Webentwicklung

Einführung: Organisatorisches & Überblick

Inhalt dieser Einheit

1. **Organisatorisches**
   - Wie läuft das hier alles ab?
2. **Web und Internet**
   - Da gibt es einen Unterschied?
3. **Vogelperspektive**
   - Was umfasst Webentwicklung alles?
   - Was werden wir behandeln, und was nicht?
4. **Ein Blick auf den Stoffplan**

Organisatorisches

Wie läuft das Ganze hier ab?

Willkommen!

- Dozent: **Franz Zieris**
  - Lehrbeauftragter an der HTW *(d.h. kein Büro, Sprechstunde, …)*
    - sonst: Mitarbeiter der AG Software Engineering, Freie Universität Berlin
    - seit 2006: freiberuflicher Softwareentwickler, Fokus Webentwicklung
  - Kontakt: zierisf@htw-berlin.de, oder sprechen Sie mich an
- Kurs-Format:
  - Mittwoch: Seminaristischer Lehrvortrag + Übung (→ **C 625**)
  - Dazwischen: Bearbeitung von Übungsaufgaben
  - Am Ende: Klausur
- www.zieris.net/webdev
  - Termine
  - Präsentationsfolien
  - Übungsaufgaben
  - Zusatzmaterial

Disclosure

- Basis für die meisten Vortragsfolien: Max Beier
  - bereitgestellt unter CC BY-SA

Modulprüfung

- Zwei Prüfungskomponenten (§9 RStPO):
  1. Praktische **Übungsaufgaben**, verteilt über das Semester
     - ("Modulbegleitend geprüfte Studienleistung", §12 RStPO)
     - Bearbeitung allein, keine Gruppenarbeit (§13 RStPO)
  2. **Klausur** in Schriftform, am 2018-01-31
     - ("Schriftliche Modulabschlussprüfung", §10 RStPO)

- Gewichtung für Modulnote: jeweils 50%

Übungsaufgaben

- alle zwei Kurs-Wochen eine neue Aufgabe
  - Ausnahme: 1. Aufgabe ist klein, nur eine Woche Zeit
- Bearbeitung:
  - allein, keine Gruppenarbeit
  - Fragenstellen in der Übung, Bearbeitung dazwischen
  - mehrmalige Abgabe/Verbesserung im Zeitrahmen möglich
- Bewertung:
  - persönliche Vorführung
    - in der Übung am jeweiligen Abgabetag
  - Aufgaben geben unterschiedlich viele Punkte
    - volle Punktzahl, wenn alle Anforderungen erfüllt sind

Übungsaufgaben: Plagiate

- **Plagiate**:
  - Fremde Codeschnipsel ohne Quellenangabe
  - Fremde Lösungen

- Konsequenzen:
  1. Mal: Abzug der möglichen Punktezahl
  2. Mal: Kurs nicht bestanden

Übungsaufgaben: Technisches

- Lösungen werden über Git verwaltet
  - dezentrale Versionskontrolle (toll und relevant)
- Git-Repositories und Datenbanken auf studi.f4...
  - mit Lesezugriff für Dozenten (siehe 1. Übungsaufgabe)
  - vorzuführende Arbeitsergebnisse müssen vor der Übung im Repository liegen

- Bei Fragen zwischen den Übungen:
  - Bittebittebitte, *keinen* Code per E-Mail verschicken,
  - lieber einen Git-Branch anlegen und URL zusenden.

Software

- Folgende Software wird zum Einsatz kommen
  - Apache HTTP-Server
  - MySQL-Datenbank
  - PHP
- Sonst freie Wahl
  - Betriebssystem, Webbrowser, Editor
- Empfehlung für Laborrechner:
  - Linux, Firefox/Chromium, Netbeans IDE

Was ist denn nun eigentlich

Webentwicklung?

Web und Internet

Gibt es da etwa einen Unterschied?

