

Multimedia Engineering II

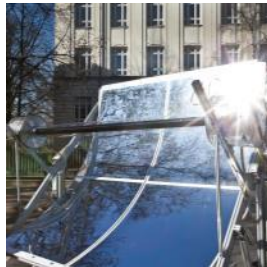
01 Überblick

Johannes Konert



BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN

University of Applied Sciences



Überblick und Start

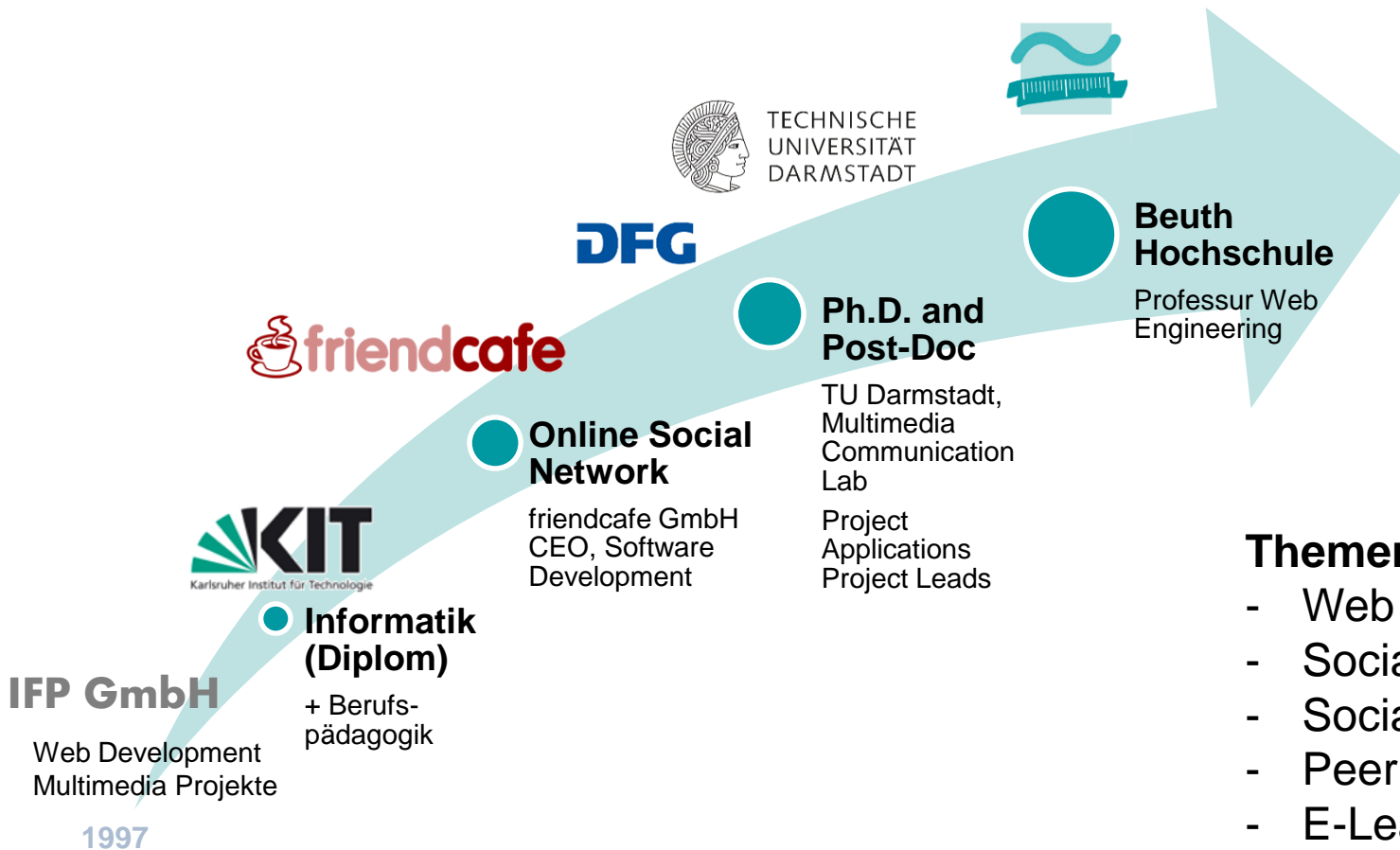
- Wer bin ich?
- Wer sind Sie?
- (Lern)ziele von MME 2
- Organisatorisches
 - Zeitplan und Stundenplan
 - Benotung (Übungen und Klausur)
 - Kommunikation
 - Vorwissen
- Crashkurs
 - HTML5
 - CSS3
 - JavaScript (bzw. ECMAScript 5)
- Übungsblatt 1 und Zusammenfassung

Überblick und Start

■ Wer bin ich?

- Wer sind Sie?
- (Lern)ziele von MME 2
- Organisatorisches
 - Zeitplan und Stundenplan
 - Benotung (Übungen und Klausur)
 - Kommunikation
 - Vorwissen
- Crashkurs
 - HTML5
 - CSS3
 - JavaScript (bzw. ECMAScript 5)
- Übungsblatt 1 und Zusammenfassung

Johannes Konert



Themen:

- Web Anwendungen
- Social Media
- Social Learning
- Peer Learning
- E-Learning

Beispiele meiner Arbeit



■ friendcafe

■ Technologie:

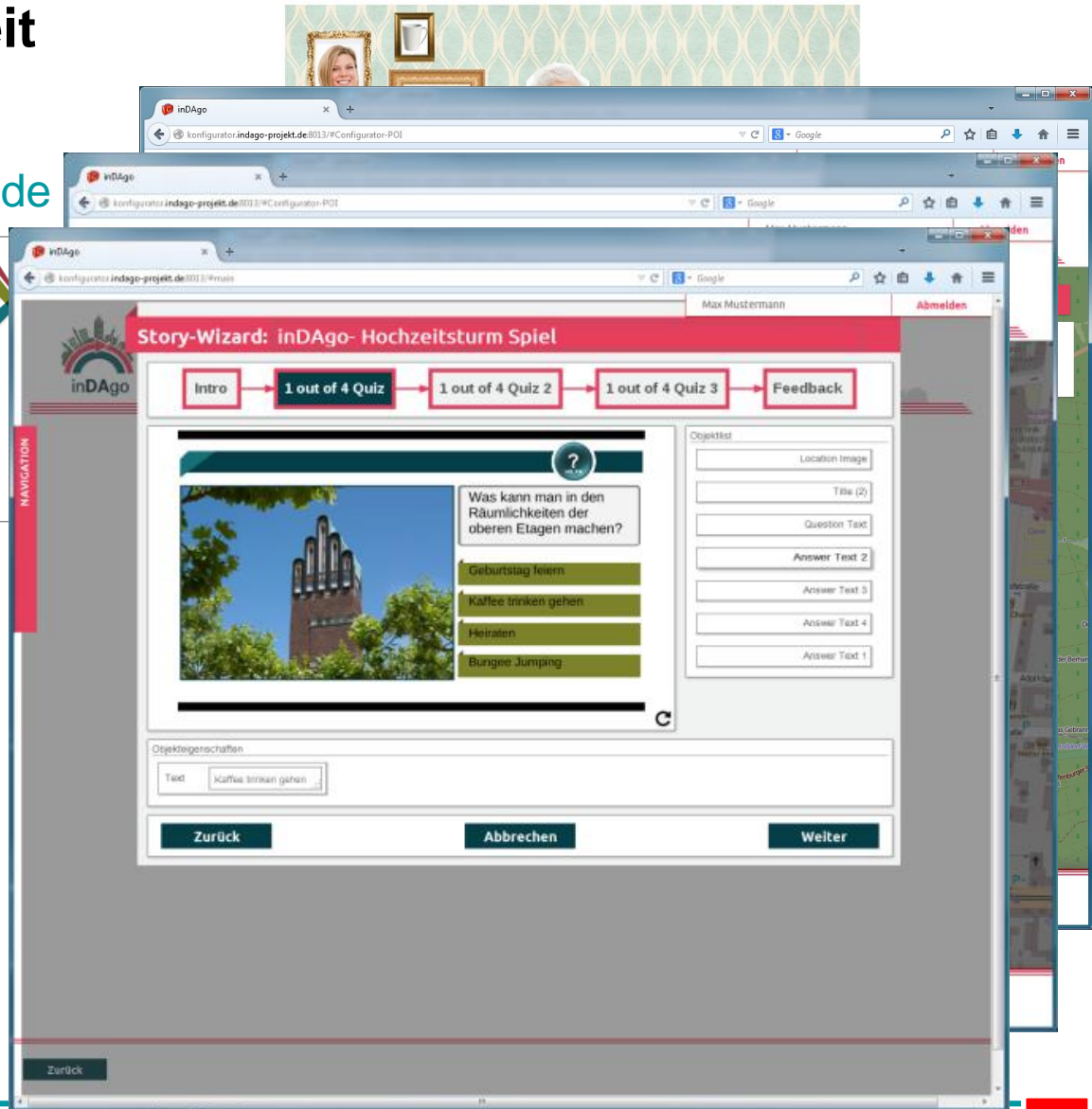
- JavaServlets
- Webserver:
Apache Tomcat
- Eigene
TemplateEngine
- Eigene verteilte
Architektur
- YUI Client-JS



