



# Einführung in die Informationstechnik I

Veranstaltungsskript

Dozent: Axel Dürkop

Institut für Technische Bildung und Hochschuldidaktik (iTBH)

TU Hamburg

Wintersemester 2016/17

---

# Inhaltsverzeichnis

## Allgemeines

Über diese Veranstaltung	1.1
Konzept der Veranstaltung	1.2
Ziele und Inhalte	1.3
Prüfung	1.4

## Voraussetzungen

Das Werkzeug	2.1
Mailadresse	2.2
GitHub-Account	2.3
TU-Account	2.4
Hypothes.is-Account	2.5
Zotero-Account	2.6
Bibliotheksausweis	2.7
Eigener Rechner	2.8
Software	2.9
Zusammenfassung	2.10

## Sitzungen

Fr, 04.11.2016	3.1
Sa, 05.11.2016	3.2
Fr, 18.11.2016	3.3
Fr, 25.11.2016	3.4
Fr, 02.12.2016	3.5
Fr, 09.12.2016	3.6
Fr, 16.12.2016	3.7
Fr, 23.12.2016	3.8
Fr, 13.01.2017	3.9
Fr, 20.01.2017	3.10
Fr, 27.01.2017	3.11
Fr, 03.02.2017	3.12

---

# Material

Webtechnologien	4.1
HTML5	4.1.1
CSS3	4.1.2
Analyse Datenverkehr	4.1.3
Atom	4.2
Packages	4.2.1
Linux-Quickstart	4.3

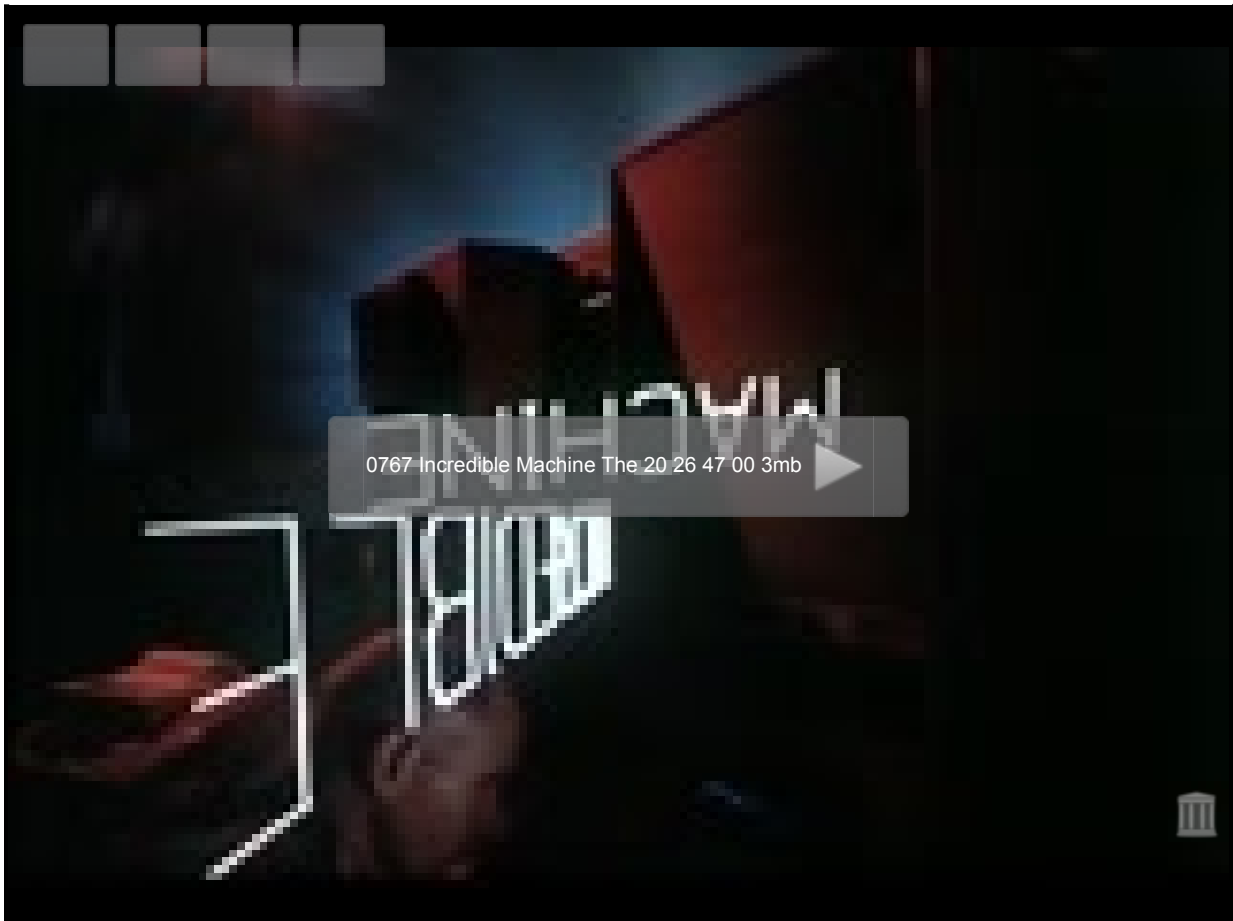
# Literatur

Bücher	5.1
--------	-----

# Einführung in die Informationstechnik I

Begleitendes Skript für die [Veranstaltung Einführung in die Informationstechnik](#) an der [TU Hamburg](#),  
Fachbereiche Elektrotechnik/Informationstechnik und Medientechnik am [Institut für Technische Bildung  
und Hochschuldidaktik \(iTBH\)](#).

Dozent: [Axel Dürkop](#)



Film: "Incredible Machine, The" by [Murphy \(Owen\) Productions](#), veröffentlicht 1968, Lizenz: [Public Domain Mark 1.0](#)

# Konzept der Veranstaltung

Die konsekutiven Veranstaltungen *Einführung in die Informationstechnik I* und *Einführung in die Informationstechnik II* nähern sich dem Themenkomplex *Informationstechnik* aus verschiedenen Perspektiven.

## Allgemeine Perspektive

So sollen zunächst **aktuelle Trends der Informations- und Telekommunikationstechnik** identifiziert werden. Dabei steht das Internet in seinen zahlreichen Facetten als technisches Konstrukt im Mittelpunkt. Relevante Schnittmengen von Gesellschaft/Ethik/Politik/Informationstechnik werden ebenfalls ermittelt.

## Berufliche Perspektive

Nachfolgend sollen **Ordnungsmittel der Berufe aus den Fachrichtungen** gesichtet werden, um die Bedeutung informationstechnischen Wissens aus den Perspektiven von Lehrenden und Lernenden zu ermitteln.

## Persönliche Perspektive

Aus einem weiteren Blickwinkel betrachten die Studierenden den Themenkomplex Informationstechnik **vor dem Hintergrund persönlicher Neigungen und Interessen**. Sie wählen ein Thema aus, das sich sowohl fachlich-theoretisch wie praktisch erschließen lässt.

## Teilen und Kollaboration

Ein Thema, das die beiden Semester überspannen wird, sind *Teilen* und *Kollaboration*. Im Hinblick auf das Studienziel *Gewerbelehramt* werden wir an uns und mit uns selbst die Vor- und Nachteile internetgestützter Zusammenarbeit ausprobieren. Dazu werden verschiedene (Internet)tools vorgestellt und eingeführt, die einer kritischen Betrachtung während und nach dem Ausprobieren unterzogen werden.

# Veranstaltungsziele und -inhalte

Veranstaltungsziele und -inhalte richten sich in dieser Veranstaltung nach den [Fachspezifischen Bestimmungen \(FSB\)](#) der TUHH für diesen Studiengang.

Dort sind zum einen Qualifikationsziele angegeben, die sich über die beiden Veranstaltungen *Einführung in die Informatik* erstrecken. Außerdem finden sich dort Inhalte, aus denen abgelesen werden kann, in welcher Breite und Tiefe Informatik im Bezug auf das Studienziel *Gewerbelehramt* behandelt werden wird.

## Auftrag

Beschäftigen Sie sich mit den FSB insofern, als dass Sie

- Ihre Vorkenntnisse mit den Anforderungen in Beziehung setzen. Was können und kennen Sie schon?
- unbekannte Begriffe recherchieren und sich ein grobes Verständnis von ihrer Bedeutung erarbeiten
- die [Prüfungsformen](#) zur Kenntnis nehmen und sich überlegen, was das für Ihre individuelle Vorbereitung bedeutet

# Prüfungsformen und -voraussetzungen

## Prüfungsformen

### Mündliche Prüfung

Gemäß den [FSB](#) ist folgende Prüfungsleistung zu erbringen:

- mündliche Prüfung (20-30 Min.) *Einführung in die Informationstechnik I*

Die Prüfung ist bezogen auf die Inhalte, Fragestellungen, Arbeitsaufträge und Projekte, die im Kontext der Veranstaltung bearbeitet wurden.

### Studienleistung

Als Studienleistung ist die termingerechte Erledigung der gestellten Aufgaben und Aufträge zu erbringen. Von einer aktiven Mitarbeit an der Veranstaltung wird ausgegangen. Das erfolgreiche Bestehen der Studienleistungen (nicht benotet) ist notwendig für das Bestehen der Prüfung.

## Empfehlungen

### Empfehlung zur Mitarbeit

Das zeitnahe, gewissenhafte und kontinuierliche **Nachbereiten** der Sitzungsinhalte, das zeitnahe und gründliche **Bearbeiten** der Arbeitsaufträge sowie eine rechtzeitige Vorbereitung auf die mündliche Prüfung sind für das erfolgreiche Absolvieren des Teilmoduls zu empfehlen.

### Empfehlung zur Dokumentation

Auf die Inhalte und Themen aus den beiden Informatiksemestern wird in den kommenden Semestern ggf. Bezug genommen werden. Daher ist eine gute **schriftliche Dokumentation** der Inhalte ratsam.

## Werkzeug und Material

Die Veranstaltung hat keine Zugangsvoraussetzungen. Allerdings brauchen wir einiges, um sinnvoll und effektiv zu arbeiten. Die notwendigen Voraussetzungen sind im folgenden genannt und mit Arbeitsaufträgen versehen.

## Abgabetermin

Die **kompletten Aufträge** des Abschnitts *Voraussetzungen* sind bis zum ~~Vorabend der dritten Veranstaltung (11.11.2016)~~ 17.11.2016 zu erledigen.



## Mailadresse für Lernzwecke

Da wir uns mit verschiedenen Tools und Plattformen beschäftigen werden, wird empfohlen, hierfür eine spezielle Mailadresse einzurichten. Manche Plattformen erzeugen viele Benachrichtigungen, die bisherige Teilnehmende nicht gern in ihrem regulären Postfach haben wollten.

## Auftrag

- Legen Sie sich eine spezielle Mailadresse an, Sie werden sie zukünftig häufiger nutzen können.
- Wählen Sie ggf. auch einen unverständlichen Benutzernamen, wenn Ihnen ein gewisser Grad von Anonymität wichtig ist. Klarnamenzwang gibt es bei mir nicht.
- Dienste, die einen hohen Schutz der Privatsphäre garantieren, finden sich z.B. in [dieser guten Sammlung freier Tools und Dienste](#).

# GitHub-Account

[GitHub](#) ist eine Website, auf der Menschen aus der ganzen Welt kollaborativ an Code und Texten arbeiten können. Viele Programme, die wir täglich nutzen und die aus dem Internet nicht wegzudenken sind, [werden mit Hilfe von GitHub entwickelt](#). Viele davon sind Open Source oder sogar [Freie Software](#).



Abbildung 2.3.1 - Wie heißt dieses Tier?

Ich bin überzeugt, dass jeder, der heute beginnt, sich für informatische Inhalte und Zusammenhänge zu interessieren, an GitHub nicht vorbei kommt. Denn hinter GitHub steckt nicht nur eine weitere Internetplattform. Vielmehr etablieren die Konzepte hinter GitHub eine neue Kulturtechnik der Kollaboration und des Teilens, die auch für Lehr-Lernzusammenhänge nutzbar gemacht werden kann. Nicht ohne Grund gibt es daher auch ein [GitHub Education Programm](#).

Wir werden uns mit GitHub vertraut machen und uns langsam die Funktionen dieser Plattform erschließen. Dafür ist es notwendig, dass sich alle Studierenden ein GitHub-Account zulegen.

## Auftrag

- Registrieren Sie sich bei GitHub für einen freien Account.
  - Wir werden GitHub in einer geschützten Umgebung verwenden. D.h. Ihre Arbeit darin wird im Zusammenhang mit der Veranstaltung nicht öffentlich sein.
  - Wählen Sie ggf. eine [Email-Adresse, die Sie nur für die Arbeit mit Plattformen und Tools im Studium verwenden werden](#).
  - Lesen Sie dabei die Terms of Service sowie die Datenschutzerklärung von GitHub gründlich

durch, wenn Sie sich registrieren.

