

# **WBE: WEB-ENTWICKLUNG**

## **EINFÜHRUNG**

# ÜBERSICHT

- Worum geht's in WBE?
- Organisatorisches
- Internet und WWW
- Client-Server-Architektur
- Die Sprachen des Web: HTML, CSS, JavaScript
- Web-Standards und APIs

# ÜBERSICHT

- Worum geht's in WBE?
- Organisatorisches
- Internet und WWW
- Client-Server-Architektur
- Die Sprachen des Web: HTML, CSS, JavaScript
- Web-Standards und APIs

# WORUM GEHT'S?

- Web-Technologien...?
  - HTML, CSS, JavaScript
  - Canvas, SVG, Geolocation, DOM, Websockets, uvm.
- Web-Plattform...?
  - Web-APIs und deren Einsatzmöglichkeiten
  - z.B. Web-Apps, Electron-Apps, Webserver uvm.
- Web-Entwicklung...?
  - Tools, Frameworks, Libraries

Das ist ein riesiges Gebiet!

## Speaker notes

Curriculum Studium Informatik

<https://www.zhaw.ch/de/engineering/studium/bachelorstudium/informatik/curriculum-lehrplan/>

# ZIELE WBE

- JavaScript gut verstehen und einsetzen können
- Nebenbei: HTML und CSS verstehen (Selbststudium)
- Auswahl von Web-APIs kennen und einsetzen können
- Eigenes kleines Web-Framework erstellen können

Klar ist: Das Vorhaben ist immer noch sehr ambitioniert

## Speaker notes

- WBE wurde im HS20 zum ersten Mal durchgeführt
- Ersetzt drei Kurse: WEB1, WEB2, WEB3
- Komplett neu aufgebauter Kurs
- Seither jährlich mit kleineren Anpassungen

Falls noch nicht alles ganz rund läuft:

Beschwerden an [bkrt@zhaw.ch](mailto:bkrt@zhaw.ch) 😊

# ERWARTETE VORKENNTNISSE

- Grundkenntnisse HTML und CSS
  - Mail am Ende des letzten Semesters
  - Moodle-Kurs zum Selbststudium
  - In Tests und Praktikumsaufgaben vorausgesetzt
- Grundlegende Programmierkenntnisse
  - Programmieren 1/2
  - Software-Projekt 1/2

## Speaker notes

Der Vorbereitungskurs im Moodle enthält eine Menge an Informationen, mehr, als Sie für WBE benötigen. So können Sie selbst entscheiden, wie tief Sie sich in die Grundlagen von HTML und CSS einarbeiten möchten. Klar, je tiefer desto besser, da Web-Kenntnisse in vielen Informatik-Disziplinen von Vorteil sind. Der beim Vorbereitungskurs verlinkte [HTML & CSS Crash Course](#) (Tutorial auf Youtube) liefert in wenigen Stunden einen guten Überblick.

# ÜBERSICHT

- Worum geht's in WBE?
- Organisatorisches
- Internet und WWW
- Client-Server-Architektur
- Die Sprachen des Web: HTML, CSS, JavaScript
- Web-Standards und APIs

# ABLAUF

- Zwei Unterrichtslektionen pro Woche
- Zwei Praktikumslektionen pro Woche

Zeit	Art
100..120h	ca. Arbeitsaufwand 4 Credits (25..30h pro Credit)
42h	Präsenzlektionen ( $14 * 4 * 0.75$ )
58..78h	Selbststudium, Prüfungsvorbereitung, Prüfung

# KURSNOTE

Anteil	Art
20%	Leistungsnachweise im Semester
80%	Semesterendprüfung (SEP)

# LEISTUNGSNACHWEIS IM SEMESTER

- Kurztests
- Bewertete Abgaben im Praktikum

Leistungsnachweise im Semester können die Kursnote verbessern, nicht aber verschlechtern – für misslungene oder verpasste Leistungsnachweise wird die Note der SEP eingesetzt

Tipp: Nutzen Sie die Gelegenheit, sich durch die Leistungsnachweise im Semester ein Polster für die SEP zu verschaffen

## Speaker notes

Leistungsnachweise im Semester können die Kursnote verbessern, nicht aber verschlechtern. Leistungsnachweise, welche schlechter ausfallen als die SEP gehen mit der SEP-Note in die Bewertung ein. Verpasste Leistungsnachweise werden mit 1.0 bewertet.

Konsequenz: Für verpasste Leistungsnachweise wird die Note der SEP eingesetzt. Dafür gibt es für verpasste Leistungsnachweise keine Nachholtermine. Ausnahme: Krankmeldung mit Arztzeugnis.

# SEMESTERENDPRÜFUNG

- Mix aus verschiedenen Aufgaben zu den WBE-Themen
  - Multiple-Choice (Variante Kprim)
  - Fragen mit kurzen Antworten
  - Code-Aufgaben
- Wir haben noch keine komplette Musterprüfung 😕
- Tests können aber als Muster für die SEP angesehen werden

## MULTIPLE-CHOICE MIT KPRIM

Aus <https://teachingtools.uzh.ch/de/tools/multiple-choice-co-aufgaben-erstellen>:

*Der Aufgabentyp Kprim ist vor allem für Prüfungssettings interessant. Er besteht immer aus einer Vignette und vier Antwortalternativen, die jede für sich entweder richtig oder falsch sein kann. Dieser Typ hat sich testtheoretisch bewährt (gute Trennschärfe, geringe Ratewahrscheinlichkeit). Per Definition gibt es nur die volle Punktzahl, wenn alle vier Antwortalternativen richtig beurteilt wurden. Bei drei von vier korrekten Entscheidungen gibt es die halbe Punktzahl, alle anderen Lösungsmuster geben 0 Punkte.*

Dazu noch ein Hinweis: Wenn zu einer Antwortalternative keine Auswahl getroffen wird, wird dies als falsch gewertet. Es ist daher sinnvoll, alle Antwortalternativen zu beurteilen.

## ANDERE FRAGEN

- Fragen mit kurzen Antworten können zum Beispiel sein: „Welchen Wert liefert die folgende Funktion, wenn sie mit ... aufgerufen wird?“
- Code-Aufgaben wird es nur vereinzelt geben, zum Beispiel: „Schreiben Sie eine Funktion, welche...“

# UNTERLAGEN

In elektronischer Form (PDF) im Moodle:

- Die im Unterricht gezeigten Folien
- Ergänzende Hinweise zu den Folien (Lecture Notes)
- Praktikums-Aufgaben und Beispiel-Code

<https://moodle.zhaw.ch/course/view.php?id=20705>

# UNTERRICHT UND PRAKTIKUM

## Praktikumslektionen

- An den Notebooks gearbeitet
- Gespräche, Diskussionen, Teamarbeit erwünscht

## Unterrichtslektionen

- Vortrag und Demos, Diskussionen mit ganzer Klasse
- Gespräche untereinander bitte vermeiden – führen zu schlechterem Unterricht und stören alle
- Notizen machen empfohlen

## NOTEBOOKS IM UNTERRICHT

Students are Better Off without a Laptop in the Classroom

<https://www.scientificamerican.com/article/students-are-better-off-without-a-laptop-in-the-classroom/>

Bleibt natürlich jedem selbst überlassen...

## KURSVORGABEN

- Wichtige Regelungen in den Kursvorgaben (bitte lesen)
- Sollten Änderungen nötig sein, werden die Kursvorgaben angepasst
- Das wird aber vorgängig in den Lektionen diskutiert

Bitte erkundigen Sie sich bei den anderen Kursteilnehmern, wenn Sie einmal nicht anwesend sein können

# PRÜFUNGSRELEVANTER STOFF

Das ist nicht immer ganz klar abgrenzbar. Wenn Sie den Stoff im Wesentlichen verstanden haben, sollte eine genügende Note erreichbar sein, wenn Sie die Themen gut beherrschen eine gute Note, und für eine 6 ist die Erwartung, dass Sie sich über den vorgegebenen Stoff hinaus mit den Themen auseinandersetzen.

Der behandelte Stoff umfasst:

- Die geforderten Vorkenntnisse
- Die behandelten Themen
  - Material im Moodle
  - Erklärungen dazu in den Lektionen
  - Lösen der Praktikumsaufgaben
  - Aufgaben fürs Selbststudium

# HINWEIS

Zum Einstieg in die WBE-Themen sind in den folgenden Abschnitten einige Eckpunkte der Web-Grundlagen noch einmal zusammengestellt

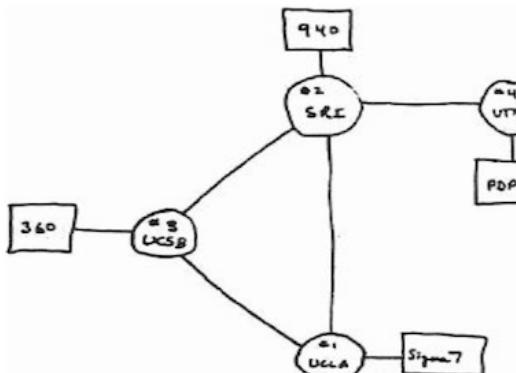
Diese sind Gegenstand des Vorbereitungskurses (Moodle), wo Sie weiteres Material zu diesen Themen finden

# ÜBERSICHT

- Worum geht's in WBE?
- Organisatorisches
- Internet und WWW
- Client-Server-Architektur
- Die Sprachen des Web: HTML, CSS, JavaScript
- Web-Standards und APIs

# INTERNET

- Weltweites Netzwerk, bestehend aus vielen Rechnernetzwerken
- Ursprünglich: ARPANET (1969: vier Knoten)
- Als Internet ab 1987 bezeichnet (ca. 27 000 Knoten)



THE ARPA NETWORK

DEC 1969

4 NODES

Dienste im Internet:  
E-Mail, Usenet, S/FTP, World Wide Web,...