Sputnik

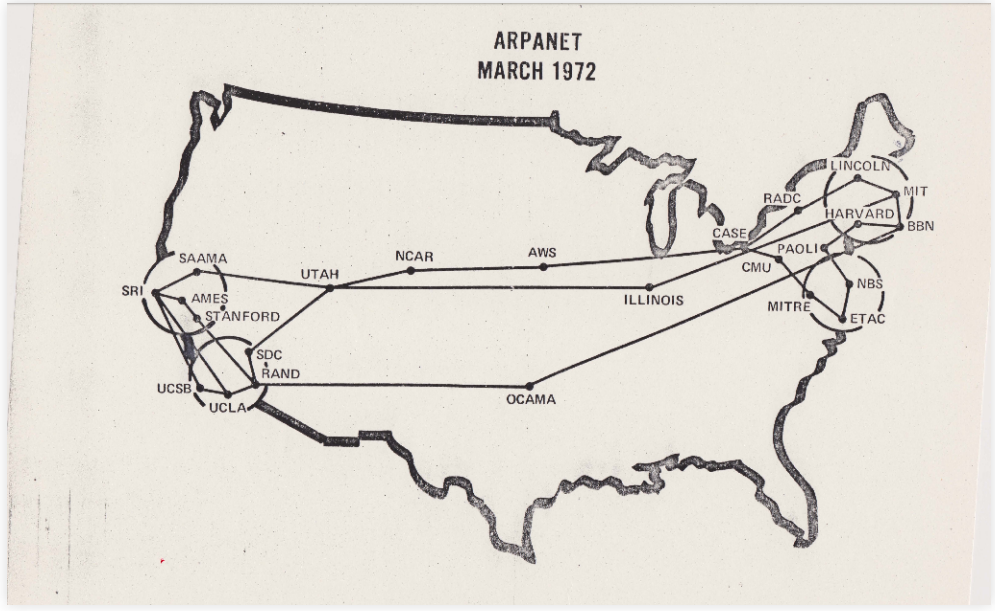
- 4. Oktober 1957: Sowjetunion bringt erfolgreich den ersten Satelliten “Sputnik 1” in Erdumlaufbahn.

- 7. Februar 1958: Gründung der *Advanced Research Projects Agency*, ARPA in den USA, um gezielte Technologieforschung zu betreiben.

Quelle: http://de.wikipedia.org/wiki/Datei:Sputnik_asm.jpg

Internet-Vorfahr: Arpanet

- Mitte/Ende 1960er: erste ARPANET-Pläne
- 1969: Betrieb mit 4 Knoten
- 1971 (Bild): 15 Knoten; Telnet und FTP werden entwickelt

