



Web-basierte Anwendungen

Studiengänge AI (4140) & WI (4120)



Organisatorisches

Platzvergabe
Regeln zur Teilnahme und Scheinvergabe
Gliederung
Literatur

Spielregeln

- Leistungsnachweis
 - Klausur (80%) als PL
 - Praktikumsübungen (20%) als separate SL, bestanden bei $\geq 50\%$
- Praktikum
 - Mindestens 75% Anwesenheit (hier: **9** von 11)
 - Aber: Ausnahme aufgrund der Corona-Regeln
 - Anreiz zum regelmäßigen Mitmachen:
 - Übungen werden – moderat – bepunktet (s.u.)
 - **Start des Praktikums: Ab KW 15, also ab kommenden Dienstag**
 - Bitte zuvor Vorübung ausführen, s.u.!
- Belegung
 - Wer einen Platz erhielt, aber doch nicht teilnimmt: Bitte melden!
 - Wichtig (alle): Erst alle LVen der Programmierausbildung abschließen!

Spielregeln

- Praktikumsablauf
 - Vorher (!): Abgabe der alten Aufgaben
 - Die ersten ca. 15': Besprechung der Musterlösungen
 - Vorstellung der neuen Aufgabe(n)
 - Bearbeitung der neuen Aufgaben
 - Ggf. parallel dazu: Abnahme der alten Aufgaben
 - Empfehlung: Zweier-Teams bilden, paarweise Abnahmen in Breakout-Räumen
- Anreiz-System
 - Mit jeder Aufgabe können Sie 1...2 "sichere" Punkte erwerben
 - 2 Punkte bei umfangreichen Aufgaben, manchmal Sonderpunkte
 - "Großzügige" Bewertung - auf's Mitmachen kommt es an.
 - Voraussetzungen
 - **Rechtzeitige Abgabe**
 - **Selbständige Bearbeitung**
 - Klare Kennzeichnung mit Name/MatNr
 - Faustregel:
 - Hohe Punktzahl aus Praktikum → I.d.R. keine Probleme mit der Klausur!

"Support"

- Web-Unterstützung: **Stud.IP** nutzen!
 - Skripte (Kopien der Folien, PDF)
 - Praktikumsaufgaben
 - Aktuelle Mitteilungen
 - Linksammlung
- E-Mail nutzen
 - Für Fragen an den Dozenten oder aktuelle Anliegen
 - Verteiler für alle Kursteilnehmer: Über Stud.IP
- Sprechstunde:
 - Nach Vereinbarung
 - Online unter <https://greenlight.cs.hs-rm.de/b/wer-4bv-0be-3gu>

Nacharbeiten und vertiefen - womit?

- Skript oder Präsentation?
 - Präsentation – stichwortartig, erfordert mündliche Ergänzung
 - Skript - zum Nachlesen. Siehe Literaturhinweise!
 - Kompromiss:
Einige Folien sind dicht beschrieben und zum Nachlesen gedacht.
- Hinweise zu den Folien
 - Folien werden inkrementell bereitgestellt
 - Mit Änderungen rechnen → falls drucken, dann möglichst spät
 - PDFs sind für Lesen am Bildschirm optimiert (1 Folie pro Seite)
 - Tipp: Zum Ausdrucken 2 oder 4 Folien pro Druckseite montieren
- Nutzen Sie die zahlreichen *on-line* verfügbaren Spezifikationen, Dokumentationen und Hilfen!
 - Einige Online-Dokumente sollten Sie bereits vor dem Termin erarbeiten, zu dem sie benötigt werden (“flipped classroom”). Sie werden ggf. angekündigt.

