

# Digitale Kommunikation und Internetdienste 1

Wintersemester 2004/2005 – Termin 15

Belegnummer    Vorlesung:    39 30 02

                          Übungen:    39 30 05

*Jan E. Hennig*

AG Rechnernetze und Verteilte Systeme (RVS)

Technische Fakultät

Universität Bielefeld

`jhennig@rvs.uni-bielefeld.de`

basierend auf den Arbeiten von Michael Blume, Heiko Holtkamp, Marcel Holtmann und I Made Wiryana

- HTTP
- HTML (Grundlagen)
- CSS (Grundlagen)

- *Hyper Text Transfer Protocol*
- über dieses Protokoll werden nahezu alle Daten im *World Wide Web* übertragen
- also: HTML-Dateien, Bilder, Anfragen und -ergebnisse, ...
- HTTP setzt dabei auf TCP auf (bzw. Erweiterungen wie T/TCP)
- die Kommunikation läuft zwischen einem HTTP-Client und einem HTTP-Server (Webserver) ab
- Standardport ist 80/tcp

- HTTP überträgt *Ressourcen*
- dies können reguläre Dateien sein
- oder auch dynamisch erzeugte Daten
- eine Ressource wird über einen *Uniform Resource Locator (URL)* identifiziert
- Beispiel:  
`http://www.rvs.uni-bielefeld.de/cms/lecture/W04`

- Struktur: Client-Server-Modell
- ein Client öffnet eine Verbindung und sendet eine *Anfragennachricht (request message)* an einen HTTP-Server
- der Server antwortet mit einer *Antwortnachricht (response message)*
- diese enthält üblicherweise die angeforderte Ressource
- anschließend wird die Verbindung geschlossen (HTTP 1.0)

- Sowohl Anfrage wie auch Antwort bestehen aus folgenden Teilen:
- Startzeile
- Null oder mehr *Headerzeilen*
- eine Leerzeile
- ein optionaler *Body* (mit der Ressource als Inhalt)

- eine Anfrage besteht aus drei Teilen:
- Anweisung
- Pfad zur Ressource
- Version von HTTP (HTTP/x.y)
- Beispiel:  
`GET /index.html HTTP/1.0`

- GET
- ordert eine Ressource (häufigste Anweisung)
- HEAD
- wie GET, aber ohne den Inhalt der Ressource (nur Header)
- POST
- sendet Daten an ein Programm (das über die URL identifiziert wird)
- üblicherweise zusammen mit Header-Angaben `Content-Type:` und `Content-Length:`



- eine Antwort besteht ebenfalls aus drei Teilen:
- Statuszeile
- Header-Zeilen
- Body (mit Ressourceninhalt)

- die Statuszeile besteht aus drei Teilen:
- HTTP-Version (z.B. HTTP/1.0)
- Statuscode
- Beschreibung des Statuscodes

- 1xx zeigt eine Informationsnachricht an
- 2xx zeigt Erfolg an
- 3xx zeigt eine Umleitung zu einer anderen URL an
- 4xx zeigt einen Fehler auf Client-Seite an
- 5xx zeigt einen Fehler auf Server-Seite an

- HTTP/1.0 200 OK
- HTTP/1.0 404 Not Found
- HTTP/1.0 301 Moved Permanently
- HTTP/1.0 302 Moved Temporarily
- HTTP/1.0 500 Server Error

- Header basieren auf RFC 822
- dies ist die Definition, die auch für E-Mails gilt
- Bezeichner gefolgt von einem Doppelpunkt, anschließend der Wert
- Beispiel:  
`Last-Modified: Fri, 31 Dec 1999 23:59:59 GMT`

- HTTP/1.0 war nie ein offizieller Standard (aber ein de-facto-Standard)
- beschrieben in RFC 1945
- HTTP/1.1 wurde von einer IETF-Arbeitsgruppe entworfen
- beschrieben in RFC 2616
- veröffentlicht im Juni 1999

- schnellere Antworten durch Möglichkeit eine bestehende Verbindung weiterzubenutzen
- Bandbreiteneinsparung durch Cacheunterstützung möglich
- effiziente Nutzung von IP-Adressen durch Möglichkeit mehrere Domains auf einem Server unterzubringen

- HTTP/1.1 bringt eine Pflichtangabe mit: die Host-Angabe im Header
- Requests sehen somit z.B. so aus:  
GET /path/file.html HTTP/1.1  
Host: www.host1.com:80  
<Leerzeile>
- die Portangabe :80 ist optional (außer wenn anderer Port)



