leipzig school of media

Web-Engineering, Web-Anwendungssysteme und Mobile Computing I

Eine Vorlesung mit integrierter Übung im Modul
„Einführung in Aufbau und Nutzung von Internettechnologien (IT)“
des Studienganges
„Crossmedia Management“

Dr. Sebastian Tramp

Ziele des Moduls (offiziell)

leipzig school of media

"

Internettechnologien bilden die Basis crossmedialer Strategien. Die Studierenden erlernen diese komplexen Technologien und erlangen Kompetenzen zum technischen Design multifunktionaler Web-Systeme und erweitern ihr Wissen über Einsatzmöglichkeiten wichtiger Klassen von Web-Anwendungssystemen. Nach erfolgreichem Abschluss des Moduls kennen die Studierenden die technischen Grundlagen der Internettechnologien. Sie verfügen über Grundwissen im Bereich Softwaretechnologien für Webanwendungen und Mobile Computing. Sie können das Erlernte selbständig anwenden.

- Dozentenplan_IT-2_Tramp

Zusammenfassung (offiziell)

school
leipzig of media

„ *Baustein IT 2 vermittelt die Softwaretechnik für die Entwicklung von Web-Applikationen. Im Rahmen des Web-Engineering werden relevante Standards und Technologien vorgestellt. Darüber hinaus werden wichtige Klassen von Web-Anwendungssystemen eingeführt. Dazu zählen insbesondere ECommerce-Systeme, Virtuelle Welten, Lehr- und Lernsysteme. Ein weiterer Schwerpunkt ist die Vorstellung von Technologien zur Entwicklung ubiquitärer Anwendungen wie Apps und Informationssystem für mobile Endgeräte, Tablet-PCs etc.*

- https://www.leipzigschoolofmedia.de/…/aufbau-und-nutzung-von-internettechnologien.html

„ *Prüfungsleistung: Projektarbeit; Bearbeitungszeit: 4 Wochen; Themenausgabe: 16.03.2019; Abgabetermin: 13.04.2019*

Relevante Standards und Technologien

- HTML*, CSS* (+ SASS, LESS), SVG, JavaScript / ECMAScript, DOM, Flash, jquery + jquery ui, ⋯
- Semantic Web, RDFa, Microdata, schema.org
- HTTP, Ajax / Ajaj, JSON, JSONP, CORS, *RPC, ⋯
- PHP, Ruby, node.js, Java, ⋯

Wichtige Klassen von Web-Anwendungssystemen

- E-Commerce, Social Web / Social Network, CMS, Blogs, E-Learing, Virtuelle Welten, ⋯

Technologien zur Entwicklung von ubiquitärer Anwendungen

- iOS, Android, Windows Mobile, BlackBerry, Chrome OS, ⋯
- Native vs. Framework, PhoneGap / Apache Cordova, Jo, jQuery Mobile, Sencha Touch, M-Project, jQTouch, Titanium, ⋯

Themen dieses Bausteins - nach Tag

leipzig school of media

18.01.2018

- The Web / Big Picture / HTTP (Wie funktioniert das Web)
- HTML(5) (Wie erstellt man Webseiten)
- CSS (Wie styled man Webseiten)

08.02.2018 (?)

- Semantic Web Primer, schema.org
- Apache / PHP / WordPress Installation
- JavaScript

03.03.2018

- PHP / Datenbanken
- Magento
- Entwicklung für Mobile Devices
- Management von Software-Projekten
- Projektbesprechung

Doch bevor wir anfangen ...

leipzig school of media

Q/A

- Vorstellungsrunde
- Motivation
- Erwartungen
- Berufliche Ziele

The Web

In 1989, Tim Berners-Lee implemented the first successful communication between a Hypertext Transfer Protocol (HTTP) client and server via the Internet creating the world wide web. His visionary decision to make it a free and accessible resource for all, sparked a global revolution in communication … foreveryone.net