Vorbereitungen für HTML und CSS

- Vorlesungskonzept
 - HTML- und CSS-Dokumentation gibt es in hoher Qualität frei verfügbar im Web.
 - Details zu HTML und CSS erarbeiten Sie sich selbstständig!
 - Wichtigste Quelle: [w3schools.com](https://www.w3schools.com)
 - „Treibende Kraft“: Die Praktikumsaufgaben
 - In der Vorlesung besprechen wir Zusammenhänge und Einsichten, die sich nicht so einfach in der Online-Doku finden. Sie bietet auch Gelegenheit zur Klärung von Fragen.
 - Je besser vorbereitet Sie zur Vorlesung kommen, desto mehr haben Sie von dieser Zeit!
- Empfohlener Lesestoff für die nächsten zwei Vorlesungen bei [w3schools.com](https://www.w3schools.com):
 - Learn HTML: Abschnitte von “HTML Home” bis “HTML Id”
 - Learn CSS: Tutorial (ohne „CSS Advanced”)

Vorsicht – viel Stoff. Verschaffen Sie sich erst mal eine Übersicht, um Details bei Bedarf leichter zu finden.



Inhalte, Termine



Webbasierte Anwendungen

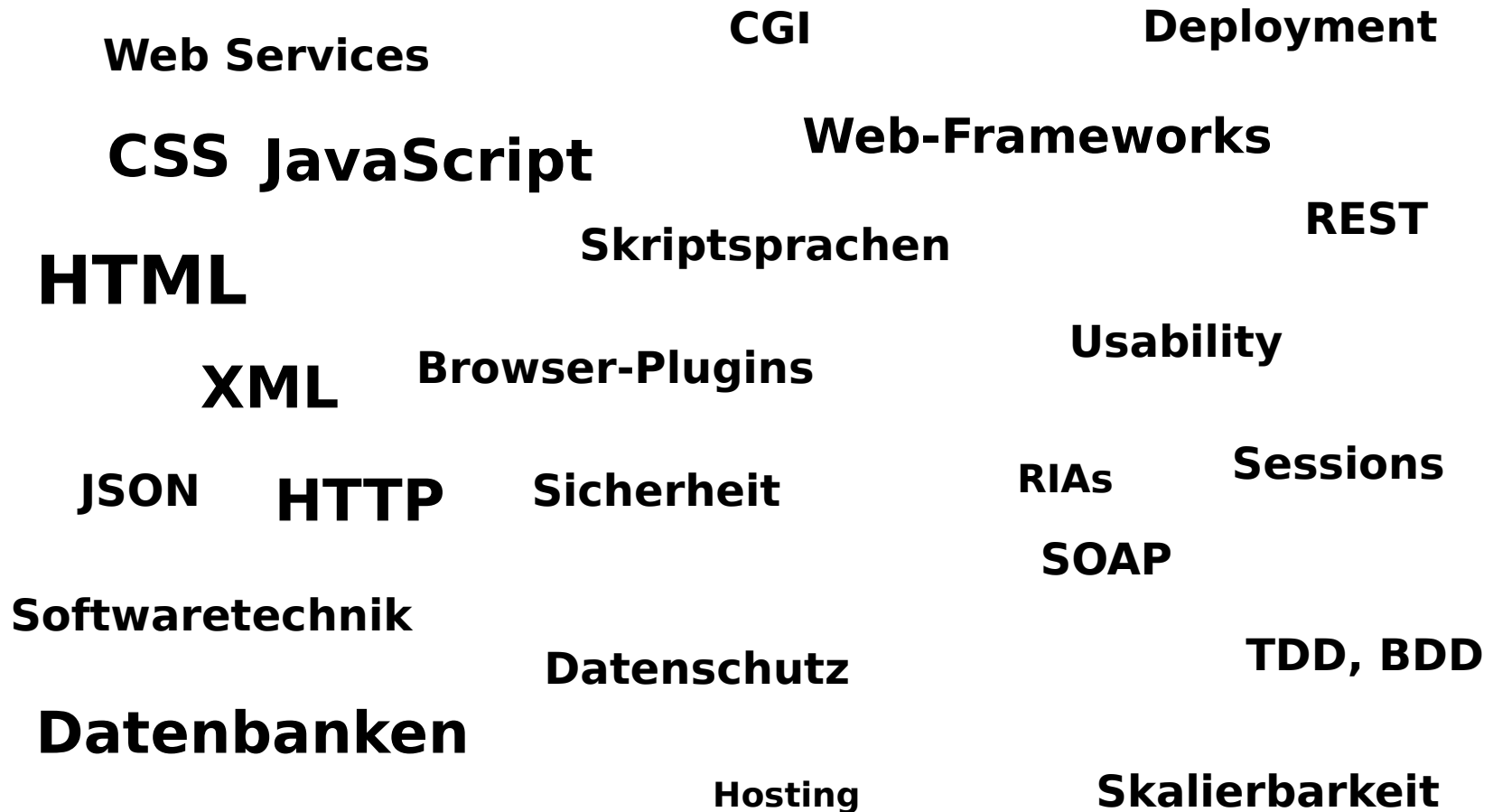
- Was ist eine Webbasierte Anwendung?
 - Word, Excel, OpenOffice?
 - Ihre Smartphone-Software?
 - Die Software Ihres WLAN-Routers?
 - Der Amazon-Shop?
 - Google-Anwendungen?
 - Facebook?
 - SAP?
- WBA-Kennzeichen?
 - Client/Server-Architektur
 - Client = Browser (aber auch andere Software)
 - (Primäres) Protokoll: HTTP
 - (Primärer) Inhalt zur Gestaltung der Benutzerschnittstelle: (X)HTML