- standardmäßig wird die Verbindung offengehalten
- bei der letzten Anfrage wird spezieller Header gesetzt:  
`Connection: close`
- ein Server darf eine Verbindung auch von sich aus schließen
- es ist dann Aufgabe des Clients darauf zu achten evtl. eine Anfrage nochmal zu stellen

- HTTP/1.2 noch nicht offiziell
- soll absolute Pfadangaben im Request unterstützen
- Beispiel:  
GET `http://www.rvs.uni-bielefeld.de/cms/` HTTP/1.2

- bislang: Übertragung von Daten
- Daten – passend interpretiert – liefern Informationen
- früher: z.B. Mailboxsysteme, schwarze Bretter
- heute häufige Anwendung: Webseiten
- Analogie: Plakat oder Anzeige

- Grundsätzliche Fragen:
- Was darf ich im WWW veröffentlichen?
- Was soll ich im WWW veröffentlichen?



- wenn keine Zugriffsbeschränkungen installiert sind, ist eine Webseite eine Art der allgemeinen Veröffentlichung
- die Veröffentlichung findet im Internet weltweit statt
- daher Beachten der entsprechenden Gesetzgebungen wichtig
- mindestens: nationale Gesetzgebung am Serverstandort und am „Einspielort“
- das Speichern von Informationen im World Wide Web bedeutet nicht den Verzicht des Autors auf das Urheberrecht und die Nutzungsrechte (Copyright)

- das Urheberrecht
- der Datenschutz
- das Strafrecht
- verschiedene Geheimhaltungspflichten
- firmeninterne Regeln
- ...



- das WWW eignet sich für Informationen, die von den Interessenten *freiwillig* aufgesucht und abgefragt werden
- also für Informationen, die nützlich und interessant sind oder die unterhalten und Freude bereiten
- es ist wichtig, die Informationen so zu strukturieren und anzulegen, daß sie von den Interessenten auch gefunden werden
- falsche oder nicht mehr aktuelle Informationen sind meist schlechter als gar keine Information
- eine nicht ganz so schön formatierte, aber verfügbare Information ist oft besser als gar keine Information

- *Hyper Text Markup Language*
- Sprache, um Text mit Struktur zu beschreiben
- Standard für Webseiten (in verschiedenen Versionen)
- Formatanweisungen sind in den Text integriert
- dient dazu den Inhalt in seiner logischen Struktur zu beschreiben (nicht (nur) Layout)



- Formatanweisungen sind in den Text „eingestreut“
- diese werden *HTML-Tags* genannt (tag = Markierung, Etikett)
- mit diesen Anweisungen wird von einem Webbrowser der Inhalt interpretiert
- und anschließend der formatierte Text dargestellt (inkl. eingebundenen Bildern, Ton, ...)
- der Benutzer hat Einfluß auf die Art und Weise der Darstellung, die logische Struktur wird hingegen vom Ersteller der Webseite vorgegeben

- verbreitet und weithin verstanden wird HTML 3.2 und 4.01
- neuer Standard ist XHTML 1.0
- das *World Wide Web Consortium (W3C)* pflegt diese Sprache und deren Weiterentwicklung
- es gibt Validatoren, mit denen die Standardkonformität getestet werden kann, z.B. <http://validator.w3.org/>

- Grundlage ist die Idee einer weltweiten Zusammenarbeit
- World Wide Web:
- alle Benutzer mit allen Clients sollen auf alle Informationen und Services auf allen Servern zugreifen können
- es gibt herstellerspezifische Erweiterungen, die *nicht* von jedem Client verstanden werden
- solche sollte man üblicherweise meiden

- die Markup-Befehle (tags) werden vom normalen Text dadurch unterschieden, dass sie zwischen „spitzen Klammern“ (tatsächlich Kleiner- und Größerzeichen) stehen: `<xxx>`
- manche haben auch Parameter (Attribute und Argumente): `<xxx yyy=zzz>`
- wenn die Argumente Sonderzeichen enthalten, müssen sie in Anführungszeichen (Quotes) eingeschlossen werden: `<xxx yyy="zzz">`