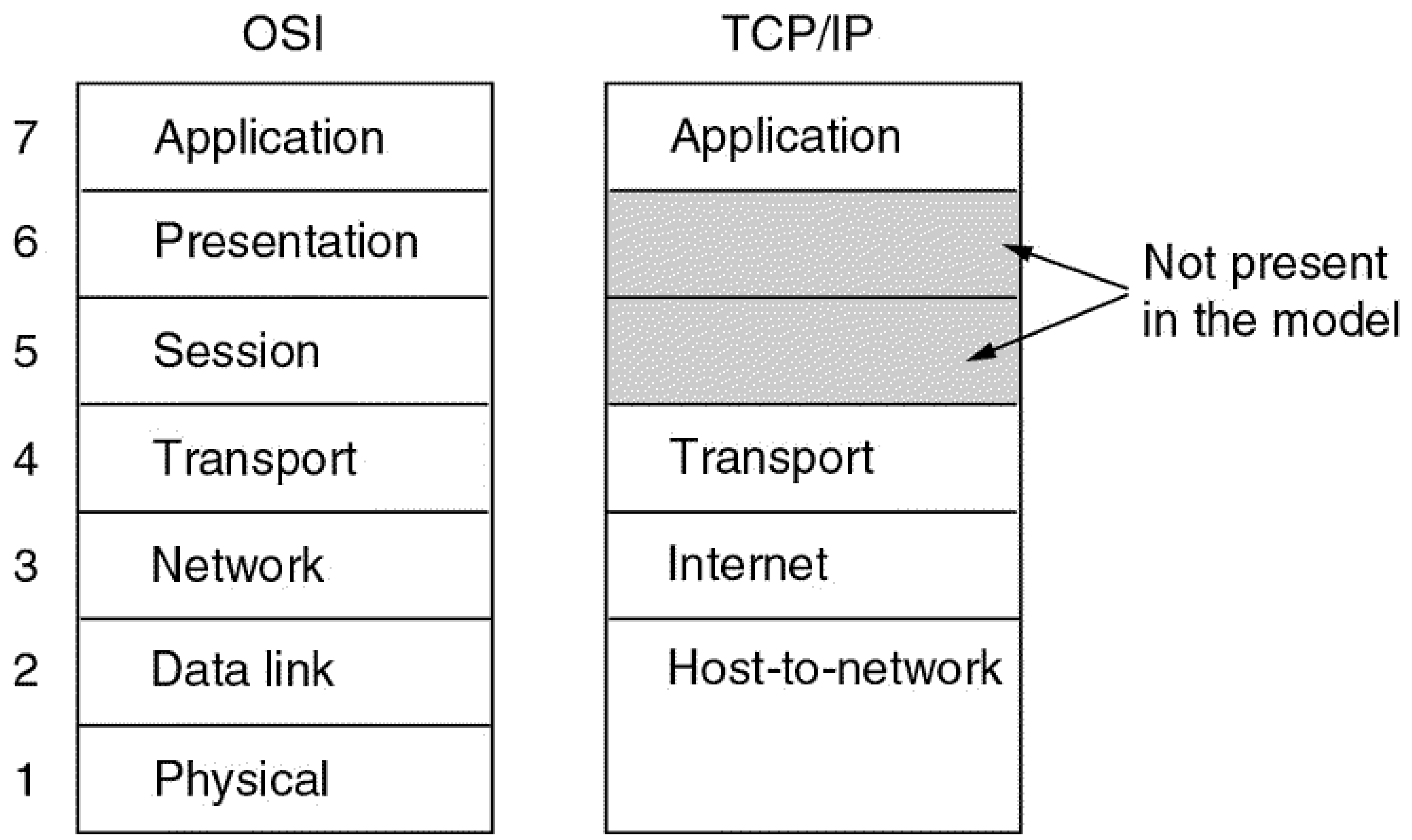
leipzig school of media

Hypertext Transfer Protocol (HTTP)

··· the mother of all communication on the World Wide Web ··· wait - not true anymore!