# WORLD WIDE WEB

- Service, der auf dem Internet aufbaut
- Tim Berners-Lee  
(Anfang der 90er Jahre am CERN in Genf auf NeXT Computer)



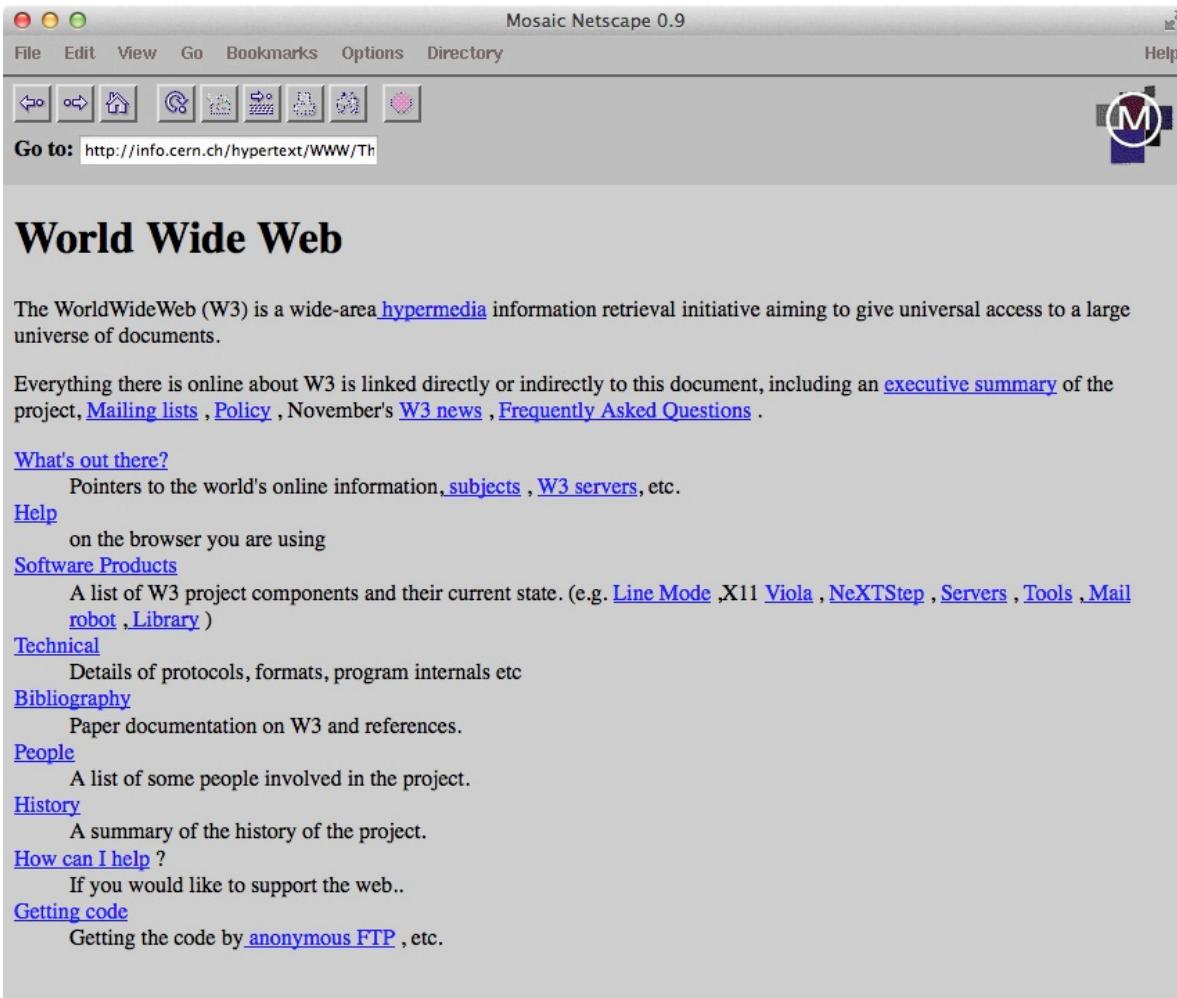
Tim's erster Webserver

Ankündigung des WWW im Usenet

## Speaker notes

Das World Wide Web (WWW) wird heute meist einfach als das *Web* bezeichnet.

# WEB-GESCHICHTE 1992



The screenshot shows the Mosaic Netscape 0.9 browser window. The title bar reads "Mosaic Netscape 0.9". The menu bar includes "File", "Edit", "View", "Go", "Bookmarks", "Options", "Directory", and "Help". Below the menu is a toolbar with various icons. The address bar says "Go to: http://info.cern.ch/hypertext/WWW/Th". The main content area displays the text "World Wide Web" and a paragraph about the W3 project, followed by several links: "What's out there?", "Help", "Software Products", "Technical", "Bibliography", "People", "History", "How can I help?", and "Getting code".

**World Wide Web**

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

[What's out there?](#)  
Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

[Help](#)  
on the browser you are using

[Software Products](#)  
A list of W3 project components and their current state. (e.g. [Line Mode](#) ,[X11 Viola](#) ,[NeXTStep](#) ,[Servers](#) ,[Tools](#) ,[Mail robot](#) ,[Library](#) )

[Technical](#)  
Details of protocols, formats, program internals etc

[Bibliography](#)  
Paper documentation on W3 and references.

[People](#)  
A list of some people involved in the project.

[History](#)  
A summary of the history of the project.

[How can I help ?](#)  
If you would like to support the web..

[Getting code](#)  
Getting the code by [anonymous FTP](#) , etc.

The birth of the Web  
(CERN)

Ankündigung NCSA  
Mosaic

# WEB-GESCHICHTE 1996...



## Speaker notes

# WAYBACKMACHINE

Alte Websites ansehen ist möglich mit der WayBackMachine:

<https://archive.org>

Altavista 1996

The screenshot shows the Altavista homepage from 1996. At the top, there's a navigation bar with icons for back, forward, search, and other functions. The URL in the address bar is <http://www.altavista.com/>. The main header features the Altavista logo with the tagline "Technology, Inc." and "View Multimedia From Our Vantage Point". Below the header is a search bar containing the text "AutoByTel". A link below the search bar says "Click here for advertising information - reach millions every month!". The search form includes dropdown menus for "Search the Web" and "Display the Results", and a "Submit" button. Below the search form is a link to "Search with Digital's Alta Vista". There are two buttons at the bottom left: one for "Contests" with the text "Make Me Laugh..." and another for "Creative Web" with the text "Create a Site...". At the bottom center, there's a link to "Download free demo versions of AltaVista Technology software". The footer contains the Altavista logo again and a horizontal line followed by the text "[Creative][Search][Humor][Email]".

http://www.twi.ch/

**TWI**  
TECHNIKUM WINTERTHUR  
INGENIEURSCHULE DES KANTONS ZÜRICH

-  [Allgemeine Informationen und Aktualitäten](#)
-  [Abteilungen / Diplomstudien](#)
-  [Nachdiplomstudien / Weiterbildung](#)
-  [Forschung und Entwicklung / Dienstleistungen](#)
-  [Zentrale Dienste \(Mensa ... Informatik\)](#)
-  [Personen und schulinterne Organisationen](#)
-  [Partnerorganisationen](#)
-  [WWW-Werkzeuge und Startseiten](#)

---

last modified: Sunday September 21 1997 14:30 MET DST by [TWI Webmaster](#)

# Netscape 1997

<https://web.archive.org/web/19970220012313/http://home.netscape.com:80/>

The screenshot shows the Netscape homepage from July 9, 1997. At the top, there's a banner with the text "WELCOME TO NETSCAPE" and "YOUR GUIDE TO NAVIGATING THE NET". Below the banner is a navigation menu with links: GUIDE TO INTERNET, COMPANY & PRODUCTS, ENTERPRISE SOLUTIONS, DEVELOPERS, NETSCAPE STORE, ASSISTANCE, and ONE STOP SOFTWARE. A large headline "200 LEADING COMPANIES CHOOSE NETSCAPE" is prominently displayed. On the left, a news item dated JULY 9, 1997, discusses Netscape surpassing 2 million seats. A list of news items follows, including the release of Communicator 4.01 and the developer release of Java Foundation Classes. To the right, there's a section titled "GET THE LATEST NETSCAPE SOFTWARE" with links to Tune Up to Communicator, Get Any Netscape Software, and Netscape Now for Subscribers Only. Below that is a "NETSCAPE SPONSORS" section with a placeholder box. Further down, a "NETSCAPE STORE SPECIAL" section offers savings on FastTrack and LiveWire Servers, and Communicator 4.01. At the bottom, there are sections for "FOCUS ON SOLUTIONS" (In-Box Direct, ISP Select, Product Update), "IN THE ENTERPRISE" (Marc Andreessen's vision), and links to "Visit a Netscape International Site", "Today's Tip", "PowerStart", "Feedback", and "Net Search". The footer includes links for Netscape Home, Download Software, Customer Service, Technical Support, Search & Contents, and Web Site Advertising, along with copyright information and server details.