The screenshot displays the friendcafe website interface. At the top, there's a navigation bar with the site logo, user greeting 'Hallo Johannes Konert', and links for 'HILFE' and 'LOGOUT'. Below this is a secondary navigation bar with 'MEINE SEITE', 'SUCHEN', 'STÖBERN', 'EINLADEN', and 'KENNENLERNEN'. The main content area is titled 'Annkathrins Seite' and features tabs for 'Profil', 'Freunde', 'Gruppen', 'Statements', 'Fotos', and 'Blog'. The 'Profil' tab is active, showing a user profile for 'Annkathrin Wulff'. The profile includes a photo, age '22 Jahre, aus Heidelberg', and status 'Offline seit 1T 15h'. To the right, there's a section 'Über Annkathrin' with details like 'Name: Annkathrin Wulff', 'friendcafe-Ämter: Ehrenamts-Team, Guide', 'Verbindung:', 'Familienstand: Ich geh ins Kloster', 'Hobbies: in der Sonne liegen, essen, trinken, schlafen...!', 'Sportarten: Fußball, Schwimmen, Schießen', 'Bands & Co: Kanye West, Fettes Brot, Die Ärzte, Maroon 5', 'Schauspieler: Eric Bana, Ron Rifkin, Colin Farrell', 'Filme: Gladiator', 'Reiseziele: Rom', 'Job: Special Operations', and 'Hochschule: Uni Heidelberg in Heidelberg, Studienbeginn: WS / 2005, Studienfach: Lehramt Gymnasium'. Below the profile, there's a 'Freunde' section showing a list of friends with their photos and names. On the right side, there's a 'Messenger' sidebar with 'Meine Mails' and 'Readme ...' buttons, and a list of 'Freunde' and 'Chats'.

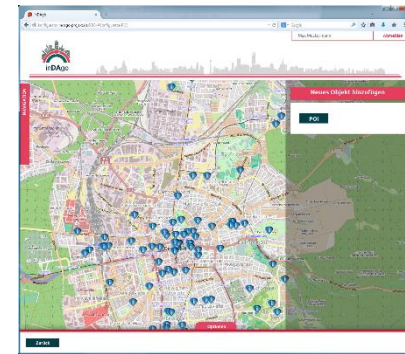
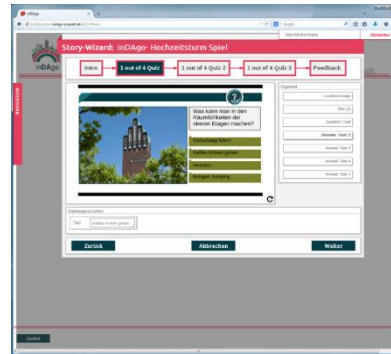
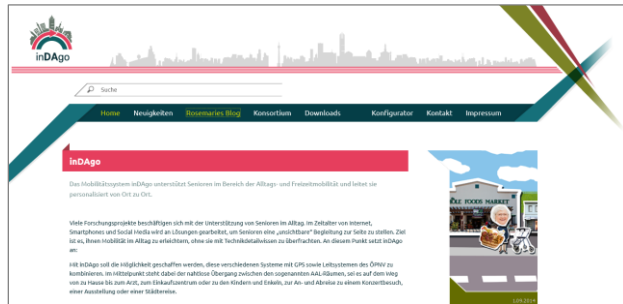
Beispiele meiner Arbeit

■ inDAgo <http://www.indago-project.de>

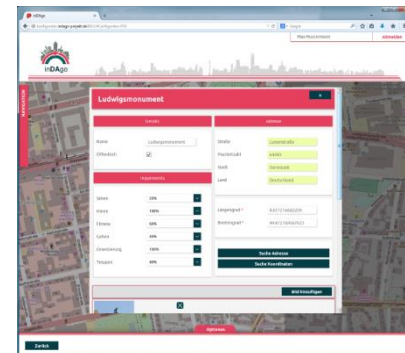


Beispiele meiner Arbeit

■ inDAgo <http://www.indago-project.de>

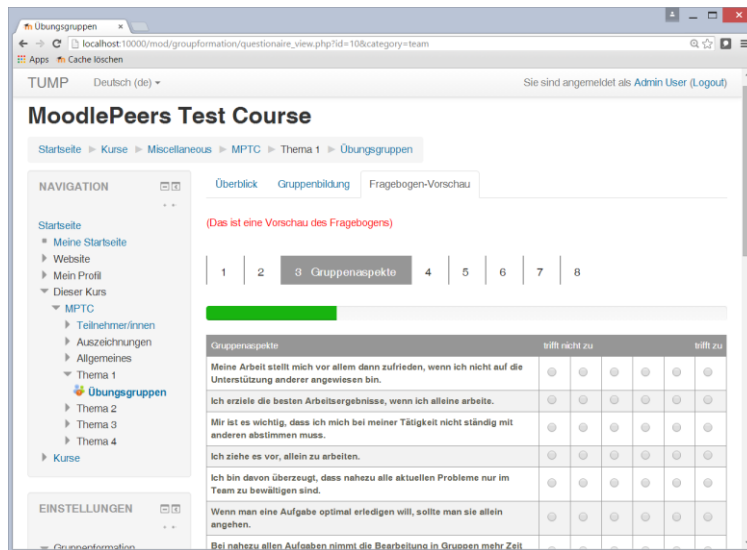


- Technologie:
 - SinglePageApp (SPA)
 - Google Web Toolkit
 - Server: Apache Tomcat