ISO/OSI vs. TCP/IP Protocol Stack

leipzig school of media



Via: marco.panizza.name

Basic Attributes

- HTTP is a stateless request - response protocol
- HTTP is an application layer protocol
- HTTP is a protocol to fetch and manipulate resources

Basic Workflow

- The client sends a request message
- The server answers with a response message
- since HTTP 1.1: persistent connections and pipelining
- since HTTP 2.0: multiplexing / simultaneous streams / server push and other speed enhancements (based on SPDY)

Typical Stakeholder

- Typical clients: browser or applications on your desktop computer or mobile device
  - but also HTTP(S) in file sync, operating system updates and other parts of your software stack
- Typical server: web server application on a host in a data center

HTTP Request Message: Overview

- Request line

```
GET /index.html HTTP/1.1
```

- Header

```
Accept-Language: en
```

- Message body (optional)

HTTP Request Line

leipzig school of media

```
GET /index.html HTTP/1.1
```

Verbs

- GET, HEAD, POST, OPTIONS (the most important ones)
- PUT, DELETE, TRACE, CONNECT, PATCH

Requested Resource

The local part of an URL (without fragment identifier)

- URL: Uniform Resource Locator

  ```
  scheme://domain:port/path?query_string#fragment_id
  ```

  ```
  http://www.bing.com/search?q=Leipzig+School+of+Media
  ```

Concepts

- An URL identifies a "resource"
- A resource is anything identified by an URL
- Resources may have representations, which can be received by dereferencing the URL
- Representations are concrete data and may vary with time
- Resources are abstact
- Please distinguish between information resources vs. non-information resources !!!

Usage

- Resources are used to define an information space
- URIs do not only identify resources; they can be used to access and interact with resources as well

HTTP Request Header

- name-value pairs
- colon separated
- clear text
- used for content negotiation (see: quality values)
- List of HTTP header fields

Example Request

```
HEAD / HTTP/1.1
User-Agent: curl/7.21.3 (i686-pc-linux-gnu)
Host: google.de
Accept: */*
```

HTTP Response Example

Tool: cURL

```
curl(.exe) -X HEAD -v http://google.de
```

Output

```
HTTP/1.1 301 Moved Permanently
Location: http://www.google.de/
Content-Type: text/html; charset=UTF-8
Date: Thu, 17 Nov 2011 00:21:37 GMT
Expires: Sat, 17 Dec 2011 00:21:37 GMT
Cache-Control: public, max-age=2592000
Server: gws
Content-Length: 218
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
```

