

# Webprogrammieren I

# Informationen

## Allgemein

- Alain M. Lafon
- E-Mail: [xlaf@zhaw.ch](mailto:xlaf@zhaw.ch)
- Webentwickler und Berater bei [panter.ch](http://panter.ch)
- Tägliche Arbeit mit: Ruby on Rails, HTML, CSS, Javascript, HAML, SASS, CoffeeScript, jQuery, Ajax, ...
- Im letzten Jahr alleine >15 abgeschlossene Web-Projekte

# Informationen

## Allgemein

- Folien, Aufgaben & Code:
  - *Moodle: Webprogrammieren I FS12*
- Literatur:
  - *Eric Freeman, Elisabeth Freeman: Head First HTML with CSS & XHTML, O'Reilly 2005*
  - *Eric Freeman, Elisabeth Robson: Head First HTML5 Programming, O'Reilly 2011*
  - *warum zwei Bücher??*

# Informationen

## Lernziele

- Verstehen der Grundlagen von
  - *HTML5*
  - *CSS3*
  - *Javascript*
- Erstellen einer einfachen Webseite
  - *Beherrschen von HTML5 in eigenen Projekten*

# Informationen

## Lerninhalte

- Grundlagen HTML(Geschichte und Herkunft), DOM
- HTML5 Syntax, Einsatzgebiete
- Parsen und schreiben von HTML5 Dokumenten mit Javascript, Styling mit CSS3
- Werkzeuge zum Analysieren von Web Protokollen verwenden
- Manipulation des DOM
- Praktische Übung, um die Themen zu erproben

# Informationen

## Besprechung Ablauf

- Lektionenplanung
  - *Kontaktunterricht 20 Lektionen*
  - *Begleitetes Selbststudium 30 Lektionen*
  - *Individuelles Selbststudium 30 Lektionen*
- Implikationen Lektionenplanung
  - *Von gesamt 60h (80 Lektionen) nur 15h (20 Lektionen) Vorlesung*
  - *=> Hohe Eigenverantwortung*

# Informationen

## Besprechung Ablauf

- 2 ECTS
- Projektarbeit
  - *Teil des begleiteten Selbststudiums*
  - *Umfang 10h*
- 1 schriftliche Klausur am Ende des Semesters
- optional: Klausuren inmitten des Semesters
  - *gewünscht?*

# Informationen

## Besprechung Ablauf

- Ich halte die Veranstaltung zum ersten Mal
- Vorgänger ist nicht gut angekommen, deshalb wird sie neu ausgelegt
- Sukzessive Vorbereitung
  - *Einzelne Themen können stärker oder schwächer ausgearbeitet werden*
  - *=> Kritik erwünscht und nötig*
- Aktive Partizipation notwendig
  - *=> Laptops mitbringen*



**Fragen?**

**Vorkenntnisse?**

**Professionelle Webentwickler im  
Raum?**

**Los geht's!**

# Das Web

## Historische Einbettung

- 1989 als Projekt im CERN in Genf
- Erfinder: Tim Berners-Lee
- Erste Webseite: [info.cern.ch](http://info.cern.ch)

## Public Announcement im Usenet:

This project is experimental and of course comes without any warranty

whatsoever. However, it could start a revolution in information access. We are

currently using WWW for user support at CERN. We would be

very interested in

comments from anyone trying WWW, and especially those making other data available, as part of a truly world-wide web.

[Originaltext groups.google.com/group/alt.hypertext](https://groups.google.com/group/alt.hypertext)