Webbasierte Anwendungen

- Vielfalt der Einflüsse

Mobile Devices





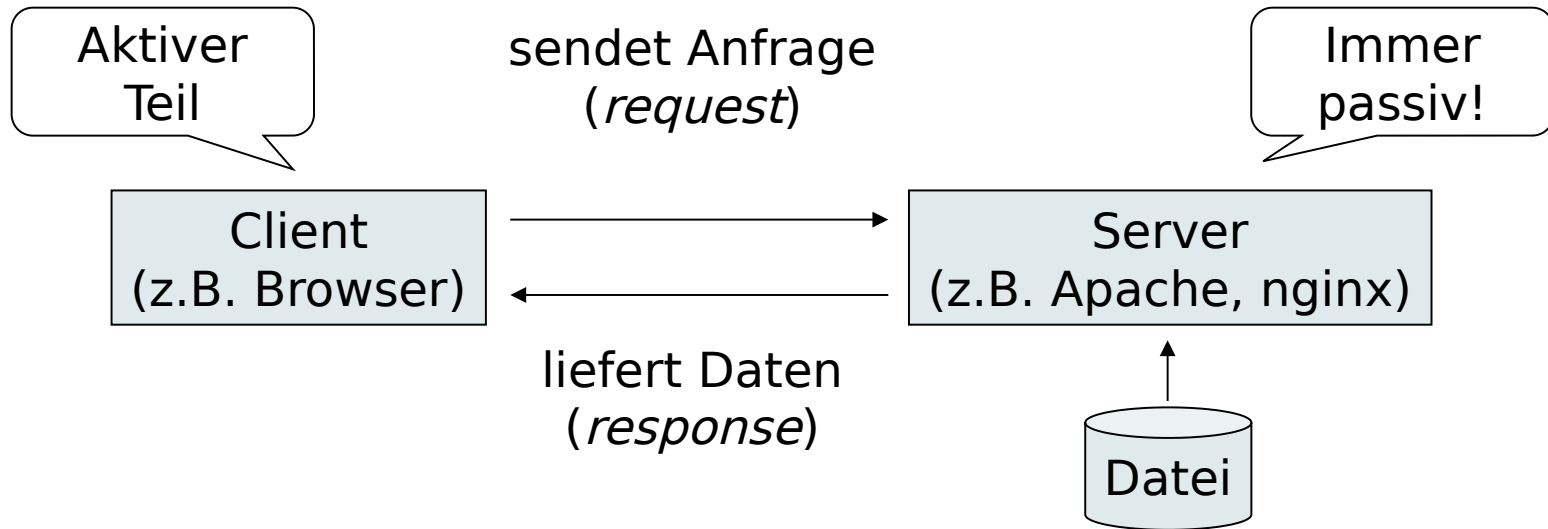
Webbasierte Anwendungen

- Folgen
 - In einem 5cp-Grundkurs zu WBA können wir nur einen kleinen Teil der Aspekte behandeln
 - Wir konzentrieren uns auf die wichtigsten Techniken und vor allem Prinzipien
 - Serverseitige Implementierungssprachen spielen keine große Rolle. Wir verwenden hier eine leicht erlernbare Skriptsprache: Ruby
 - Die Wahl der Tools richtet sich nicht nach ihrer momentanen Verbreitung, sondern nach ihrer Eignung zur Demonstration von Prinzipien und Trends
 - Die hohe Komplexität webbasierter Anwendungen erfordert gute Grundlagen in Programmierung (und Softwaretechnik)!



Webbasierte Anwendungen

- Einfaches Client-Server-Modell



- Grundlagen webbasierter Anwendungen
 - Bestehende Netzwerkverbindung, hier: TCP/IP
 - Gemeinsames Protokoll, hier: **HTTP**
 - Flexible Adressierung, hier: **URL** bzw. URI
 - Standardisierte Inhalte, hier: **(X)HTML, CSS, ...**