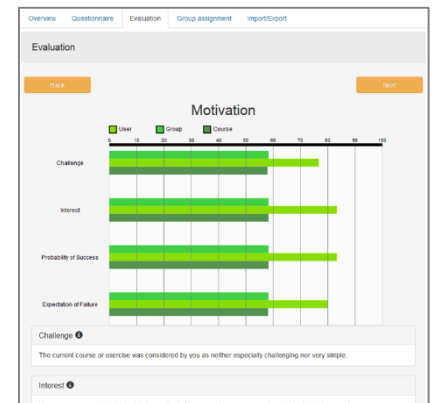
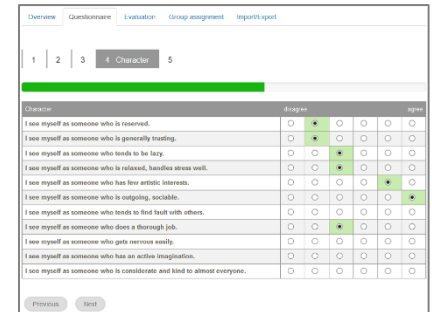
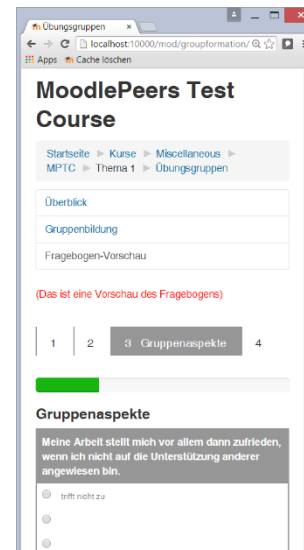


Beispiele meiner Arbeit

- MoodlePeers
<http://sourceforge.net/projects/moodlepeers/>



- Technologie
 - Moodle/PHP
 - jQuery



Überblick und Start

- Wer bin ich?
- **Wer sind Sie?**
- (Lern)ziele von MME 2
- Organisatorisches
 - Zeitplan und Stundenplan
 - Benotung (Übungen und Klausur)
 - Kommunikation
 - Vorwissen
- Crashkurs
 - HTML5
 - CSS3
 - JavaScript (bzw. ECMAScript 5)
- Übungsblatt 1 und Zusammenfassung

Wer sind Sie?

- Vorwissen: Tools, Programmiersprachen aus Projekten oder Praktika



Kurzer Moodlefragebogen als Teil der Übung1



- **Was erwarten Sie von ME2? (2min Zeit)**

- Je 1 Moderationskarte
- “Ich möchte etwas lernen über ...”
- 2-3 Stichpunkte
- Wird eingesammelt



Feedback beim nächsten Mal

Überblick und Start

- Wer bin ich?
- Wer sind Sie?
- **(Lern)ziele von MME 2**
- Organisatorisches
 - Zeitplan und Stundenplan
 - Benotung (Übungen und Klausur)
 - Kommunikation
 - Vorwissen
- Crashkurs
 - HTML5
 - CSS3
 - JavaScript (bzw. ECMAScript 5)
- Übungsblatt 1 und Zusammenfassung

Ziele von Multimedia Engineering 2

Auszüge aus dem Modulhandbuch

Alternativen Server-seitiger Programmierung mit Vertiefung

Modulare Programmierung

Vorgehensmodelle, Architektur-Muster

Design und Architekturmuster für Client-seitige Programmierung

Multi-Device und Cross-Media Szenarien
(AV-Streaming-Technologien)



Literatur im Modulhandbuch

...veraltet...bitte nicht benutzen!



Ziele von Multimedia Engineering 2

- **Schwerpunkt liegt auf Server-seitiger Implementierung**
 - JavaScript
 - API-Implementierung
 - Testen
 - Datenbank-Anbindung
- **Kommunikation Client-Server** via JSON
- Keine Basics in HTML5, CSS3, JavaScript
- Keine Rich Client Applications



Ziele von Multimedia Engineering 2

- Am Ende von MME2 können Sie...

..aus der Vielfalt an Server-Architekturen anhand bekannter Vor- und Nachteile für Ihre Anforderungen ein passende auswählen.

..moderne REST APIs entwerfen, implementieren und testen.

..die Client-Server-Kommunikation via JSON konzipieren und mittels node.js und backbone.js implementieren



Überblick und Start

- Wer bin ich?
- Wer sind Sie?
- (Lern)ziele von MME 2
- **Organisatorisches**
 - Zeitplan und Stundenplan
 - Benotung (Übungen und Klausur)
 - Kommunikation
 - Vorwissen
- Crashkurs
 - HTML5
 - CSS3
 - JavaScript (bzw. ECMAScript 5)
- Übungsblatt 1 und Zusammenfassung

Zeitplan Zug 1, Dienstag (vorläufig)

	Datum	Thema
1	05.04.2016	
2	12.04.2016	Einführung, Ziele, Ablauf, Benotung
3	19.04.2016	Client-Server Architektur
4	26.04.2016	REST-APIs
5	03.05.2016	REST in node.js
6	10.05.2016	Debugging und Testen
7	17.05.2016	Strukturierung, Modularisierung
8	24.05.2016	Vertiefung einzelner Themen
9	31.05.2016	Datenhaltung, SQL, NoSql, primär mit MongoDB
10	07.06.2016	backbone.js als Gegenpart zu REST/node
11	14.06.2016	Authentifizierung und Patterns
12	21.06.2016	Mobile Development/Cross-Plattform-Development
13	28.06.2016	Gastdozent(en)
14	05.07.2016	Klausurvorbereitung
15	12.07.2016	Klausur PZR1 (Di, 12.07. 14:00 Uhr, Ingeborg-Meising-S.)
16	19.07.2016	Klausureinsicht
	21.09.2016	Klausur PZR2 (Mi, 21.09. 12:00 Uhr, Raum B101, H Gauß)

Zeitplan Zug 2, Mittwoch (vorläufig)

	Datum	Thema
1	06.04.2016	
2	13.04.2016	Einführung, Ziele, Ablauf, Benotung
3	20.04.2016	Client-Server Architektur
4	27.04.2016	REST-APIs
5	04.05.2016	REST in node.js
6	11.05.2016	Debugging und Testen
7	18.05.2016	Strukturierung, Modularisierung
8	25.05.2016	Vertiefung einzelner Themen
9	01.06.2016	Datenhaltung, SQL, NoSql, primär mit MongoDB
10	08.06.2016	backbone.js als Gegenpart zu REST/node
11	15.06.2016	Authentifizierung und Patterns
12	22.06.2016	Mobile Development/Cross-Plattform-Development
13	29.06.2016	Gastdozent(en)
14	06.07.2016	Klausurvorbereitung
15	12.07.2016	Klausur PZR1 (Di, 12.07. 14:00 Uhr, Ingeborg-Meising-S.)
16	20.07.2016	Klausureinsicht
	21.09.2016	Klausur PZR2 (Mi, 21.09. 12:00 Uhr, Raum B101, H Gauß)

Zeitplan Zug 1 ÜBUNGEN (vorläufig)