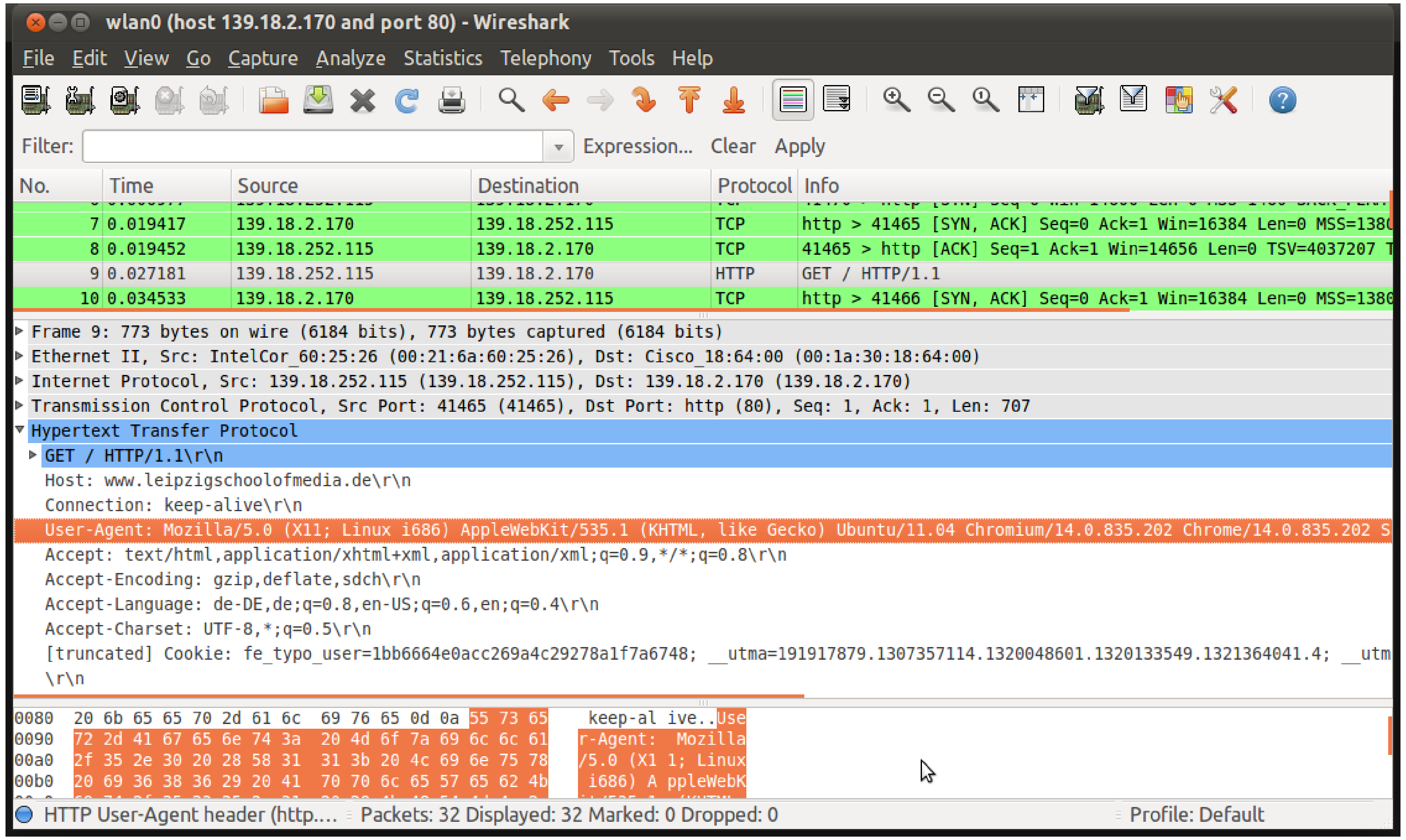
Important Status Codes

- 200 OK ··· everything is fine
- 301 Moved Permanently ··· change your bookmark
- 303 See Other ··· follow the light
- 401 Unauthorized ··· please send credentials
- 403 Forbidden ··· not for you
- 404 Not Found ··· broken link
- 500 Internal Server Error ··· something is wrong

see also: List of HTTP Status Codes

Practice: Deep-Inspection of your Browser Communication

leipzig school of media


wlan0 (host 139.18.2.170 and port 80) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply

```
No.  Time      Source          Destination     Protocol Info
  7 0.019417  139.18.2.170    139.18.252.115  TCP      http > 41465 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=138
  8 0.019452  139.18.252.115  139.18.2.170    TCP      41465 > http [ACK] Seq=1 Ack=1 Win=14656 Len=0 TSV=4037207 T
  9 0.027181  139.18.252.115  139.18.2.170    HTTP     GET / HTTP/1.1
 10 0.034533  139.18.2.170    139.18.252.115  TCP      http > 41466 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=138

▶ Frame 9: 773 bytes on wire (6184 bits), 773 bytes captured (6184 bits)
▶ Ethernet II, Src: IntelCor_60:25:26 (00:21:6a:60:25:26), Dst: Cisco_18:64:00 (00:1a:30:18:64:00)
▶ Internet Protocol, Src: 139.18.252.115 (139.18.252.115), Dst: 139.18.2.170 (139.18.2.170)
▶ Transmission Control Protocol, Src Port: 41465 (41465), Dst Port: http (80), Seq: 1, Ack: 1, Len: 707
▼ Hypertext Transfer Protocol
  ▶ GET / HTTP/1.1\r\n
    Host: www.leipzigschoolofmedia.de\r\n
    Connection: keep-alive\r\n
    User-Agent: Mozilla/5.0 (X11; Linux i686) AppleWebKit/535.1 (KHTML, like Gecko) Ubuntu/11.04 Chromium/14.0.835.202 Chrome/14.0.835.202 S
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Encoding: gzip,deflate,sdch\r\n
    Accept-Language: de-DE,de;q=0.8,en-US;q=0.6,en;q=0.4\r\n
    Accept-Charset: UTF-8,*;q=0.5\r\n
    [truncated] Cookie: fe_typo_user=1bb6664e0acc269a4c29278a1f7a6748; __utma=191917879.1307357114.1320048601.1320133549.1321364041.4; __utm
    \r\n

0080  20 6b 65 65 70 2d 61 6c  69 76 65 0d 0a 55 73 65    keep-al ive..Use
0090  72 2d 41 67 65 6e 74 3a  20 4d 6f 7a 69 6c 6c 61   r-Agent:  Mozilla
00a0  2f 35 2e 30 20 28 58 31  31 3b 20 4c 69 6e 75 78   /5.0 (X1 1; Linux
00b0  20 69 36 38 36 29 20 41  70 70 6c 65 57 65 62 4b    i686) A ppleWebK

HTTP User-Agent header (http.... Packets: 32 Displayed: 32 Marked: 0 Dropped: 0    Profile: Default
```