WELCOME TO  
NETSCAPE  
YOUR GUIDE TO NAVIGATING THE NET

GUIDE TO INTERNET COMPANY & PRODUCTS ENTERPRISE SOLUTIONS DEVELOPERS NETSCAPE STORE ASSISTANCE ONE STOP SOFTWARE

**200 LEADING COMPANIES CHOOSE NETSCAPE**

**JULY 9, 1997**

**N**etscape surpasses a key milestone of [200 design wins](#) with Netscape Communicator and SuiteSpot 3.0 software - now reaching more than 2 million seats.

- Download Netscape [Communicator 4.01](#) for Macintosh and Windows. Communicator Preview Release 6 is now [available](#) for Unix.
- To simplify the development of Internet and intranet applications, Netscape and Sun [announce](#) the developer release of Java Foundation Classes.
- The beta version of Netscape [Messaging Server 3.0](#) for Windows NT is now [available](#).
- Get the *New York Times*, SportsLine, and HotWired delivered daily via [Netscape In-Box Direct](#).
- [More news ...](#)

**HEADLINES FROM ABCNEWS.COM**

**ABC NEWS** Dover Roams Mars

**NETSCAPE COLUMNS**

In this week's [The Main Thing](#) column, Jim Barksdale talks about team collaboration. Read all the [Netscape Columns](#) to keep up with Internet issues and trends.

**FOCUS ON SOLUTIONS**

**IN-BOX DIRECT** Subscribe to [Netscape In-Box Direct](#) and receive free content from NewsHound, E! Online, and National Geographic.

**ISP SELECT** Get 30 days of [free Internet access](#) when you sign up with a participating Local Bell Internet Service Provider through Netscape ISP Select.

**PRODUCT UPDATE** Netscape [SuiteSpot 3.0](#) delivers an integrated suite of server software for web, messaging, and crossware applications.

**IN THE ENTERPRISE** Marc Andreessen outlines Netscape's future vision and new software designed to make the [Networked Enterprise](#) a reality.

Visit a Netscape International Site Today's Tip | PowerStart | Feedback | Net Search

NETSCAPE HOME DOWNLOAD SOFTWARE CUSTOMER SERVICE TECHNICAL SUPPORT SEARCH & CONTENTS WEB SITE ADVERTISING

Corporate Sales: 415/937-2555; Personal Sales: 415/937-3777; Government Sales: 415/937-3678  
If you have any questions, please visit [Customer Service](#), or contact your nearest [sales office](#).

Copyright © 1997 Netscape Communications Corporation  
This site powered by [Netscape SuiteSpot](#) servers.

## WEB WIE FRÜHER...

Wer das Web wie früher erleben möchte, kann <https://theoldnet.com/docs/httpproxy/index.html> aufrufen. Durch Setzen von *theoldnet.com* als Proxy mit der Jahreszahl als Port (1996 - 2012) kann man im Web des entsprechenden Jahres surfen. Wozu ist das gut? Auch hier gilt: Je besser man die Vergangenheit kennt, desto besser wird man die Zukunft meistern.

# WEB HEUTE

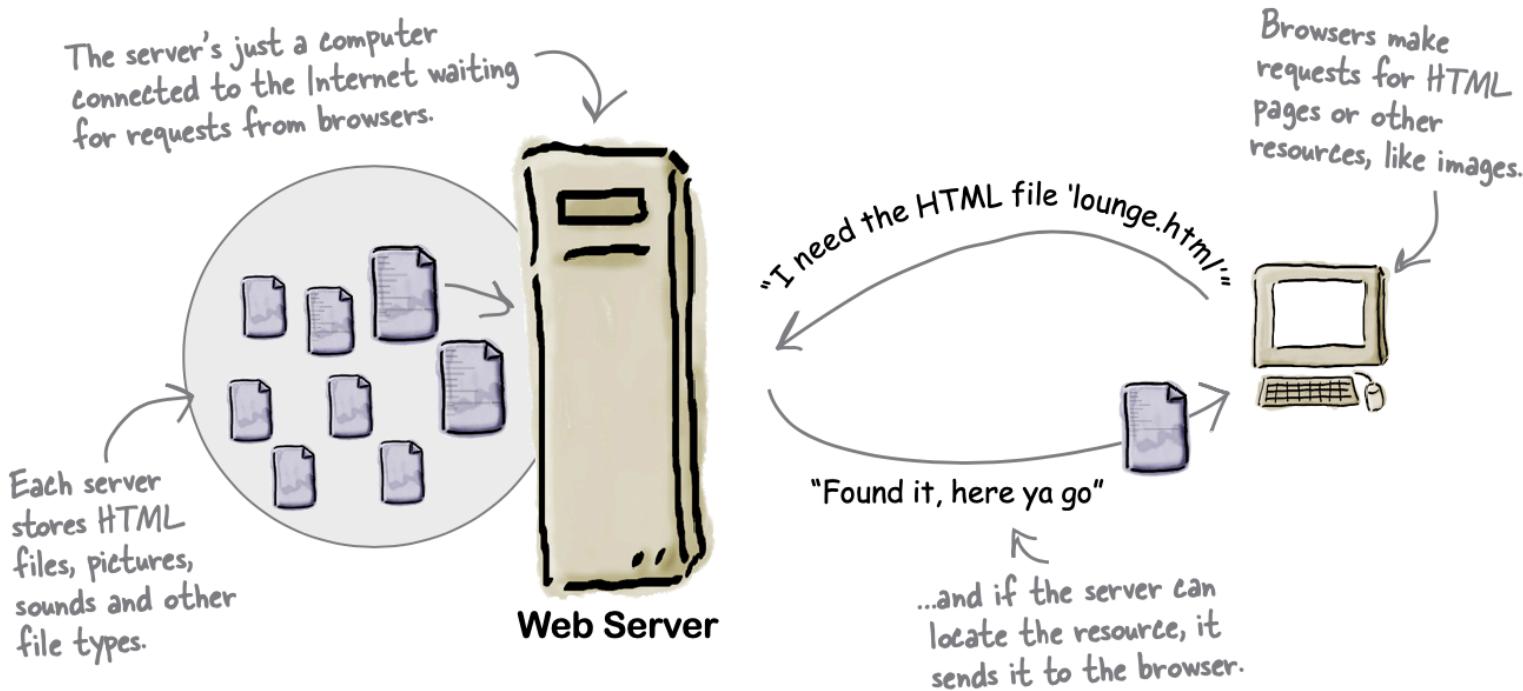
- Wichtige Applikations- und Informationsplattform
- Unzählige Technologien und Spezifikationen
- Vieles davon rund um „HTML Living Standard“
- Zahlreiche weitere APIs und Standards
- Grundlagen: HTML + CSS + JavaScript

# ÜBERSICHT

- Worum geht's in WBE?
- Organisatorisches
- Internet und WWW
- Client-Server-Architektur
- Die Sprachen des Web: HTML, CSS, JavaScript
- Web-Standards und APIs

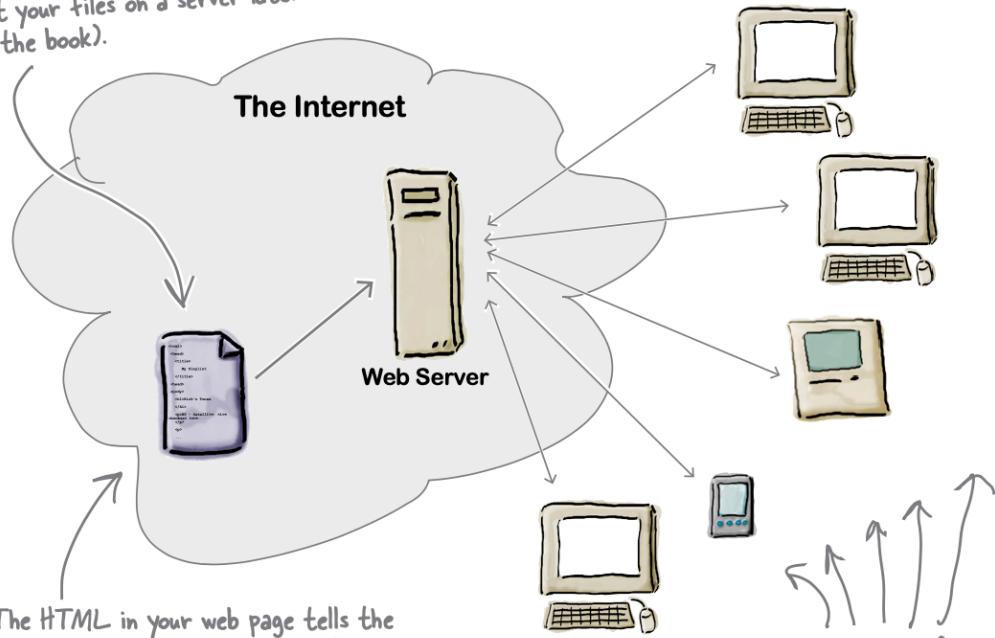
# WEB-ARCHITEKTUR

- Client-Server
- Interaktion: Request/Response



## Speaker notes

To make web pages, you create files written in the HyperText Markup Language (HTML for short) and place them on a web server (we'll talk about how to get your files on a server later in the book).