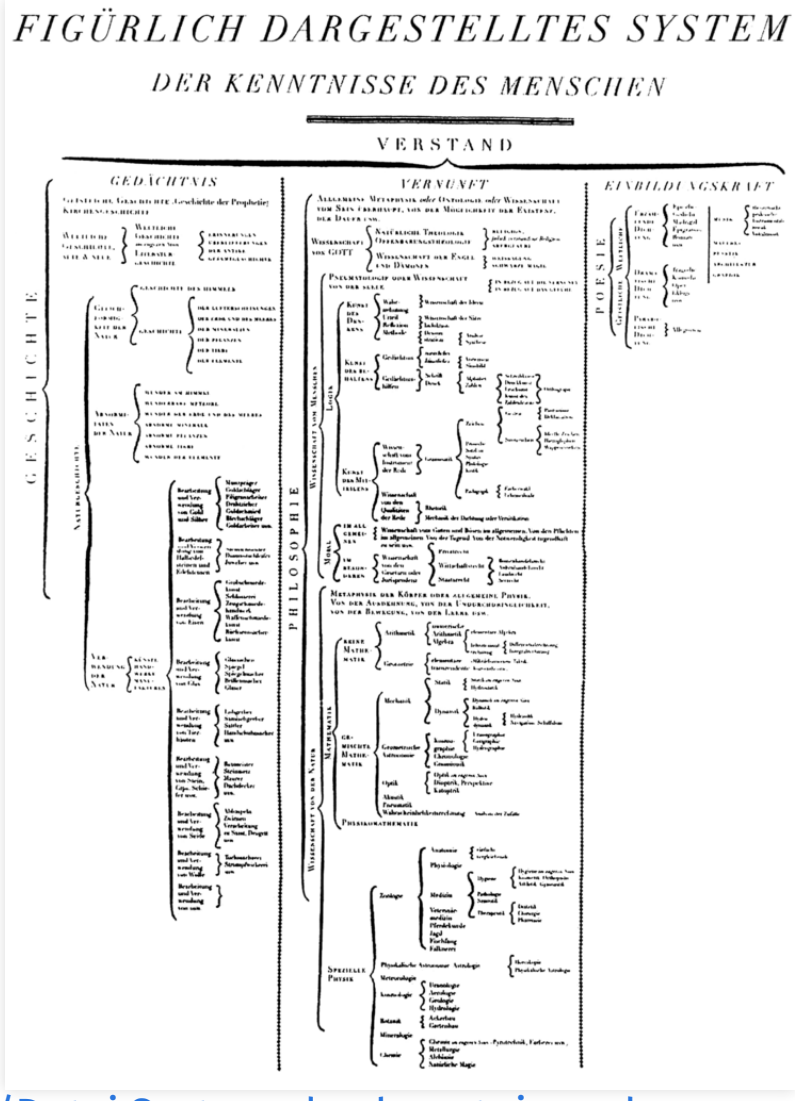


- Eigene ähnliche Netze in Europa (GB, FR)
- Ende 1970er/Anfang 1980er: Netze verbinden (Tiefseekabel, Satelliten)
- Internationale Bemühungen → *Interconnected Networks*, kurz *Internet* (via TCP/IP)

Quelle: https://commons.wikimedia.org/wiki/File:Arpanet_1972_Map.png

Wissenrepräsentation: Baum

- Baum-Modelle: hierarchisch, linear
  - z.B. klassische Bibliotheken, Enzyklopädien
- Encyclopédie (D'Alembert und Diderot, 1751-1780):
  - 35 Bände, 70.000 Artikel: Baum des Wissens
  - *mit* Querverweisen



Quelle: https://de.wikipedia.org/wiki/Datei:System-der-kenntnisse-des-menschen-retouched.png

Wissensrepräsentation: Netz

- Europäische Organisation für Kernforschung (CERN)
  - Viele Leute, viele Projekte
  - Informationen darüber waren zwar im Internet, aber hierarchisch strukturiert

- **Tim Berners-Lee**: unzufrieden damit
  - 1990: Paper *“Information Management: A Proposal”*
    - beschreibt ein System zur besseren Verwaltung von Informationen
    - statt einer Baumstruktur: Netz, da dieses besser das menschliche Denken widerspiegelt

Quelle: http://cds.cern.ch/record/2259198

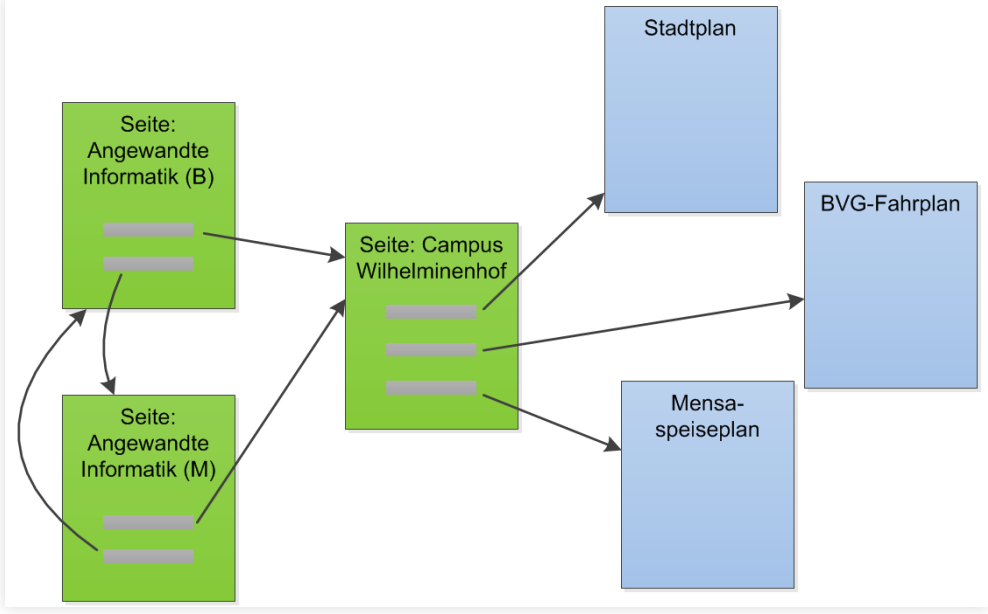
Die Geburt des Web

- **Tim Berners-Lee** (→): entwickelte Hypertext-System “WorldWideWeb”
  - auf Basis des Internets (hier schon 22 Jahre alt) damit weltweit
- Technisch:
  - Dokumente anreichern um Querverweise, um zu anderen Dokumenten zu springen
  - er entwickelte das Kommunikationsprotokoll HTTP
    - und die Auszeichnungssprache HTML
    - und einen Browser
    - und einen Server, der Hypertext-Dokumente auslieferte