	Datum	Thema	Übung
1	05.04.2016		Ü1: Client-Website
2	12.04.2016	Einführung, Ziele, Ablauf, Benotung	Ü1
3	19.04.2016	Client-Server Architektur	Ü2: Server mit node.js
4	26.04.2016	REST-APIs	Ü2
5	03.05.2016	REST in node.js	- Feiertag - (5.5.)
6	10.05.2016	Debugging und Testen	Ü3: API mit node.js
7	17.05.2016	Strukturierung, Modularisierung	Ü3
8	24.05.2016	Vertiefung einzelner Themen	Ü4: Umfangreiche REST API
9	31.05.2016	Datenhaltung, SQL, NoSql, primär mit MongoDB	Ü4
10	07.06.2016	backbone.js als Gegenpart zu REST/node	Ü4
11	14.06.2016	Authentifizierung und Patterns	Ü5: mongoDB-Anbindung
12	21.06.2016	Mobile Development/Cross-Plattform-Development	Ü5
13	28.06.2016	Gastdozent(en)	Ü6: Backbone.js
14	05.07.2016	Klausurvorbereitung	Ü6
15	12.07.2016	Klausur PZR1 (Di, 12.07. 14:00 Uhr, Ingeborg-Meising-S.)	-
16	19.07.2016	Klausureinsicht	-
	21.09.2016	Klausur PZR2 (Mi, 21.09. 12:00 Uhr, Raum B101, H Gauß)	-

Zeitplan Zug 2 ÜBUNGEN (vorläufig)

	Datum	Thema	Übung
1	06.04.2016		Ü1: Client-Website
2	13.04.2016	Einführung, Ziele, Ablauf, Benotung	Ü1
3	20.04.2016	Client-Server Architektur	Ü2: Server mit node.js
4	27.04.2016	REST-APIs	Ü2
5	04.05.2016	REST in node.js	- Feiertag - (5.5.)
6	11.05.2016	Debugging und Testen	Ü3: API mit node.js
7	18.05.2016	Strukturierung, Modularisierung	Ü3
8	25.05.2016	Vertiefung einzelner Themen	Ü4: Umfangreiche REST API
9	01.06.2016	Datenhaltung, SQL, NoSql, primär mit MongoDB	Ü4
10	08.06.2016	backbone.js als Gegenpart zu REST/node	Ü4
11	15.06.2016	Authentifizierung und Patterns	Ü5: mongoDB-Anbindung
12	22.06.2016	Mobile Development/Cross-Plattform-Development	Ü5
13	29.06.2016	Gastdozent(en)	Ü6: Backbone.js
14	06.07.2016	Klausurvorbereitung	Ü6
15	12.07.2016	Klausur PZR1 (Di, 12.07. 14:00 Uhr, Ingeborg-Meising-S.)	-
16	20.07.2016	Klausureinsicht	-
	21.09.2016	Klausur PZR2 (Mi, 21.09. 12:00 Uhr, Raum B101, H Gauß)	-

Stundenplan Zug 1

	Dienstag	Mittwoch	Donnerstag
08:00-09:30			Übung Gruppe 2a B223
10:00-11:30			Übung Gruppe 2b B223
12:15-13:45	Unterricht Zug 1 B301	Unterricht Zug 2 B042	Übung Gruppe 1a B223
14:15-15:45			Übung Gruppe 1b B223
16:00-17:30			
17:45-19:15			

Stundenplan Zug 2

	Dienstag	Mittwoch	Donnerstag
08:00-09:30			Übung Gruppe 2a B223
10:00-11:30			Übung Gruppe 2b B223
12:15-13:45	Unterricht Zug 1 B301	Unterricht Zug 2 B042	Übung Gruppe 1a B223
14:15-15:45			Übung Gruppe 1b B223
16:00-17:30			
17:45-19:15			

Überblick und Start

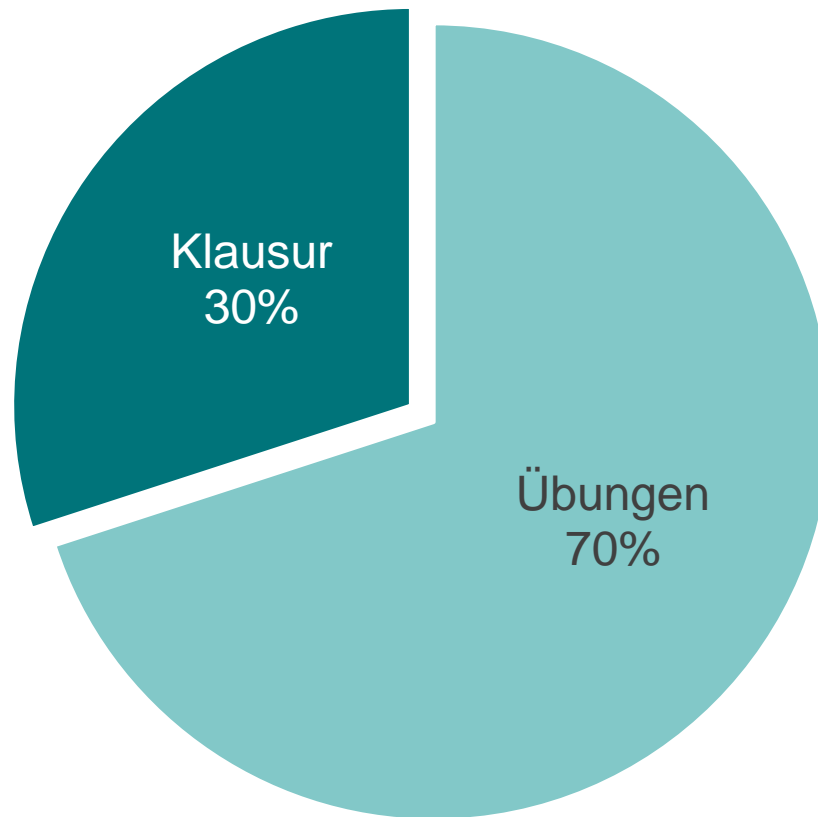
- Wer bin ich?
- Wer sind Sie?
- (Lern)ziele von MME 2
- Organisatorisches
 - Zeitplan und Stundenplan

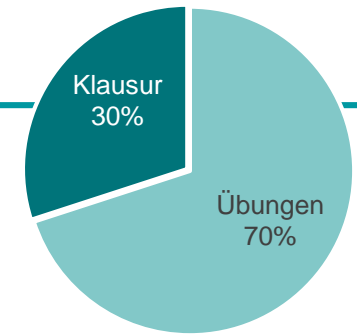
Benotung (Übungen und Klausur)

- Kommunikation
- Vorwissen
- Crashkurs
 - HTML5
 - CSS3
 - JavaScript (bzw. ECMAScript 5)
- Übungsblatt 1 und Zusammenfassung

Benotung

Endnote





Benotung – Übungen

- Eine Übung gilt als bestanden, wenn mind. 50% der Pflichtpunkte erreicht wurden
- Bonusaufgaben mit extra Punkten
- Pflicht+Bonus = Punkte der Übung (maximal so viele wie Pflichtteil)
- Verspätete Abgabe einer Übung:
bis 1Woche: max. 75% der Punkte,
bis 2Wochen: max. 50% der Punkte
- Summe aller Übungen = 60 Punkte

Ü-Note	ab einschl. (Punkte)	
1,0	48	80%
1,3	46	77%
1,7	44	73%
2,0	42	70%
2,3	40	67%
2,7	38	63%
3,0	36	60%
3,3	34	57%
3,7	32	53%
4,0	30	50%
5,0	<30	

	Thema	Dauer (Wochen)	Punkte Pflicht- teil	Punkte Bonus- teil
Ü1	Client-Website (miniYouTube)	2	5	2
Ü2	Server mit node.js	2	10	5
Ü3	API mit node.js	2	10	5
Ü4	REST-API mit node.js	3	15	5
Ü5	mongoDB	2	10	5
Ü6	Backbone.js	2	10	5

Die Angaben in den Tabellen sind vorläufig (Änderungen möglich)

Übungen: Ausgabe, Bearbeitung und Abgabe

Ausgabe

- (spätestens) Mittwochs nach Zug2-Unterricht in Moodle als PDF

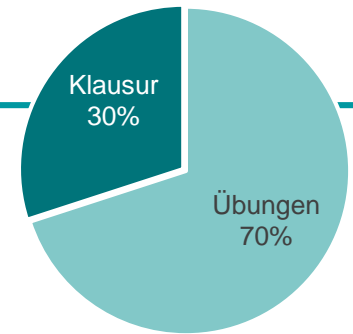
Bearbeitung in Gruppen

- 2-3 Personen
- **Bis Freitag abend**, 15.04.2016: In Moodle selbst Gruppe anlegen oder beitreten. Wer keine Gruppe hat, sendet mir eine E-Mail und wird zugeteilt.