Once you've put your files on a web server, any browser can retrieve your web pages over the Internet.

The HTML in your web page tells the browser what it needs to know to display your page. And, if you've done your job well, your pages will even display well on cell phones and mobile devices, and work with speech browsers and screen magnifiers for the visually impaired.

And there are a lot of PCs and devices connected to the Internet all running web browsers. More importantly, there are friends, family, fans, and potential customers using those devices!

# CLIENT: BROWSER / USER AGENT



Firefox



Safari



Chromium



Microsoft Edge



Firefox Developer Edition



Safari Technology Preview



SeaMonkey



Vivaldi



<https://gs.statcounter.com>

## Speaker notes

Solche Angaben sind immer mit Vorsicht zu geniessen. Eigentlich müsste genauer geschaut werden, wie die Daten erhoben wurden. Wenn zum Beispiel die Zugriffe auf eine bestimmte Menge von Websites ausgewertet werden, wird das Ergebnis durch Art und Inhalte der ausgewählten Websites beeinflusst.

Sehr stark hängen die Anteile auch von der ausgewerteten Region ab. Weltweit hat Chrome zum Beispiel einen höheren Marktanteil als in der Schweiz.

Auf Mobilgeräte beschränkt hat in der Schweiz aufgrund der starken Stellung von iOS der Safari einen guten Marktanteil:



# USER AGENT: LYNX

The screenshot shows a terminal window titled "Schreibtisch — lynx — 91x24" running the Lynx text-based web browser. The title bar also displays "LYNX – The Text Web-Browser (p1 of 2)". The content area shows the homepage of <http://lynx.invisible-island.net/>. It includes copyright information from 1997-2017, 2018 by Thomas E. Dickey, and a menu with links to the home page, current development, stable release, and resources. A link to "LYNX – The Text Web-Browser" is present. The text "Lynx is the text web browser." is displayed. Below it, the message "This is the toplevel page for the Lynx software distribution site." is shown. Further down, it says "The current development sources have the latest version of Lynx available (development towards 2.9.0)." and "The main help page for lynx-current is online; the current User Guide is part of". At the bottom, there is a blue status bar with the text "-- Leertaste für nächste Seite --", keyboard instructions ("Pfeile: Auf/Ab: andere Seite im Text. Rechts: Verweis folgen; Links: zurück. H)ilfe O)ptionen P) Druck G)ehe zu M) Hauptseite Q) Beenden /=Suche <=History"), and a small grey vertical bar.

## Speaker notes

Bei der Bezeichnung *Browser* denkt man zuerst einmal an einen grafischen Browser wie Chrome oder Firefox. Das World Wide Web Consortium (W3C) bevorzugt statt *Browser* die Bezeichnung *User Agent*, um klarzustellen, dass es ganz unterschiedliche Arten von Clients gibt, mit denen man auf das Web zugreifen kann.

Hier einige Beispiele:

- Grafischer Browser auf Notebook (div. Bildschirmauflösungen,-größen)
- Grafischer Browser auf Smartphone (div. Bildschirmauflösungen,-größen)
- Grafischer Browser auf Tablet
- Einfacher Browser auf Feature Phone
- Sprachausgabe oder tastbare Braille-Ausgabe
- Browser auf Fernsehgerät
- Ausgedruckte Webseite
- Text-only-Browser
- Programm, das Webinhalte lädt und verarbeitet, z.B. Suchmaschine

Es kann durchaus nichts schaden, Ihre eigenen Seiten einmal mit dem *Lynx* anzusehen. Das Ergebnis ist etwa das, was auch Google auf Ihren Seiten „sieht“.

<https://lynx.invisible-island.net>

# BROWSER WARS

- Zunächst: Mosaic
- Später: Netscape
- Dann: Internet Explorer
- Heute: Chrome

Problematisch waren immer die Phasen, in denen ein Browser deutlich marktbeherrschend war

„This page is best viewed with Internet Explorer 6 on Windows.“

## Speaker notes

Wann immer ein Browser deutlich marktbeherrschend war, haben dessen Entwickler gerne neue Features in den Browser eingebaut, ohne Rücksicht auf deren Standardisierung zu nehmen. Dadurch entsteht leicht einen Tendenz weg vom offenen Web hin zu einem proprietären Client-Server-System.

Aktuell erleben wir wieder so eine Phase: Das W3C ist weitgehend entmachtet und nur noch eingeschränkt involviert in die weitere Entwicklung der Webstandards. Zudem dominiert der Browser Chrome die Landschaft der User Agents. Viele alternativen Browser bauen zudem auf der Rendering Engine Blink des Chromium-Projekts auf, auf dem auch der Chrome basiert.

Als Web-Entwickler sollten wir uns für eine möglichst grosse Browser-Vielfalt und eine gemeinsame und überlegte Weiterentwicklung der Webstandards einsetzen. Nachdem Microsoft für der Browser Edge die eigene Rendering Engine aufgegeben hat und ebenfalls auf Blink setzt, gibt es nur noch wenige Alternativen:

- Firefox: verschiedene Plattformen
- Safari: Apple-Plattformen
- Alle Browser unter iOS, da Apple vorgibt, dass nur Webkit als Rendering Engine unter iOS zulässig ist (auch nicht gerade im Sinne eines offenen Web und gerät immer mehr unter Druck)

# WEBSERVER

- Reagieren auf Anfragen von User Agents
- Senden diesen die angefragten Dateien
- Beispiele
  - Apache HTTP Server
  - Nginx (ausgesprochen „engine-ex“)
  - Microsoft Internet Information Services (IIS)
  - Apache Tomcat (Java)

# WEB-PLATTFORM

## Drei Grundpfeiler

- Sprachen: Rolle von HTML, CSS und JavaScript
- Adressierung: URI/URL, IP-Adresse, Domain-Name
- Protokoll: HTTP

Gleich mehr zu den Sprachen. Schauen wir zuerst kurz auf Adressierung und Protokoll...

# ADRESSIERUNG

## Uniform Resource Locator, URL

"http://" [user[:password]@] host [:port] path [ "?" query ] [ "#" fragment ]

scheme-specific-part → → → |  
|  
**http://hans:geheim@example.org:80/demo/example.cgi?land=de&stadt=aa#geschichte**  
| | | | | | | |  
| | | host | url-path | query | fragment  
| | password | port |  
| user |  
scheme (hier gleich Netzwerkprotokoll)

(Quelle: Wikipedia)

## Speaker notes

Uniform Resource Locators (URL) (RFC 1738)

<https://www.ietf.org/rfc/rfc1738.txt>

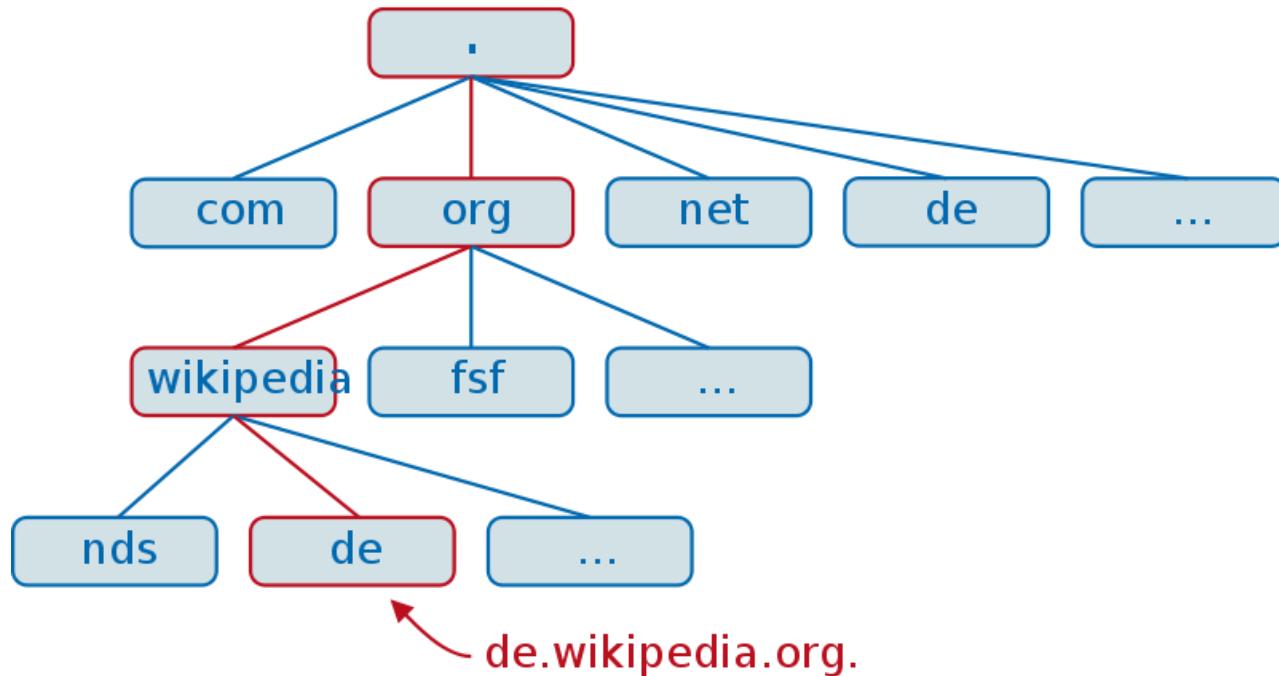
# IP-ADRESSE

- Host-Angabe in URL: IP-Adresse oder Domain-Name
- IPv4: 32 Bit, IPv6: 128 Bit
- Beispiele:

173.194.40.95

243f:6a88:85a3:08d3:1319:8a2e:0370:7344

# DOMAIN NAME SYSTEM (DNS)



IP-Adressen schwer  
zu merken, Abhilfe:  
Domain-Namen

# PROTOKOLL: HTTP

- Einfaches, textbasiertes Protokoll
- Beispiel: Anfrage an Server

```
GET /~bkrt/hallo.html HTTP/1.1
```

Host: dublin.zhaw.ch

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X...) Gecko/20100101 Firefox/22.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: de-de,de;q=0.8,en-us;q=0.5,en;q=0.3

Accept-Encoding: gzip, deflate

Connection: keep-alive

## Speaker notes

### Antwort vom Server

```
HTTP/1.1 200 OK
Date: Mon, 15 Jul 2013 17:10:56 GMT
Server: Apache/2.2.15 (CentOS)
Last-Modified: Wed, 17 Oct 2012 08:10:22 GMT
ETag: "5b018a-af-4cc3ccd575780"
Accept-Ranges: bytes
Content-Length: 175
Connection: close
Content-Type: text/html; charset=UTF-8
```

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Hallo</title>
  </head>
  <body>
    <h1>Hallo</h1>
    <p>Ich bin eine Webseite</p>
  </body>
</html>
```

Hypertext Transfer Protocol - HTTP/1.1 (RFC 2616)

<https://www.ietf.org/rfc/rfc2616.txt>

# ÜBERSICHT

- Worum geht's in WBE?
- Organisatorisches
- Internet und WWW
- Client-Server-Architektur
- Die Sprachen des Web: HTML, CSS, JavaScript
- Web-Standards und APIs

# DIE SPRACHEN DES WEB

- HTML und CSS bereits bekannt  
(Vorkenntnisse, Vorkurs)
- Daher hier nur eine schnelle Zusammenfassung...

# ÜBERSICHT

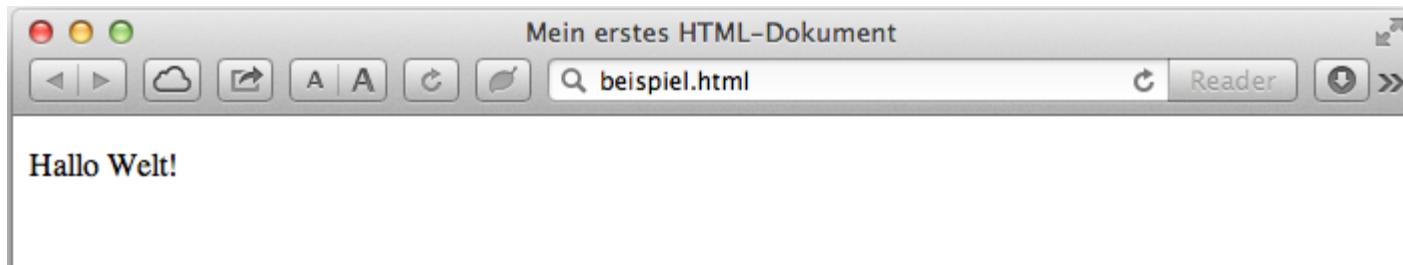
- Internet und WWW
- Client-Server-Architektur
- Die Sprachen des Web: HTML, CSS, JavaScript
  - Markup mit HTML
  - Darstellung mit CSS
  - Verhalten mit JavaScript
- Web-Standards und APIs

# MARKUP MIT HTML

- **Markup Language:** Tags (`<p>`)
- **Hypertext:** Links
- Multimedia: eingebettete Multimedia-Elemente
- Gestaltung: verbundene CSS-Datei(en)
- Verhalten: verbundene JavaScript-Datei(en)
- Verschiedene Content-Types: `text/html`, `image/jpeg`, ...

# MARKUP MIT HTML

```
1 <!DOCTYPE html>
2 <html lang="de">
3   <head>
4     <title>Mein erstes HTML-Dokument</title>
5     <meta charset="utf-8" />
6   </head>
7
8   <body>
9     <p>Hallo Welt!</p>
10  </body>
11 </html>
```



# MARKUP MIT HTML

- Es geht um **Inhalt** und **Struktur** von Dokumenten
- Was bedeuten die einzelnen Teile (Überschrift? Absatz?)
- Darstellung und Verhalten explizit ausgeklammert
- Ohne CSS-Stylesheet: Standard-Darstellung im Browser

## Speaker notes

Eine grosse Diskussion derzeit unter Webentwicklern ist, ob es eine gute Idee ist, Tools wie **Tailwind CSS** einzusetzen. Kurz: Tailwind CSS stellt eine grosse Zahl von CSS-Klassen zur Verfügung, um bestimmte Darstellungen zu erreichen, z.B.:

```
<figure class="md:flex bg-slate-100 rounded-xl p-8 md:p-0 dark:bg-slate-800">
  ...
</figure>
```

Damit nicht bei jedem Laden die gesamte Liste der CSS-Regeln geladen werden muss, kann die CSS-Datei in einem Build-Schritt auf die tatsächlich verwendeten Klassen reduziert werden.

Viele Entwickler sehen diesen Ansatz als Rückschritt um ein paar Jahrzehnte. Damals hat man Darstellung und Inhalt/Struktur ebenfalls vermischt – es gab Attribute wie font – oder man hat die Styles inline in den HTML-Code übernommen:

```
<figure style="display:flex, border-radius:5px, ...">
  ...
</figure>
```

Im Gegensatz zu diesem Beispiel bezieht Tailwind die Display-Grösse mit ein, was sicher ein Vorteil ist. Eigentlich sollte es aber beispielsweise so sein:

```
<figure class="employee-image">
  ...
</figure>
```

Damit hat man den Inhalt beschrieben, die Darstellung sollte an anderer, zentraler Stelle beschrieben werden. Vorteile des Tailwind-Ansatzes: man hat schnell eine Seite mit guter Darstellung zusammengeschustert, muss sich kaum mit CSS auseinandersetzen, und kann Code, der sich bewährt hat, einfach per Copy&Paste übertragen. Nachteil dieses Ansatzes: Trennung von Inhalt und Darstellung gehen verloren (wie wenn man in Word die Darstellungsoptionen für jeden Textbereich angibt statt Styles wie Titel, Text oder Liste zu verwenden) und man hat unnötig viel Redundanz im Code.

Tailwind, and the death of web craftsmanship

<https://pdx.su/blog/2023-07-26-tailwind-and-the-death-of-craftsmanship/>

# HTML: NOCH EIN BEISPIEL

```
1 <!DOCTYPE html>
2 <html lang="de">
3   <head> ... </head>
4   <body>
5     <section>
6       <header>
7         
8       </header>
9       <article>
10        <h1>Webseiten</h1>
11        <p>Die Grundidee ist ziemlich einfach:</p>
12        <p>Das <strong>World Wide Web</strong> (heute meist kurz: das Web)
13          basiert auf einer <em>Client-Server-Architektur</em>. Der Client
14          ist meist ein grafischer <em>Browser</em>. ...</p>
15        </article>
16      </section>
17    </body>
18 </html>
```

## Speaker notes

Beispiel komplett:

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <title>Mein zweites HTML-Dokument</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="styles/screen.css">
  </head>

  <body>
    <section>
      <header>
        
      </header>

      <article>
        <h1>Webseiten</h1>

        <p>Die Grundidee ist ziemlich einfach:</p>

        <p>Das <strong>World Wide Web</strong> (heute meist kurz: das Web)
          basiert auf einer <em>Client-Server-Architektur</em>. Der Client
          ist meist ein grafischer <em>Browser</em>. ....</p>
      </article>
    </section>
  </body>
</html>
```