# **Das Web**

**Was macht das Web aus?**

**Wie funktioniert es?**

# Das Web

## Topologie

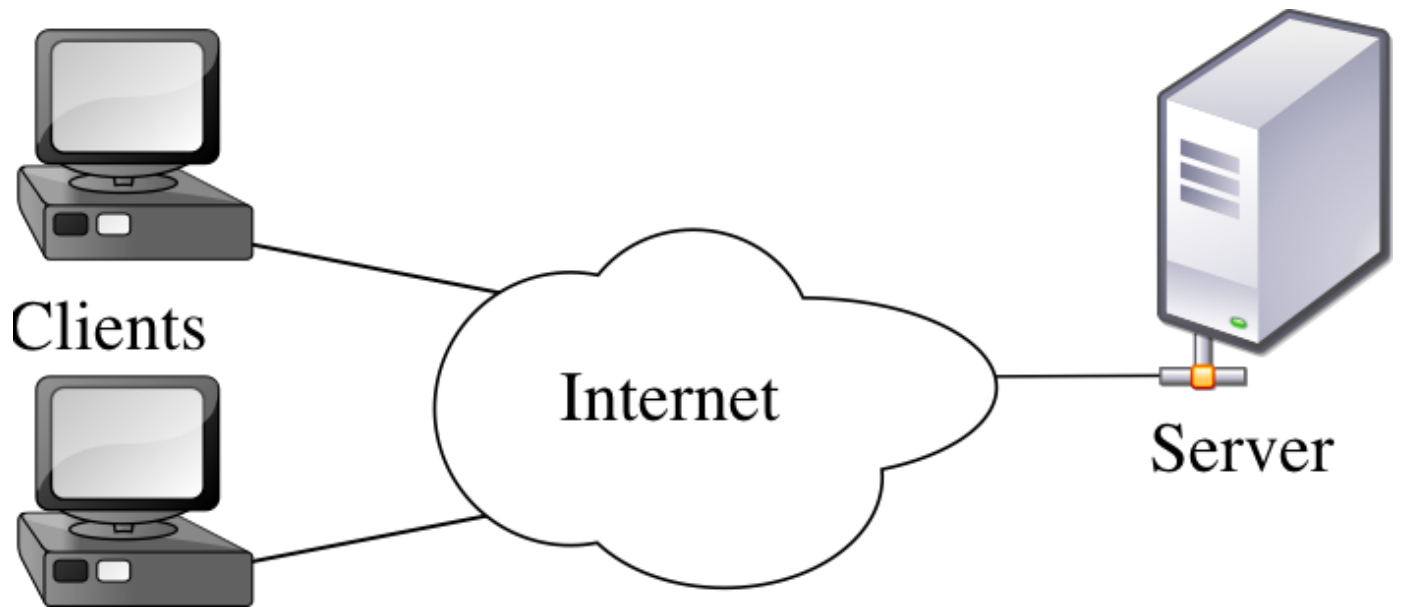
### Client - Server

- Aufgaben und Dienste werden innerhalb eines Netzes verteilt
- Clients fordern von Servern gezielt Aufgaben an

# **Das Web**

## **Topologie**

### **Client - Server**



Quelle: [wikipedia.org](https://www.wikipedia.org)



# Das Web

## Topologie

### Definition Server

- Ein Server ist ein Programm (Prozess), das mit einem anderen Programm (Prozess), dem Client, kommuniziert, um ihm Zugang zu einem Dienst zu verschaffen.

### Definition Client

- Ein Client kann auf Wunsch einen Dienst bei dem Server anfordern, der diesen Dienst bereitstellt.

# Topologie

## Protokolle

### Namensauflösung: DNS

- Assoziiert Informationen zu Hostnamen
- Hauptsächlich genutzt als "forward-lookup"
  - *Aus Domain Name IP Adresse nachschlagen*
- Weltweit verteilter Verzeichnisdienst
- Vorteile
  - *Namen sind einfacher zu merken als Zahlenkolonnen*
  - *IP Adressen(sprich physikalische Server) können einfach getauscht werden*