Abgabe

- Upload als ZIP-Archiv in Moodle mit Namensgebung
Ü[Nr]__Nachnamen__Matrikelnummern.ZIP
Beispiel: Ü2__Mueller_Meier__12345678__87654321.ZIP
- Persönliche Demonstration und Erklärung in der Übung
 - durch alle Gruppenmitglieder
 - Jeder kann die Abgabe erläutern, sonst keine Punkte

Benotung - Klausur



- Die Klausur dauert 90min
- Es gibt maximal 100 Punkte (90 Punkte reichen für die 1.0)
- Die Klausur gilt als bestanden, wenn mind. 45% der Punkte erreicht wurden
- Sie können an den Punkten je Aufgabe ablesen, wie viel Zeit ca. veranschlagt ist.
- Kein auswendig gelernter Code zu schreiben, aber ggf. Code-Teile korrigieren
- Bei Multiple-Choice wird angegeben, wie viele Antworten richtig sind. Pro MC-Aufgabe kann es kein negatives Ergebnis geben.
- Für das Bestehen von ME2 müssen Übung und Klausur bestanden werden
- **(Rücktritt? Nicht vergessen bis 19.04.)**

K-Note	ab einschl. (Punkte)	
1,0	90	90%
1,3	85	85%
1,7	80	80%
2,0	75	75%
2,3	70	70%
2,7	65	65%
3,0	60	60%
3,3	55	55%
3,7	50	50%
4,0	45	45%
5,0	<45	

Die Angaben in der Tabelle sind vorläufig
(Änderungen möglich)

Überblick und Start

- Wer bin ich?
- Wer sind Sie?
- (Lern)ziele von MME 2
- Organisatorisches
 - Zeitplan und Stundenplan
 - Benotung (Übungen und Klausur)

Kommunikation

- Vorwissen
- Crashkurs
 - HTML5
 - CSS3
 - JavaScript (bzw. ECMAScript 5)
- Übungsblatt 1 und Zusammenfassung

Kommunikation

■ Sie als Lerngemeinschaft untereinander

- Forum in Moodle
- ..

■ Studierende -> Dozent

- Forum in Moodle !
- Email od. Sprechstunde
 - bei vertraulichen Infos (Attest, ..)
 - Terminkonflikten, Gruppenwechsel, etc.
- Sprechstunde: nach Vereinbarung. Üblich: Mittwoch 15:00-16:00 Uhr, Raum: 218, unbedingt vorab per Email abstimmen

■ Dozent -> Studierende

- Moodle Nachrichtenforum: für alles Organisatorische, Downloads, etc.
 - **Spätestens Mittwoch nachmittag die Übungsaufgaben in Moodle als PDF**
- **Lehrkraftnews** Moodle für Klausurankündigungen und offizielle Meldungen



<https://lms.beuth-hochschule.de/moodle/course/view.php?id=8190>

Einschreibeschlüssel:
me2

Überblick und Start

- Wer bin ich?
- Wer sind Sie?
- (Lern)ziele von MME 2
- Organisatorisches
 - Zeitplan und Stundenplan
 - Benotung (Übungen und Klausur)
 - Kommunikation

Vorwissen

- Crashkurs
 - HTML5
 - CSS3
 - JavaScript (bzw. ECMAScript 5)
- Übungsblatt 1 und Zusammenfassung

Was Sie mitbringen sollten - Vorwissen

Software Engineering

Web Tech. Client
(HTML, CSS etc.)

JavaScript

Zustimmung zu und Einhaltung
der **Unterrichts-Prinzipien**

Was Sie mitbringen sollten – MME2

- Konstruktive und lebendige Lernatmosphäre
- Das Beste aus der Zeit rausholen

Ich

1.) Sorgfältig vorbereiten auf jeden Unterricht und Übung

2.) Fragen häufig stellen und passend beantworten

3.) Pünktlich sein

4.) Ihnen sagen, wenn etwas falsch ist

Sie

1.) Sorgfältig vorbereiten auf jeden Unterricht und Übung

2.) Fragen häufig beantworten und passend stellen

3.) Pünktlich sein

4.) Mir sagen, wenn etwas falsch ist

Was Sie mitbringen sollten - Zeitinvestment

MME2

- 4 SWS (5CP)
- Lehre Zeit: 16 Wochen x 4SWS x 45min = 48h
- Gleiche Zeit zur Vor/Nachbereitung: 3h je Woche (48h)



Hardware

- Einen Laptop oder Smartphone für Internetzugriff
- Einen Block und Stift

Überblick und Start

- Wer bin ich?
- Wer sind Sie?
- (Lern)ziele von MME 2
- Organisatorisches
 - Zeitplan und Stundenplan
 - Benotung (Übungen und Klausur)
 - Kommunikation
 - Vorwissen
- **Crashkurs**
 - HTML5
 - CSS3
 - JavaScript (bzw. ECMAScript 5)
- Übungsblatt 1 und Zusammenfassung

HTML5, CSS3, ECMAScript5: Zuständigkeiten

- Ordnen Sie die folgenden Elemente den 3 rechts zu.
 - 1) **Alleine** in Stillarbeit durchgehen (2min)
 - 2) Mit 2-3 Nachbarn diskutieren und begründen (2min)
- Unklare Fälle klären wir anschließend mit allen

Asynchrones Laden	Layout
Responsive Design	Video-Einbindung
Animation	Canvas
Semantik	Interaktionssteuerung



HTML5

CSS3

ECMAScript5

HTML5, CSS3, ECMAScript5: Zuständigkeiten

- Ordnen Sie die folgenden Elemente den 3 rechts zu.
 - 1) **Alleine** in Stillarbeit durchgehen (2min)
 - 2) Mit 2-3 Nachbarn diskutieren und begründen (2min)
- Unklare Fälle klären wir anschließend mit allen