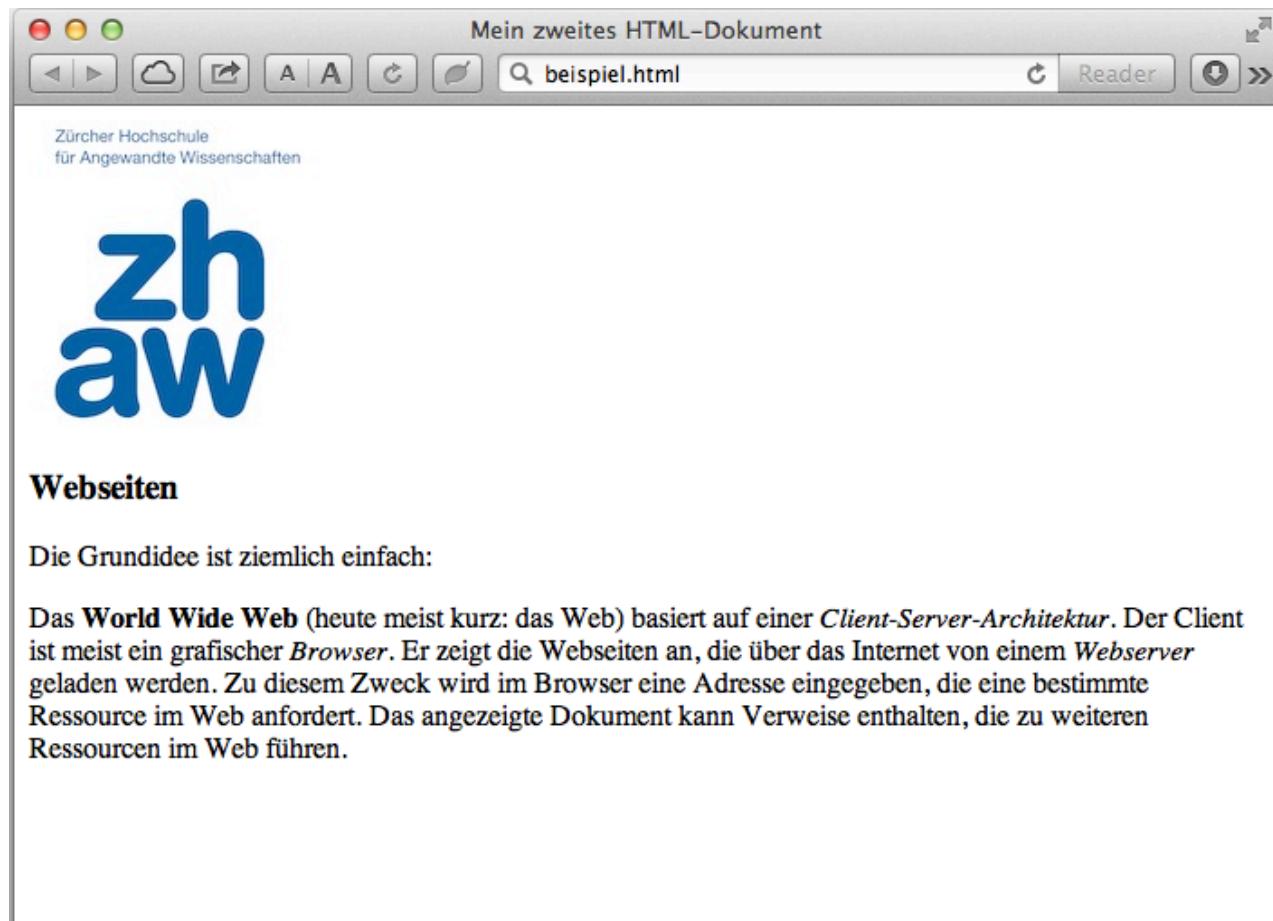
HTML ist eine Auszeichnungssprache (Markup Language). Für die Auszeichnung (das *Markup*) der Inhalte werden *Tags* verwendet, etwa <h1>...</h1> für Überschriften oder <p>...</p> für Absätze. Die meisten HTML-Elemente haben ein *Start-Tag* und ein *End-Tag* und umklammern damit den Inhalt, der weitere Tags enthalten kann. Dadurch entsteht eine hierarchische Struktur der Dokuments. *Leere Elemente* haben keinen Inhalt, zum Beispiel <meta ...>, <img ...>.

Das *Start-Tag* (oder im Fall eines leeren Elements das *Tag*) kann auch Attribute haben, welche das Element näher beschreiben. Die korrekte Schreibweise ist der Name des Attributs, ein =-Zeichen ohne Leerzeichen davor oder danach, und schliesslich der Attributwert in Anführungszeichen. Nach einem Leerzeichen können weitere Attribute folgen.

```

```

# STANDARD-DARSTELLUNG (OHNE CSS)

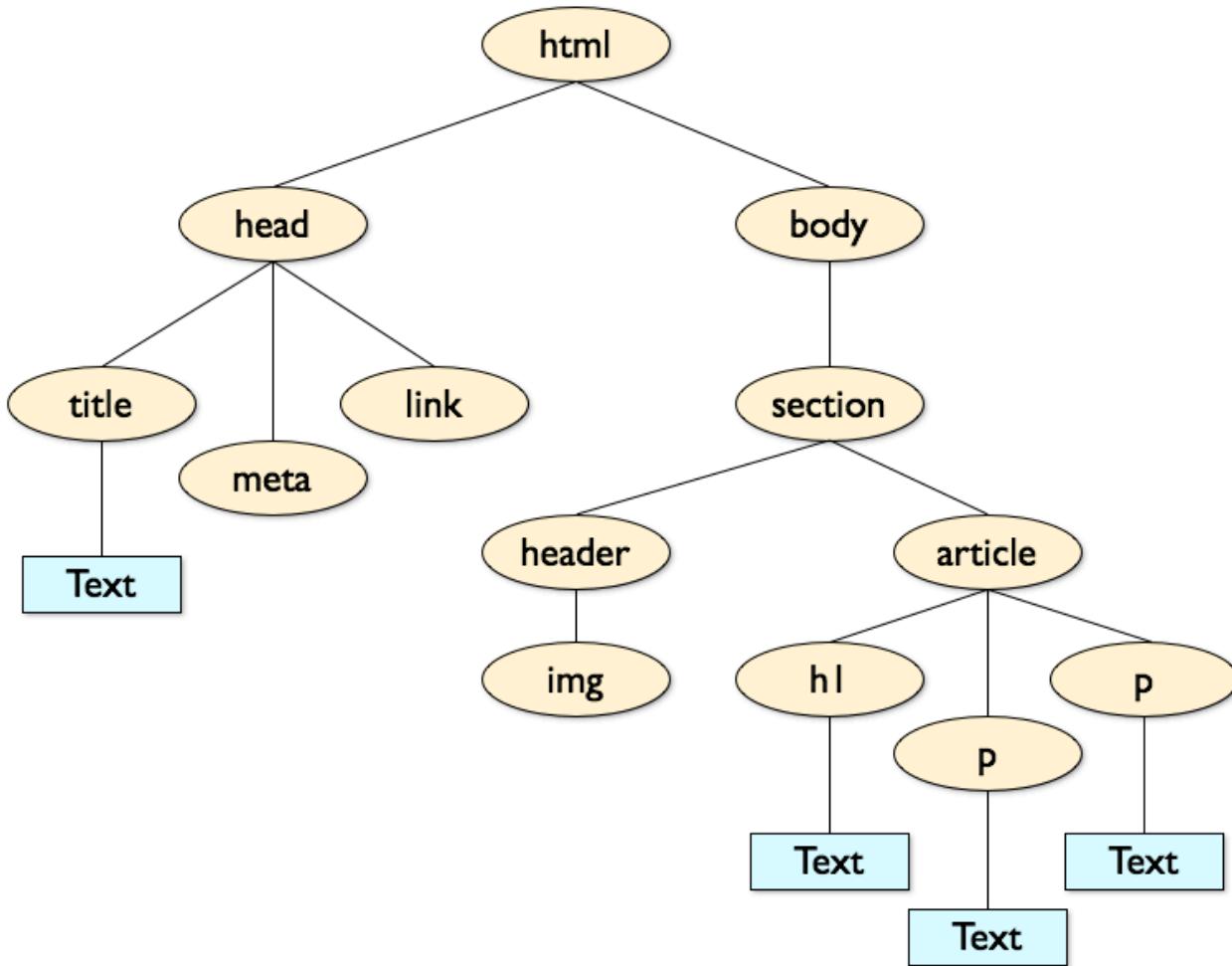


## Webseiten

Die Grundidee ist ziemlich einfach:

Das **World Wide Web** (heute meist kurz: das Web) basiert auf einer *Client-Server-Architektur*. Der Client ist meist ein grafischer *Browser*. Er zeigt die Webseiten an, die über das Internet von einem *Webserver* geladen werden. Zu diesem Zweck wird im Browser eine Adresse eingegeben, die eine bestimmte Ressource im Web anfordert. Das angezeigte Dokument kann Verweise enthalten, die zu weiteren Ressourcen im Web führen.

# HIERARCHISCHE STRUKTUR



## Speaker notes

Bestimmte Entwickler-Tools im Browser können diese Hierarchie auch explizit im Browser hervorheben, wie das folgende Bild zeigt.

```
<!DOCTYPE html>
<HTML lang="de">

<head>
<title>
Mein zweites HTML-Dokument
</title>
<meta charset="utf-8">
<link rel="stylesheet" href="styles/screen.css">
</head>

<body>
<section>
<header>

</header>

<article>
<h1>
Webseiten
</h1>

<p>
Die Grundidee ist ziemlich einfach:
</p>

<p>
...
</p>
</article>
</section>
</body>
</HTML>
```

# ÜBERSICHT

- Internet und WWW
- Client-Server-Architektur
- Die Sprachen des Web: HTML, CSS, JavaScript
  - Markup mit HTML
  - Darstellung mit CSS
  - Verhalten mit JavaScript
- Web-Standards und APIs

# CASCADING STYLE SHEETS (CSS)

```
1 @charset "utf-8";
2
3 body {
4     background-color: #999;
5 }
6 article {
7     margin-left: 200px;
8     font-family: 'Georgia Pro', georgia, serif;
9     line-height: 140%;
10 }
11 p {
12     font-size: 3em;
13     color: blue;
14 }
15 ...
```

# DARSTELLUNG OHNE/MIT CSS

The screenshot displays a web browser window with two tabs. The left tab, titled "Mein zweites", shows the Zürcher Hochschule für Angewandte Wissenschaften (zhaw) website. It features the zhaw logo in blue, the university's name in black, and a section titled "Webseiten" with the text: "Die Grundidee ist ziemlich einfach: Das World Wide Web (heute meist kurz: das Web) ist meist ein grafischer Browser. Er zeigt die Webseiten geladen werden. Zu diesem Zweck wird im Browser eine Adresse eingegeben, die eine bestimmte Ressource im Web anfordert. Das angezeigte Dokument kann Verweise enthalten, die zu weiteren Ressourcen im Web führen." The right tab, titled "Mein zweites HTML-Dokument", shows a simplified version of the same content, missing the logo and some styling.

