

# W3DSKI 104 – Fortgeschrittene Programmierung

Dozent: Prof. Dr. Michael Eichberg  
Kontakt: [michael.eichberg@dhbw.de](mailto:michael.eichberg@dhbw.de), Raum 149B  
Version: 1.0

# Verteilte Systeme - Kerninhalte gem. MHB

- Terminologie, Konzepte, Architekturen, Anforderungsprofile und Architekturmodelle für verteilte Systeme
- Synchrone und asynchrone Kommunikation und entfernter Methodenaufruf (RMI, RPC, Web-Services, ...), Übertragungsprotokolle und APIs zwischen Client und Server (z.B. HTTP, HTTPS, WebSockets, Fetch API, etc.), Einführung in das RESTful API-Design
- Herausforderungen in verteilten Systemen
- High Performance Computing und Distributed Computing
- Sicherheitsaspekte bei der Verarbeitung von verteilten Anwendungen; Authentifizierung, Autorisierung, Rollenkonzepte
- Grundlagen Ubiquitous Computing, Internet der Dinge, MQTT, Edge Computing, Streaming & Messaging

# Web-Entwicklung - Kerninhalte gem. MHB

- Frontend-Technologien, HTML, CSS, JavaScript
- Konzepte, Entwurfsmuster und Werkzeuge für die Entwicklung von Web-Anwendungen - Entwurf und Umsetzung von Responsive Web-Design und zustandsbehafteten Web-Anwendungen
- Abgrenzung client-side und Server-Side-Rendering

Labor Web Projekt: - Die theoretischen Inhalte sollen jeweils auch mit aktuellen Technologien beispielhaft umgesetzt werden. Es soll eine übergreifende Anwendung entwickelt werden anhand derer das Zusammenspiel deutlich wird.

# Prüfungsleistung - Portfolio

## Hintergrund

- Das Modul hat 5 ECTS
- Insgesamt gibt es max. 120 Punkte
- 3 Bestandteile:

1. Vorträge - max. 30 Punkte
2. Schriftlicher Zwischenleistungsnachweis - max. 40 Punkte

Der schriftliche Zwischenleistungsnachweis wird ähnlich zur einer Klausur sein und am 24. Juni erfolgen. Die Dauer ist 40 Minuten. Inhalt sind alle bis zum 18. Juni behandelten Themen. Schwerpunkt bilden Themen aus dem Bereich der verteilten Systeme. Es wird aber auch Fragen aus dem Bereich der Web-Entwicklung geben.

3. Programmierprojekt - max. 50 Punkte

# Vorträge - organisatorische Rahmenbedingungen


- Vortragsdauer pro Person: 10 Minuten.
- Jeder hält nur einen Vortrag.
- Die Vortragsthemen werden sukzessive vergeben.

Vortragsthemen

# 1. Vorträge am 4. Juni

# Barrierefreiheit und HTML (2 Personen)[1]

4. Juni

- Allgemeine Einführung in das Thema Barrierefreiheit ( Accessibility)
- Barrierefreie HTML Webseiten mit HTML5 Gestalten
- Erste Quellen:
  - <https://developer.mozilla.org/en-US/docs/Learn/Accessibility>
  - <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA>
  - <https://web.dev/learn/accessibility>
  - <https://www.w3.org/TR/wai-aria/>

Achtung: *CSS und JavaScript sollen nicht behandelt werden.*

---

[1] Präsentation nach der Einführung in HTML möglich.

# HTML Forms (2 Personen)[2]

4. Juni

- Formulare mit HTML; insbesondere Input Typen und Validierung von Daten
- Erste Quellen:
  - <https://developer.mozilla.org/en-US/docs/Learn/Forms>
  - <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/form>
  - <https://web.dev/learn/forms>

*Achtung: CSS und JavaScript soll nicht behandelt werden.*

---

[2] Präsentation nach der Einführung in HTML möglich.