- einfache oder mehrfache Leerstellen oder Zeilenwechsel wirken jeweils so wie eine Leerstelle
- die meisten HTML-Befehle treten paarweise auf, mit einem Start-Tag der Form `<xxx>` und einem End-Tag der Form `</xxx>`
- diese Befehlspaare geben jeweils die Bedeutung des dazwischen liegenden Textes an
- wichtig ist, daß paarweise Befehle immer richtig geschachtelt werden



```
<html>
  <head>
    <title> Der Titel des HTML-Files
  </title>
  </head>
  <body>
    <h1> Die Überschrift
  </h1>
    <p> Der erste Absatz.
  </p>
    <p> Ein
      <em> betontes
    </em>
      Wort im zweiten Absatz.
      Nach diesem Absatz kommt eine Linie:
  </p>
    <hr>
    <p> Das ist der letzte Absatz.
  </p>
  </body>
</html>
```

- der gesamte Inhalt wird zwischen `<html>` und `</html>` eingeschlossen
- zwischen `<head>` und `</head>` stehen die Angaben, die für die Verwaltung der HTML-Datei notwendig sind (Titel, Autor, ...)
- zwischen `<body>` und `</body>` steht die Information, die am Bildschirm des Benutzers dargestellt werden soll
- zwischen `<title>` und `</title>` steht die Überschrift der HTML-Datei, die in Bookmarks und Suchhilfen verwendet wird

- nicht alle Zeichen sind erlaubt
- im Titel sind nur ASCII-Zeichen gestattet
- jedoch viele Browser können auch mit 8-Bit-Zeichen umgehen (aber nicht alle)
- nötig daher: Umkodieren von nicht im ASCII-Zeichensatz vorkommenden Zeichen wie Umlauten, Ligaturen, ...



- Umlaute werden als Kombination von Grundbuchstaben mit `uml` angegeben
- als Escape-Hinweis (analog zum Bytestopfen) gelten einleitend `&` und beendend ein Semikolon
- `ä = &auml;`      `Ä = &Auml;`
- `ö = &ouml;`      `Ö = &Ouml;`
- `ü = &uuml;`      `Ü = &Uuml;`

- Ligaturen werden ähnlich angegeben als Kombination der beiden Zeichen
- anstelle von `uml` wird `lig` geschrieben
- z.B. ß = `&szlig;`

- `<h1>...</h1>` bis `<h5>...</h5>` stellen verschiedene Überschriftsebenen dar
- `<strong>...</strong>` sorgt für hervorgehobenen Text
- `<em>...</em>` sorgt für betonten Text
- Absätze werden von `<p>...</p>` eingeschlossen
- Bilder werden mit `` eingebunden

- entscheidend für den Erfolg von Hypertext sind (und waren) Hyperlinks
- damit lassen sich verschiedene Dokumente direkt miteinander verknüpfen
- eine Sprungadresse wird mit `<A href="http://ziel.url">...</A>` angegeben
- innerhalb eines Dokuments können Ankerpunkte mit `<A name="Anker1">...</A>` gesetzt werden
- auf diese kann mittels `"http://ziel.url#Anker1"` direkt verwiesen werden

- drei Arten von Listen sind definiert:
- eine unsortierte Liste wird mit `<ul>...</ul>` erstellt, Listenelemente in `<li>...</li>` eingebettet
- eine sortierte Liste wird mit `<ol>...</ol>` erstellt, Listenelemente in `<li>...</li>` eingebettet
- eine Definitionsliste wird mit `<dl>...</dl>` erstellt, die Begriffe in `<dt>...</dt>` und die eigentliche Definition in `<dd>...</dd>` eingebettet

```
<ul>
  <li>ein Punkt in der Liste</li>
  <li>noch ein Punkt
    <ol>
      <li>nummerierter Unterpunkt</li>
      <li>noch ein nummerierter Unterpunkt</li>
    </ol>
  </li>
</ul>
```

- Tabellen werden mit `<table border="Dicke">...</table>` gekennzeichnet
- für `Dicke` muß eine Zahl eingesetzt werden
- `<tr>...</tr>` kennzeichnet eine Zeile (table row)
- `<th>...</th>` kennzeichnet eine Kopfzelle (table head)
- `<td>...</td>` kennzeichnet eine Datenzelle (table data)

```
<table border="3">
  <tr>
    <th>Kopfzelle</th>
    <th>Kopfzelle</th>
  </tr>
  <tr>
    <td>Datenzelle</td>
    <td>Datenzelle</td>
  </tr>
</table>
```



- um den Browsern das Interpretieren einer HTML-Seite zu erleichtern gibt es die Möglichkeit Dokumenttypangaben zu machen
- man zeigt damit, auf welche HTML-Version man sich bezieht
- für XHTML sind diese Angaben bereits fest vorgeschrieben
- Beispiel (vor dem `<html>`-Tag anzugeben):  