Zürcher Hochschule  
für Angewandte Wissenschaften

**zhaw**

**Webseiten**

Die Grundidee ist ziemlich einfach:

Das **World Wide Web** (heute meist kurz: das Web) ist meist ein grafischer *Browser*. Er zeigt die Webseiten geladen werden. Zu diesem Zweck wird im Browser eine Adresse eingegeben, die eine bestimmte Ressource im Web anfordert. Das angezeigte Dokument kann Verweise enthalten, die zu weiteren Ressourcen im Web führen.

Zürcher Hochschule  
für Angewandte Wissenschaften

**Webseiten**

Die Grundidee ist ziemlich einfach:

Das **World Wide Web** (heute meist kurz: das Web) basiert auf einer *Client-Server-Architektur*. Der Client ist meist ein grafischer *Browser*. Er zeigt die Webseiten an, die über das Internet von einem *Webserver* geladen werden. Zu diesem Zweck wird im Browser eine Adresse eingegeben, die eine bestimmte Ressource im Web anfordert. Das angezeigte Dokument kann Verweise enthalten, die zu weiteren Ressourcen im Web führen.

# ÜBERSICHT

- Internet und WWW
- Client-Server-Architektur
- Die Sprachen des Web: HTML, CSS, JavaScript
  - Markup mit HTML
  - Darstellung mit CSS
  - **Verhalten mit JavaScript**
- Web-Standards und APIs

# VERHALTEN MIT JAVASCRIPT

```
1 let pElem = document.querySelector('p')
2 pElem.addEventListener('click', function () {
3     let newText = document.createTextNode("removed!")
4     pElem.replaceChild(newText, pElem.childNodes[0])
5 })
```

- Klick auf den Text ersetzt ihn durch "removed!"
- Element des Dokuments auswählen
- Reaktion auf Ereignis festlegen

# JAVASCRIPT

- Veröffentlich 1995 in Vorversion des Netscape Navigators 2.0
- Unter Zeitdruck entwickelt von Brendan Eich
- Ziel: Scripts um Webseiten dynamischer zu machen
- Zunächst: **LiveScript**, dann **JavaScript** (Marketing)
- **JavaScript und Java haben wenig gemeinsam (!)**

# JAVASCRIPT HEUTE

- Web-Apps mit vielen tausend Codezeilen (z.B. Maps)
- Einsatz in nativen Applikationen (z.B. VSCode)
- Performant: Just-in-time Compiler (JIT)
- Zielsprache für andere Sprachen (z.B. TypeScript)

JavaScript ist heute eine der wichtigsten und meistverwendeten Programmiersprachen

# JAVASCRIPT PLATTFORMEN

- **Web-Browser** mit verschiedenen JavaScript-Engines und diversen APIs für die Browser-Plattform (DOM, Storage, ...)
- **Node.js** als browserunabhängige JavaScript-Plattform, ebenfalls mit speziellen APIs (Filesystem, Web-Server, ...)
- Alternativen zu Node.js wie **Deno** oder **Bun**

# JAVASCRIPT-ALTERNATIVEN

- Sprachen, welche nach JavaScript übersetzt werden, u.a.
  - [TypeScript](#)
  - [ReScript](#)
  - [ClojureScript](#)
  - neueres JavaScript zu älterem JavaScript
- Transpiler (z.B. [Babel](#))

# WEB-ASSEMBLY

- Neuere Entwicklung
- Virtuelle Maschine
- Diverse Sprachen zu Web-Assembly kompilierbar

## Speaker notes

Damit auch möglich, Sprachen wie Rust oder Kotlin im Web einzusetzen.

Dabei soll aber nicht verschwiegen werden, dass JavaScript und das Web gemeinsam gewachsen und somit eng verzahnt sind. Das gilt vor allem für das DOM, das Document Object Model. DOM-Manipulationen wird man auch in Zukunft eher in JavaScript oder einer eng verwandten Sprache machen.

# ÜBERSICHT

- Worum geht's in WBE?
- Organisatorisches
- Internet und WWW
- Client-Server-Architektur
- Die Sprachen des Web: HTML, CSS, JavaScript
- Web-Standards und APIs

# WORLD WIDE WEB CONSORTIUM

- Kurz: W3C
- Standardisiert Technologien des World Wide Web
- Gegründet 1994 am MIT
- Gründer und Vorsitzender: Tim Berners-Lee

Leider schwindet in letzter Zeit der Einfluss des W3C auf die weitere Entwicklung des Web, da die Browser-Hersteller (dominanter Player: Google) mehr und mehr die Richtung vorgeben

# WHATWG

- Web Hypertext Application Technology Working Group
- Gründer: Apple, Mozilla, Opera, später: Microsoft, Google
- Grund: Unzufriedenheit mit W3C-Ausrichtung
- Eigene Standards der WHATWG
- Zeitweise parallel: HTML des W3C, HTML der WHATWG
- Heute: **HTML Living Standard** der WHATWG

<https://html.spec.whatwg.org/multipage/>

# TECHNOLOGIEN

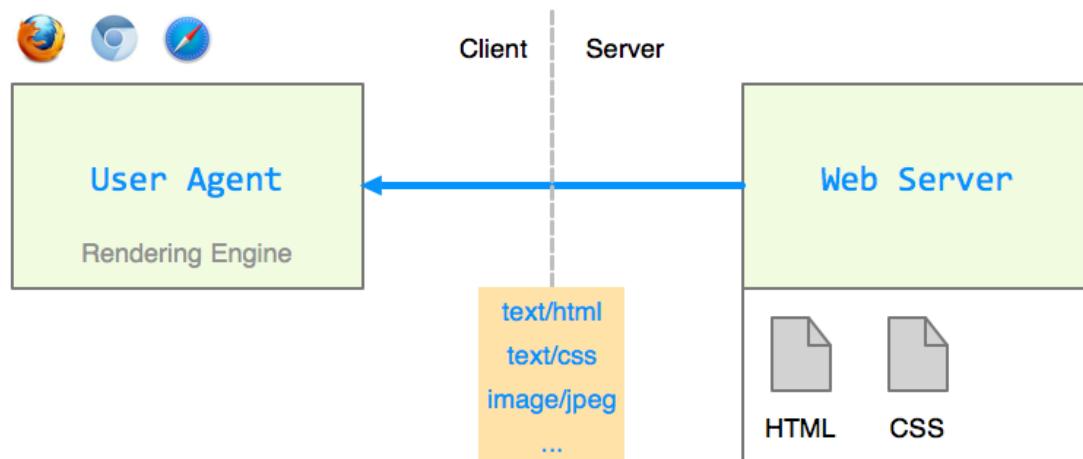
## Client-seitig

- Beschränkt auf das, was der Browser kann
- HTML + CSS + JavaScript + noch ein paar Sachen
- → Front-end Entwickler

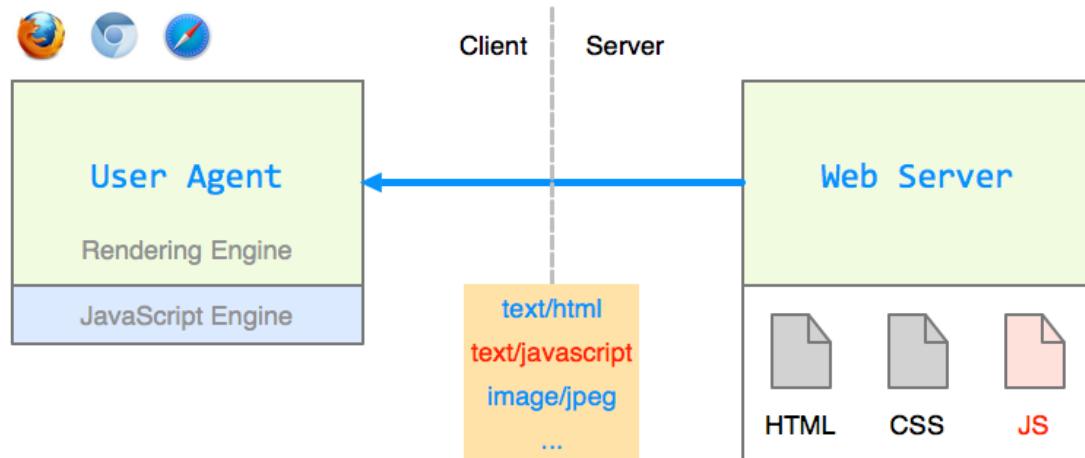
## Server-seitig

- Praktisch unbeschränkt: Plattform, Programmiersprache, ...
- Erzeugt und gesendet wird das, was der Browser kann
- → Back-end Entwickler