- **Sichern Sie Ihre Zugangsdaten! Wir werden viel mit GitHub arbeiten.** Da wir noch weitere Account anlegen werden, organisieren Sie die Verwaltung Ihrer Zugangsdaten gleich richtig, z.B. mit [KeepassX](#).
- Lesen Sie nach der Registrierung den [Guide](#), um sich einen Überblick zu verschaffen, was der hauptsächliche Zweck von GitHub ist.
- **Wichtig!** Senden Sie mir per Mail Ihren Nutzernamen bei GitHub zu, damit ich Sie zu einer Arbeitsgruppe hinzufügen kann.

# TU-Account

Für die Nutzung bestimmter Dienst im Rahmen der TU sind **eine TU-Mailadresse** sowie ein individueller **Nutzeraccount an der TU** notwendig.

Sofern nicht schon vorhanden, erhalten Sie beides beim [User Service Center \(USC\) der TU](#).

## Auftrag

Besorgen Sie sich einen Nutzeraccount der TU. Halten Sie ihn für den Gebrauch im Rahmen der Veranstaltung bereit.

# Hypothes.is-Account

[Hypothes.is](#) ein Browsertool, mit dem sich prinzipiell *alle* Webseiten unseres Internets *annotieren* lassen. Konkret: Man macht eine Markierung auf einer Webseite und schreibt etwas dazu, einen Kommentar, eine Anmerkung oder einen Verbesserungsvorschlag. Der Clou: Man bekommt so beim Browsen auch die Gedanken anderer zu lesen, kann darauf eingehen und die Inhalte einer Webseite zum gemeinsamen Nachdenken nutzen.

Das Potenzial von Hypothes.is für Unterricht und Lehre ist groß, die Homepage [gibt eine Einführung in die Möglichkeiten für Lernende und Lehrende](#). Da wir gemeinsam Grundlagentexte zur Informationstechnik und zum Internet lesen und darüber sprechen werden, möchte ich Hypothes.is in der Veranstaltung ausprobieren.

Hypothes.is wird als Dienst in den USA angeboten, was kritisch zu sehen ist (vgl. [Terms of Service](#)). Daher bitte ich um eine Diskussion im Rahmen der Veranstaltung, ob wir uns darauf einlassen, oder eine eigene Installation von Hypothes.is im deutschen Rechtsraum wollen, denn Hypothes.is wird als Freie Software auf [GitHub](#) entwickelt.

## Auftrag

Diskutieren Sie mit Kommilitonen, wie wir Hypothes.is einsetzen wollen:

- als gehostete Version in den USA
- als selbst installierte Version am Rechenzentrum der TUHH

# Zotero-Account

[Zotero](#) ist ein Recherche- und Literaturverwaltungsprogramm. Es kann als Browserplugin oder Standalone-Programm verwendet werden, ist [Freie Software](#) und läuft auf allen drei gängigen Betriebssystemen (Linux, Mac OS, Windows).

Zotero bietet zum einen die Möglichkeit, vom ersten Semester an Literatur, die man lesen sollte oder gelesen hat, zu erfassen und zu kommentieren. Zum andern kann Literatur über [zotero.org](https://zotero.org) auch geteilt werden.

Wir werden eine Gruppe für unsere Veranstaltung anlegen und gemeinsam Literatur zu unseren Themen zusammentragen.

## Auftrag

Registrieren Sie sich bei Zotero und teilen Sie dem Dozenten die verwendete Mailadresse mit.

# Bibliotheksausweis

Die [Bibliothek der TUHH \(TUB\)](#) hält zahlreiche Bücher und Publikationen zu technischen Themen vor. Viele davon sind mittlerweile in digitalen Formaten erhältlich.

Mit einem Benutzerausweis der TUB erhalten Sie Zugang zu diesem Bestand, sogar von zu Hause.

## Auftrag

Besorgen Sie sich einen Bibliotheksausweis der TUB.

# Eigener Rechner

Sicherlich haben Sie einen eigenen Rechner für das Studium. In den beiden Informatikveranstaltungen wird dieser eine entscheidende Rolle spielen, da Sie mit ihm viele Aufträge erledigen werden müssen.

Es ist daher notwendig, dass Sie Ihren Rechner gut kennen bzw. kennen lernen. In späteren Unterrichtssituationen sind Sie immer wieder gefragt, mit unterschiedlichsten Rechnertypen, Betriebssystemen und Softwareprogrammen umzugehen. Daher ist die sehr gute Kenntnis des eigenen Systems in den Bereichen Software und Hardware unabdingbar.

## Rechner im Seminarraum

Es ist auf jeden Fall sinnvoll und ratsam, den eigenen Rechner für die Veranstaltung zu nutzen, da oft Aufträge in der Veranstaltung angefangen werden, die dann zu Hause beendet werden sollen. Die Erfahrung zeigt, dass der Austausch von Daten zwischen mehreren Rechnern fehlerbehaftet, unpraktisch und langwierig sein kann.

Sollten Sie über keinen eigenen Rechner verfügen, können Sie ein Windows-System im Seminarraum nutzen.

## Auftrag

- Bringen Sie Ihren Rechner stets mit in die Veranstaltung, sofern er tragbar ist :-).
- Erstellen Sie ein Datenblatt zu Ihrem persönlichen "Gerätepark":
  - Rechnertyp und -modell
  - Betriebssystem und -version
  - weitere technische Spezifikationen wie Arbeitsspeicher, Prozessorleistung etc.
  - Auf dem Rechner installierte Software und Software, mit der Sie im Beruf hauptsächlich gearbeitet haben
  - Kleinstcomputer wie Tablets, Handys und Smartphones, die Sie verwenden
  - Betriebssysteme und -versionen auf diesen Geräten
  - Router und Internetzugang zu Hause



## Eingesetzte Software

Im Kontext der Veranstaltung setze ich weitestgehend auf [Open-Source-Software](#) und [Freie Software](#), weil es für mich selbstverständlich ist, dass Software, die in Lehr-Lernzusammenhängen eingesetzt wird, kostenlos und frei ist. Daher werde ich zur Erledigung der Arbeitsaufträge immer ein Programm vorschlagen, dass diese Kriterien erfüllt.

## Software, mit der wir arbeiten werden

- [Atom](#)
- [Zotero](#)
- [Chromium](#)
- [GIMP](#)
- [Inkscape](#)

## Offene Standards

- [HTML5](#)
- [CSS3](#)
- [Markdown](#)

## Auftrag

Installieren Sie [Atom](#) auf Ihrem Rechner.

# Zusammenfassung

Wenn Sie alle Aufträge aus dem Bereich *Voraussetzungen* erledigt haben, sollte bei Ihnen folgendes vorhanden sein:

- ☐ Mailadresse für Lernzwecke\*
- ☐ GitHub-Account\*
- ☐ TU-Account
- ☐ eine Meinung zur Nutzung von Hypothes.is
- ☐ Zotero-Account\*
- ☐ Bibliotheksausweis
- ☐ Datenblatt der persönlichen technischen Umgebung\*
- ☐ Atom auf dem eigenen Rechner

\*) Mitteilung an das Dozententeam

## **[01] Fr, 04.11.2016**

### **Agenda**

- Vorstellungsrunde und Organisatorisches
- Gruppenarbeit: Nutzung digitaler Medien in Alltag und Beruf
- Arbeitsaufträge für die nächste Sitzung

### **Gruppenarbeit**

Tragen Sie mindestens zu zweit zusammen, was Ihnen zu folgender Fragestellung einfällt:

"Welche (digitalen) Medien nutzen Sie privat und im Beruf?"

Halten Sie das Ergebnis digital fest und senden Sie es per Mail an das Dozententeam.

## [02] Sa, 05.11.2016 (Doppelsitzung)

```
<!DOCTYPE HTML>
<html lang="de" >
  <head>
    <meta charset="UTF-8">
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
    <title>HTML5 · GitBook</title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="description" content="">
    <meta name="generator" content="GitBook 3.2.2">
```

Abbildung 3.2.1 - Markup des Kopfteils eines HTML5-Dokuments

## Ziel

- den eigenen Rechner für die Webentwicklung bereit machen
- Verständnis des Unterschieds von HTML und CSS
- gute Praxis beim Arbeiten mit Atom
- erste Kenntnisse in der Entwicklung mit HTML und CSS

## Inhalte

Die Doppelsitzung gibt uns die Möglichkeit, den eigenen Rechner für die Entwicklung mit HTML und CSS einzurichten. Anschließend werden wir erste einfache HTML-Dokumente entwickeln und miteinander verlinken. Dabei nutzen wir [Atom](#) als Entwicklungsumgebung und [Chromium](#) als Entwicklungsbrowser.

## Angesprochene Themen in der Sitzung

In der Sitzung wurden im Gespräch mit den Studierenden weitere Themen aufgeworfen:

- [Owncloud/NextCloud](#) als Alternative zu Dropbox
- Email-Verschlüsselung
- Suchmaschinenoptimierung (SEO)
- Encoding/Decoding von Zeichen
  - [ASCII-Tabelle](#)
  - [UTF-8](#)
- [Obsoleszenz](#) bei technischen Geräten und Software am Beispiel von Apples Softwareupdates
- Anonymes Browsen mit [TOR](#) und [Tails](#)
- Künstliche Intelligenz (KI), auf Englisch Artificial Intelligence (AI)

# Auftrag

Entwickeln Sie eine Webpräsenz bestehend aus mindestens zwei HTML-Dokumenten. Das Thema können Sie frei wählen. Es darf auch humorvoll und spielerisch sein.

## Startseite

Die Startseite oder Homepage der Website enthält folgende Elemente:

- Überschrift
- großes Bild
- Text
- ungeordnete Liste (Aufzählung)

## Unterseite(n)

Von der Startseite ist mindestens eine Unterseite verlinkt. Sie enthält folgende Elemente:

- Tabelle
- geordnete Liste
- verlinktes Bild
- Text
- Verlinkung zurück zur Homepage

Machen Sie sich um das gegenwärtige Aussehen nicht zuviel Sorgen. Als nächstes beschäftigen wir uns mit CSS zur Formatierung der Inhalte.

# Entwickeltes HTML/CSS-Konstrukt aus der Veranstaltung

Zum Kennenlernen HTML und CSS haben wir gemeinsam folgendes Konstrukt entwickelt, das ich im Anschluss an die Veranstaltung nochmal überarbeitet und kommentiert habe.

*Snippet:* index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML lernen ohne Brötchen</title>
  <!-- Notwendig, damit der Browser Umlaute korrekt anzeigt. -->
  <meta charset="utf-8">
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div id="wrapper">
    <header>
      
    </header>
    <!--Hier steht ein Kommentar.-->
    <section id="content">
```

```

<h1>HTML lernen ohne Brötchen</h1>
<ul>
  <li>
    <a href="http://pro-linux.de" target="_blank">Hund</a>
  </li>
  <li>Katze</li>
  <li>Maus</li>
</ul>
<!-- Beispiel für die Anwendung einer CSS-ID -->
<p id="first-paragraph">Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation
nullamco laboris nisi ut aliquip ex ea <a href="#">commodo</a> consequat. Duis aute irure dolor in reprehenderit
in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident,
sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

<!-- Beispiel für Inline CSS -->
<p style="color: red;">Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation
nullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate
velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident,
sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua.</p>
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua.</p>

<!-- Beispiel für die Anwendung einer CSS-Klasse -->
<p class="ausgezeichnet">Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua.</p>
<p class="ausgezeichnet">Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua.</p>
<p class="ausgezeichnet">Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua.</p>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua.</p>
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua.</p>
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua.</p>

<a href="http://placekitten.com/600/600">
  
</a>

<table>
  <tr>
    <td>links oben</td>
    <td>rechts oben</td>
  </tr>
  <tr>
    <td>links unten</td>
    <td>rechts unten</td>
  </tr>
</table>
</section>
</div>
</body>

```

&lt;/html&gt;

**Snippet:** style.css

```

/*
Die folgenden Farben wurden aus dem Headerbild errechnet. Der genutzte Service ist http://www.lavishbootstrap.com/?utf8=%E2%9C%93&image_url=http%3A%2F%2Fplacekitten.com%2F700%2F104&commit=Go+Lavish
Für Menschen, die keine Gestalter_innen sind, kann so ein Tool helfen, ein stimmiges Design zu erarbeiten.