# Topologie

## Protokolle

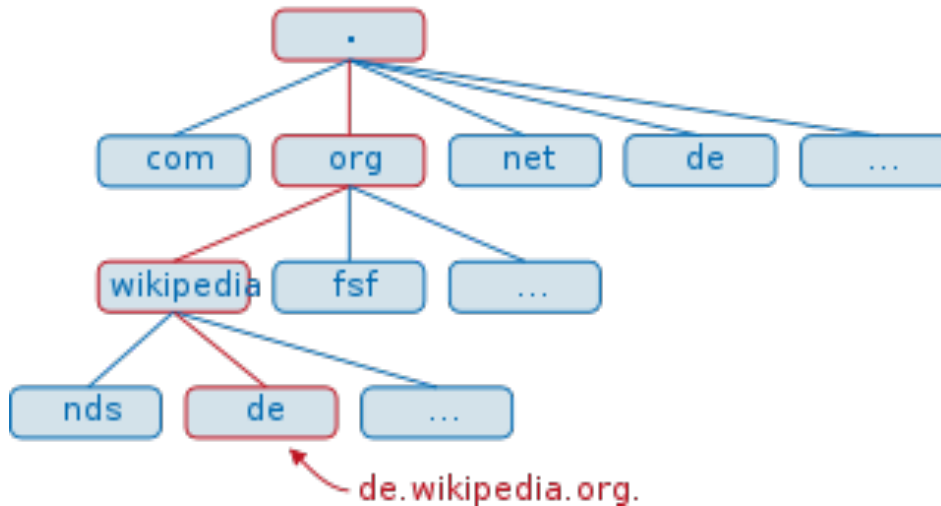
### Namensauflösung: DNS

- Funktionsweise
  - *Dezentrale Verwaltung*
  - *Eindeutige Namen*
  - *Erweiterbarkeit*
  - *Hierarchische Baumstruktur des Namensraums*

# Topologie

## Protokolle

### Namensauflösung: DNS



Quelle: [wikipedia.org](https://www.wikipedia.org)

# Topologie

## Protokolle

### Namensauflösung: DNS

- IPv4
  - *32 bit Adressraum*
  - $\Rightarrow 2^{32}$  max IP Adressen ( $4.3 \cdot 10^9$ )
    - NB: Seit dem 03.02.2011 keine weiteren IPv4 Adressen mehr
  - *Grund für die Limitierung*
    - IPv4 war nie für die Öffentlichkeit gedacht
    - War ein Projekt der DARPA (Defense Advanced Research Projects Agency)

# Topologie

## Protokolle

### Namensauflösung: DNS

- IPv6
  - *Inkompatibel zu IPv4*
  - *Grösserer Adressraum*
  - *$2^{128}$  oder  $3.4 \cdot 10^{38}$  Adressen*
  - *=> Jeder Rechner, jedes Stück Ware bekommt lebenslange Adresse*
  - *Zwingende Verschlüsselung auf IP Ebene (IPsec)*
  - *Effizientere Header-Daten Verarbeitung als IPv4*

# Topologie

## Protokolle

### N.B. IPv6

- Einige Provider sind noch nicht bereit
- WinXP kann (von Haus aus) nur IPv4
- Privacy Extensions notwendig für Betriebssysteme, um Tracking auszuschliessen
  - *Das OS generiert temporäre IPv6 Adressen*
  - *Default in OSX, iOS, Windows sowie einige Linux Distros*

# Beispiel DNS

```
$dig google.com
```

```
; <>> DiG 9.7.3-P3 <>> google.com
```

```
;; global options: +cmd
```

```
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1038
```

```
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 0
```

```
;; QUESTION SECTION:
```

```
;google.com.      IN  A
```

```
;; ANSWER SECTION:
```

```
google.com.      29  IN  A 209.85.148.138
```

```
google.com.      29  IN  A 209.85.148.139
```



```
google.com.  29  IN  A 209.85.148.101  
google.com.  29  IN  A 209.85.148.102  
google.com.  29  IN  A 209.85.148.113  
google.com.  29  IN  A 209.85.148.100
```

```
;; Query time: 199 msec
```

```
;; SERVER: 10.9.11.21#53(10.9.11.21)
```

```
;; WHEN: Mon Feb 27 07:47:33 2012
```

```
;; MSG SIZE  rcvd: 124
```

# Topologie

## Protokolle

### HTTP (Hypertext Transfer Protocol)

- Überträgt Daten über ein Netzwerk
- Meist gebraucht im Browser
  - *Aber auch Transportschicht in **RESTful** Schnittstellen (vgl. **SOA/SOAP**)*
- Zustandlos
  - *Sessions sind Anwendungssache*