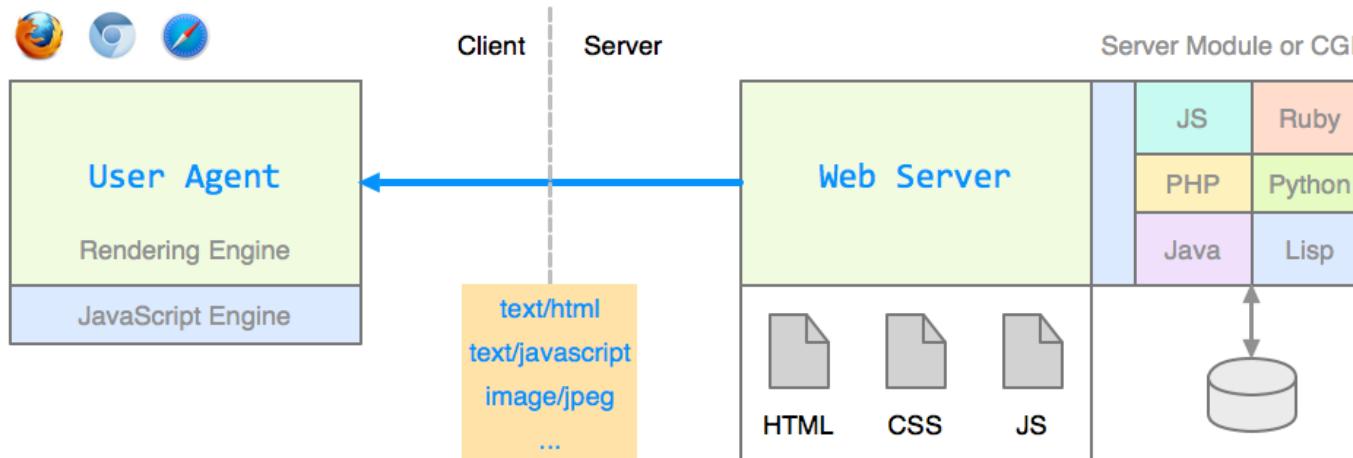
# STATISCHE WEBSEITEN



# CLIENTSEITIGE PROGRAMMLOGIK



# SERVERSEITIGE PROGRAMMLOGIK



# ENTWICKLUNG

- Statische Webseiten
- Generierte Inhalte (CGI z.B. Shell Scripts, Perl)
- Serverseitig eingebettete Scriptsprachen (PHP)
- Client Scripting und Applets (JavaScript, Java Applets, Flash)
- Enterprise Application Server (Java, Java EE)
- MVC Serveranwendungen (Rails, Django)
- JavaScript serverseitig

# UMFANGREICHES GEBIET

The screenshot shows the Can I Use website interface. At the top, there's a navigation bar with links for Home, News, and a date (June 26, 2020). To the right of the navigation is a "Compare browsers" button. The main content area has three columns: "CSS", "HTML5", and "SVG". Each column contains a list of features with a small square icon indicating support. On the far right, there's a sidebar titled "Can I Use" with sections for CSS, HTML5, and SVG, each listing many specific features with their own support status icons.

## CSS

- ::first-letter CSS pseudo-element selector
- ::placeholder CSS pseudo-element
- ::selection CSS pseudo-element
- :dir() CSS pseudo-class
- :has() CSS relational pseudo-class
- :in-range and :out-of-range CSS pseudo-classes
- :is() CSS pseudo-class
- :placeholder-shown CSS pseudo-class
- @font-face Web fonts
- Blending of HTML/SVG elements
- calc() as CSS unit value
- Case-insensitive CSS attribute selectors
- ch (character) unit
- CSS 2.1 selectors
- CSS ::marker pseudo-element
- CSS :any-link selector

## HTML5

- accept attribute for file input
- Add to home screen (A2HS)
- async attribute for external scripts
- Attributes for form submission
- Audio element
- Audio Tracks
- Autofocus attribute
- Canvas (basic support)
- Canvas blend modes
- classList (DOMTokenList)
- Color input type
- contenteditable attribute (basic support)
- Custom Elements (V1)
- Custom protocol handling
- Datalist element
- dataset & data-\* attributes
- Date and time input types

## SVG

- SVG (basic support)
- SVG effects for HTML
- SVG filters
- SVG fonts
- SVG fragment identifiers
- SVG in HTML img element
- SVG SMIL animation
- **All SVG features**

## JS API

- AbortController & AbortSignal
- Accelerometer
- Ambient Light Sensor
- Auxclick
- Base64 encoding and decoding
- Basic console logging functions

## Speaker notes

Die Web-Plattform umfasst eine ausserordentlich grosse Zahl von Technologien, APIs, Frameworks und Bibliotheken. Dazu kommt, dass sich das ganze Gebiet schnell entwickelt.

Hier einen vernünftigen Auschnitt zu definieren und in einem Semester zu vermitteln, ist eine Herausforderung. Wir versuchen unser Bestes ... :)

- <https://html-now.github.io>
- <https://caniuse.com/#index>

# WEB OMNIPRÄSENT

- Web-Seiten oder Web-Apps im Browser
- Auf Desktop, Notebook, oder Smartphone
- Native Mobilapps mit Web-Kern (PhoneGap, Cordova)
- Native Desktop-Apps auf Web-Basis (Electron)
- Apps, welche REST-basierte Webservices nutzen

## Speaker notes

Webtechnologien findet man heute überall. Klar, zunächst einmal Webangeboten, die man im Browser auf dem Desktop-PC oder dem Notebook, ebenso wie auf dem Smartphone, nutzt. Aber auch in nativen Applikationen steckt häufig Web-Technologie. Sei es, dass die ganze App als Web-Applikation entwickelt wurde und dann in eine native Hülle gesteckt wurde (PhoneGap, Cordova), sei es, dass nur ein Teil einer App eine WebView verwendet, etwa für das Hilfesystem. Applikationen wie VSCode verwenden mit Electron ein Plattform basierend auf Chromium und Node.js.

- <https://phonegap.com>
- <https://cordova.apache.org>
- <https://www.electronjs.org>

# ZIELE WBE (WH)

- JavaScript gut verstehen und einsetzen können
- Nebenbei: HTML und CSS verstehen (Selbststudium)
- Auswahl von Web-APIs kennen und einsetzen können
- Eigenes kleines Web-Framework erstellen können

# ABGRENZUNG

- Schwerpunkt Front-end, wenig Back-end
- Beschränkung auf ausgewählte Standards und Web-APIs
- Nur ansatzweise: mobiles Web und zugehörige APIs
- Kein Thema: Gestaltung, Usability und Accessibility

## Speaker notes

Einige Themen müssen explizit aussen vor bleiben, obwohl sie wichtige Themen für Web-Entwickler sind.

*Back-end:* Bis auf wenige Ausnahmen werden wir uns vor allem mit dem Front-end beschäftigen. Die Backend-Entwicklung ist selbst ein riesiges Gebiet und würde Stoff für mindestens ein weiteres Semester umfassen.

*Mobiles Web:* Ein grosser Teil der Web-Nutzung entfällt heute auf Smartphones. Wie Webseiten und Anwendungen so umgesetzt werden, dass sie auf verschiedenen Gerätetypen genutzt werden können, werden wir ansehen. Eine vertiefte Behandlung mobil-spezifischer Themen ist im Rahmen von WBE aber nicht möglich, etwa die Entwicklung hybrider Apps (PhoneGap/Cordova) oder von Progressive Web Apps (PWAs). Hier müssen wir auf das Wahlfach "Mobile Applications" verweisen, in dem neben dem mobilen Web auch native Apps für Android und iOS entwickelt werden.

*Web-APIs:* Nur eine kleine Zahl der WEB-APIs können wir im Rahmen von WBE ansehen, viele interessante APIs bleiben leider unberücksichtigt.

*Gestaltung, Usability und Accessibility:* Das sind ebenfalls sehr wichtige Themen, speziell für Front-end-Entwickler. Leider reicht auch für diese Themen die Zeit in WBE nicht.

Auch wenn die oben genannten Themen in WBE nicht vertieft behandelt werden können, wird es doch für Interessierte zum einen oder anderen Thema Tipps, Quellen und Material für die Einarbeitung geben.

# FRONT-END DEVELOPMENT

- A practical guide to learning front end development for beginners  
<https://www.freecodecamp.org/news/a-practical-guide-to-learning-front-end-development-for-beginners-da6516505e41/>
- Developer Roadmap  
<https://github.com/kamranahmedse/developer-roadmap>
- Learn to become a modern Frontend Developer in 2020  
<https://medium.com/@kamranahmedse/modern-frontend-developer-in-2018-4c2072fa2b9c>
- MDN: Learn Web Development  
<https://developer.mozilla.org/en-US/docs/Learn>
- Frontend Masters Books  
<https://frontendmasters.com/guides/>