Background: #AEAFAD
Text: #A18C70
Headlines: #24251C
Links: #7B8183
Besuchte Links: #8A755B
*/

body {
    background: #AEAFAD;
}

/* "Wickel" um den Inhalt, der weitere Formatierungsmöglichkeiten bietet.
So kann nämlich die Farbe des Hintergrund hinter dem Inhalt anders gewählt werden, als die Hintergrundfarbe des gesamten sichtbaren Bereichs (s. body-Formatierung) */
#wrapper {
    /* beschränkt die Breite des Inhaltsbereichs */
    max-width: 700px;
    /* positioniert den schmalen Inhalt immer mittig im Browser*/
    margin: 0 auto 0 auto;
    background: #ddd;
    /* Damit der Innenabstand vom Padding nicht der Breite des Wrapper zugeschlagen wird: */
    box-sizing: border-box;
    /* Runde Ecken */
    /*border-radius: 8px;*/
}

/* Innerer Wickel, um den Innenabstand setzen zu können */
#content {
    /* Innenabstand von den Inhalten des Wrappers zu seinen Außengrenzen */
    padding: 10px 20px;
}

h1, h2, h3 {
    color: #504834;
}

a {
    color: #7B8183;
}

/* Farbe für besuchte Links. Die Schreibweise mit dem Doppelpunkt implementiert eine -> Pseudoklasse, hier "visited" */
a:visited {
    color: #8A755B;
}

p {

```

```
    color: #A18C70;
}

/* Beispiel für die Definition einer CSS-ID */
#first-paragraph {
    font-style: italic;
    font-size: 20px;
}

table {
    color: green;
    border: 1px solid #aaa;
    border-collapse: collapse;
}

td {
    border: 1px solid #aaa;
}

/* Beispiel für eine CSS-Klassendefinition */
.ausgezeichnet {
    font-weight: bold;
}

p a {
    font-size: 25px;
}

ul li a {
    color: green;
}
```

## Fertigstellung

Arbeiten Sie im Anschluss an die Veranstaltung weiter an dem Projekt und nutzen Sie die [Materialien](#), um Ihre Fertigkeiten zu erweitern.

## Abgabe

Senden Sie ein ZIP mit allen zugehörigen Dateien bis zum **17.11.2016** an das Dozententeam.



## [03] Fr, 18.11.2016

Die heutige Veranstaltung geht auf zwei Themen ein: **Informationsarchitektur und Navigation** auf Websites sowie **technische Zusammenhänge des World Wide Web**.



*Abbildung 3.3.1 - Absolut notwendig: Verständliche Navigationselemente auf Websites*

## Ziele

- die Bedeutung einer begründeten Informationsarchitektur für Websites verstehen
- Aufbau und Formatierung von Menüs und Navigationen in HTML/CSS verstehen und umsetzen können
- Chrome Developer Tools kennen und zielgerichtet einsetzen
- den Zusammenhang der Kommunikation zwischen Client und Webserver kennen und verstehen

## Inhalte

- Listen in HTML und CSS
- Navigationen und Benutzerführung auf Webseiten
  - Informationsarchitektur
  - Layout

- Usability
- Geschichte, technische Komponenten und Zusammenhänge des World Wide Web (WWW)

## Auftrag

1. **Erweitern Sie Ihr bisheriges HTML-Konstrukt um eine Navigation.** Voraussetzung dafür ist, dass Sie auch entsprechende Unterseiten haben.

- **Analyse:** Schauen Sie sich bekannte Webseiten<sup>1</sup> im Hinblick auf die Navigation an:
  - Wieviele Navigationsbereiche gibt es?
  - Welches Ordnungssystem liegt den Bereichen zugrunde, also: Welche Arten von Einträgen befinden sich wo?
  - Wie ist die Navigation angeordnet, vertikal, horizontal?
  - Was passiert mit der Navigation, wenn Sie in der Seite scrollen?
  - Wie verändert sich die Navigation auf mobilen Endgeräten? Sie können das mit [Google Chromium testen](#).
  - Welches HTML-Markus liegt der Navigation zugrunde, welches CSS?
    - Bringen Sie sich bei, mit den [Chromium Developer Tools](#) die Seite zu inspizieren.
  - **Halten Sie Ihre Erkenntnisse fest, wir werden in der folgenden Sitzung darauf eingehen.**
- **Anfertigung:** Bauen Sie eine eigene Navigation für Ihre Website.
  - Nehmen Sie sich nicht zu viel vor! Gerade an der Navigation können Sie einige grundlegende Konzepte von HTML und CSS erarbeiten und verstehen lernen.
  - Machen Sie sich ggf. vorher eine Skizze auf Papier oder mit einem Tool, das Sie gut beherrschen, um festzulegen, wie die Navigation aussehen soll.
  - **Vorschlag zum Spaß haben:** Arbeiten Sie "im Auftrag"! Lassen Sie sich von einer Kommilitonin oder einem Kommilitonen vorschreiben, wie sie oder er die Navigation gern hätte. Dabei lernen Sie, sich mit dem "Kunden" auszutauschen und auf Anregungen und Kritik zu reagieren.

2. **Veröffentlichen Sie Ihre Website im World Wide Web!**

- Da sich in unserem Veranstaltungszusammenhang ergeben hat, dass einer mit einem Raspberry Pi "Provider spielt", veröffentlichen Sie zunächst Ihr aktuelles Konstrukt auf diesem Rechner. Der Vorgang ist nicht trivial, wenn man das noch nie gemacht hat. Gehen Sie es daher mit Ruhe und Präzision an. Folgende Fragestellungen sollen Sie leiten:
  - Was ist "das Internet"?
  - Worin besteht der Unterschied zwischen *Internet* und *World Wide Web (WWW)*?
  - Wie muss ein Gerät beschaffen sein, das am Internet teilnehmen soll?
  - Welche Software ist notwendig, um ein HTML-Dokument im WWW zu veröffentlichen?
- **Machen Sie zu allem, was Ihnen einfällt, Notizen (Begriffe, Konzepte, Handlungen etc.)**

---

<sup>1</sup>. Analysieren Sie nicht nur "die Großen", denn dort ist meist eine sehr spezielle Technik im Einsatz, die viel Vorwissen erfordert. In den bekannten sozialen Netzwerken und Webshops kommt in der Regel JavaScript zum Einsatz, um die User Experience (UX) zu verbessern. Schauen Sie sich also auch Seiten vom Sportverein und vom Bäcker um die Ecke an. ↩

## Quellen

## Bücher

- Jacobsen, J. (2014). Website-Konzeption: erfolgreiche Websites planen, umsetzen und betreiben (7., überarb. und Aufl.). Heidelberg: Dpunkt.
- Duckett, J. (2013). HTML & CSS: erfolgreich Websites gestalten & programmieren. München u.a.: Pearson.
- Friedman, V. (2009). Praxisbuch Web 2.0: moderne Webseiten programmieren und gestalten. (2., aktualisierte und Aufl.). Bonn: Galileo Press. S. 221-228.

## Web

- [Listamatic](#) | systematisches Onlinetutorial zur Gestaltung von Listen als Menüs und Navigationselemente. Veraltet, in den Grundlagen der Formatierung von HTML-Listen immer noch gültig.
- [CSS Navigation Bar](#) | Erstellung horizontaler und vertikaler Navigationsleisten mit CSS
- [Create a Nav Bar | HTML & CSS Tutorial](#) | Videotutorial
- [Simple Horizontal Navigation](#) | Videotutorial und Transkript
- [Die Maus erklärt das Internet](#) | Wenn auch sehr unwissenschaftlich, so doch ein Klassiker
- [The Birth of the Web](#) | Website des CERN mit einer anschaulichen Aufbereitung der Entstehungsgeschichte
- [Wie Tim Berners-Lee das Web erfand](#) | deutschsprachiger Artikel aus dem Tagesspiegel

## [04] Fr, 25.11.2016

In der heutigen Veranstaltung werden wir uns weiter der Frage widmen, wie wir die unsere HTML-Seite ins Internet stellen. Statt sie aber einfach bei irgendeinem Provider hochzuladen, nähern wir uns weiter dem technischen Konzept des World Wide Web und planen seinen Nachbau im Seminarraum!

Natürlich werden wir das World Wide Web nicht in seiner technischen Komplexität abbilden können. Ausgehend von der Client/Server-Architektur können wir jedoch erste Komponenten in Stellung bringen, die uns helfen, konzeptionelle und technische Zusammenhänge zu verstehen. Mal sehen, wie groß unser Nachbau wird...

## Ziele

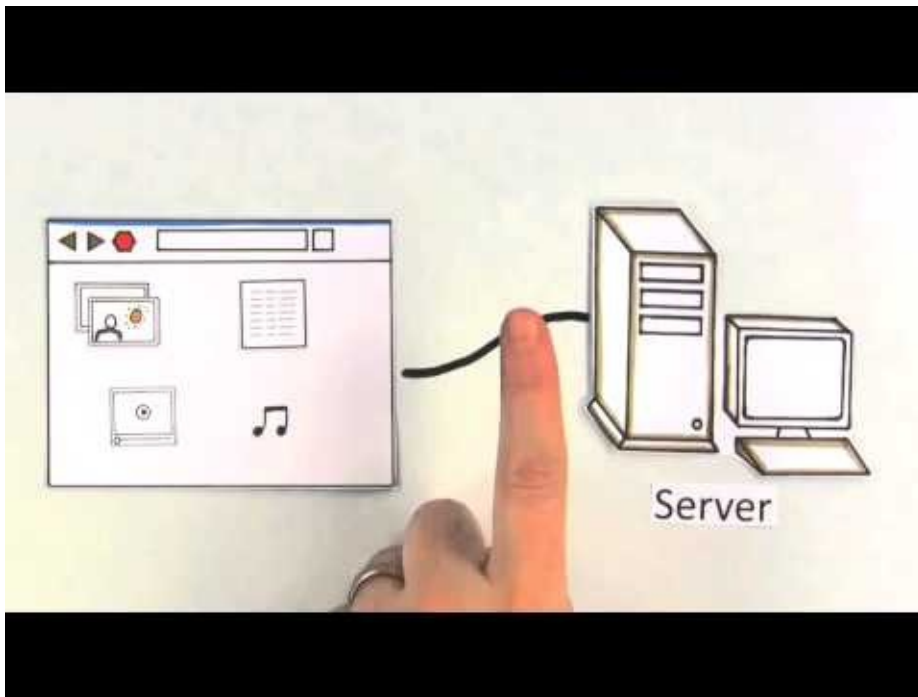
- Einen Nachbau des World Wide Web planen

## Inhalte



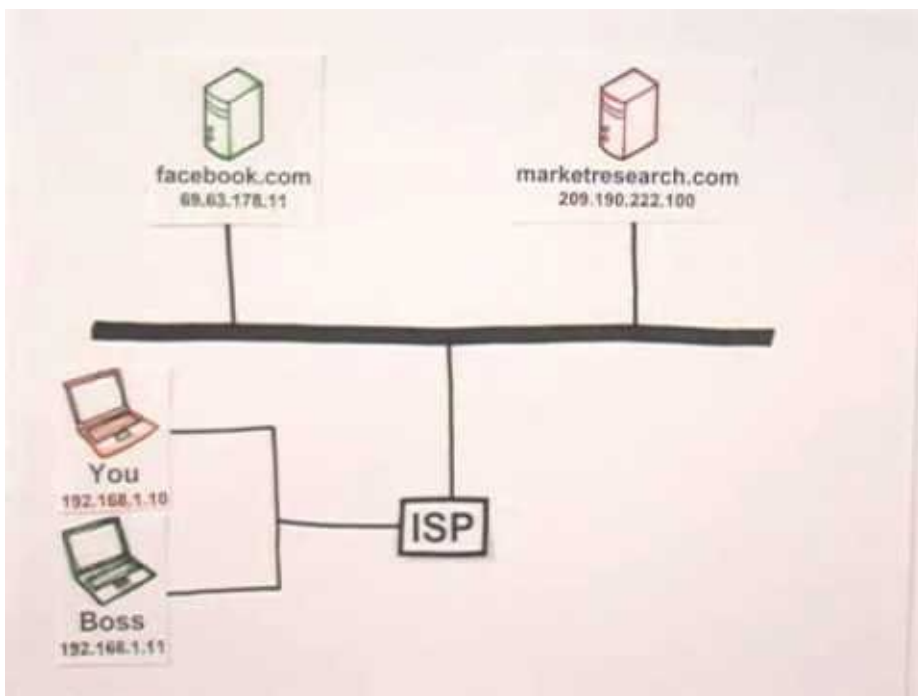
[Video link](#)

**Video 1:** ["Die Sendung mit der Maus - Wie funktioniert das Internet"](#)



[Video link](#)

**Video 2:** "The World Wide Web in Plain English"



[Video link](#)

**Video 3:** "How the Internet Works in 5 Minutes"

## Themen in der Veranstaltung

- Keylogger
- Phishing
- Geschäftsmodelle für digitale Artefakte (Musik, Literatur)

- Cory Doctorow
- Paolo Coelho
- div. Bands, die ihre Musik frei zum Download zur Verfügung stellen
- [iRights](#)

## Auftrag

1. Sehen Sie sich die Videos an.
2. Geben Sie gemeinsam in der Gruppe wieder, was Sie gesehen und verstanden haben.
3. Welche Hard- und Softwarekomponenten brauchen Sie, um Webseiten zu veröffentlichen und abzurufen?
4. Nehmen Sie bei Ihren Überlegungen an, dass der Raspberry Pi die Rolle eines Servers einnimmt.
5. Recherchieren Sie, skizzieren Sie, bauen Sie, testen Sie!

Zur Sammlung von Fragen und Notizen verwenden wir ein [Etherpad](#).

Ausgehend von den Fragen und dem "Maus"-Film bereiten die Teilnehmenden zum nächsten Mal fachlich die folgenden Themen vor:

- Server
- Router
- DNS ("Auskunft")
- "Bote"/Daten
- Modem
- Provider

## Quellen

### Bücher

- Steyer, R. (2016). WordPress. Wiesbaden: Springer Fachmedien Wiesbaden. Zugriff am 24.11.2016. Verfügbar unter: <http://link.springer.com/10.1007/978-3-658-12830-2>, S. 33-42
- Meinel, C. (2004). WWW : Kommunikation, Internetworking, Web-Technologien ; mit 106 Tabellen (Xpert.press). (H. Sack, Hrsg.). Berlin u.a.: Springer. Verfügbar unter: <http://d-nb.info/968931863/04>

### Videos

- ["Was ist das Internet? \(ZDF - 1996\)"](#) | Das Internet, erstmals im deutschen Fernsehen erklärt
- ["3sat neues Doku: 40 Jahre Internet - 20 Jahre www"](#)

## [05] Fr., 02.12.2016

Die heutige Veranstaltung zielt darauf ab, einen ersten Plan für ein "abgespecktes" World Wide Web zu entwickeln. Mittelfristig wollen wir unsere HTML-Seiten auf einen Server hochladen und von dort über HTTP abrufen - wie im "echten" WWW.

### Ziel

- Minimalsetup für eine Client-/Server-Architektur

### Inhalte

Folgende Inhalte sind im Laufe der Veranstaltung thematisiert worden:

- Die Problematik von Umlauten, Leerzeichen und Sonderzeichen in Dateinamen
- Chrome Developer Tools für das Entwickeln mobiler und reaktionsfähiger Design (responsive webdesign)
- Media Queries im Zentrum von Responsive Webdesign
- Der DHCP-Server: Verleih von IP-Adressen
- MAC-Adressen, IP-Adressen, Subnetzmasken
- Router, Switches und Hubs

### Auftrag

- Formulieren Sie in, zwei kurze Sätze zu den Kernkonzepten und -begriffen des World Wide Web:
- Erstellen Sie ein HTML-Dokument - neu oder als Teil ihrer schon entstandenen Webpräsenz - und fügen Sie dort die Begriffserklärungen ähnlich einem Glossar ein.
- Machen Sie sich mit dem Listenelement `<d1></d1>` bekannt. Es eignet sich hervorragend für ein Glossar!

### Ablauf der Veranstaltung im Rückblick

Die Teilnehmenden haben zu ihren Themen kurz referiert und ihr Wissen im Laufe der Veranstaltung in die Konzeption eines ersten Aufbaus des WWW im Seminar einfließen lassen.

Durch die **Maieutische Methode** haben wir in Gruppenarbeit herausgefunden, wie eine rudimentäre Client-/Server-Architektur aufgebaut sein könnte.

Als Ziel für die nächste Veranstaltung wurde definiert, dass wir in der Lage sein müssen, unsere Webseitenkonstrukte auf ein Zielserver hochzuladen. Dies geschehen, ohne dass ein DHCP-Server die IP-Adressen zuweist. Stattdessen werden die Teilnehmenden die IP-Einstellungen manuelle vornehmen.

Für die Veranstaltung waren einige Fragen leitend:

- Woher bekommen wir das neue Wissen her? (Sammeln von Quellen in Zotero)
- Welche weiteren Fragen sind aufgetaucht (Eintrag ins Etherpad)?
- Wie klären wir weitere Fragen, um zu verlässlichen und befriedigenden Ergebnissen zu kommen?

Am Ende der Veranstaltung wurden alle Teilnehmenden nochmals aufgefordert, sich bei Zotero zu registrieren und ihren Nutzernamen/Emailadresse an das Dozententeam zu schicken.



## **[06] Fr., 09.12.2016**

Die heutige Veranstaltung ist praktisch ausgerichtet. Wir werden "unser" geplantes LAN aufbauen, Software installieren, manuell Netzwerkadressen vergeben und unsere Dateien verschlüsselt auf den Raspberry Pi übertragen

### **Ziele**

- Dateien (HTML/CSS) auf einen Server hochladen können

### **Inhalte**

In der heutigen Veranstaltung werden wir

- den Raspberry Pi in Betrieb nehmen
- ein lokales Netzwerk aufbauen und konfigurieren
- Protokolle zur Datenübertragung zwischen Client und Server kennenlernen und ausprobieren
- in die Administration eines entfernten Rechners einsteigen
- Daten verschlüsselt in eigenes Benutzerverzeichnis auf dem Server hochladen

### **Inhalte aus der Veranstaltung**

- Linux-Distributionen
  - Ubuntu
  - Linux Mint
  - Redhat
  - Suse Linux
  - Debian
- DNS-Server

### **Inputelemente**

#### **Vorbereitungen**

- Teilnehmer\_innen installieren FileZilla auf ihrem eigenen Rechner

#### **Der Raspberry Pi mit Graphical User Interface (GUI)**

- Vorstellung des Raspberry Pi auf der grafischen Oberfläche (Raspian)
- Zeigen der Konfigurationsmöglichkeiten
- Anlegen von Teilnehmenden-Accounts durch den Provider

- Konfiguration des Raspberry Pi mit einer statischen Adresse aus einem privaten Netz (grafische Benutzeroberfläche)