# SVG - Vektorgrafik auf Webseiten (1 Person)[3]

4. Juni

- Einführung in SVG und Vektorgrafiken
- HTML5 und eingebettete SVGs
- Erste Quellen:
  - <https://developer.mozilla.org/en-US/docs/Web/SVG>
  - <https://svg-tutorial.com>
  - [https://www.w3schools.com/graphics/svg\\_intro.asp#:~:text=SVG%20stands%20for%20Scalable%20Vector,supported%20by%20all%20major%20browsers.](https://www.w3schools.com/graphics/svg_intro.asp#:~:text=SVG%20stands%20for%20Scalable%20Vector,supported%20by%20all%20major%20browsers.)

Achtung: *JavaScript und CSS sollen nicht behandelt werden.*

---

[3] Präsentation nach der Einführung in HTML möglich.

## 2. Vorträge am 12. Juni

# Common.js Modules (1 Person)

12. Juni

- Modulsystem (insbesondere) für NodeJS
- <https://nodejs.org/api/modules.html>

# ECMAScript Modules (1 Person)

12. Juni

- Modulsystem für JavaScript (NodeJS und Browser)
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules>

# Saas (1 Person)

12. Juni

■ CSS Erweiterung (CSS Preprocessor)

■ <https://sass-lang.com/>

# Less (1 Person)

12. Juni

■ CSS Preprocessor

■ <https://lesscss.org/>

## 3. Vorträge am 17. Juni

# Bootstrap (2 Personen)

17. Juni

■ Entwicklung von *responsive* Webseiten

■ <https://getbootstrap.com/>

Achtung: Sass (und Less) werden getrennt behandelt werden.



## 4. Vorträge am 18. Juni

# Express (3 Personen)

18. Juni

- Einführung in Express
- *Programmierung von Endpunkten*
- <https://expressjs.com/>

## Hinweise

- *Grundlagen des RESTful Designs brauchen nicht in der Präsentation behandelt werden; d. h. die Präsentation soll sich darauf konzentrieren, wie man Express verwendet (Hands-on.)*
- WebSockets sollen nicht behandelt werden (es gibt einen weiteren Vortrag dazu).

## 5. Vorträge am 24. Juni

# Electron (2 Personen)

24. Juni

- Framework zur Entwicklung von Cross-Platform Desktop-Anwendungen mit Webtechnologien.
- <https://www.electronjs.org/>

# WebSockets (und Express) (1 Person)

24. Juni

- Einführung in WebSockets
- Verwendung von WebSockets in Browsern

[https://developer.mozilla.org/en-US/docs/Web/API/WebSockets\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API)

- Programmierung von WebSockets in Express

<https://www.npmjs.com/package/express-ws?activeTab=readme> (Abstimmen mit der Gruppe, die Express präsentiert.)

# WebRTC (1 Person)

24. Juni

■ Echtzeitkommunikation für das Web

■ <https://webrtc.org/>

# Vue.js (3 Personen)

24. Juni

■ JavaScript Framework für Web User Interfaces

■ <https://vuejs.org/>

## 6. Vorträge am 25. Juni



# Typescript (2 Personen)

25. Juni

- JavaScript with types

(Der Vortrag sollte sich auf die Vorteile von statischer Typisierung konzentrieren; insbesondere für das *Programming in the large.*)

- <https://www.typescriptlang.org/>

# Angular (4 Personen)

25. Juni

■ Web-Application Framework

■ <https://angular.io/>

# Svelte (4 Personen)

25. Juni

- Javascript Web Framework
  - Svelte
  - SvelteKit („Server Side Rendering“)
- <https://svelte.dev/>

# React (4 Personen)

25. Juni

■ Frontend orientierte Bibliothek für die Entwicklung von Benutzeroberflächen

■ <https://react.dev/>

# ***Consensus Algorithms (2 Personen)*****[4]**

25. Juni

- Einführung in Paxos
  - Einführung in Raft
  - Erste Quellen:
    - [https://en.wikipedia.org/wiki/Paxos\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Paxos_(computer_science))
    - <https://raft.github.io>
- 

**[4]** Präsentation nach den grundlegenden Konzepten (insbesondere 2PC) von verteilten Systemen.