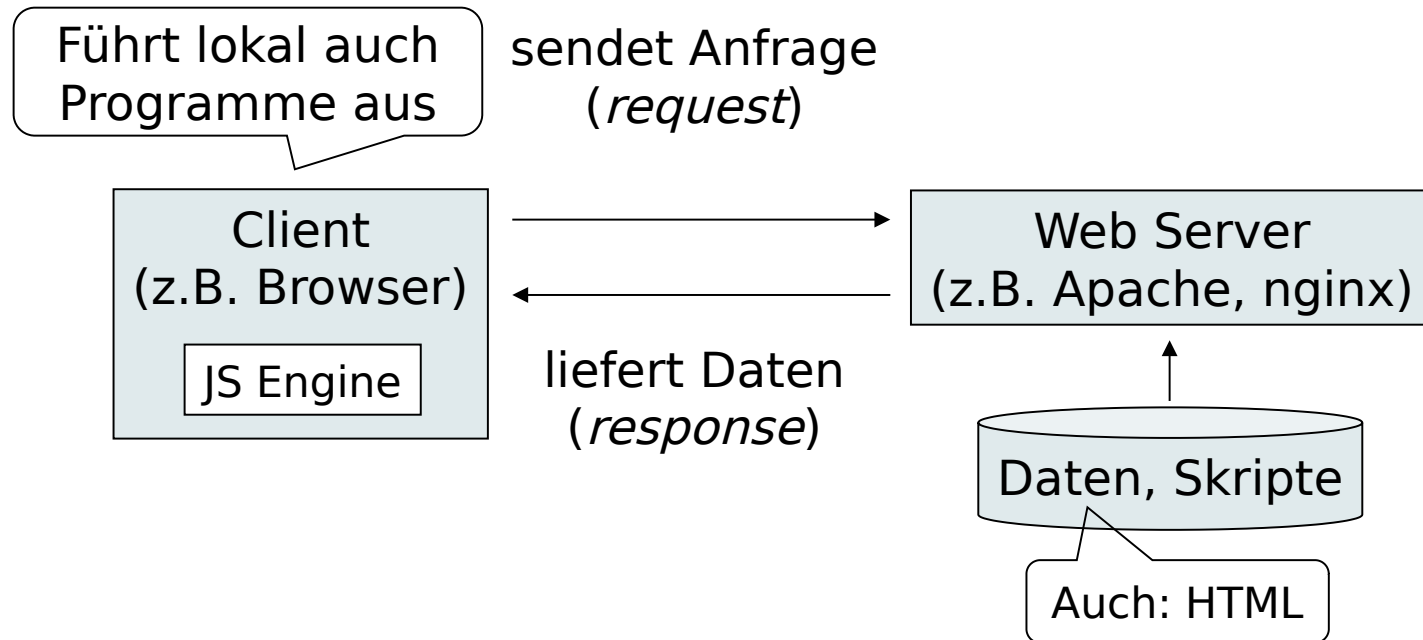
Webbasierte Anwendungen

- WWW \neq Internet
 - Das „Internet“ (Netz von Netzwerken) beruht heute auf IP-Technologie (IPV4 oder IPV6)
 - Es unterstützt viele Protokolle, die Dienste zur Übertragung von Daten nutzen, z.B.
 - E-Mail: SMTP, POP3, IMAP
 - Remote Login: Telnet, ssh
 - File Transfer: ftp, sftp
 - Instant Messaging: IRC
 - Skype: Eigenes Protokoll
 - **Web: HTTP**
 - Wir unterscheiden TCP- und UDP-basierte Protokolle
 - Ports: Viele Protokolle verwenden Standard-Ports, etwa
 - Telnet: 23/TCP, SMTP: 25/TCP, ftp: 20,21/TCP
 - **Web: 80/TCP** (http) bzw. **443/TCP** (https)
- Fazit: **WWW \subset Internet**