```
IP:          192.168.178.100
Maske:       255.255.255.0
Broadcast:   192.168.178.255
Gateway:     192.168.178.1
```

## Teilnahme am Netzwerk

Manuelle Konfiguration der Rechner der Studierenden für die Teilnahme an diesem Netz:

```
IP:          192.168.178.[...]
Maske:       255.255.255.0
Broadcast:   192.168.178.255
Gateway:     192.168.178.100
```

## IPs der Teilnehmer\_innen

```
guenther: 192.168.178.55
kamerakind: 192.168.178.15
katzenkuchen: 192.168.178.22
xldrkp: 192.168.178.101
vollgut: 192.168.178.110
keineahnung: 192.168.178.25
aufstpauli: 192.168.178.75
```

## Kommunikation zwischen Client und Server

- Besprechung der gängigen Protokolle FTP, SFTP bzw. SSH
- Einrichtung des Raspberry Pi mit einem SSH-Server
- Anmeldung als User auf dem Raspberry Pi
- Verwendung von FileZilla für den Upload der HTML-Seiten auf den Raspberry Pi

## Auftrag

Schreiben Sie einen Text von ca. einer Dreiviertel-seite Länge, der folgende Punkte thematisiert:

- Aufbau unseres lokalen Netzwerks mit seinen Komponenten
- Übertragung von Dateien vom Client auf den Server

Verfassen Sie Ihren Text in HTML und verlinken Sie verwendete Fachtermini mit aussagekräftigen und seriösen Quellen. Dabei soll sich die verlinkte Quelle in einem neuen Tab öffnen, wenn der Link angeklickt wird.

## Ablauf der Veranstaltung im Rückblick

Eingangs wurde die mündliche Prüfung klarer gefasst und formuliert, dass die Teilnehmenden sich im letzten Drittel der Veranstaltung ein Thema greifen können, das sie vertiefen wollen. Ferner wurde der weitere Verlauf der Veranstaltung im zweiten Semester skizziert und die Anforderungen an die dort zu verfassende Hausarbeit genauer umrissen.

Im inhaltlichen Teil der Veranstaltung wurde zunächst der Raspberry Pi vorgestellt, wobei auf verschiedene Linux-Distributionen und den Vorgang des Kopieren eines Images auf die SD-Karte eingegangen wurde. Der RPI wurde hochgefahren, die Konfigurations- und Installationsmöglichkeiten gezeigt.

Anschließend haben wir erarbeitet, wie das LAN konfiguriert sein muss, damit alle Rechner der Teilnehmenden (2 Linux, 3 Mac, 2 Windows) auf den Raspberry zugreifen können. Die Teilnehmenden haben dafür manuell die Netzwerkkarten konfiguriert bzw. USB-LAN-Sticks des Instituts verwendet.

Mit Filezilla wurden anschließend Dateien zunächst auf den pi-Account kopiert. Anschließend wurden Nutzeraccounts eingerichtet, die in Zukunft auch für den Webservice der Nutzer verwendet werden sollen. Dabei wurden verschiedene Problematiken angesprochen, die bei Mehrbenutzersystemen auftreten können.

## **Ausblick**

In der kommenden Veranstaltung wird der RPI als Webserver eingerichtet werden.

## [07] Fr., 16.12.2016

### Ziele

- Die Teilnehmer\_innen können Dateien auf einem Host bereitstellen und diese über HTTP abrufen

### Inhalte

- Exkurs: Bilder fürs Web aufbereiten
  - Exkurs: Freie/offene Software für Web und Print
    - [Inkscape](#)
    - [GIMP](#)
    - [Scribus](#)
- Absicherung der Nutzeraccounts auf dem Raspberry Pi, sodass keiner in den Ordner des anderen schauen kann
- Bereitstellung und Abruf eigener Websites vom Raspberry Pi
- Details der Protokolle [SSH](#) und [HTTP/HTTPS](#)

### Auftrag

Versammeln Sie das neu erworbene Wissen aus den vergangenen Wochen zum Abschluss des Jahres nochmal in einer integrierten Übung.

Erstellen Sie eine **weihnachtliche Webpräsenz** mit mehreren Unterseiten. Dabei sollen folgende Spezifikationen implementiert werden:

- Die Site soll eine **fixe Navigation** am oberen Rand des Viewports erhalten
- Das Layout soll sich über die **gesamte Breite des Viewports** erstrecken. Als Beispiel hierfür können Sie sich [die Seite des iTBH](#) ansehen.
- Die Site soll mehrere Unterseiten enthalten:
  - 1 Weihnachtsgeschichte mit zwei Bildern, die unter [Creative Commons](#) (CC) lizenziert sind. [Korrekte Quellenangabe für CC nicht vergessen!](#)
  - 1 Plätzchenrezept, formatiert wie bspw. bei [Chefkoch](#). Korrekte Quellenangabe für das Rezept nicht vergessen.
  - 1 Weihnachtslied, formatiert wie bspw. [auf dieser Seite](#) (nur den Text), Quellenangabe nicht vergessen
  - 1 animiertes Weihnachts-GIF, Quellenangabe nicht vergessen
  - Gegen die Langeweile: Schnee auf der Seite
  - Gegen die Langeweile: fliegender Weihnachtsmann, von rechts nach links oder umgekehrt

### Abgabe

Senden Sie das Konstrukt als gepacktes Archiv bis Mittwoch, den 21.12.2016, an das Dozententeam.

## [08] Fr., 23.12.2016



**Abbildung:** Filming Jeopardy! von [Steve Jurvetson](#) unter CC-BY

## Ziele

- Die Teilnehmer\_innen haben verstanden, was ein Webserver (Apache/nginx) ist und wie dieser funktioniert.
- Entspannt in die Weihnachtsfeiertage einsteigen.

## Inhalte

- Wiederholung: Sind alle Fragen aus dem [Etherpad](#) beantwortet?
- Installation und Konfiguration eines Webserver
- Details von [HTTP/HTTPS](#)
  - [Analyse des Netzwerkverkehrs mit den Chrome Developer Tools](#)
- Arbeiten auf dem Raspberry Pi über die Kommandozeile

## Ablauf

## Der Ernst

- Die Teilnehmer\_innen bauen das lokale Netzwerk im Seminarraum auf.
- Sie stellen ihre Weihnachtswebsites auf dem Server bereit und informieren die anderen über das neue Angebot.
- Die Teilnehmer\_innen arbeiten in Paaren zusammen und setzen sich mit ihren Weihnachtswebsites im *peer review* auseinander. Dabei verwenden Sie die *Chrome Developer Tools*.

## Der Spaß

Im zweiten Teil der Veranstaltung bereiten wir uns auf die mündliche Prüfung am Semesterende mit Jeopardy vor<sup>1</sup>.



**Video:** Beispielerpisode der amerikanischen Ausgabe von Jeopardy

Für das Spiel in unserer Veranstaltung verwenden wir Jeopardy in der [Version von Sven Kammerer](#). Es war nicht einfach, eine lauffähige Version des Spiels zu finden, obwohl es auf [GitHub über 800 Projekte zu dem Thema](#) gibt.

Um eine Runde Jeopardy spielen zu können, braucht man in sechs Kategorien jeweils fünf Fragen (30 insgesamt). Diese müssen [in einer Textdatei](#) gespeichert werden:

```
HTML
100: [nE]<p></p> ##Mit welchem Element werden Textabsätze ausgezeichnet?##
200:
300:
400:
500:
CSS
100: margin: 15px 0;
200:
300:
400:
500:
Politik
100:
200:
300:
400:
500:
Betriebssystem
100:
200:
300:
400:
500:
Internet
100:
200:
300:
400:
500:
Personen
100: Tim-Berners-Lee ##Wer hat das WWW erfunden?##
200:
300:
400:
500:
```

## Vorbereitung

- JedeR Teilnehmer\_in denkt sich jeweils eine Frage/Antwort-Kombination pro Kategorie aus und notiert diese in einer Datei 1.jrf auf seinem Rechner.
- Alle Teilnehmer\_innen laden ihre Dateien in ihr Webverzeichnis auf dem Raspberry hoch.
- Das Dozententeam sammelt die Fragmente ein und erstellt daraus eine gemeinsame Datei, mit der wir dann spielen.

## Auftrag für die Nachweihnachtszeit

- Genießen Sie die Feiertage.
- Lesen Sie den Text "Technische Fundamente des World Wide Web" (als Kopie verteilt) unter der Fragestellung, was von den genannten Begriffen und Konzepten bisher in ihrer Arbeit vorgekommen ist.
- Bauen Sie weiter an Ihrer Website.



<sup>1</sup>. Im Zusammenhang mit Jeopardy ist der Wettkampf zwischen Menschen und [Watson](#) interessant.  
Das Thema [Künstliche Intelligenz \(KI/AI\)](#) gewinnt ja gerade an Fahrt. [↔](#)

## [09] Fr., 13.01.2017

### Inhalte

Entgegen der Planung hat sich diese Sitzung in eine andere, gute Richtung entwickelt. Angesprochen wurden u.a. die folgenden Themen:

- Konzept und Details von Creative Commons
  - Verwenden von Artefakten
  - Veröffentlichen von Artefakten
- Grundlagen wissenschaftlichen Arbeitens
  - Sinn und Zweck von Hausarbeit, Bachelor- und Masterarbeit; Promotion
- Stärken und Herausforderungen der Teilnehmenden in Bezug auf die Veranstaltungsthemen
- Ausrichtung, Ablauf und Gestalt der Fachgespräche

### Auftrag

- Setzen Sie sich systematisch mit den Inhalten der Veranstaltung auseinander.
- Nutzen Sie dafür dieses Skript sowie Bücher und Quellen, die Ihnen helfen, Ihr erworbenes Wissen in **mündliche** Worte zu fassen.
- Sprechen Sie **miteinander** über die Themen. Stellen Sie sich gegenseitig Fragen.
- Ergreifen Sie die Initiative!

## [10] Fr., 20.01.2017

### Ziele

- Die Studierenden können per SSH auf einen entfernten Rechner zugreifen.
- Die Studierenden beherrschen [grundlegende Linuxbefehle](#) und sind in der Lage, einfache Dateioperationen durchzuführen.
- Die Studierenden sind in der Lage, Dateien per `scp` auf einen entfernten Rechner zu kopieren.

### Inhalte

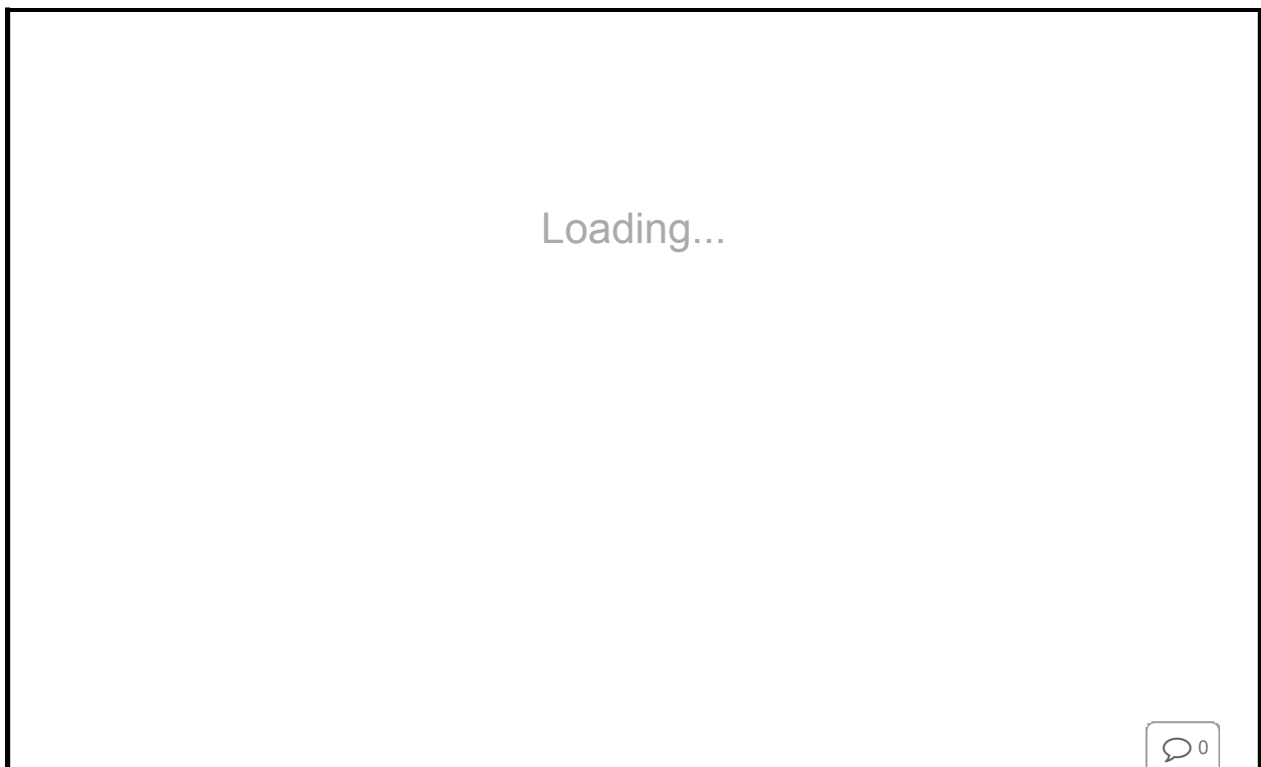
#### Prüfungsvorbereitung

- Gemeinsame Sammlung von möglichen Fragen eines Fachgesprächs in einem [neuen Etherpad](#). Konsultieren Sie auch [das erste Etherpad](#), in dem wir uns Fragen zum Internet gestellt haben.

#### Prüfungstermin

Der Prüfungstermin für die Fachgespräche ist der **10. Februar 2017**. Die Prüfungen finden in der Zeit von 09:00 bis 12:00 statt (jeweils zwei Studierende pro Stunde).

#### Etherpad: Vorbereitung auf die Fachgespräche



## Fachliches

- gemeinsame Erinnerung an Themen und Stoff aus 2016
- Installation und Verwendung einer Linux-Kommandozeile für den Zugriff auf den entfernten RPI
  - [PuTTY](#)
  - [Git for Windows](#)
  - Terminal auf dem Mac
    - [OS X: Das Terminal für Einsteiger](#)
    - [Einstieg ins Terminal auf dem Mac \(en\)](#)
  - Terminal unter Linux
    - Hier gibt es zahlreiche Einführungen, Tutorials und Dokumentationen, z.B.:
    - [Introduction to Bash](#) (Bash ist eine Version der Shell unter UNIX/Linux)
    - [An Introduction to the Linux Terminal](#)
  - [Fingerübungen](#) für die Shell

## Szenarien

Die folgenden Szenarien skizzieren das Hin- und Herkopieren von Dateien zwischen zwei Rechnern.

### Szenario 1: Datei lokal anlegen, auf entfernten Rechner *ins Heimatverzeichnis* kopieren

lokal	remote
1. Anlegen eines Ordners mit <code>mkdir scp-uebung</code>	
2. Wechseln in den Ordner mit <code>cd scp-uebung</code>	
3. Anlegen und Füllen einer Datei mit <code>nano scp.html</code>	
4. Hochkopieren der Datei mit <code>scp scp.html account@192.168.178.100:.</code>	

### Szenario 2: Ordner remote anlegen, Datei vom lokalen Rechner in diesen neuen Ordner kopieren

lokal	remote
	1. Anlegen eines Ordners mit <code>mkdir uebung</code>
	2. Verlassen des Rechners mit CTRL-D
3. Ggf. wechseln in das Verzeichnis mit der Datei, z.B. <code>cd scp-uebung</code>	
4. Hochkopieren der Datei mit <code>scp scp.html account@192.168.178.100:./uebung</code>	

## Tipp

- Mit `cd` wechselt man immer ins Homeverzeichnis!
- `cp` nutzt man, um Dateien und Ordner auf ein und demselben Rechner zu kopieren, `mv`, um sie zu verschieben
- `rm` verwendet man zum Löschen von Ordnern und Dateien. **Vorsicht:** Hier gibt es kein *Undo*!

## Auftrag

- Registrieren Sie sich bei [Uberspace](#) für einen kostenlosen, 30-tägigen Account.
- Veröffentlichen Sie dort eine einfach gestaltete Website, die **in jeder Hinsicht rechtlich unbedenklich** ist.
- Der Inhalt Ihrer Seite ist eine detaillierte Anleitung, wie Sie per SSH auf den Server gelangen und dort eine HTML-Seite hinterlegen.
- **Wichtig:** Schreiben Sie die Anleitung so, wie Sie **sprechen**, nicht wie Sie schreiben. Stellen Sie sich beim Schreiben vor, dass Sie einen Vortrag halten.
- Senden Sie bis Mittwoch, den 25.01.2017, die URL ihrer Seite an das Dozententeam.

## **[11] Fr., 27.01.2017**

Die Veranstaltung wurde vom unserem Tutor gehalten. Angaben zu den Inhalten folgen.

## **[12] Fr., 03.02.2017**

### **Inhalte**

- Veranstaltungsevaluation
- Rückblick auf die Veranstaltung

### **Auftrag**

Die Studierenden bereiten sich auf die Fachgespräche vor.

# Webtechnologien

Das Thema Webtechnologien ist mittlerweile sehr komplex geworden. Die folgenden Sprachen und Begriffe mögen einen ersten Einstieg in das Thema bieten und bewusst machen, was man alles lernen könnte/müsste/sollte:

## Sprachen

- [HTML5](#)
- [CSS3](#)
- [JavaScript/ECMAScript](#)
- [PHP](#)
- [Python](#)
- [Ruby On Rails](#)
- [Markdown](#)

## Frameworks

- [Symfony](#)
- [Django](#)
- [Flask](#)
- [React](#)
- [Ionic](#)

## Software

- [Apache](#)
- [nginx](#)

Die Liste ist in keiner Weise abschließend, sondern zieht lediglich einige Komponenten zusammen, mit denen das Internet, wie wir es kennen und benutzen, gegenwärtig gebaut wird.



# HTML5



Abbildung 4.1.1.1 - Offizielles HTML5-Logo

HTML ist keine Programmiersprache. Das Akronym bedeutet *Hypertext Markup Language* und ist damit eine Auszeichnungssprache für Text. Durch die Möglichkeit, Dokumente miteinander zu verknüpfen - durch Hyperlinks - entsteht Hypertext. [Tim Berners Lee](#) hat 1989 diese bahnbrechende Entwicklung veröffentlicht, die nach wie vor das World Wide Web (WWW) bestimmt.

Wer das WWW mitgestalten will, kommt an HTML nicht vorbei. HTML5 ist die aktuelle Version von HTML und zeichnet sich durch [neue und spezielle technische Eigenschaften](#) aus, die den Ansprüchen der heutigen Zeit gerecht werden sollen.

## Lernressourcen

HTML5 kann man mit jedem Texteditor notieren, als `dateiname.html` speichern und dann als Dokument im Browser öffnen. Wir werden in der Veranstaltung [Atom](#) verwenden, einen Editor, der unter der [MIT Lizenz](#) veröffentlicht wurde und damit als [Freie Software](#) gilt.

## Referenzen

- Ein sehr guter Anlaufpunkt in Sachen Referenz und Dokumentation ist das **Mozilla Developer Framework**. Hier gibt es mittlerweile viel zu [CSS](#) in deutscher Sprache. Jedoch gilt auch hier, dass die englische Version hinsichtlich Aktualität und Vollständigkeit der deutschen vorzuziehen ist.

## Webseiten und Tutorials

- Eine aktuelle deutschsprachige Website, die auch Stilfragen und gute Praxis vermittelt, ist das [HTML-Seminar](#) von Axel Pratzner.
- Auf dieser Seite ist besonders [der Vergleich zwischen der "althergebrachten Art" und der neuen zu erwähnen](#), mit der Webseiten strukturiert werden.
- Die W3School hat [viele Lerneinheiten](#) zu aktuellen Sprachen des Webs.

## Videos

- Eine gute deutschsprachige Einführungsreihe findet sich [bei Youtube](#) unter dem Titel "HTML5 Tutorial".
- In englischer Sprache, dafür sehr langsam und ausführlich und auf sechs Teile verteilt: [HTML5 Tutorial for Beginners](#).

## Cheat Sheets

- kleingedruckt und auf Englisch, dafür aber sehr vollständig: <http://websitesetup.org/html5-cheat-sheet/>
- eher ein kleines Heft, Englisch, mit weiteren hilfreichen Informationen: <https://makeawebsitehub.com/the-html-5-mega-cheat-sheet/>
- Englisch, dafür aber mit viel erklärendem Text: <https://hostingfacts.com/html-cheat-sheet/>
- eine Sammlung von Cheat Sheets, die u.a. auch welche zu HTML5 enthält: <https://envato.com/blog/cheatsheets-web-designers/>
- gute deutschsprachige Übersicht aller Tags: <http://www.html-seminar.de/befehlsuebersicht.htm>

## Bücher

Einige Buchempfehlungen zu HTML5 und CSS3 finden sich bei [Zotero](#). Mein Favorit ist Duckett (2014), weil es sehr ansprechend gestaltet ist und neben HTML5 und CSS3 auch noch Grundlagen der Client-/Serverkommunikation anschaulich erklärt. Es ist in Hamburger Bibliotheken verfügbar.

# CSS3

CSS dient dazu, Dokumente zu formatieren, die mit einer Markup-Sprache wie HTML ausgezeichnet wurden.

