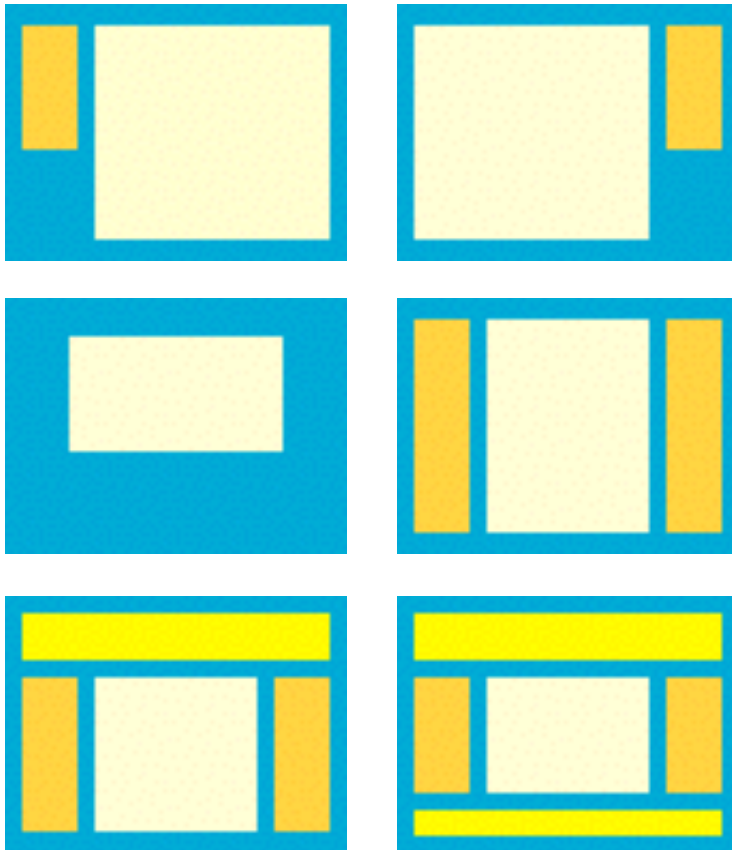
-- 1 --

Grid-Layout

Beispiel



Layout mit div



CSS

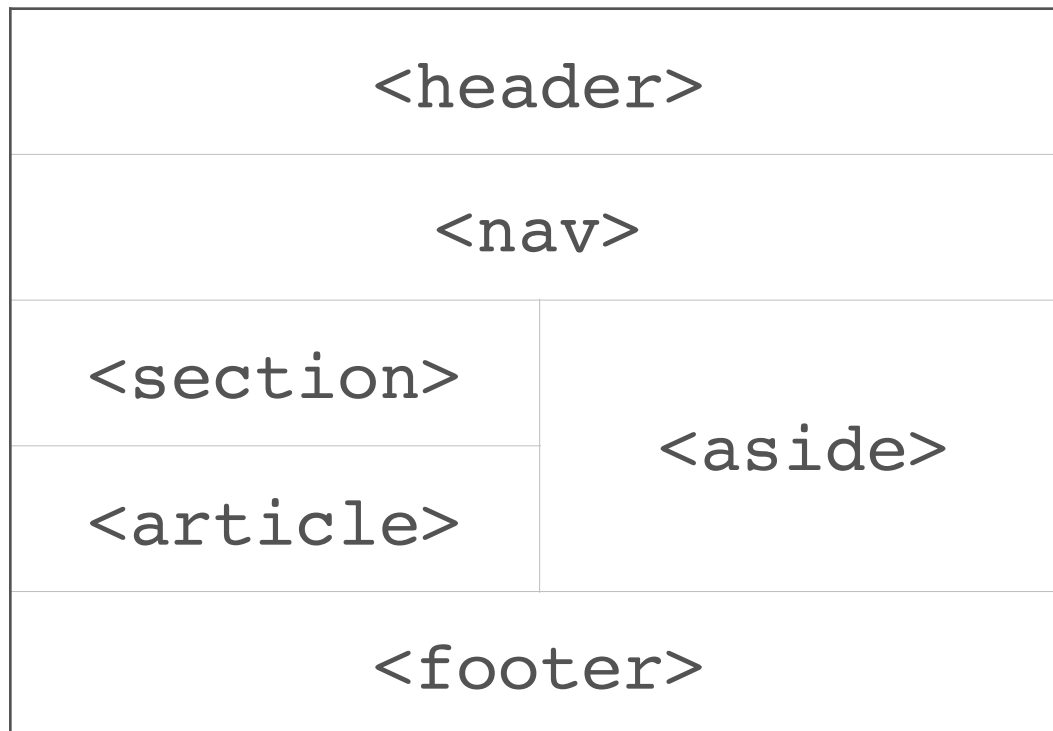
```
#header {  
  background-color:black;  
  color:white;  
  text-align:center;  
  padding:5px;  
}  
#section {  
  width:350px;  
  float:left;  
  padding:10px;  
}  
...
```