Challenge: Use wireshark to analyze what your browser sends and receives.
Limit the network traffic capturing to HTTP communication with the University of Leipzig server.
Review the sent and received HTTP packets and notice, which HTTP header fields your browser creates.

Resources: Web Architecture / HTTP

- ForEveryone.net | The web, past and future; video by the web foundation
- Architecture of the World Wide Web, Volume One; W3C Recommendation 15 December 2004 „… *this architecture document discusses the core design.“*
- Web Architecture and Information Management; Course from the School of Information, UC Berkeley
- Wikipedia pages about:
    - Uniform Resource Locator
    - Hypertext Transfer Protocol
    - WebDAV (to look beyond basic HTTP)
- wireshark - network protocol analyzer
- curl - command line networking tool
- Live HTTP headers - Firefox plugin to watch HTTP headers

HTML

Wakeup please! :-)

HTML Introduction

HTML is the acronym for HyperText Markup Language

- Hypertext: text with hyperlinks
  - Hyperlinks: references which can be used for more than text references (comes from the Greek prefix "υπερ-" and means "over" or "beyond")
  - Both terms where coined by Ted Nelson and Douglas Engelbart influenced by the article As We May Think by Vannevar Bush from 1945 (!)
- Markup: annotations in a document's content
- Language: an artificial language designed to express computations

HTML was invented by Tim Berners-Lee in 1991 and is now maintained by the W3C and (since HTML5) the WHATWG.

HTML Markup Basics: Tags

Plain text is annotated with tag elements:

```
not annotated text <tagname>annotated text</tagname> not annotated text
```

Tag elements can be used nested:

```
<ol>
    <li>plain list item</li>
    <li>list item <q>with <span>wrong</q></span> nesting</li>
</ol>
```

HTML Markup Basics: attributes

Tag elements can be extended by attributes

```
<tagname attribute1="value1" attribute2="value2">text</tagname>
<a href="http://aksw.org/">AKSW Homepage</a>
```

Some attributes can be used with all tags, some only with specific tags:

- Global attributes:
  - class, id, title, style, dir, …
  - + data-* (custom data attributes)
  - + on* (event handler attributes)
- Attributes for specific tag elements:
  - e.g. href, media, rel are specifically allowed for the anchor tag a
  - please refer this list of tag elements

HTML Document Structure

leipzig school of media

A basic HTML(5) document is structured like this:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Sample page</title>
  </head>
  <body>
    <h1>Sample page</h1>
    <p>This is a <a href="demo.html">simple</a> sample.</p>
    <!-- this is a comment -->
    </body>
</html>
```

document = doctype declaration + HTML document (= HTML head + HTML body)

Physical vs. Logical Markup

Physical Markup

- Tag elements which do strictly define the style: `b` (bold), `big` (font bigger)

Logical / Semantic Markup

- Tag elements which annotate a meaning and foster separation between style and structure
- Example: `em` or `strong` instead of `b`
- HTML5 introduces a lot new semantic tag elements: article, section, header / footer, nav, aside, figure / figcaption
- `<time datetime="2014-02-13">13.02.2014</time>`

Try to avoid physical markup completely!

- Logical Inline Tags vs. Physical Inline Tags
- 10 HTML Tag Crimes You Really Shouldn't Commit
- Tableless Web Design

HTML Entities

Reserved characters must be replaced with character entities

Most important entities

- `<` (less than): `&lt;`
- `>` (greater than): `&gt;`
- `&` (ampersand): `&amp;`

Other enties

List of HTML Character entity references

Specific Tags: h*, p, a

Headings

Headings are defined with the `h1` (most important) to `h6` (least important) tags.