Webbasierte Anwendungen

- Rein Client-seitige Web-Anwendungen

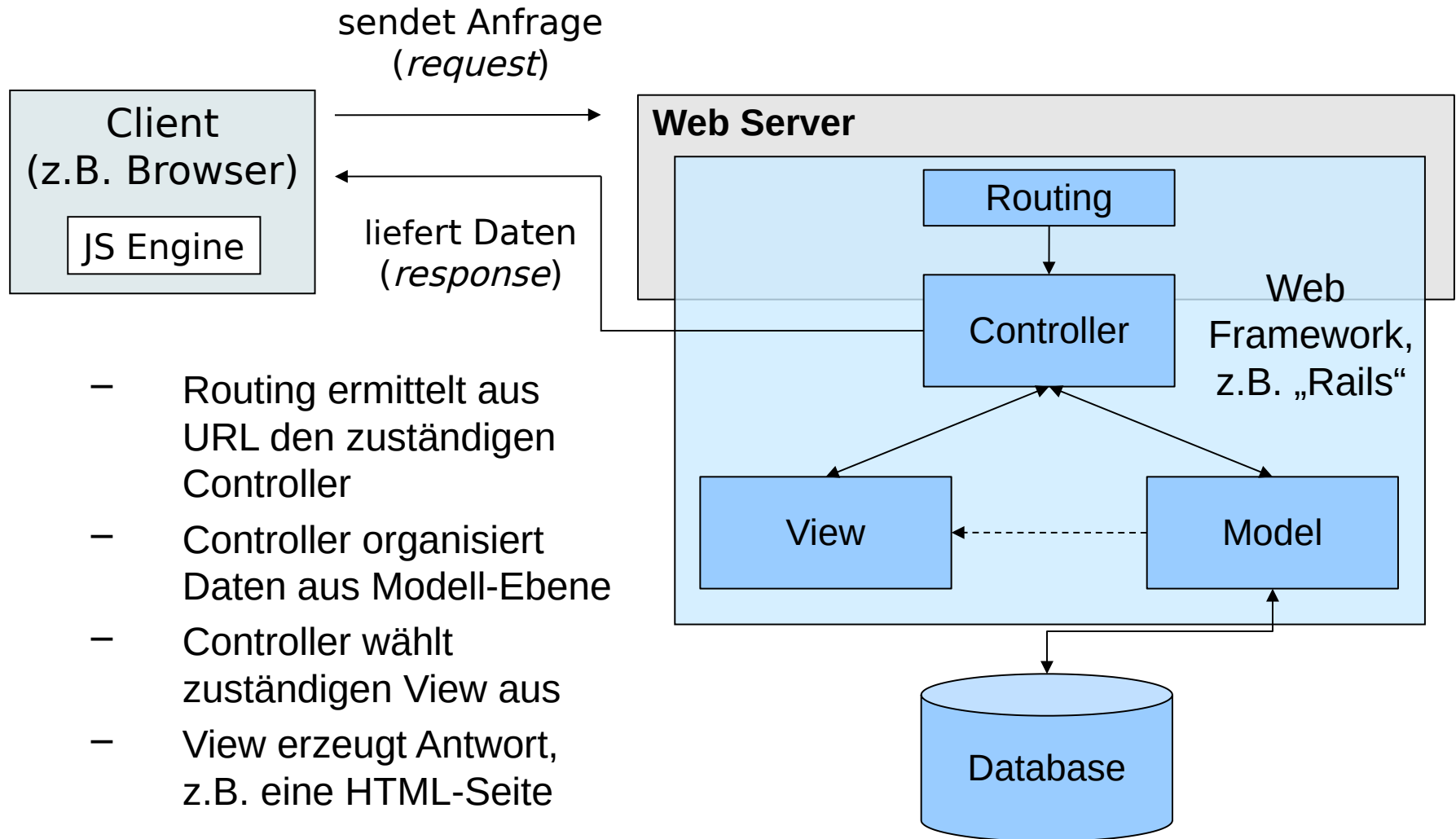


- Client-seitige Anwendungen
 - Der Client enthält eine Komponente zur Ausführung von Programmcode, i.d.R. eine „JavaScript-Engine“
 - Der Client erhält ein API („DOM“), über das Programme Einfluss nehmen können auf die Anzeige wie auch auf den Datenaustausch
 - Der Server liefert JS-Skripte wie andere Daten auf Anfrage aus



Webbasierte Anwendungen

- Serverseitige Web-Anwendung, hier: MVC-Entwurfsmuster





Webbasierte Anwendungen

- Unser Kursprogramm
 - HTML: Grundlagen
 - Zu ergänzen durch Eigenstudium!
 - CSS: Auszüge
 - Zu ergänzen durch Eigenstudium!
 - JavaScript:
 - Grundsätzliches zur Sprache
 - DOM, jQuery: Beispiele
 - Client-seitiger Einsatz: Taschenrechner, Grafiken
 - JS-Frameworks:
 - Rails als organisatorischer Rahmen
 - Ruby-Grundlagen & Rails-Tutorial „Getting started“
 - Kleines Abschluss-Projekt im letzten Drittel
- **Ihre Aufgabe bis zum ersten Praktikum** (ab Dienstag):
 - Installieren Sie die erforderliche Software! → Siehe Vorübung!



Datum (Do) / KW	Vorlesung
15.04.21 / 14	Regeln; Überblick, HTML
22.04.21 / 15	(X)HTML, HTML5; HAML; Layout
29.04.21 / 16	CSS2.1, CSS3; SCSS
06.05.21 / 17	HTML5: Neues, insb. UJS-Vorb.
13.05.21 / 18	Template-Techniken, SSI, JSP
20.05.21 / 19	JavaScript; CoffeeScript
27.05.21 / 20	JavaScript: Objekte; JQuery
03.06.21 / 21	JS: DOM; AJAX
10.06.21 / 22	HTTP 1.1, Proxies & Caches
17.06.21 / 23	Server-Seite: Modelle, DB-Anbindung
24.06.21 / 24	REST; Controller; Sessions
01.07.21 / 25	Controller; JS/Ajax
08.07.21 / 26	Sicherheit im Web. WASP Top 10
Ab 12.07.21 / 27	(Prüfungswochen)