HTML

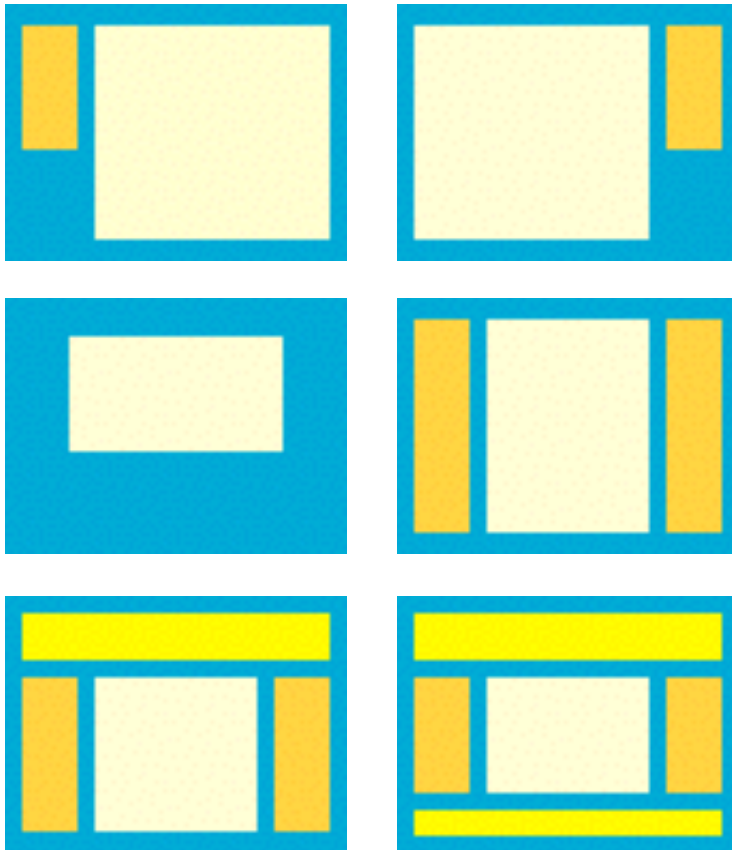
```
<div id="header">...</div>  
<div id="section">...</div>  
<div id="footer">...</div>
```

Layout mit HTML5

- Einführung neuer semantischer Elemente:



Layout mit HTML5



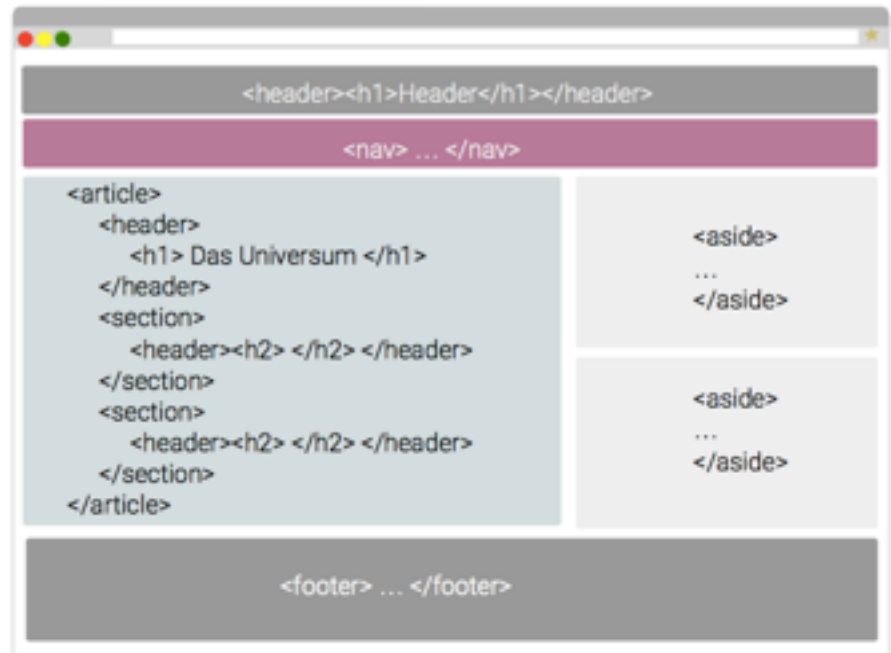
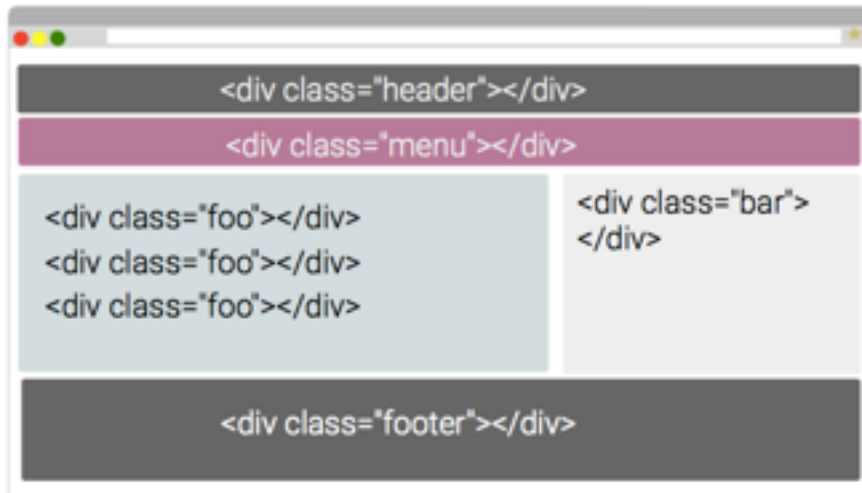
CSS

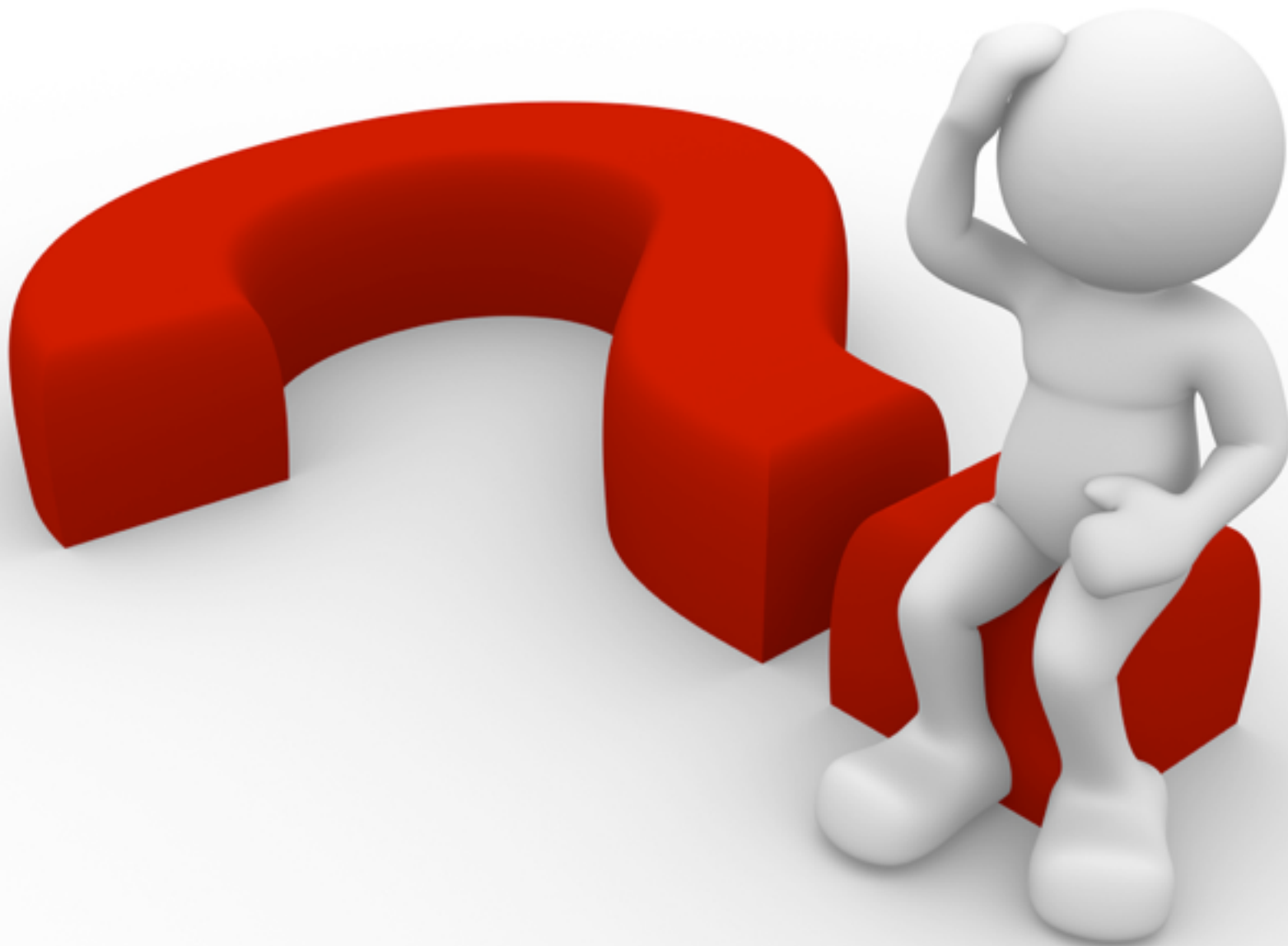
```
header {  
  background-color:black;  
  color:white;  
  text-align:center;  
  padding:5px;  
}  
section {  
  width:350px;  
  float:left;  
  padding:10px;  
}  
...
```