# Topologie

## Protokolle

### HTTP (Hypertext Transfer Protocol)

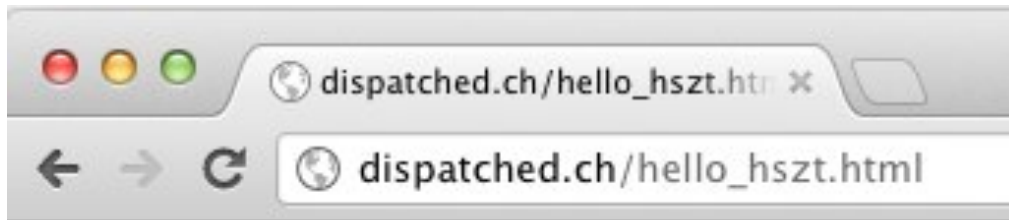
- Kennt pro Request Header Informationen und Status Codes
  - *Dadurch nicht beschränkt auf Hypertext*
  - *WebDAV etwa wird für den Austausch von Daten im Netz genutzt (Folder-Shares)*

# Topologie

## Protokolle

### HTTP (Hypertext Transfer Protocol)

#### Beispiel



Hello Students!

- Was geschieht hier im Hintergrund?

# Demo

```
$ telnet dispatched.ch 80
```

```
Trying 81.20.129.176...
```

```
Connected to dispatched.ch.
```

```
Escape character is '^['.
```

```
GET /hello_hszt.html HTTP/1.1
```

```
Host: dispatched.ch
```

```
HTTP/1.1 200 OK
```

```
Date: Wed, 29 Feb 2012 11:15:16 GMT
```

```
Server: Apache/2.2.16 (Debian) DAV/2 mod_fcgid/2.3.6 PHP/5.3.3-7 with
```

```
Suhosin-Patch mod_python/3.3.1 Python/2.6.6 mod_ssl/2.2.16 OpenSSL/0.9.8o
```

```
mod_wsgi/3.3 mod_scgi/1.13 Phusion_Passenger/3.0.7 mod_perl/2.0.4 Perl/v5.10.1
```

```
Last-Modified: Wed, 29 Feb 2012 11:12:34 GMT
```

```
ETag: "4828c49-10-4ba186b98cc80"
```

Accept-Ranges: bytes

Content-Length: 16

Vary: Accept-Encoding

Content-Type: text/html

Hello Students!

Connection closed by foreign host.

# Demo WTF?

## Exkurs telnet

- telnet ist ein Netzwerkprotokoll
- Es ermöglicht zeichenorientierten Datenaustausch zwischen Client und Server
  - *So kann man mit HTTP, SMTP (Mail), FTP (File Transfer), manuell sprechen*

# Was zeigt die Demo?

## Anfrage

- "telnet dispatched.ch 80"
  - *Verbindet zum Server auf Port 80 via TCP (out of scope -> Netzwerk Vorlesung)*
- "GET /hello\_hszt.html HTTP/1.1"
  - *Fordert die Datei 'hello\_hszt.html' vom Server an*
  - *Unter Verwendung von HTTP Version 1.1*
- "Host: dispatched.ch"
  - *Vervollständigt die Anfrage der Resource:  
[http://dispatched.ch/hello\\_hszt.html](http://dispatched.ch/hello_hszt.html)*



# Was zeigt die Demo?

## Antwort

- "HTTP/1.1 200 OK"
  - *Antwortet in HTTP Version 1.1 mit Status Code 200*
- Informationen zum Server: Datum, Version des Servers
- "Last-Modified" und "ETag"
  - *Caching*
- "Content-Type: text/html"
  - *Könnte sonst auch Javascript Bilder, Audio, Video, etc, senden*
- "Hello Students!"
  - *Eigentliche Nutzlast*