```
<h1>Leipzig School of Media<h1>
```

Paragraphs

```
<p>A nice little paragraph<p>
```

Links

```
<a href="http://aksw.org">AKSW</a>
```

Specific Tags: div, span

leipzig school of media

div

div elements are used to structure the page (where no other structure element is suitable)

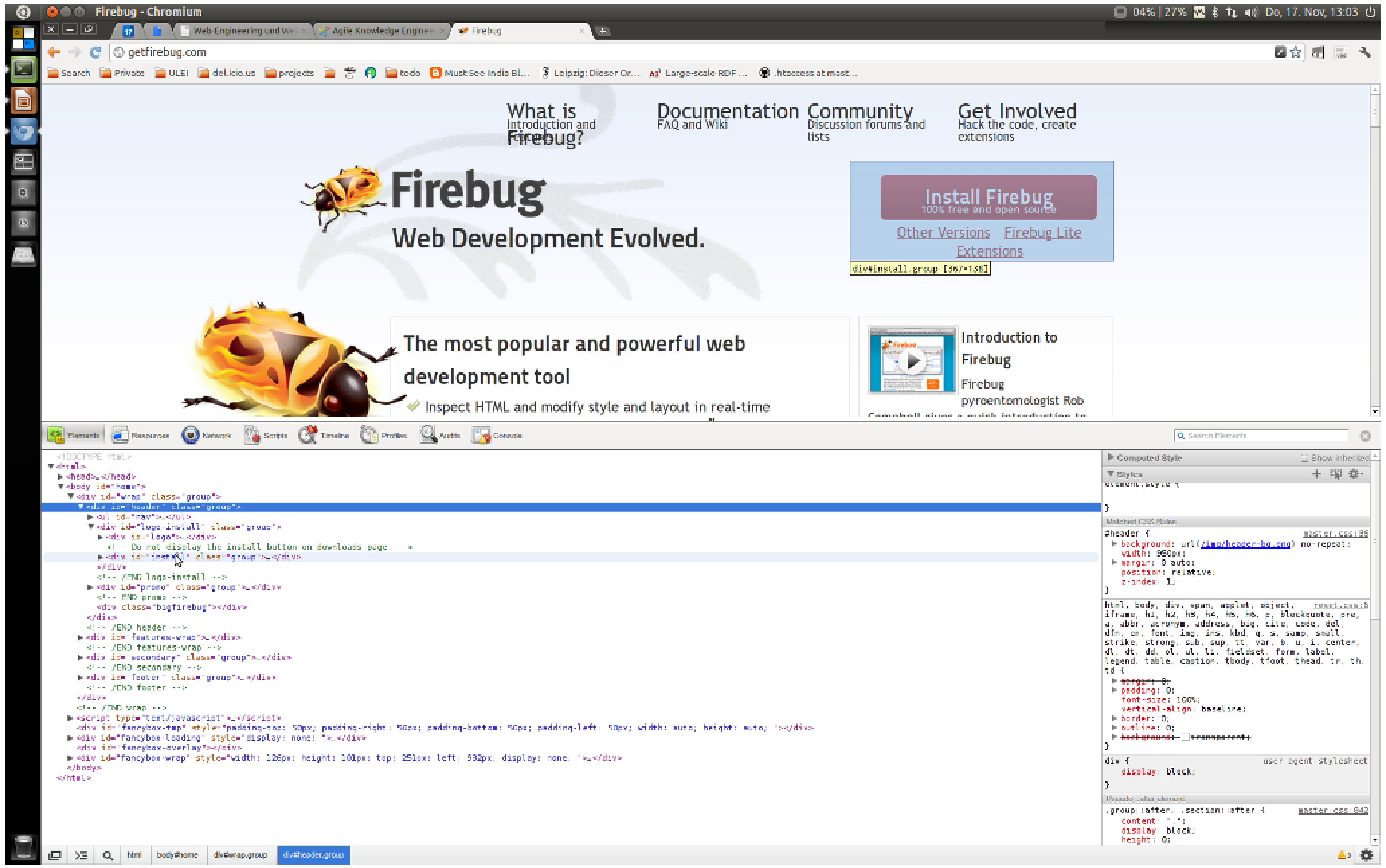
```
<div class="conclusion"><p>...</p><p>...</p></div>
```

span

span elements are used to annotate the text inline (where no other inline element is suitable)