HTML

```
<header>...</header>  
<section>...</section>  
<footer>...</footer>
```

Layout div vs. HTML5







Interaktive Medien

Kapitel Mediensysteme & World Wide Web

Dynamik im WWW

Dynamische Webseiten

- **Dynamische Webseiten** werden vor jeder Anzeige **neu** berechnet
- **Statische Webseiten** besitzen festgelegte Inhalte
 - **Interaktivität** kann nur durch interaktive Multimedia-Dokumente umgesetzt werden, z.B. Flash-Dokumente ...

Dynamische Webseiten

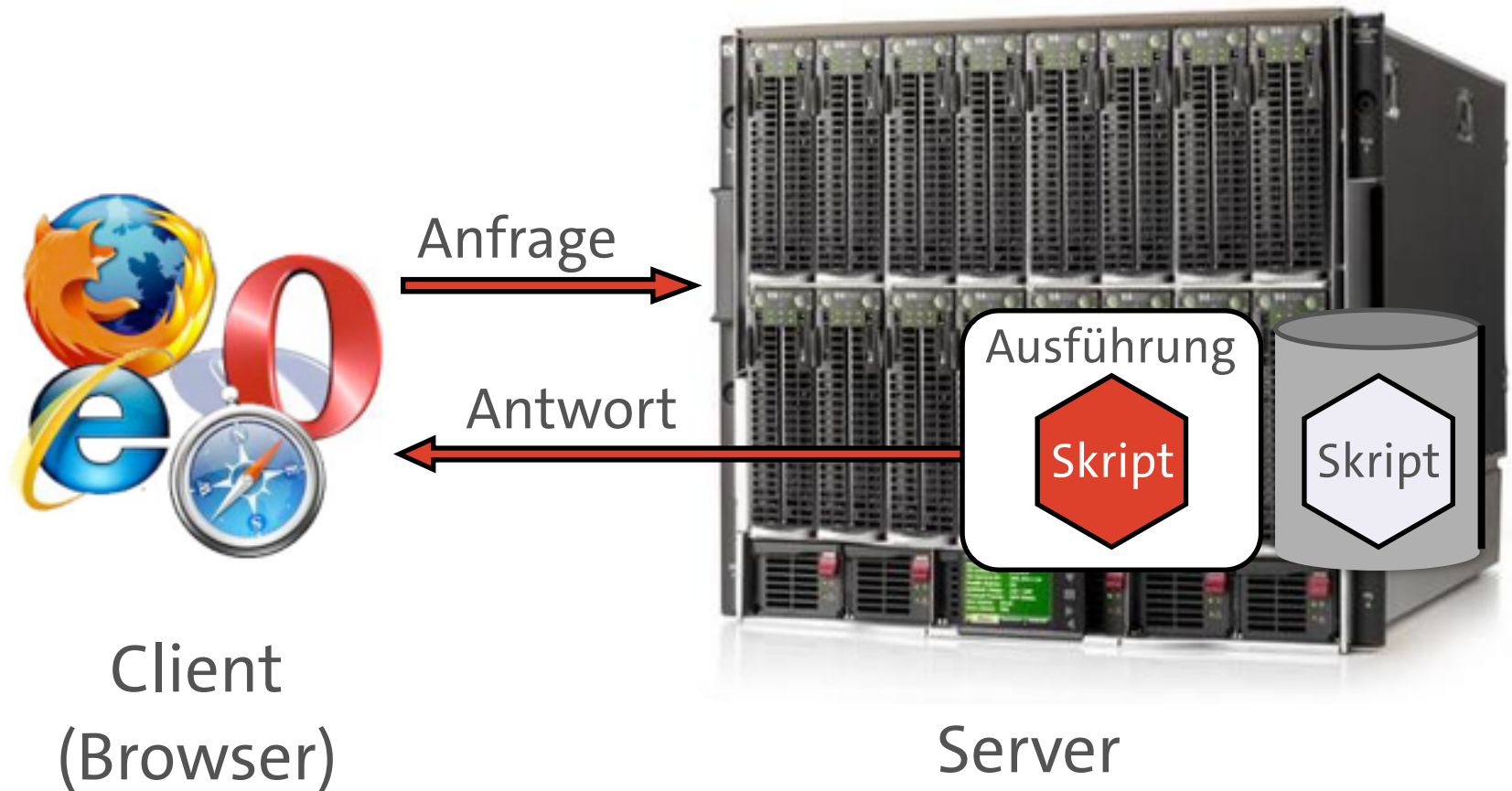
- Auswertung zur **dynamischen Anzeige** findet i.d.R. anhand von **Skriptsprachen** statt
- Skript ist Bestandteil der vom Browser aufgerufenen Datei
- Beispiele:
 - Inhalt eines Online-Warenkorbs
 - Verwendung aktueller Uhrzeit
 - ...