HTML5

Semantik

Canvas

Video-Einbindung

CSS3

Responsive Design

Layout

Animation

Interaktionssteuerung

ECMAScript5

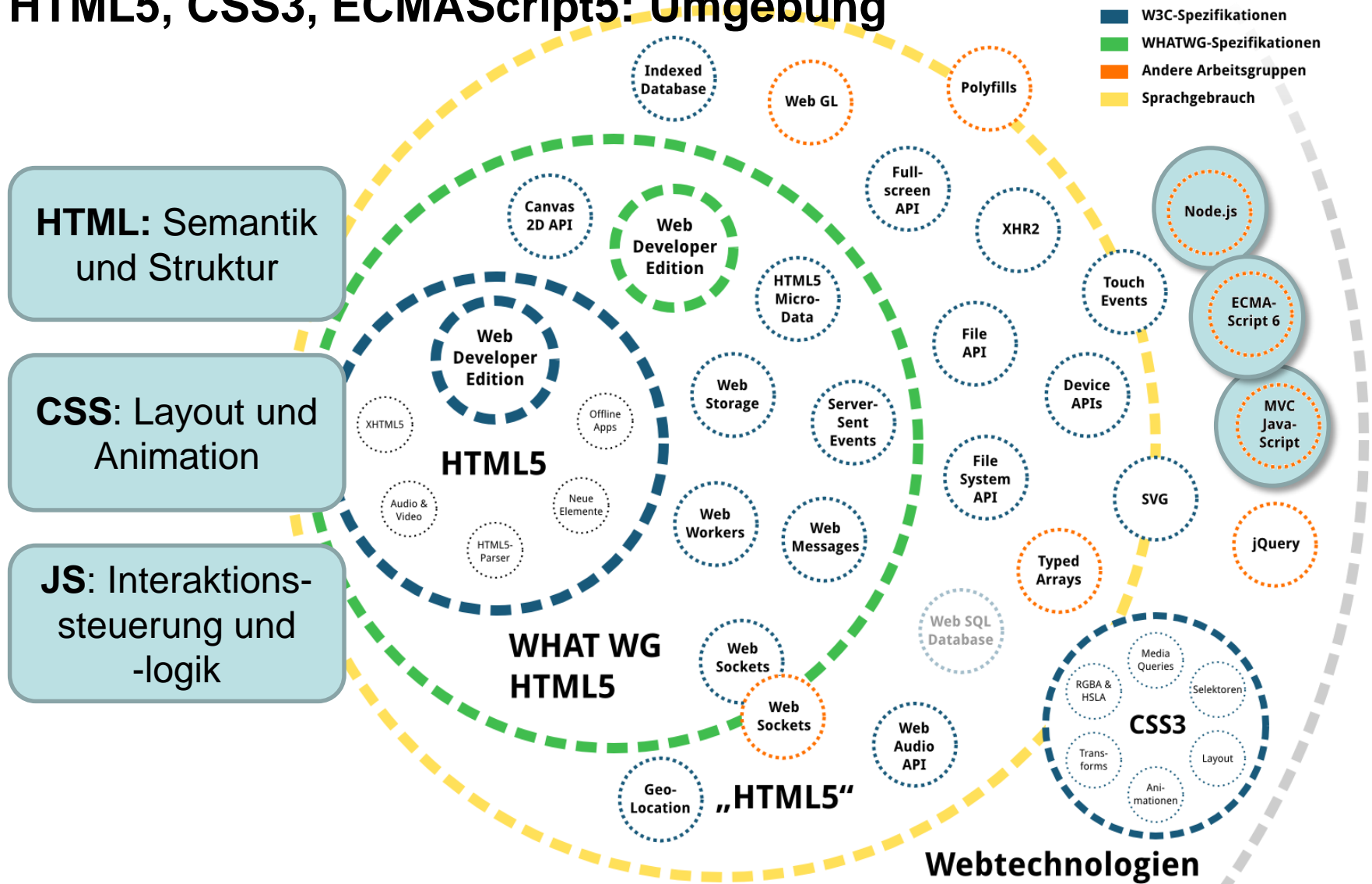
Interaktionssteuerung

Asynchrones Laden

Canvas-Steuerung

Animation

HTML5, CSS3, ECMAScript5: Umgebung

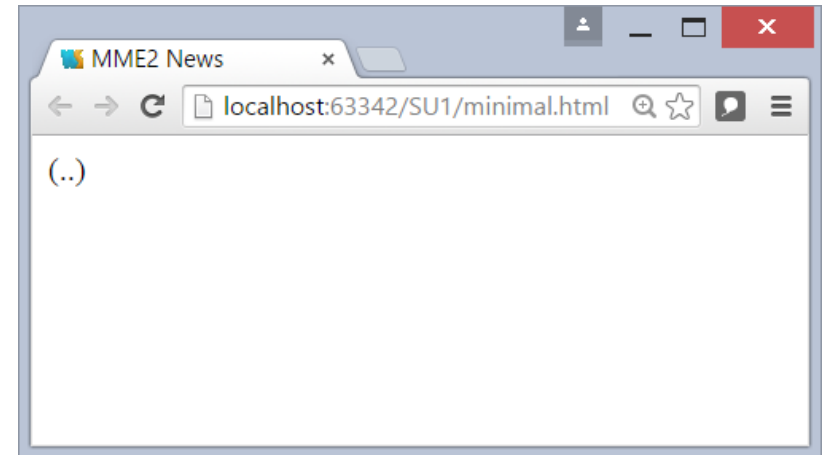


HTML5 Tags und Struktur

```
<!DOCTYPE html>
<html>
<head lang="de">
  <meta charset="utf-8">
  <title>MME2 News</title>
  <link href="base.css" rel="stylesheet" type="text/css">
  <script src="app.js"></script>
</head>
<body>

(..)

</body>
</html>
```



HTML5

■ Welches Tag wofür?

- Bestimmen Sie in Teams (2-3)
für die Tags einen Verwendungszweck (3min)
- Achtung: Einige Tags sind nicht zu verwenden



1. `<td>`
2. `<a>`
3. `<aside>`
4. ``
5. `<body>`
6. `<c>`
7. `<center>`
8. `<canvas>`
9. `<div>`
10. `<input>`
11. `<main>`
12. `<nav>`
13. `<output>`
14. `<section>`
15. ``
16. `<video>`

HTML5






1. `<td>`
2. `<a>`
3. `<aside>`
4. ~~``~~
5. `<body>`
6. ~~`<e>`~~
7. ~~`<center>`~~
8. `<canvas>`
9. `<div>`
10. `<input>`
11. `<main>`
12. `<nav>`
13. `<output>`
14. `<section>`
15. ``
16. `<video>`

1. Tabellenzelle
2. Verlinkung (Hypertext)
3. Zusatzinformation mit Bezug zur Umgebung
4. Ungültig! (früher mal für Fettdruck)
5. Definiert den sichtbaren Dokumentbereich im Browser
6. Ungültig!
7. Ungültig! (früher mal für Zentrierung)
8. Zeichenfläche
9. Blockelement
10. Eingabeelement
11. Hauptteil des Dokumentes (Einzigartig pro URL)
12. Navigationselemente der Seite/App
13. Zur Ausgabe von Berechnungen in `<form>` Elementen
14. Abschnitt der Seite (geschachtelt, ggf. mit `<article>`)
15. Inline-Element
16. Video-Einbettung (optional mit Steuerelementen)