Quelle: http://cds.cern.ch/record/39437

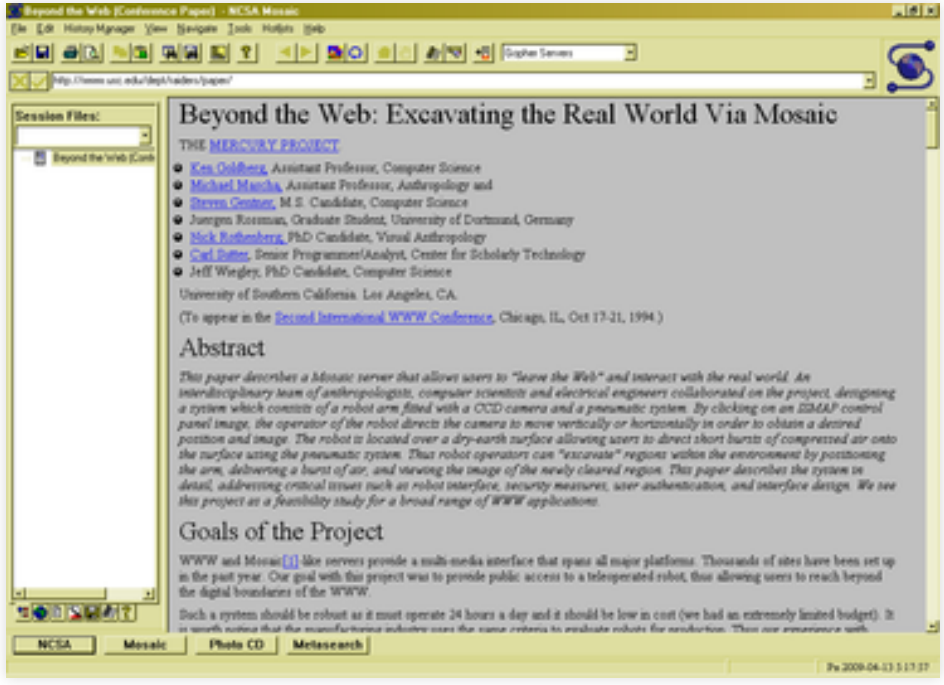
Hypertext

- Hypertext-Systeme (gr. “*hyper*”, über etwas):
  - Dokumente nicht linear verknüpft
  - gab es als Idee schon länger
- Hypertext-Dokumente
  - Links: unidirektionale Verbindung zw. Dokumenten
    - leichte Vernetzung, keine zentrale Stelle mit Meldepflicht
  - Navigation mit visuellen Browsern (Point-and-Click) erleichterten Zugang
  - senken Redundanz: einmal schreiben, dann verlinken



Erste Browser

- 04/1993: CERN gibt World-WideWeb-Quelltext frei
- 1993: NCSA veröffentlicht Mosaic (→)
  - Name: neben HTTP- auch FTP-, NNTP-, und Gopher-Client
  - Unterstützte Inline-Bilder
  - Für Abenteurer: https://github.com/alandipert/ncsa-mosaic
- Ende 1993: Mosaic für Apple Macintosh & Windows
- Danach: immer mehr Browser
  - häufig aus Forschungsprojekten entstanden, etwa 1994: Opera