Dynamische Webseiten

- Zwei verschiedene Ansätze:
 1. **Serverseitige Dynamik:** Berechnung findet auf dem **Server (Rechner mit den anzuzeigenden Informationen)** statt
 2. **Clientseitige Dynamik:** Berechnung findet auf dem **Rechner des Webnutzers** statt

Serverseitige Dynamik

Ablauf



Serverseitige Dynamik

Ablauf

1. Zugriff auf angeforderte Datei
2. Standardkonformes HTML-Dokument wird direkt beim Server erzeugt
3. HTML-Dokument wird an Client übertragen
4. Anzeige im Browser basierend auf dem erhaltenen HTML-Dokument

Diskussion



Was sind die Vor- und Nachteile von
serverseitiger Dynamik?

Serverseitige Dynamik

- Vorteil:
 - Konfiguration des clientseitigen Browsers spielt keine Rolle
 - Abstimmung zwischen Serversoftware und Dateien einfacher
 - einzige Möglichkeit für Zugriff auf Datenbestände (z.B. Datenbanken) auf Server
- Nachteil:
 - dynamische Vorgänge hängen von Internetverbindung ab

Serverseitige Dynamik

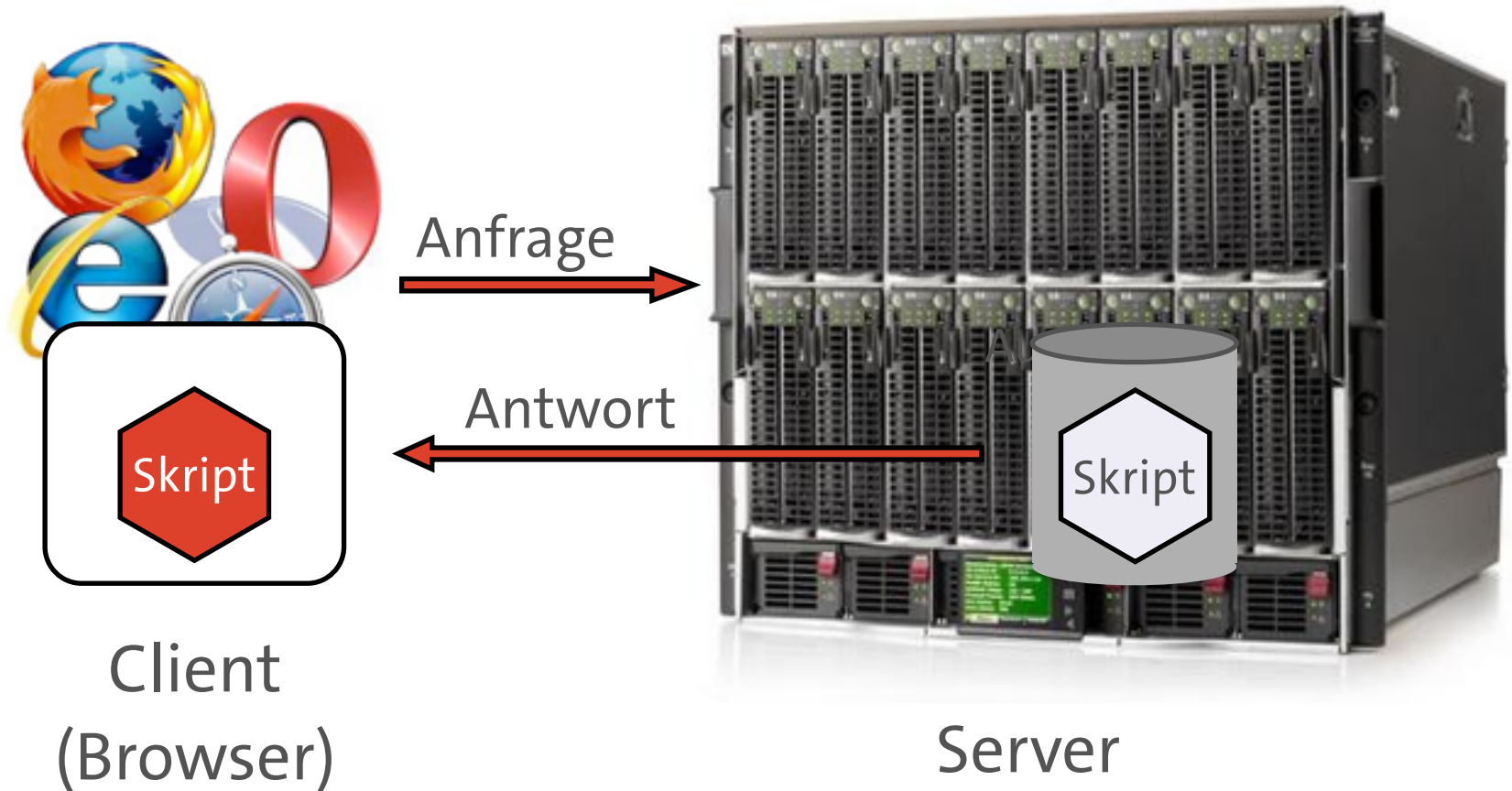
Sprachen

- Beispiele für typische Sprachen serverseitiger Ausführung:
 - PHP
 - ASP
 - Java Server Pages (JSP)
 - ASP.Net



Clientseitige Dynamik

Ablauf



Clientseitige Dynamik

Ablauf

1. Datei mit Skriptinformationen liegt auf Server
2. Server leitet Datei an Client
3. Skript wird beim Client ausgeführt
4. Berechnung startet
5. Anzeige wird je nach Berechnung modifiziert

Diskussion



Was sind die Vor- und Nachteile von
clientseitiger Dynamik?

Clientseitige Dynamik

- Vorteil:
 - Ausführung des Skripts benötigt keinen Internetzugriff mehr
 - ➔ sehr schnelle Reaktionszeit
- Nachteil:
 - verwendeter Browser (beim Nutzer) muss Skriptsprache unterstützen und zulassen

Clientseitige Dynamik

Sprachen

- Beispiele für typische Sprachen clientseitiger Ausführung:
 - JavaScript
 - ECMAScript
 - VBScript



JavaScript

- **JavaScript (JS)** ist Skriptsprache, die hauptsächlich für **dynamisches HTML** in Web-Browsern eingesetzt wird
- Ursprung 1995 bei Firma Netscape
- Inzwischen internationaler Standard für JavaScript:
ECMAScript bzw. **ISO-16262**

JavaScript

Typische Anwendungsgebiete

- Dynamische Manipulation von Webseiten
- Plausibilitätsprüfung (Datenvalidierung) von Formulareingaben noch vor der Übertragung zum Server
- Sofortiges Vorschlagen von Suchbegriffen
- Banner oder Laufschriften
- ...

JavaScript

Einbindung in HTML

- Einbindung über `script`-Element
- Direkte Anbindung an HTML-Elemente als Ereignisbehandlung (***Event-Handler***) möglich
 - mehr dazu in Abschnitt „Events“
- Einbindung externer JavaScript-Datei

JavaScript

Einbindung: Beispiel

```
<html>
  <head>
    <title>Test</title>
    <script>
      alert("Hallo Welt!");
    </script>
  </head>
  <body>
  </body>
</html>
```