```
The concert took place in <span class="location">Leipzig</span>.
```

Practice: HTML Inspection with Firebug or Chrome Developer Tools



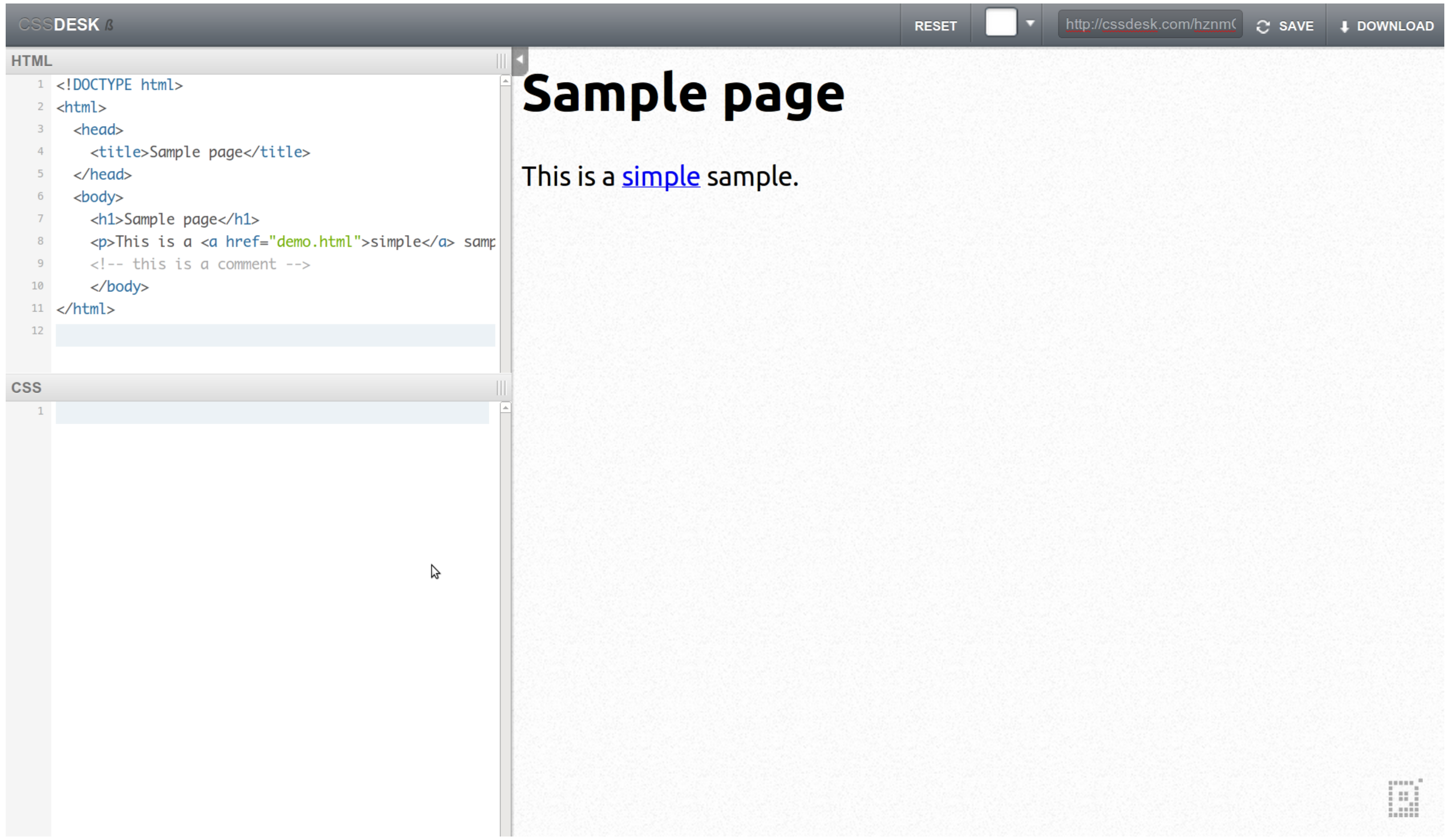
```
<!DOCTYPE html>
▼<html>
  ▶<head>…</head>
  ▼<body id="home">
    ▼<div id="wrap" class="group">
      ▼<div id="header" class="group">
        ▶<ul id="nav">…</ul>
        ▼<div id="logo-install" class="group">
          ▶<div id="logo">…</div>
           <!-- Do not display the install button on downloads page. -->
          ▶<div id="install" class="group">…</div>
         </div>
         <!-- /END logo-install -->
        ▶<div id="promo" class="group">…</div>
         <!-- END promo -->
         <div class="bigfirebug"></div>
       </div>
       <!-- /END header -->
      ▶<div id="features-wrap">…</div>
       <!-- /END features-wrap -->
      ▶<div id="secondary" class="group">…</div>
       <!-- /END secondary -->
      ▶<div id="footer" class="group">…</div>
       <!-- /END footer -->
     </div>
     <!-- /END wrap -->
    ▶<script type="text/javascript">…</script>
     <div id="fancybox-tmp" style="padding-top: 50px; padding-right: 50px; padding-bottom: 50px; padding-left: 50px; width: auto; height: auto; "></div>
    ▶<div id="fancybox-loading" style="display: none; ">…</div>
     <div id="fancybox-overlay"></div>
    ▶<div id="fancybox-wrap" style="width: 126px; height: 101px; top: 251px; left: 932px; display: none; ">…</div>
   </body>
</html>
```