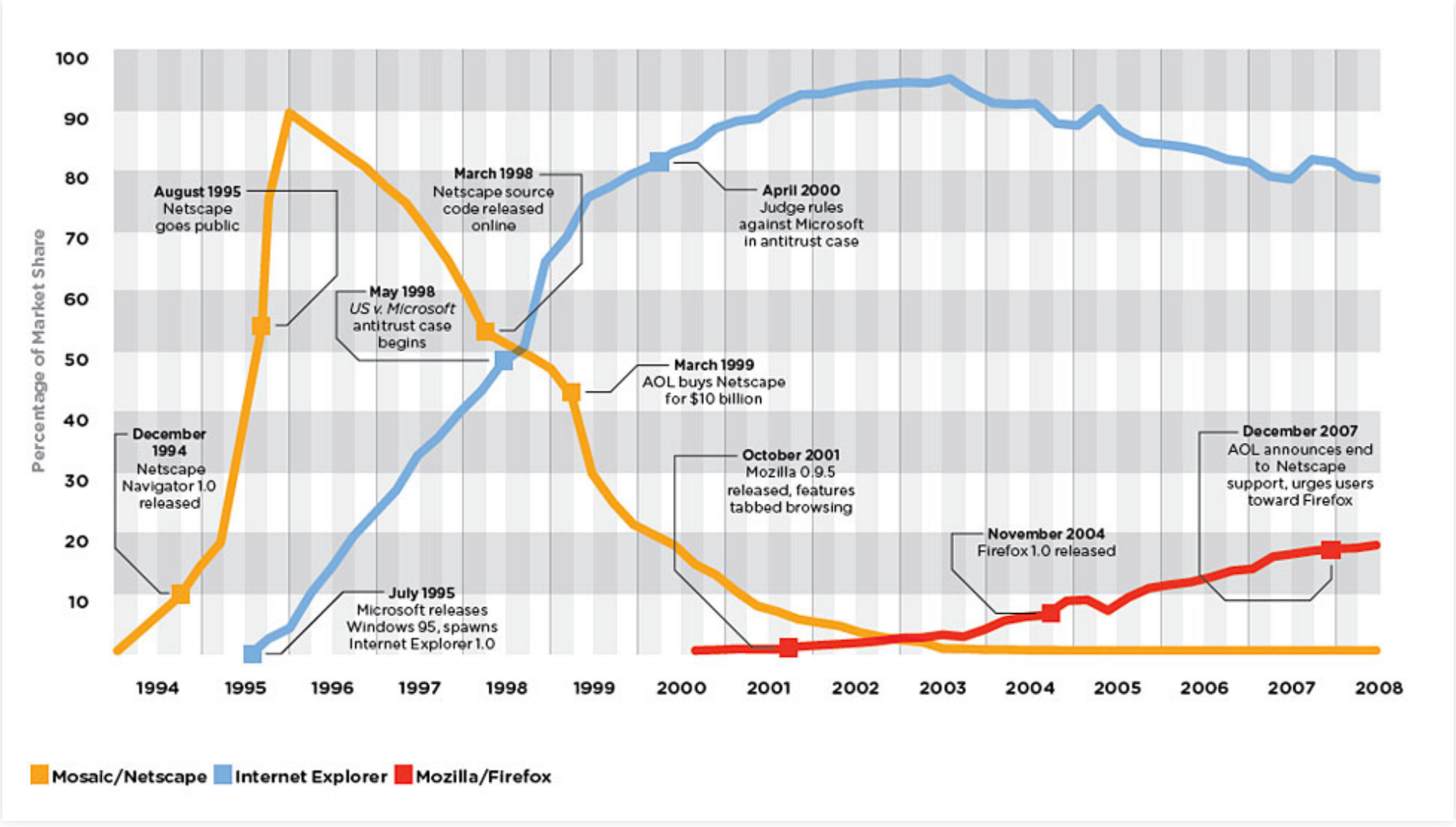


Quelle: https://en.wikipedia.org/wiki/File:NCSA_Mosaic.PNG

Browser Wars

- NCSA Mosaic 3.0 für Windows
  - Popularität zieht Kommerz an
  - Mosaic wurde kommerziell, umbenannt in “Netscape Navigator”
  - Implementierte “LiveScript” (heute JavaScript) und Cookies
- Microsoft
  - anfangs Kooperation mit Netscape, dann aber Konkurrenzprodukt Internet Explorer
  - **Positiv**: Konkurrenz befördert den Fortschritt, neue Features, …
  - **Negativ**: Feature-Wildwuchs ohne Spezifikation statt Bugfixes
  - Folge: Entwicklung mehrerer Versionen von Webseiten für verschiedene Browser
    - Zum Glück fast ausgestorben: Browserweichen

Browser Wars



Quelle: https://www.wired.com/2008/09/the-browser-wars/

Standardisierung durch W3C

- 1994: Berners-Lee gründet das W3C (*World Wide Web Consortium*)
  - Ziel: Standardisierung der Web-Protokolle und -Technologien
- Veröffentlichte Spezifikationen zu HTML 4.01, Bildformat PNG, Cascading Style Sheets, …
  - Forcierte aber nie deren Einhaltung: Standards sind “Recommendations”, *Empfehlungen*