HTML5 Tags und Struktur

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="utf-8">
  <meta name="description" content="..">
  <meta name="keywords" content="MME2">
  <meta name="author" content="J.Konert">
  <title>MME2 News</title>
  <link href="base.css" rel="stylesheet" type="text/css">
  <!--[if lt IE 9]>
  <script
src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
  <![endif]-->
  <script src="app.js"></script>
</head>
<body>
<nav>
  <a href="#home">Willkommen</a>
  <a href="#news">Neuste Meldungen</a>
  <a href="#imprint">Impressum</a>
</nav>
<section id="home"><h1>Willkommen</h1> (..)</section>
<main>
  <section id="news">
    <article>(..)</article>
  </section>
</main>
(..)
</body>
</html>
```

HTML5 Video - Kompatibilität (erst seit 2015!)

					
H.264					
OggTheora					
WebM					

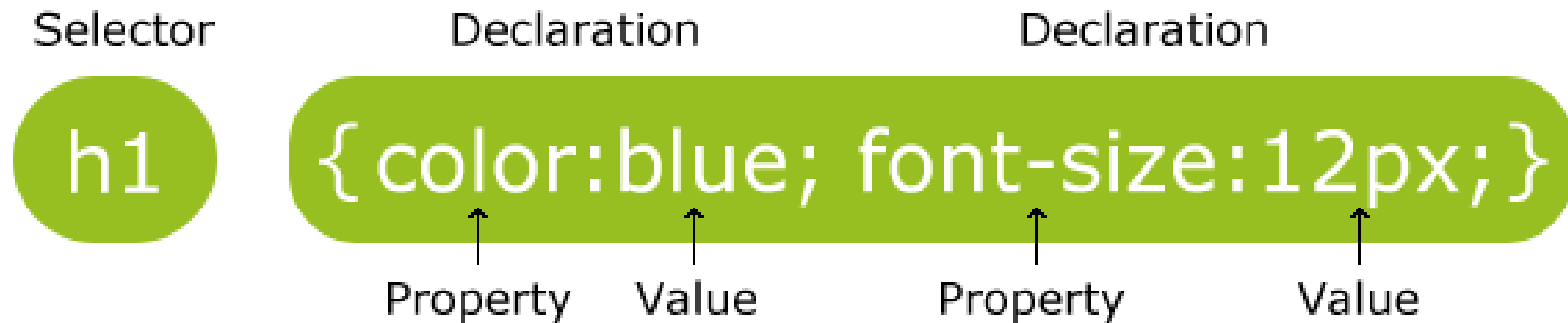
HTML5 Video

```
<!DOCTYPE html>
<html>
<head lang="de">
  <meta charset="utf-8">
  <title>MME2 Video Example</title>
  <link href="base.css" rel="stylesheet" type="text/css">
  <script src="app.js"></script>
</head>
<body>
(..)
<main>
  <section id="topvideo">
    <video controls>
      <source src="http://.../small.mp4" type="video/mp4">
    </video>
  </section>
</main>
(..)
</body>
</html>
```

Überblick und Start

- Wer bin ich?
- Wer sind Sie?
- (Lern)ziele von MME 2
- Organisatorisches
 - Zeitplan und Stundenplan
 - Benotung (Übungen und Klausur)
 - Kommunikation
 - Vorwissen
- Crashkurs
 - HTML5
 - CSS3**
 - JavaScript (bzw. ECMAScript 5)
- Übungsblatt 1 und Zusammenfassung

CSS3



- Selektor: welche DOM-Elemente geändert werden
- Deklaration: wie geändert werden soll

CSS3 Selektor-Varianten

- Es geht fast alles. Vieles (zu kompliziertes) ist nicht sinnvoll.
- Komplette Vielfalt im Web http://www.w3schools.com/cssref/css_selectors.asp

Einfachselektoren

Syntax	Beispiel	Wirkung
*	* { }	Alle Elemente“
.Klasse	.intro	Alle Elemente mit class=„intro“
#id	#navi	Das Element mit der id=„navi“
Element	h1	Alle Überschriften <h1>
Selektor, Selektor	h1, h2	Sowohl alle Überschriften <h1> als auch <h2>
[Attribut]	[target]	Alle Elemente die ein Attribut <...target=„..“> haben
[Attribut*="Wert"]	[title*="Beispiel"]	Alle Elemente, deren title die Zeichen „Beispiel“ enthält

Kombination:

```
div#navi a.beuth-intern[href*="fb6."] {font-weight: bold}
```

CSS3 Selektor-Varianten

- Es geht fast alles. Vieles (zu kompliziertes) ist nicht sinnvoll.
- Komplette Vielfalt im Web http://www.w3schools.com/cssref/css_selectors.asp

Kombinationsselektoren

Syntax	Beispiel	Wirkung
Selektor Selektor	main p	Alle Paragraphen die (beliebig tief) als Kindelemente des <main>-Tags vorkommen
Selektor > Selektor	ol > li	Alle Listenelemente die direkte (!) Kindelemente einer sortierten Liste sind
Selektor + Selektor	h1 + p	Jeder Absatz der direkt(!) als nachfolgender Knoten nach einer H1 Überschrift kommt

CSS3 Selektor-Varianten

- Es geht fast alles. Vieles (zu kompliziertes) ist nicht sinnvoll.
- Komplette Vielfalt im Web http://www.w3schools.com/cssref/css_selectors.asp

Pseudoklassenselektoren

Syntax	Beispiel	Wirkung
:active	a:active	Aktuell angeklickter Link
:hover	a:hover	Aktueller Link mit mouseover
:focus	input:focus	Aktuelles Eingabefeld mit Fokus (Tab-Steuerung)
:target	a:target	Element, welches das Ziel des Links ist (Nützlich bei)
:disabled	input:disabled	Deaktivierte Eingabeelemente
:required	form *:required	Alle Elemente innerhalb einer Form die Attribut „required“ gesetzt haben.
:not (Selektor)	a:not (.beuth-intern)	Alle Links, die nicht die Klasse beuth-intern haben

CSS3 Selektor-Varianten

- Es geht fast alles. Vieles (zu kompliziertes) ist nicht sinnvoll.
- Komplette Vielfalt im Web http://www.w3schools.com/cssref/css_selectors.asp

Pseudoelementselektoren (Elemente, die nicht im HTML vorhanden sind)

Syntax	Beispiel	Wirkung
<code>::before</code> <code>::after</code>	<code>h1::before {content: url(bild.png) }</code>	Setzt vor jede Überschrift H1 das bild.png davor
<code>::first-letter</code>	<code>p::first-letter</code>	Der erste Buchstabe eines jeden Paragraphen

CSS3

```
h1 { font-family: sans-serif }  
h2 { font-family: sans-serif }  
h3 { font-family: sans-serif }
```



Diese zwei Schreibweisen sind äquivalent

```
h1, h2, h3 { font-family: sans-serif }
```

■ Allgemeine Form:

```
Selektor [, Selektor2, ...] {  
    Eigenschaft1: Wert1;  
    ... Eigenschaftn: Wertn[;]  
}  
/* Kommentar */  
/* In eckigen Klammern stehen optionale Angaben */
```

CSS3 Kaskade

- Treffen auf ein HTML-Element mehrere Selektoren zu, dann werden alle ausgeführt.
- Bestimmte, kaskadierende Reihenfolge, damit es eindeutig ist
 1. Selektor-Spezifität (wie genau passt es)
gibt es mehrere, dann:
 2. Nähe zum Element (wo ist es definiert)

1. Selektor-Spezifität

```
main h1[class*=„long“] { color: red }
```

ist spezifischer als 

```
main h1 { color: blue }
```

ist spezifischer als 

```
h1 { color: green; font-size: 2em }
```

- **Achtung:** Schriftgröße 2em bleibt beim Zusammenführen auch bestehen

CSS3 Kaskade

- Treffen auf ein HTML-Element mehrere Selektoren zu, dann werden alle ausgeführt.
- Bestimmte, kaskadierende Reihenfolge, damit es eindeutig ist
 1. Selektor-Spezifität (wie genau passt es)
gibt es mehrere, dann:
 2. Nähe zum Element (wo ist es definiert)

1. Selektor-Spezifität (Allgemein)

#id

ist spezifischer als 

[Attribut] (bspw. .Klasse)




ist spezifischer als 

Element

CSS3 Kaskade

- Treffen auf ein HTML-Element mehrere Selektoren zu, dann werden alle ausgeführt.
- Bestimmte, kaskadierende Reihenfolge, damit es eindeutig ist
 1. Selektor-Spezifität (wie genau passt es)
gibt es mehrere, dann:
 2. Nähe zum Element (wo ist es definiert)

2. Nähe zum Element

- a. **Inline Style:** Direkt am Element schlägt alles (wegen Spezifität!)
 ist näher als
- b. **Internal Style:** Im <head> eingebundener Stil
 ist näher als
- c. **External Style:** In extra Datei <link ..> xyz.css definierter Stil
 ist näher als
- d. **Browser Style:** Vom Browser(hersteller) vordefinierter Stil (schwächster)
(gibt es mehrere vom gleicher Nähe, dann zählt die Ladereihenfolge im Dokument)

CSS3 Kaskade - Beispiel

■ Welche Farbe hat die Überschrift des Artikels?