Im Laufe der Zeit haben sich Arbeitsgruppen um das [W3C](#) herum darauf geeinigt, die Auszeichnung und Formatierung von Inhalt strikt zu trennen. Das heißt, mit HTML kennzeichnet man Informationen in einem Dokument entsprechend ihres Sinns. Das nennt man auch *semantisches Markup*. Man verwendet also das `table`-Element für tabellenartige Inhalte und nicht zum Layouten.

Mit CSS werden die Inhalte hinsichtlich Größe, optischer Anordnung, Abstand, Farbe etc. formatiert.

Allgemein wird von CSS gesprochen. Soll jedoch auf die besonderen Eigenschaften und Neuerungen verwiesen werden, die sich in den vergangenen 10 Jahren angesichts neuer Anforderungen an Webseiten und -applikationen ergeben haben, nennt man CSS3 im Besonderen. Die Entwicklungsgeschichte von CSS wird [in der Wikipedia](#) nachgezeichnet.

## Lernressourcen

### Referenzen

- Ein sehr guter Anlaufpunkt in Sachen Referenz und Dokumentation ist das **Mozilla Developer Framework**. Hier gibt es mittlerweile viel zu [CSS](#) in deutscher Sprache. Jedoch gilt auch hier, dass die englische Version hinsichtlich Aktualität und Vollständigkeit der deutschen vorzuziehen ist.

### Videos

- Eine deutschsprachige Tutorialreihe, die auch Layouts mit CSS behandelt, gibt es auf [Youtube](#).
- [CSS-Tricks Videos](#) | Die Videos sind kein Lehrgang, den man nacheinander durcharbeiten kann. Vielmehr werden Themen und Tricks behandelt, die im Alltag des Webseitenbauens immer wieder aufkommen.

### Cheat Sheets

- gute Sammlung auf <https://envato.com/blog/cheatsheets-web-designers/> Eine Websuche nach "cheat sheet css3" hilft.

### Webseiten

- [A List Apart](#) | einschlägige Seite für Designer\_innen, die in der Vergangenheit oftmals zur Weiterentwicklung von Theorie und Praxis rund um CSS beigetragen hat. So ist hier zuerst der einflussreiche Artikel von Ethan Marcotte zu [Responsive Webdesign](#) erschienen.
- [CSS-Tricks](#) | lange existierende Seite von Chris Coyier, von der man sehr viel lernen kann.

## Bücher

- Ein Klassiker in deutscher Sprache ist das Buch "CSS. Das umfassende Handbuch" von Kai Laborenz. Es ist zwar verhältnismäßig alt (letzte Auflage 2011), enthält jedoch geballtes Wissen zum Formatieren und Layouten von Webseiten, das nach wie vor aktuell ist.

# Analyse der Kommunikation zwischen Client und Server

Im folgenden Beispiel wird beschrieben, wie man die Kommunikation zwischen dem Browser auf dem Clientrechner und der Webserversoftware sichtbar machen und analysieren kann.

## Voraussetzungen

- Nutzung von Google Chrome oder Chromium in einer aktuellen Version (im Beispiel Version 49.0.2623.108 Built on Ubuntu 14.04, running on LinuxMint 17.3 (64-bit))
- Server mit Python und dem Modul `SimpleHTTPServer` (im Beispiel Raspberry Pi mit Raspian Debian Jessie)
- Hosting einer mittelkomplexen Website auf dem Server (im Beispiel die Seite *Secondhandblumen Petersen*, frei verfügbar unter <https://github.com/xldrkp/secondhandblumen>)

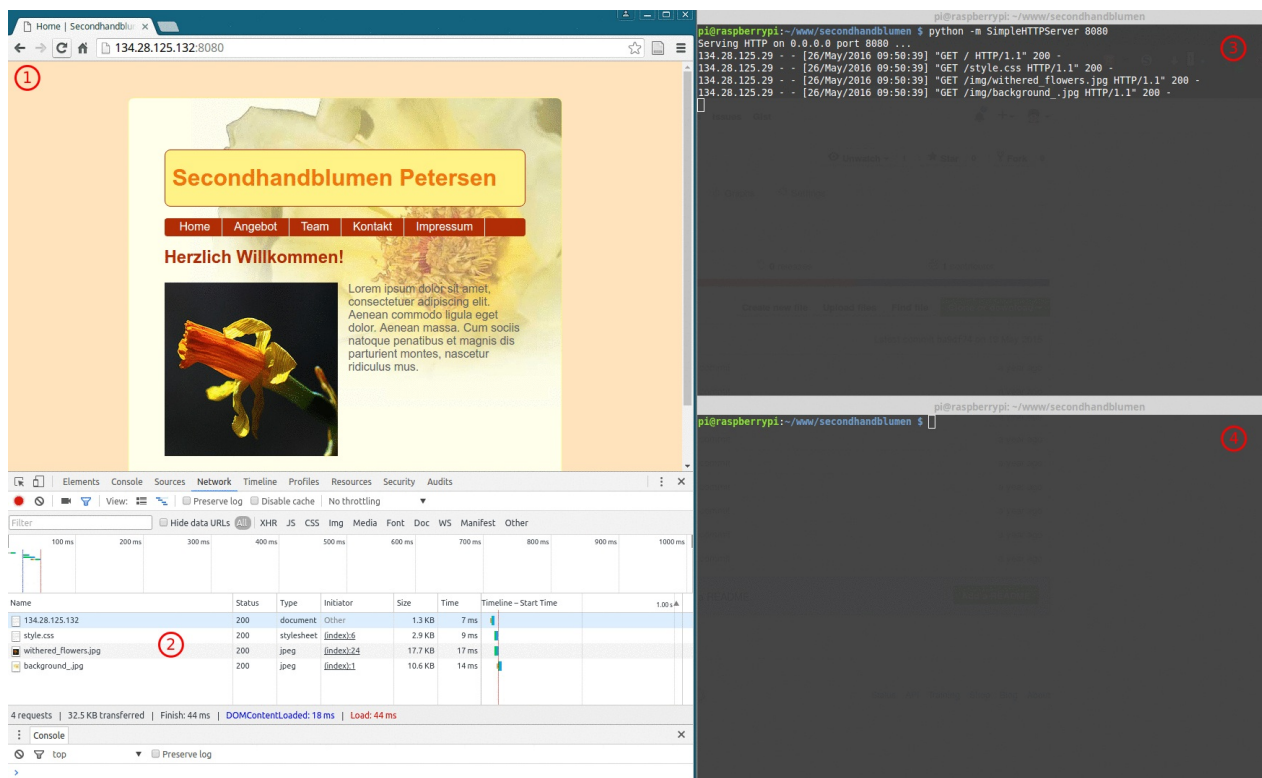


Abbildung 4.1.3.1 - Sinnvolle Anordnung von Fenstern zur Analyse des Netzwerkverkehrs

## Ablauf des Experiments

Das Experiment wird von Bob und Alice durchgeführt.

Mit einer Anwendung wie PuTTY oder dem Terminal verbindet sich Alice per SSH mit dem Server (vgl. Abb. 1, (3)). Dort navigiert sie in den Ordner, der die Website hostet und startet den Server.

Im Browser ruft Bob nun die Adresse auf, unter der die Website zu erreichen ist. Sofern er erfolgreich ist, schaltet er die *Entwicklertools* dazu, erreichbar über das Burgermenü des Chrome-Browsers oder die Tastenkombination `STRG+SHIFT+I` oder die Funktionstaste `F12`. Standardmäßig wird die Ansicht der Entwicklertools unten angezeigt (vgl. Abb. 1, (2)).

Bob wechselt anschließend in den Tab "Netzwerk" und ruft mit `F5` erneut die Website im Browser auf. In der Folge können Bob und Alice den Netzwerkverkehr zwischen Client und Server beobachten (vgl. (2) und (3)). Der Server schreibt sein Logfile direkt ins Terminal und zeigt damit an, welche Befehle er über HTTP empfängt. Die Statuscodes zeigen an, wie er die Anfrage (Request) behandelt und ob er dem anfragenden Client eine Antwort (Response) geben kann.

Das Navigieren in der Website zeitigt weiteren Netzwerkverkehr.

## Erweiterung des Experiments

Alice schlägt vor, sich den *Server Roundtrip* genauer anzusehen. Bob fragt, was sie damit meint. Alice erklärt: Der Server Roundtrip wird auch *Request-Response-Cycle* genannt. Beides bedeutet im Grunde dasselbe, nämlich dass der Client eine Anfrage an den Server stellt und dieser eine Antwort gibt - Request, Response. Dieser Trip wird oftmals wiederholt, weil nicht nur die HTML-Seite vom Server geholt werden muss, sondern auch alle Ressourcen, die darin eingebunden sind, also Bilder, CSS, JavaScript und andere Daten. Die Entwicklertools des Browsers geben genaueren Einblick in die Kommunikation, wenn man eine Resource anklickt.

Bob navigiert auf die Homepage, aktualisiert die Browseransicht und klickt anschließend auf `index.html` im Netzwerktab (vgl. Abb. 2, (1)).

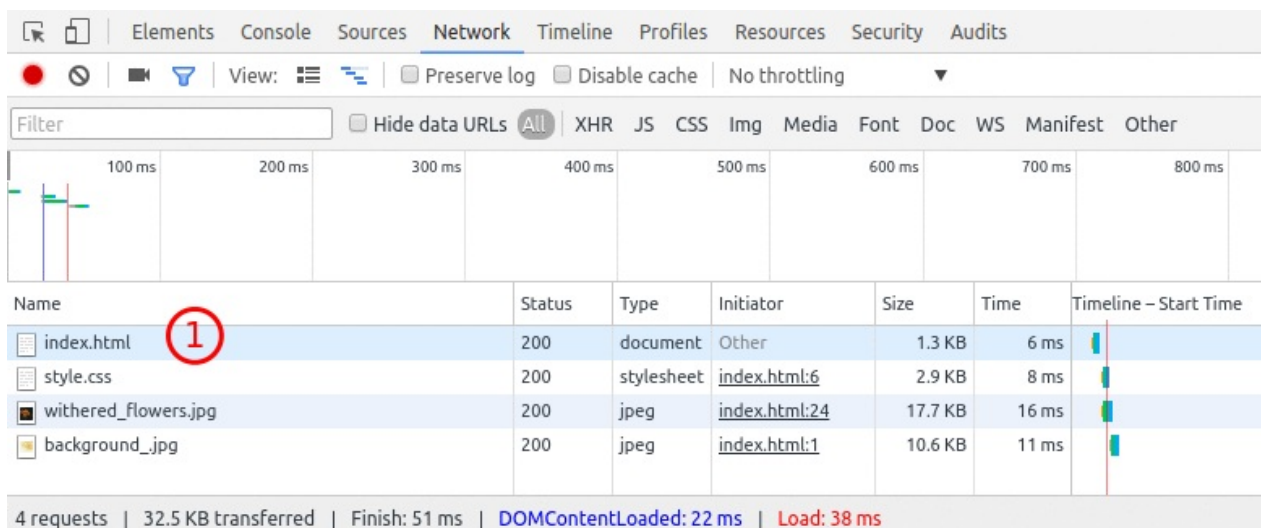


Abbildung 4.1.3.2 - Auswahl der Datei `index.html` im Netzwerktab

Die folgende Ansicht zeigt die Daten, die bei Request und Response jeweils am Kopfe der HTTP-Kommunikation mitgeschickt wurden. Sie werden daher auch im Tab *Headers* angezeigt. Im Bereich (1) kann die Datei ausgewählt werden, die vom Server abgerufen werden soll. Für die Seite `index.html` werden insgesamt vier Requests gestellt, wie (2) zeigt. Bereich (3) zeigt generelle Informationen zu der Anfrage. (4) zeigt Informationen zur Antwort des Servers, (5) listet Daten auf, die mit dem Request zum Server geschickt wurden. Hinter *view source* (6) verbirgt sich ein Umschalter, der die Daten nochmal in Rohform anzeigt.

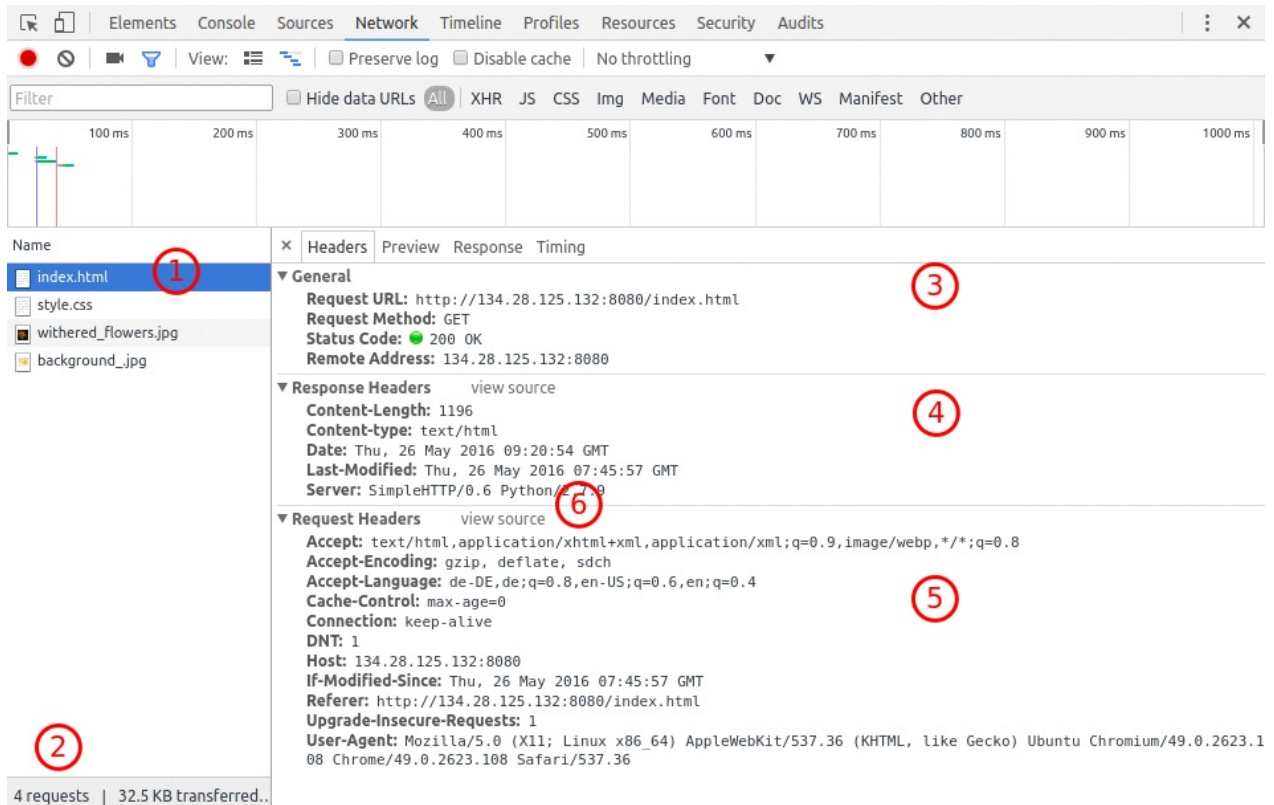


Abbildung 4.1.3.3 - Anzeige der Header-Daten aus Request und Response

Klicks auf andere Dateien in (1) bringen andere Informationen in (3-6) hervor.

## Fragen und Aufgaben

- Welche HTTP-Statuscodes gibt es? Tragen Sie bitte unter diesem Aufzählungspunkt einige einschlägige Quellen zusammen!
- Füllen Sie das Formular Ihrer Website aus. Schicken Sie es ab. Was passiert? Finden Sie heraus, wie Sie in den Entwicklertools die übertragenen Formulardaten auf dem Weg zum Server sehen können!

# Atom

Atom ist ein freier Editor für alle gängigen Betriebssysteme. Er kommt aus dem Hause [GitHub](#) und kann mit *Packages* angepasst und erweitert werden.

Im folgenden sind die Packages aufgeführt und erläutert, die in der Veranstaltung eine Rolle spielen.

Auf der [Packages-Website](#) können zahlreiche weitere Pakete heruntergeladen werden.



# Packages

Packages erweitern die Funktionalität von Atom in unterschiedlichsten Bereichen. Im folgenden sind die Packages aufgeführt, die im Kontext der Veranstaltung besprochen und konfiguriert wurden.

- Packages
  - Autosave
  - Pigments
  - remote-sync
    - Lokal arbeiten, per SSH hochladen
    - Installation des Packages *remote-sync*
    - Konfiguration von *remote-sync*
    - Die Konfigurationsdatei `remote-sync.json` erstellen
    - Konfiguration testen
    - Tipps

Den Bereich zur **Installation** von Packages wird geöffnet über `Edit -> Preferences -> Install` .

Den Bereich zur **Konfiguration** von Packages wird geöffnet über `Edit -> Preferences -> Packages` .

## Autosave

**Quelle:** Core Package

**Zweck:** Speichert die geöffnete Datei automatisch

**Konfiguration:** Es gibt eine sinnvolle Funktion von autosave, die per Default deaktiviert ist: Sie speichert automatisch den Stand der aktiven Datei in Atom, wenn das Fenster verlassen wird. So spart man sich das manuelle Speichern, wenn man bei der Webentwicklung in den Browser umschaltet.

**Zugang:** Eingeschaltet wird diese Funktion über `Settings -> Packages -> autosave -> Settings -> Enabled` (nach unten scrollen)

## Pigments

**Quelle:** <https://atom.io/packages/pigments>

**Zweck:** Legt Farben auf Farbangaben im CSS

**Konfiguration:** Nach der Installation funktioniert das Package wie gewünscht.

## remote-sync

### Lokal arbeiten, per SSH hochladen

Eine SSH-Verbindung stellt eine verschlüsselte Verbindung zum Server her, sodass wir anschließend im Terminal arbeiten können. Die Befehlseingaben finden zwar immer noch auf unserem Laptop statt, ausgeführt werden sie aber auf dem Raspberry Pi. Diese Übertragungstechnik werden wir uns nun zunutze machen, indem wir ein Plugin für Atom installieren, das bei jedem Speichern einer Datei die gemachten Änderungen an den Raspberry überträgt.

## Installation des Packages *remote-sync*

**Quelle:** <https://atom.io/packages/remote-sync>

**Zweck:** synchronisiert Dateien zwischen Client und Server

## Konfiguration von *remote-sync*

1. In den *Settings* auf *Settings* in der Zeile des Packages klicken.
2. Hier sind nicht viele Einstellungen möglich, da das Konzept für die Konfiguration des Packages eine Konfigurationsdatei vorsieht. Diese heißt standardmäßig `.remote-sync.json`. Würdest Du diesen Namen dieser Datei ändern wollen, könntest Du es hier tun.

Damit sind wir an dieser Stelle schon durch mit den Einstellungen und müssen uns stattdessen der Konfigurationsdatei widmen.

## Die Konfigurationsdatei `remote-sync.json` erstellen

Für den nächsten Schritt wird dafür ausgegangen, dass das Verzeichnis des Projekts in Atom ähnlich wie das folgende aussieht:

```
secondhandblumen_flask/
├─ app.py
├─ app.pyc
├─ dummy.txt
├─ .remote-sync.json
├─ static
│   └─ css
│   └─ img
├─ templates
│   └─ anbot.html
│   └─ .git
│   └─ .gitignore
│   └─ img
│   └─ impressum.html
│   └─ index.html
│   └─ kontakt.html
│   └─ LICENSE
└─ team.html
```

Das Package *remote-sync* tut nur klaglos seinen Dienst, wenn der Ordner, der synchronisiert werden soll, direkt in Atom geöffnet ist.

Gemäß der gezeigten Projektstruktur könnte die Konfigurationsdatei wie folgt aussehen:

```
{
  "transport": "scp",
  "hostname": "192.168.178.100",
  "port": 22,
  "username": "username",
  "password": "password",
  "target": "/home/username/www/secondhandblumen_flask",
  "uploadOnSave": true,
  "saveOnUpload": true,
  "watch": [],
  "ignore": [
    ".remote-sync.json",
    ".git/**"
  ]
}
```

Die jeweiligen Parameter sind der [Dokumentation des Packages auf der Website](#) zu entnehmen.

Die Erfahrung zeigt, dass ein Reload von Atom nach einer Änderung der Konfiguration notwendig ist.

## Konfiguration testen

Legen Sie eine neue Datei an und führe einen Rechtsklick darauf aus. Wählen Sie im Kontextmenü *Remote Sync -> Upload File*. Beobachten Sie in der Logzeile, ob der Vorgang gelingt.

## Tipps

1. Der Parameter `watch` dient der Überwachung von Dateien im Projekt. Werden sie geändert, findet automatisch ein Upload statt.

# Einige Linuxbefehle für die Kommandozeile

- Einige Linuxbefehle für die Kommandozeile
  - `ssh`
  - `sudo`
  - `man`
  - `apt-get`
- Fingerübungen für die Shell
  - Dateioperationen
  - `pwd`, `cd`
  - Anmerkung: relative und absolute Pfade
  - absoluter Pfad
  - relativer Pfad
  - `mkdir`, `mv`, `ls`
  - `touch`, `rm`
  - `nano`

## ssh

```
ssh -l loginname rechnername | Starten einer sicheren Shell auf einem anderen Rechner
```

"Wenn man mit **ssh** zum ersten Mal eine Verbindung zu einem anderen Rechner herstellt, erscheint oft eine Warnung. Dies geschieht, weil **ssh** hier nachfragt, ob es dem anderen Rechner mit seiner IP-Adresse vertrauen darf. Beantwortet man diese Frage mit `yes`, speichert **ssh** den Namen und den *RSA-Fingerprint* (Code zur eindeutigen Identifizierung des anderen Rechners) in der Datei `~/.ssh/known_hosts`, so dass diese Nachfrage beim nächsten Anmelden nicht mehr erscheint.

Danach fragt **ssh** nach dem Passwort des Benutzers *loginname*. Nach der Eingabe des richtigen Passworts beginnt die Sitzung am anderen Rechner, was durch die Ausgabe des Promptzeichens `$` angezeigt wird. [...]

Die Secure-Shell **ssh** [...] ist ein Login-Programm, das eine vollständig verschlüsselte Sitzung zwischen Rechnern unterstützt. Für jede Verbindung wird hierzu zwischen den Partnern ein neuer Sitzungsschlüssel ausgehandelt, so dass einem potentiellen Angreifer kaum Zeit bleibt, einen solchen Schlüssel schnell genug zu knacken. [...]"<sup>1</sup>

## Kommentar

Auf Windows-Rechnern bietet sich die Verwendung des Programms `PuTTY` an. Unter Mac OS und Linux öffnen Sie ein Terminal (Mac: `cmd` + `SPACE`, Eingabe von Terminal (Spotlight-Suche), `Enter`) und verbinden sich dort mit dem Befehl

```
ssh loginname@rechnername
```

Für unseren RPI könnte das so aussehen (Das letzte Byte der IP müssen Sie anpassen, wenn Sie in der Uni sind. Zuhause muss die ganze IP in Ihr Heimnetz passen!):

```
ssh pi@134.28.125.225
```

<sup>1</sup>. Herold 2004:207 [↩](#)

---

## sudo

```
sudo [optionen] kommando
```

"Das Kommando **sudo** ermöglicht es, ausgewählte Benutzer bestimmte Programme unter einem anderen Login-Namen ausführen zu lassen<sup>2</sup>. So kann z.B. *root* festlegen, dass einige Benutzer auch administrative Aufgaben durchführen, wie z.B. neue Benutzer anlegen oder neue Drucker einrichten dürfen, ohne dass sie das Passwort von *root* kennen müssen. [...]

Damit ein Benutzer ein bestimmtes Kommando, das er normalerweise nicht ausführen kann, mittels **sudo** aufrufen kann, muss *root* einen entsprechenden Eintrag in der Datei

```
/etc/sudoers
```

vornehmen. Dies geschieht üblicherweise mit dem Aufruf des Kommandos

```
visudo
```

---

## Kommentar

Wenn man nicht eingestellt hat, dass der RPI in den grafischen Modus booten soll, fragt er nach dem Hochfahren nach einem Benutzernamen und einem Passwort. Meldet man sich dann als Benutzer *pi* an, hat man nicht die Privilegien des Administrators *root*. Das ist auch gut so, denn hätte man diese Rechte ständig, könnte man leicht etwas anrichten, dass man im Nachhinein bedauert, da jeder Befehl ohne Rückfrage ausgeführt wird<sup>3</sup>. Außerdem lässt sich das System ggf. angreifen, wenn man unter *root* arbeitet.

Mit dem Kommando **sudo** kann man sich als Benutzer *pi* die Rechte des Administrator holen, um bestimmte Aktionen durchzuführen, die eigentlich nur *root* möglich sind. Das ist oft der Fall auf dem RPI. Wenn also ein Kommando nicht ausgeführt wird, kann es helfen, **sudo** davorzuschreiben.

Wir werden in der *sudoers*-Datei arbeiten, wenn wir per PHP auf die Ein- und Ausgänge des RPI zugreifen.

## Links

- [Homepage des sudo-Programms](#)

<sup>2</sup>. Herold 2004:147 [↩](#)

<sup>3</sup>. Zu Linux-Kommandos, die man besser nicht ausgeführt hätte, gibt es [eine wahre Geschichte über Toy Story 2 von Pixar](#). [↩](#)

---

## man