HTML 5, Living Standard

- **WHATWG** (*Web Hypertext Application Technology Working Group*), seit 2004
  - Entwickler und Firmen (Opera, Mozilla, Apple, …)
  - Ziel: neue Generation von Anwendungen, bessere Spezifikation
  - HTML als Living Standard
    - ohne Versionsnummern, aktuell “Last Updated 11 October 2017”
- **W3C** adaptiert seit 2007 WHATWG-Spezifikation für HTML5-*Recommendations*
  - mit Versionsnummern, aktuell “HTML 5.1 2nd Edition”
- Einige HTML5-Vorteile:
  1. native Features, die vorher nur durch Plugins möglich waren
     - Video/Animationen: **Flash**; Formularvalidierung: **jQuery**
  2. klar definierter Parsing-Algorithmus: browserübergreifend gleiche Generierung des DOMs aus dem HTML-Markup
     - (mehr dazu in der nächsten Einheit)

Web vs. Internet

- **World Wide Web** (www):
  - über das Internet abrufbares System
  - von elektronischen Hypertext-Dokumenten
  - die durch Hyperlinks untereinander verknüpft sind
  - und über das Protokoll HTTP (oder HTTPS) übertragen werden.
- Drei fundamentale Bestandteile:
  - HTML: Hypertext Markup Language
  - URL: Uniform Resource Locator
  - HTTP: Hypertext Transfer Protocol
- **Browser** als übliches Betrachtungsprogramm

- Das Internet verknüpft **Teilnehmer**
- Das Web verknüpft **Informationen**

Zurück zur Frage: Was ist denn nun

Webentwicklung?

Vogelperspektive

Was umfasst Webentwicklung alles?
Was werden wir behandeln, was nicht?

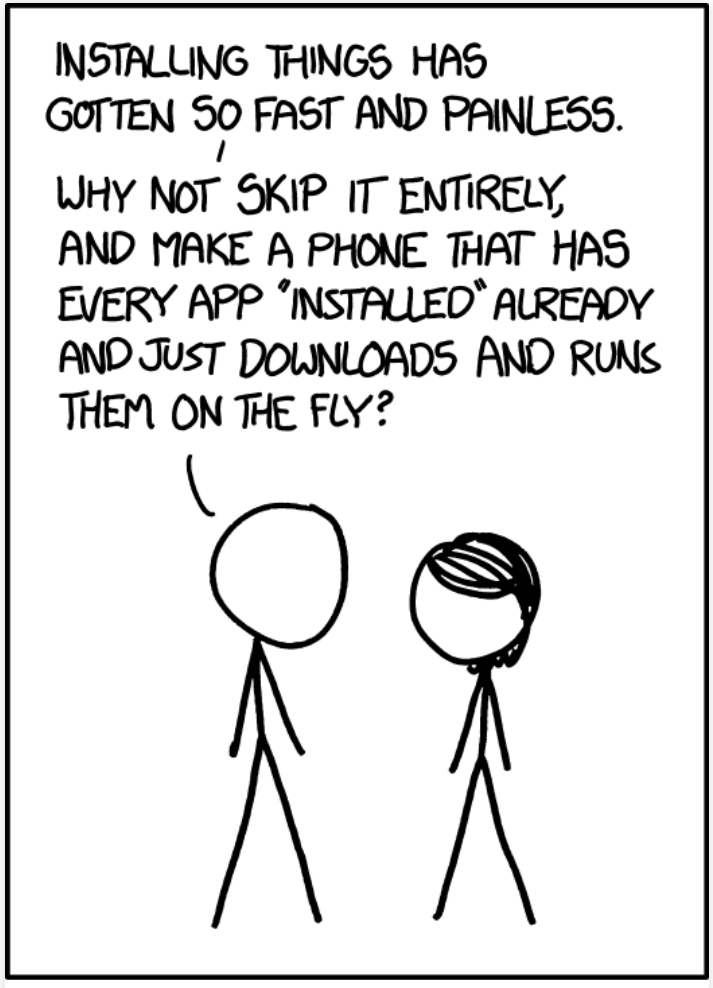
Anwendungsentwicklung allgemein

- **Nutzer/in**:
  - Hat ein Ziel, und möchte dieses effizient und angenehm erreichen
- **(Anwendungs-)Entwickler/innen** (Sie!):
  - Aufgabe: Dem Nutzer ermöglichen sein/ihr Ziel zu erreichen

Was könnten Nutzerziele sein?