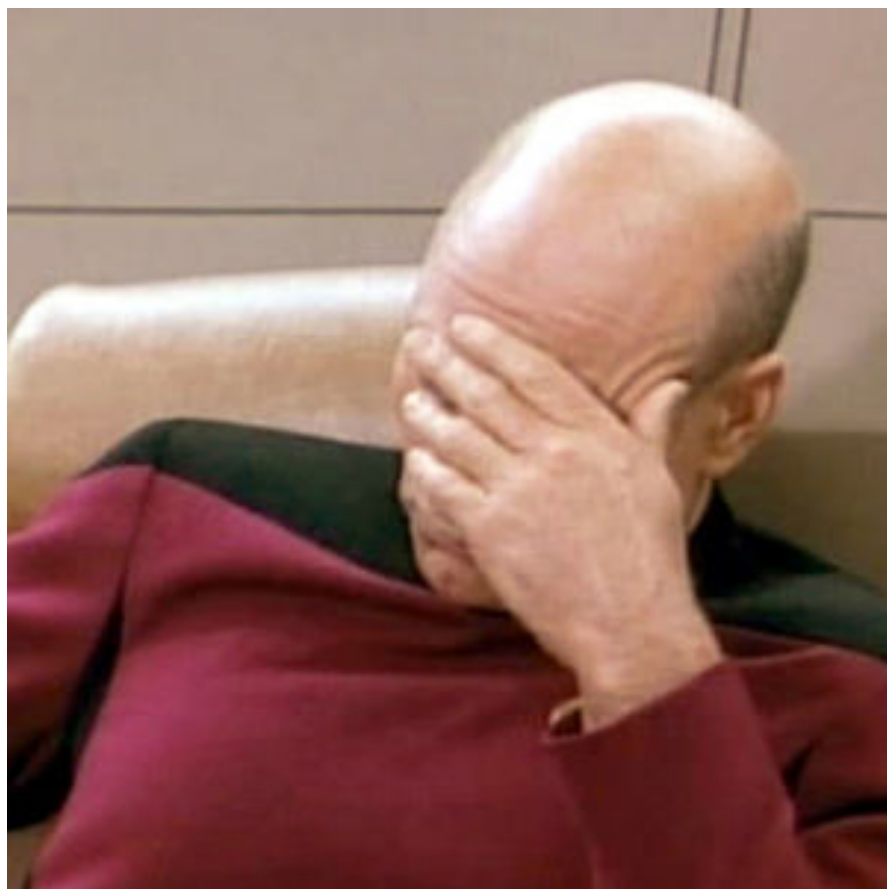
# Demo Ende

- Soweit der Ausblick zur Kommunikation mit einem Server
- Mehr dazu in Webprogrammieren II

**Engage your laptops!**

**Startet eure favourisierten  
Text-Editoren**

**Du hast keinen Text-  
Editor?!??**



# Text Editor Wahl-O-Mat

- Nutzt die Gelegenheit und lernt einen Editor wenn ihr noch keinen habt
- Bonus Punkte für VIM oder Emacs (alle Plattformen)
- OK sind auch
  - *TextMate (Mac)*
  - *gEdit (Linux)*
  - *NotePad++ (Windows)*
    - Nein, notepad.exe ist nicht ok und führt zu Punktabzug!

# An die nicht (VIM Emacs) Nutzer

- Erinnert euch an meine Worte wenn dieser Tag kommt
  - *Wichtige Semantik: 'wenn', nicht 'falls'(;*



**This doesn't work on eclipse**

**I should try vim**



# Erste HTML Datei

```
1.      <html>
2.      <head>
3.      <title>Webprogrammieren
I</title>
4.      </head>
5.      <body>
6.      <h1>Eine erste HTML Datei</h1>
7.      
8.      <p>
9.      Wat??
```

10.