```
man [optionen] [bereich] thema | Traditionelle Online-Hilfe für Unix
```

Mit **man** kann man das Manual eines Befehls anzeigen lassen. Beispiel:

```
man shutdown
```

"Das Kommando gibt die Informationen seitenweise am Bildschirm aus. Mit der ENTER-Taste kann man zeilenweise und mit der SPACE-Taste seitenweise vorwärts blättern. Mit der Eingabe von *q* wird **man** beendet. Die Beschreibung eines Kommandos im gedruckten Handbuch oder im Online-Manual nennt man *Manpage*<sup>4</sup>."

---

## Kommentar

Manpages muss man mögen, denn ohne sie bleibt einem das System schnell ein Buch mit sieben Siegeln. Daher rate ich, sich mit einfachen Befehlen wie **shutdown** oder **reboot** an das Lesen und Verstehen der Manpages heranzutasten.

Außerdem sollte man sich natürlich die Manpage von **man** auch mal ansehen...

<sup>4</sup>. Herold 2004:235 [↩](#)

## apt-get

```
apt-get | APT-Werkzeug für den Umgang mit Paketen
```

**apt-get** ist das Werkzeug, um das Betriebssystem zu aktualisieren und zusätzliche Software zu installieren. Um bspw. den Apache-Webserver zu installieren, ist folgendes Kommando einzugeben:

```
sudo apt-get install apache2
```

Die Folge ist, dass der Rechner das Programm aus dem Internet lädt und installiert. Wieso geht das so einfach?

---

## Die sources

In jedem Ubuntu/Debian-System gibt es eine Datei mit Namen

```
/etc/apt/sources.list
```

In dieser sind die Adressen der Server (Repositories oder Repos) gespeichert, die die freien Programme für die Linux-Distribution vorhalten. Der obige install-Befehl holt das gewünschte Programm von diesen Servern.

## System-Update

Das Image für die Raspian-Distribution ist schon vom Februar 2015 und wurde sicherlich in vielen Punkten verbessert. Diese Verbesserungen werden alle an das entsprechende Repo übermittelt. Durch Eingabe von

```
sudo apt-get update
```

holt sich Ihr Rechner die verfügbaren Aktualisierungen aus dem Repo. Eine Aktualisierung der Software findet aber noch nicht statt!

Durch anschließende Eingabe von

```
sudo apt-get upgrade
```

geben Sie das Kommando, die Aktualisierungen herunterzuladen und zu installieren. Das System fragt nach, ob Sie die aufgelisteten Programme wirklich aktualisieren wollen. Sagen Sie Ja, dann geht es los! Beim ersten Mal dauert die Aktualisierung sehr lange, haben Sie Geduld. Später geht es dann viel schneller.

## Aufgabe

- Führen Sie eine Aktualisierung des Betriebssystems nach den zuvor beschriebenen Schritten durch!
- Lesen die die Man Page zu apt-get. Sie ist in deutscher Sprache verfügbar.

---

## Fingerübungen für die Shell

Die folgende Übung geht davon aus, dass Sie per SSH mit Ihrem Raspberry verbunden sind. Um die einzelnen Aufgaben durchzuführen, müssen Sie sich unter Window mit PuTTY und unter Mac/Linux aus dem Terminal mit Ihrem laufenden Raspberry Pi verbinden. Dieser muss sich im selben Netzwerk befinden und vom DHCP-Server (meist Ihr Router) eine IP bekommen haben. Diese müssen Sie kennen (Administrationsoberfläche des Routers aufrufen).

Unter Mac/Linux lautet das Kommando zum Verbinden per SSH

```
ssh pi@0.0.0.0 (wobei 0.0.0.0 durch die IP des RPi zu ersetzen ist)
```

## Dateioperationen

Wenn Sie souverän im Terminal arbeiten wollen, fangen Sie am besten mit den Befehlen zur Dateioperation an.

### pwd, cd

- Wechseln Sie mit `cd` in Ihr Heimatverzeichnis.
- Zeigen Sie mit `pwd` an, in welchem Ordner Sie sich gerade befinden.
- Wechseln Sie mit `cd ..` in den übergeordneten Ordner.
- Wechseln Sie noch ein Verzeichnis höher. Verwenden Sie dazu die *history* der Kommandozeile, um zuvor eingetippte Befehle erneut aufzurufen (Cursor Up).
- Finden Sie heraus, in welchem Ordner Sie jetzt sind.
- Praktisch: mit `cd -` wechseln Sie zum vorherigen Ort zurück, ohne den Weg dorthin erneut eingeben zu müssen.

---

## Anmerkung: relative und absolute Pfade

Wie Sie schon bemerkt haben, ist das Trennzeichen von Ordnern und Dateien in einem Pfad unter Linux der Slash ( / ). Beim Navigieren im Dateisystem aber auch für das Programmieren komplexerer Webseiten ist das Verständnis von **relativen** und **absoluten** Pfaden unerlässlich. Daher einige Beispiele:

### absoluter Pfad

Der **absolute** Pfad zu der Datei `index.html`, die wir in der Veranstaltung angelegt haben, lautet

```
/home/pi/www/[ihrordner]/index.html
```

Das erklärt sich so: Der führende Slash zeigt an, dass der Pfad von der Wurzel (root) des Dateisystems aus beschrieben wird. Weiter nach oben als bis zu diesem führenden Slash kommen Sie nicht in Ihrem Dateisystem.

### relativer Pfad

Wechseln Sie mit `cd` in Ihr Homeverzeichnis. `pwd` sollte ausgeben, dass Sie sich in `/home/pi` befinden.

Wechseln Sie nun nochmal mit `cd ..` in das darüber liegende Verzeichnis.

Angenommen Sie wollten den Ordner `pi` betreten, geben Sie nun ein



```
cd pi
```

Sie sehen, es fehlt der führende Slash! Würden Sie stattdessen eingeben

```
cd /pi
```

würde dies zu einem Fehler führen. Mit dieser Eingabe eines absoluten Pfades würden Sie behaupten, es gäbe ausgehend von der Wurzel ein Verzeichnis *pi*. Das gibt es aber nicht, daher der Fehler.

**Regel:** Relative Pfade werden immer gebildet vom aktuellen Standort - ohne führenden Slash. Absolute Pfade mit führendem Slash ausgehend von der Wurzel des Dateisystems.

---

## mkdir, mv, ls

- Wechseln Sie in Ihr Homeverzeichnis.
- Legen Sie mit `mkdir` einen Ordner `Documents` an.
- Legen Sie einen weiteren Ordner `Downloads` an.
- Benennen Sie den Ordner `Documents` in `Dokumente` um.

```
mv Documents Dokumente
```

- Zeigen Sie den Verzeichnisinhalt mit `ls -la` an.
- Lesen Sie die Man Pages zu `mv` und `ls`.

## touch, rm

**Achtung!** `rm` ist ein mächtiger Befehl, mit dem Sie Ihr gesamtes System zerstören können. Also, los geht's!

- Wechseln Sie in den Ordner *Dokumente*.
- Legen Sie mit `touch` einige leere Dateien an.

```
touch Plan.txt  
  
touch Noten.txt  
  
touch Zeiten.txt
```

- Löschen Sie mit `rm` die Dateien wieder:

```
rm Plan.txt  
  
rm Noten.txt Zeiten.txt
```

- Wechseln Sie in den übergeordneten Ordner.
- Löschen Sie mit `rm -r` den Ordner `Dokumente` .

```
rm -r Dokumente
```

**Kommentar:** Ordner müssen mit der Option `-r` (für rekursiv) gelöscht werden. Es werden selbstverständlich auch alle Inhalte des Ordners gelöscht.

## nano

`nano` ist ein einfacher Texteditor, den wir zunächst benutzen werden. Sie starten die Bearbeitung einer neuen Datei z.B. mit

```
nano meine-datei.txt
```

Den Dateinamen muss man nicht gleich angeben, beim ersten Speichern werden Sie eh nochmal gefragt. Ihnen muss allerdings klar sein, dass die Datei in dem Ordner erzeugt und gespeichert wird, in dem Sie sich gerade befinden.

Für unsere Zwecke brauchen wir in `nano` zunächst nur zwei Tastenkombinationen:

```
STRG + O zum Speichern  
STRG + X zum Beenden des Programms
```

Probieren Sie's aus! Legen Sie einige Dateien an, schreiben Sie was hinein, löschen Sie die Dateien wieder, benennen Sie sie um etc. Irgendwann kommt das Gefühl auf, dass Sie die Maschine unter Kontrolle haben. Bis dahin sollten Sie üben...

## Bücher

Eine Literaturliste findet sich bei [Zotero](#). Beiträge sind dort herzlich willkommen.

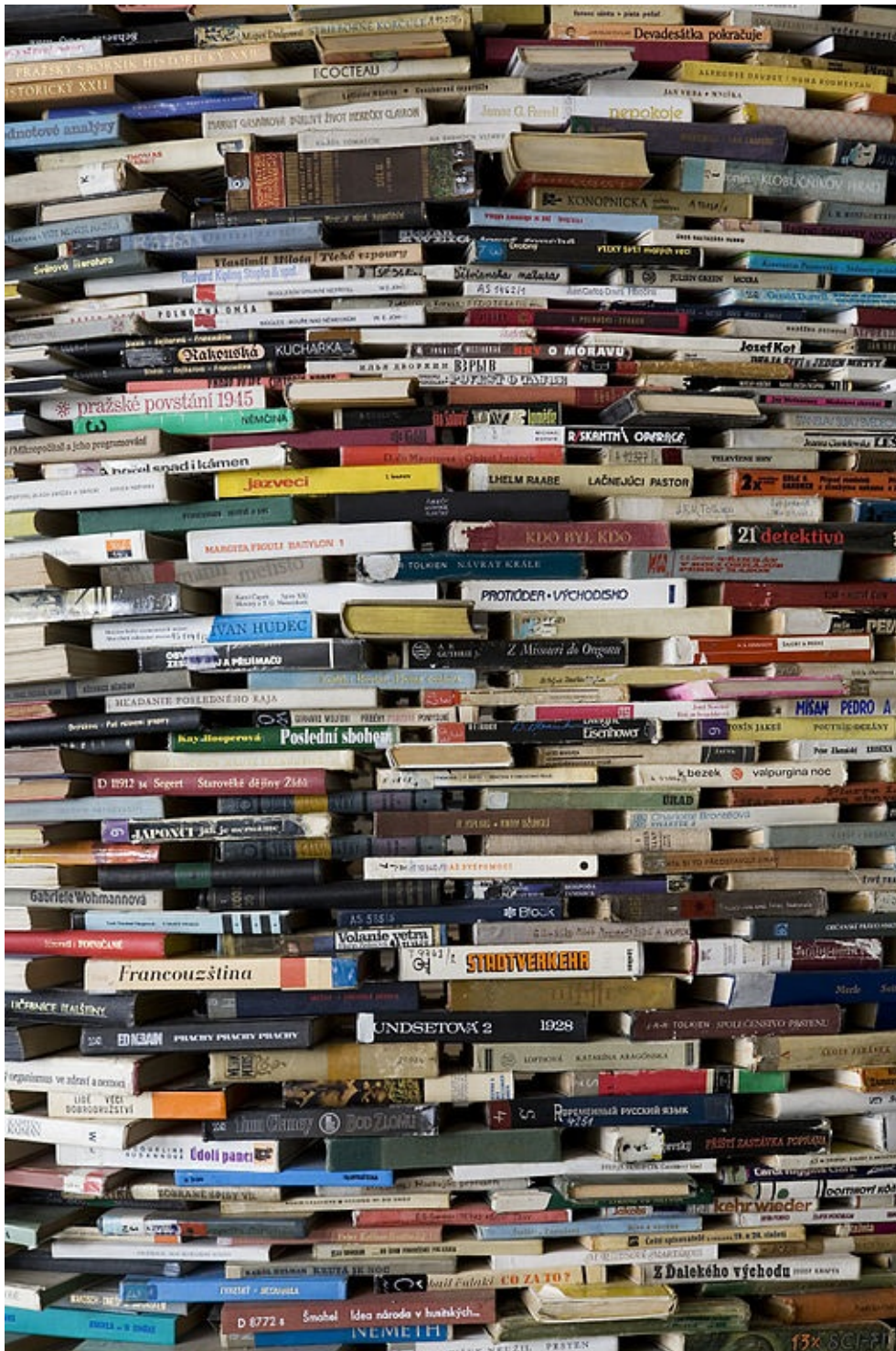


Abbildung 5.1.1 - A Tower of Used Books (© Jorge Royan / <http://www.royan.com.ar> / CC BY-SA 3.0)