```
element.style {
}
#header {                                        master.css:95
  ▶ background: url(/img/header-bg.png) no-repeat;
    width: 960px;
  ▶ margin: 0 auto;
    position: relative;
    z-index: 1;
}
html, body, div, span, applet, object,           reset.css:8
iframe, h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code, del,
dfn, em, font, img, ins, kbd, q, s, samp, small,
strike, strong, sub, sup, tt, var, b, u, i, center,
dl, dt, dd, ol, ul, li, fieldset, form, label,
legend, table, caption, tbody, tfoot, thead, tr, th,
td {
  ▶ ~~margin: 0;~~
  ▶ padding: 0;
    font-size: 100%;
    vertical-align: baseline;
  ▶ border: 0;
  ▶ outline: 0;
  ▶ ~~background: transparent;~~
}
div {                                    user agent stylesheet
    display: block;
}
.group::after, .section::after {                 master.css:842
    content: ".";
    display: block;
    height: 0;
```

Challenge: Install Firebug for Mozilla Firefox or use the Chrome Developer Tools to inspect the webpage of your choice. Try to find physical markup and other bad tag or table based layouts.

school
leipzig of media

Practice: Use Dabblet play around with HTML Tag Elements



```
<!DOCTYPE html>
<html>
  <head>
    <title>Sample page</title>
  </head>
  <body>
    <h1>Sample page</h1>
    <p>This is a <a href="demo.html">simple</a> samp
    <!-- this is a comment -->
    </body>
</html>
```

Sample page

This is a simple sample.

Challenge: Use cssdesk.com and play around with different HTML tag elements.
This should include: headings, paragraphs, links and lists

HTML Resources

These resource will help you as tutorials, references and on-line tools:

- cssdesk.com (Online HTML/CSS Sandbox)
- Detailed list of HTML elements (Reference)
- SelfHTML (Tutorial / Reference / German)
- Markup validation service @ W3C (Online service / tool)
- HTML @ w3schools.com (Tutorial)
- List of HTML Character entity references (Reference)

leipzig school of media

CSS

Cascading Style Sheets

Übung: Wie kann dieses Printmedium im Web umgesetzt werden?

leipzig school of media

- Anordnung ohne Tabellen (semantic ui, Bootstrap, 960gs, blueprint)
- Was ist im Web sinnvoll, was nicht?
- Responsive Web Design? Mobile Devices?

Süddeutsche Zeitung
AM WOCHENENDE
Einer für alle
Terrorfahndung im Alpenvorland
Kölns Polizeichef muss abtreten
„Mein Kampf“ bereits ausverkauft
Keilerei um den Keiler

CSS Resources

- LSoM Slides (PDF)
- SelfHTML (oldschool but still recommendable)
- Helper:
    - CSS Validation Service
    - CSS Desk / dabblet (text sandbox)
    - CSS 3.0 maker (graphical sandbox)
    - caniuse.com (browser issue directory)
    - Google Web Fonts

Thank you for your Attention!

My WebID:

- http://sebastian.tramp.name