`</p>`

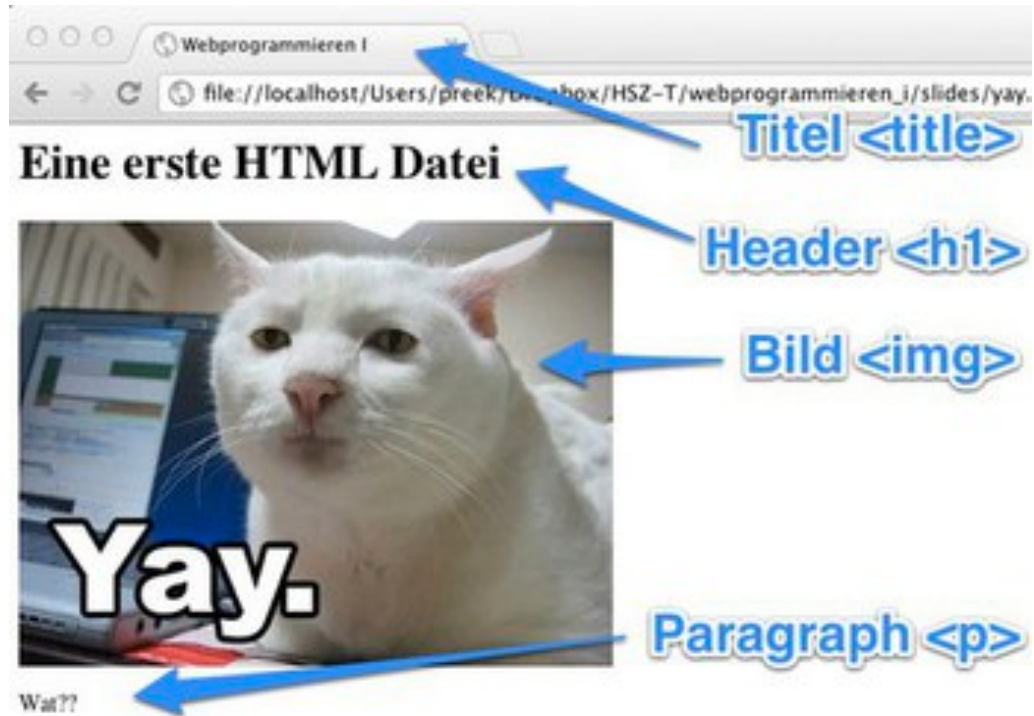
11.

`</body>`

12.

`</html>`

# Erste HTML Datei



# HTML

- HTML == Hypertext Markup Language
- Markup Language == Auszeichnungssprache
- => HTML beschreibt Semantik und Struktur von Daten
- Beides wird durch sog. "Tags" beschrieben

# HTML

## In der Demo verwandte Tags

- `<html>`
  - *Zeigt dem Browser an, dass es sich um ein HTML Dokument handelt*
- `<head>`
  - *Beinhält Meta-Information zur Webseite, etwa den Titel*
- `<body>`
  - *Beinhält den sichtbaren Inhalt der Webseite*

# HTML

## In der Demo verwandte Tags

- `<h1>`
  - *Überschrift*
- `<p>`
  - *Paragraph mit Text*

# HTML

## Tags

- `<h1>Meine Überschrift</h1>`
  - *Tags werden (immer) um Inhalt geschrieben*
  - *Tags können ineinander geschachtelt werden*
    - html -> head -> title
  - *Tags werden geöffnet und geschlossen*
    - Zugehörigkeit wird 'matching tag' genannt
  - *Tags mit ihrem Inhalt heissen Element*



# Browser Prüfung

- IE Nutzer?
- Firefox Nutzer alle Firebug installiert?

# HTML

## Style Tag

- HTML ist semantisch
- Styling/Repräsentation ist separiert

1.	<code>&lt;style type="text/css"&gt;</code>
2.	
3.	<code>&lt;/style&gt;</code>

# HTML

## Style Tag am Beispiel der ersten HTML Datei

```
1. <html>
2.   <head>
3.     ...
4.     <style type="text/css">
5.       body {
6.         font-family : "Lucida Grande";
7.         border       : solid 1px;
8.         height       : 400px;
9.         width        : 402px;
```

```
10.         padding      : 10px;
11.     }
12.     p {
13.         font-weight : bold;
14.         font-size   : 20px;
15.     }
16. </style>
17. </head>
18. ...
19. </html>
```

# Ergebnis

Eine erste HTML Datei



Wat??