Datum (Di) / KW	Praktikum
13.04.21 / 14	„Getting Started“: Bis 3.1 „Installing Rails“
20.04.21 / 15	01: HTML: HAML/Rails-Version (Abgabe in 2 Wochen)
27.04.21 / 16	02: (S)CSS für HTML (Abgabe in 1 Woche)
04.05.21 / 17	03: HTML: Formulare (Abgabe in 2 Wochen)
11.05.21 / 18	04: Rails „Getting started“; Ruby (Eigenstudium)
18.05.21 / 19	05: Form, UJS: Taschenrechner
25.05.21 / 20	05: UJS, CSS: Taschenrechner
01.06.21 / 21	06: JS: DOM, JSON: Tabelle
08.06.21 / 22	06: JS: DOM, JSON: Grafik
15.06.21 / 23	07: Projekt: Rails-Anwendung
22.06.21 / 24	Projekt: Rails-Anwendung
29.06.21 / 25	Projekt: Rails-Anwendung
06.07.21 / 26	Projekt-Präsentationen.

* Zitate zu Software-Entwicklung

- *Sometimes it pays to stay in bed in Monday, rather than spending the rest of the week debugging Monday's code. ~Dan Salomon*
- *Everyone knows that debugging is twice as hard as writing a program in the first place. So if you are as clever as you can be when you write it, how will you ever debug it? ~Brian Kernighan*
- **Controlling complexity is the essence of computer programming.**
~Brian Kernighan
- **Good code is its own best documentation.** As you're about to add a comment, ask yourself, "How can I improve the code so that this comment isn't needed?" ~Steve McConnell
- **Any fool can write code that a computer can understand. Good programmers write code that humans can understand.**
~Martin Fowler

Zitate

- Vollkommenheit entsteht offenbar nicht dann, wenn man nichts mehr hinzuzufügen hat, sondern wenn man nichts mehr wegnehmen kann.
Antoine de St. Exupéry
- **Folgen (nicht nur) für WBA**
 - **Unser Code sollte nicht nur funktionieren, sondern für Menschen möglichst leicht nachvollziehbar sein**
 - **Kürze von Code ist wichtig (bei gleicher Verständlichkeit)**
 - **Vergeben Sie Namen/Bezeichner nie leichtfertig – sie erschließen das Verständnis**
- Zum Schluss:
 - *It's the only job I can think of where I get to be both an engineer and an artist. There's an incredible, rigorous, technical element to it, which I like because you have to do very precise thinking. On the other hand, it has a wildly creative side where the boundaries of imagination are the only real limitation. ~Andy Hertzfeld, about programming*
Quelle: <http://www.quotegarden.com/programming.html>

* Quellen

- HTML
 - selfHTML: <https://wiki.selfhtml.org>
 - W3Schools: <https://w3schools.com>
- JavaScript, CoffeeScript
 - selfHTML, W3Schools (s.o.)
 - Christian Wenz: JavaScript: Grundlagen, Programmierung, Praxis. Galileo Computing, 11. Auflage, 2014. ISBN 978 3 83621 979 2
 - Philip Ackermann: JavaScript. Das umfassende Handbuch, Rheinwerk Verlag, 1. Auflage, 2016. ISBN 978 3 8362 3838 0
 - Trevor Burnham: CoffeeScript. Accelerated JavaScript Development. The Pragmatic BookShelf, Dallas & Raleigh, 2011. ISBN 978 1 93435 678 4
 - Dito, 2. Auflage (2015): ISBN 978 1 94122 226 9
 - Projekt-Webseite: <http://coffeescript.org>
- Ruby
 - <https://ruby-lang.org>, <https://ruby-doc.org>
- Wettbewerb der Frameworks/Plattformen 2011
 - <https://www.plat-forms.org/results-2011>
 - <http://www.heise.de/developer/meldung/Plat-Forms-2011-Ruby-Programme-besonders-vollstaendig-1386170.html>