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```
- und anstelle von `<html>` z.B.:  

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

- frühe HTML-Versionen haben Layout-Tags und Strukturgebende Tags gemischt verwendet
- heute läßt sich dieses getrennt behandeln
- Vorteile:
  - einheitliches Layout wiederverwendbar
  - Änderungen können an wohldefinierter Stelle vorgenommen werden
- Layout via *Content Style Sheet (CSS)*
- eigene Datei oder *inline*

- die Trennung von Layout und Struktur ist außerdem hilfreich für die Gestaltung barrierefreier Seiten, z.B. für Sehbehinderte
- früher wurden oft ineinander geschachtelte Tabellen fürs Layout verwendet
- ein Sprachsynthesizer oder eine Braille-Ausgabezeile aber können die eigentliche Struktur so kaum sinnvoll darstellen
- derartige Seiten sind nicht barrierefrei und behindertengerecht
- mit einer Trennung von Layout und Struktur hingegen kann ein Großteil der Hilfen bereits die Seiten angemessen darbieten

Die wichtigsten Regeln für Barrierefreiheit nach dem W3C:

- Bilder & Animationen: Beschreibe Funktion mit alt-Attributen
- Multimedia: Untertitel und Transkriptionen für Audio, Audio-Deskriptionen für Video
- Hypertext-Links: Link-Text sollte verständlich sein, wenn sie nicht im Textzusammenhang zu lesen sind – vermeide z.B. „hier klicken“
- Aufbau der Seite: Verwende Überschriften, Listen und eine klar gegliederte Struktur. Wenn möglich, verwende CSS zur Formatierung
- Diagramme: Beschreibe sie im Text oder verwende longdesc-Attribut

## Regeln für Barrierefreiheit nach dem W3C (Fortsetzung):

- Skripte, applets & plug-ins: Für den Fall, dass diese nicht behindertengerecht sind oder nicht unterstützt werden, mache Alternativen verfügbar
- Frames (Rahmen): Verwende noframes und verständliche Titel
- Tabellen: Sollten Zeile für Zeile lesbar sein. Fasse Inhalt zusammen
- Überprüfe die Seite: Verwende unter <http://www.w3.org/TR/WCAG> beschriebene Software & Richtlinien  
(dies sind die *Web Content Accessibility Guidelines*)

- man kann z.B. wichtige Hinweise zwischen `<p class="wichtig">` und `</p>` einschließen
- und im Style-Sheet angeben, dass alle mit `class="wichtig"` markierten Absätze eingerückt und in roter Schrift geschrieben werden sollen
- mit weiteren Klassen kann die gesamte Corporate Identity abgebildet werden
- ein und derselbe Inhalt kann leicht mit verschiedenen Designs betrachtet werden
- CSS wird von modernen Browsern unterstützt, von einigen älteren hingegen nicht oder nur unvollständig

```
<html>
<head>
<title>Die bunte Testseite</title>

<style type="text/css">
BODY { color: black; background: white; font-family: Times, serif }
A:link { color: #990000; background: white;
        text-decoration: underline }
A:visited { color: #660000; background: white;
           text-decoration: underline }
A:active { color: #FF0000; background: yellow }
.logo    { color: #009900; background: white }
.wichtig { color: red; background: white;
          font-style: normal; font-weight: bold }
EM { font-style: normal; font-weight: bold }
BLOCKQUOTE { margin-left: 10%; margin-right: 10% }
</style>
</head>
```

...

...

```
<body>
  <h1>Seite für UniBI-Studenten</h1>
  <p>
    Das Hochschulrechenzentrum der
    <em class="logo">Universität</em>
    stellt den
    <em class="logo">UniBI</em>-Studenten
    PCs in den Benutzeräumen zur Verfügung.
    <blockquote class="wichtig">
      Achtung!
      Diese PCs dürfen nur für die Lehre und Forschung der
      <em class="logo">UniBI</em>
      verwendet werden und nicht für andere Zwecke!
    </blockquote>
    Genauere Informationen ...
</body>
</html>
```



- Das Einbinden einer externen Datei geschieht mit  
`<link rel=stylesheet href="CSS-Datei-URL" type="text/css">`
- weitere CSS-Beispiele unter  
<http://csszengarden.com/>
- weitere HTML-Anleitungen z.B. unter  
<http://www.selfhtml.org/>

Übersicht für die kommende Woche:

- IRC
- Quality of Service

Ende Termin 15. Danke für die Aufmerksamkeit.