- Schier endlose Anwendungsfälle, z.B.
  - Ziel: **Zugriff auf Inhalte**
    - Lexikon
    - Video-Plattformen
  - Ziel: **Zugriff auf Anwendungen**
    - Office-Anwendung
    - Bild-Bearbeitung
  - Ziel: **Wirkung entfalten**
    - Social Media
    - Online-Banking
    - eCommerce (div. Güter, Dienstleistung)
- Die Grenzen verschwimmen beim näheren Hinsehen
  - Es gibt eher graduelle Unterschiede z.B. bei …
    - Ausmaß der Interaktivität in der Bedienschnittstelle,
    - Rechenkomplexität im Hintergrund

Webentwicklung



Quelle: https://xkcd.com/1367/

Webentwicklung?

- … die *Arten* der Anwendungen sind vielfältig.
  - Das “Web” als Plattform-, oder gar Betriebssystem
  - Einfacher: Charakterisierung über beteiligte Technologien:

- **Webentwicklung**:
  - Entwicklung von Anwendungen in **Client-Server-Architektur**
  - Client und Server kommunizieren über **HTTP**
  - (meistens:) Client läuft in einem Webbrowser

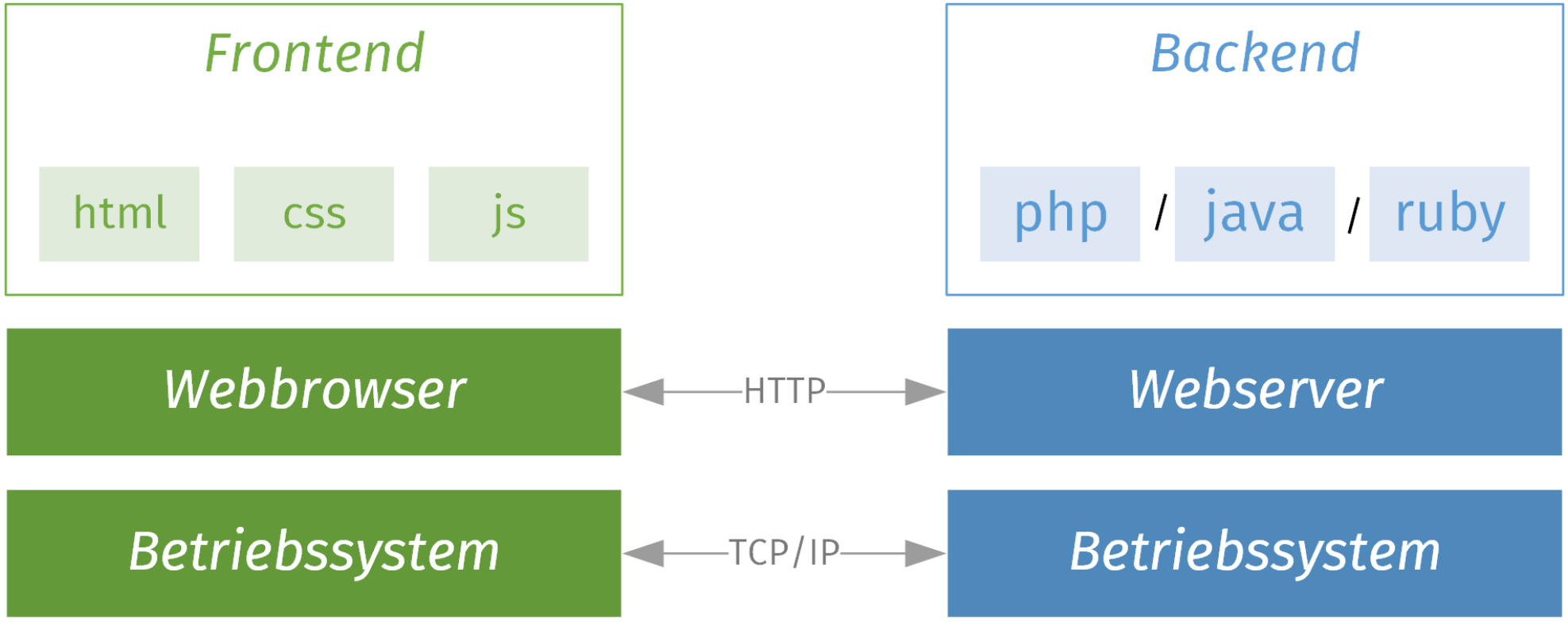
Frontend vs. Backend

- keine messerscharfe Trennlinie, aber meistens etwa so:

- **Frontend** (“Client”)
  - nutzerseitig ausgeführte/interpretierte Anwendungsteile, Bedienschnittstelle
  - Umgebung: keine Kontrolle seitens der Entwickler
    - Verlassen auf Standards (W3C, WHATWG)
  - Webbrowser: kann “nur” HTML, CSS und JavaScript

- **Backend** (“Server”)
  - stellt Inhalte/Dienste bereit, ruft weitere “Wirkungen” hervor
  - letztlich: HTML, CSS und JavaScript als Ausgabe
  - aber: volle Kontrolle seitens der Entwickler
    - Implementierung sehr flexibel (z.B. Java, Python, PHP, Ruby, ASP.NET, …)

Frontend vs. Backend



Full-Stack Web Development

- Webentwicklung unter Berücksichtigung von Frontend-, Backend- und Querschnittsbelangen, z.B.:
  - Webbasierte Routenplanung:
    - *Aktueller Standort wird vom Browser ermittelt; Nutzer fängt an, Zielbeschreibung einzugeben* (**Frontend**)
    - *Verkehrsdaten und Orte sind in einer Datenbank gespeichert* (**Backend**)
    - *Nutzer erhält passende Vorschläge für Zielort* (**Backend/Frontend**)
    - *Verbindungsanfrage wird verarbeitet und optimaler Routenvorschlag ausgewählt* (**Backend**)
    - *Textuelle Wegbeschreibung und zoom- und scrollbare Karte wird im Browser angezeigt* (**Frontend**)
    - *Routenparameter können geändert werden und führen zur serverseitigen Neuberechnung der Route* (**Frontend/Backend**)
    - *Routenplanungdienst kann Anfragen hunderter Nutzer parallel bearbeiten* (**Backend**)
    - …

Full-Stack Web Development

- Entwurf, Implementierung, Qualitätssicherung, Betrieb und Wartung von client- und serverseitigen Systemen zur Erfüllung **funktionaler Anforderungen**
  - HTTP/HTTPS, (und auch TCP/IP, DNS)
  - HTML, CSS, JavaScript
  - div. serverseitige Sprachen und Technologien
- Berücksichtigung **nicht-funktionaler Anforderungen**, wie:
  - Gebrauchstauglichkeit (Usability): Responsiveness, Mobile Web, bis hin zu gestalterische Aspekten (Übergang zum Webdesign)
  - Effizienz: Ressourcenbedarf, Antwortzeiten
  - Informationssicherheit & Datenschutz, Schutz gegen Angriffe
  - …

Inhalte und Ziele der Veranstaltung

- **Inhalte**
  - Die beste Web-Technologie und das beste Framework
    - lernen Sie hier nicht, denn die …
  - … veralten wahnsinnig schnell: Ständig gibt es eine tolle, neue Webtechnologie, oder ein neues JavaScript-Framework
    - *“Hierzulande musst du so schnell rennen, wie du kannst, wenn du am gleichen Fleck bleiben willst.”* (Lewis Caroll)
      - https://de.wikipedia.org/wiki/Red-Queen-Hypothese
- **Ziele**
  1. **Orientierung:** Grundlagen verstanden haben, für eine seriöse client- und serverseitige Technologieauswahl
  2. **Grundausstattung:** Einige konkrete Technologien beherrschen, um typische Aufgaben zufriedenstellend bearbeiten zu können
  3. **Selbsthilfe:** Sich für alles Weitere zu helfen wissen

Wichtige Hilfstechnologien

- (Nicht exklusiv für Webentwicklung …)
- Versionskontrolle: **Git** (oder SVN)
  - Historie von Quellcode-Änderungen
  - leichtere Koordination
- Datentransfer zwischen Umgebungen: **SCP** (oder FTP)
  - *lokal*: Entwicklungsrechner
  - *Produktivsystem*: für Nutzer erreichbare Instanz
    - *Deployment*: Verfügbarmachen auf Produktivsystem
  - (opt.) *Staging*: wie Produktivsystem, Vorstufe
- Kontrolle über entfernte Syteme: **SSH**
  - Neustarten von Diensten, Logfiles lesen, Wartungsarbeiten, …

Zusammenfassung: Heutige Inhalte

- Unterscheidung Internet vs. Web
- Was ist Webentwicklung?
  - Womit befasst sich Webentwicklung?
  - Was sind relevante Teilprobleme?
- Unterscheidung Frontend vs. Backend

- Blick auf den Stoffplan

Danke!