```
<head lang="de">
  <meta charset="UTF-8">
  <title>MME2 CSS3 Beispiele</title>
  <link href="base.css" rel="stylesheet" type="text/css">
  <style type="text/css">
    #first-head {color: brown}
    body main article h1 {color: blue}
    h1           {color: yellow}
    .art-head    {color: red}
  </style>
</head>
<body>
<main>
  <article id="article" class="first-art">
    <h1 id="first-head" class="art-head" style="color: pink">Newsmeldung...</h1>
  </article>
</main>
</body>
```

base.css

```
main h1 {color: aliceblue}
#first-head {color: purple}
```



CSS3 Kaskade - Beispiel

■ Welche Farbe hat die Überschrift des Artikels?

```
<head lang="de">
  <meta charset="UTF-8">
  <title>MME2 CSS3 Beispiele</title>
  <link href="base.css" rel="stylesheet" type="text/css">
  <style type="text/css">
    #first-head {color: brown}
    body main article h1 {color:blue}
    h1           {color: yellow}
    .art-head    {color: red}
  </style>
</head>
<body>
<main>
  <article id="article" class="first-art">
    <h1 id="first-head" class="art-head" style="color:pink">Newsmeldung..</h1>
  </article>
</main>
</body>
```

base.css

```
main h1 {color: aliceblue}
#first-head {color: purple}
```



Lösung (Reihenfolge der Prio): pink, brown, purple, red, blue, aliceblue, yellow, (black)

Überblick und Start

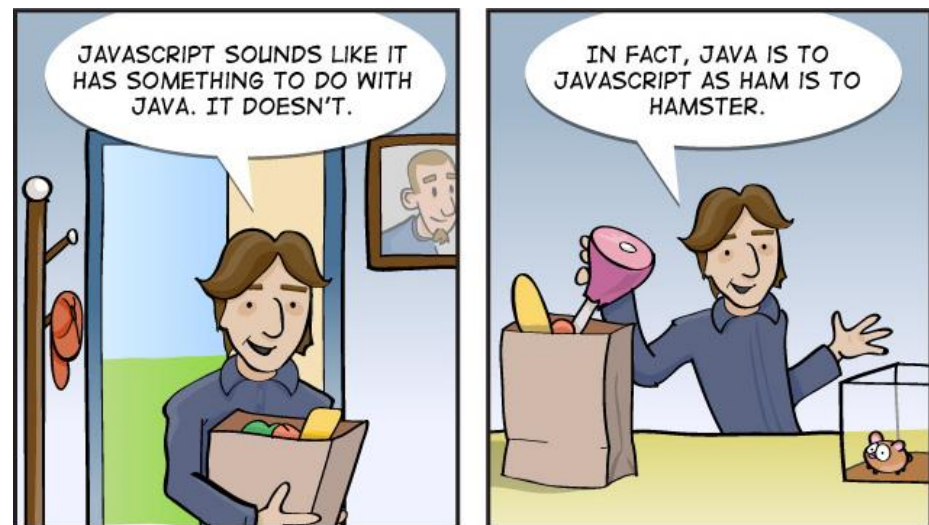
- Wer bin ich?
 - Wer sind Sie?
 - (Lern)ziele von MME 2
 - Organisatorisches
 - Zeitplan und Stundenplan
 - Benotung (Übungen und Klausur)
 - Kommunikation
 - Vorwissen
 - Crashkurs
 - HTML5
 - CSS3
- JavaScript (bzw. ECMAScript 5)**
- Übungsblatt 1 und Zusammenfassung

JavaScript bzw. ECMAScript 5

- **funktional:** mit Funktionen als First-Class-Citizens, Closures, ...
- **objektorientiert:** aber nicht mittels Klassen, sondern mittels Prototypen.
- **prinzipiell unstrukturiert:** JavaScript gibt keine Strukturen vor; diese müssen mittels Patterns und Disziplin vom Entwickler geschaffen werden.
- **kompromisslos dynamisch:** Objekte können zur Laufzeit um Methoden und Attribute erweitert werden, Quellcode kann zur Laufzeit hinzugefügt werden, ...

Insbesondere:

- Keine statische Typisierung
- Keine Sichtbarkeiten
- Kein Multi-Threading



JavaScript und DOM-Anbindung

- Browser bieten JavaScript Zugriff auf das Document Object Model (DOM) = Events und objektorientierte Struktur des HTML-Dokumentes.

index.html

```
<head>
  ...
  <script type="text/javascript" src="js/app.js"></script>
</head>
<body>
  ...
  <video id="video1">
    <source src="http://.../small.mp4" type="video/mp4">
  </video>
  <button id="play1" onclick='play("video1", "play1") '>Play</button>
  ...
</body>
```

app.js

```
function play(videoId, playId) {
  var video = document.getElementById(videoId);
  video.play(); }
```

JavaScript und DOM-Anbindung

- Browser bieten JavaScript Zugriff auf das Document Object Model (DOM) = Events und objektorientierte Struktur des HTML-Dokumentes.

Kein guter Stil:

- ✗ JavaScript direkt an die HTML-Elemente schreiben
- ✗ Globale Funktionen in JavaScript-Dateien definieren

JavaScript und DOM-Anbindung

index.html

```
<head> ...  
  <script type="text/javascript" src="js/app.js"></script>  
</head>  
<body>...  
  <video>  
    <source src="http://.../small.mp4" type="video/mp4">  
  </video>  
  <button name="play" title="Abspielen"></button>
```

app.js

```
1: window.addEventListener("load", function(event) {  
2:   var videos = document.getElementsByTagName("video");  
3:   for (var i = 0; i < videos.length; i++) {  
4:     var video = videos[i];  
5:     // ... find button objects and add listener ...  
6:     playButton.addEventListener("click", function (event) {  
7:       video.play();  
8:     }); // ...  
9:   }  
10: });
```

Achtung: Für die Übung 1 müssen Sie noch das Problem lösen, dass var video immer auf das letzte Video zeigt.

Überblick und Start

- Wer bin ich?
- Wer sind Sie?
- (Lern)ziele von MME 2
- Organisatorisches
 - Zeitplan und Stundenplan
 - Benotung (Übungen und Klausur)
 - Kommunikation
 - Vorwissen
- Crashkurs
 - HTML5
 - CSS3
 - JavaScript (bzw. ECMAScript 5)
- **Übungsblatt 1 und Zusammenfassung**

Übungsblatt 1: Client (miniYouTube)

- In Moodle zum Download

Aufgaben

- Umfrage in Moodle ausfüllen
- HTML-Seite mit Video-Einbindung erstellen
- CSS und JavaScript einbinden (**extern!**)
- Ergebnis wird gebraucht für spätere Übung.

Zusammenfassung

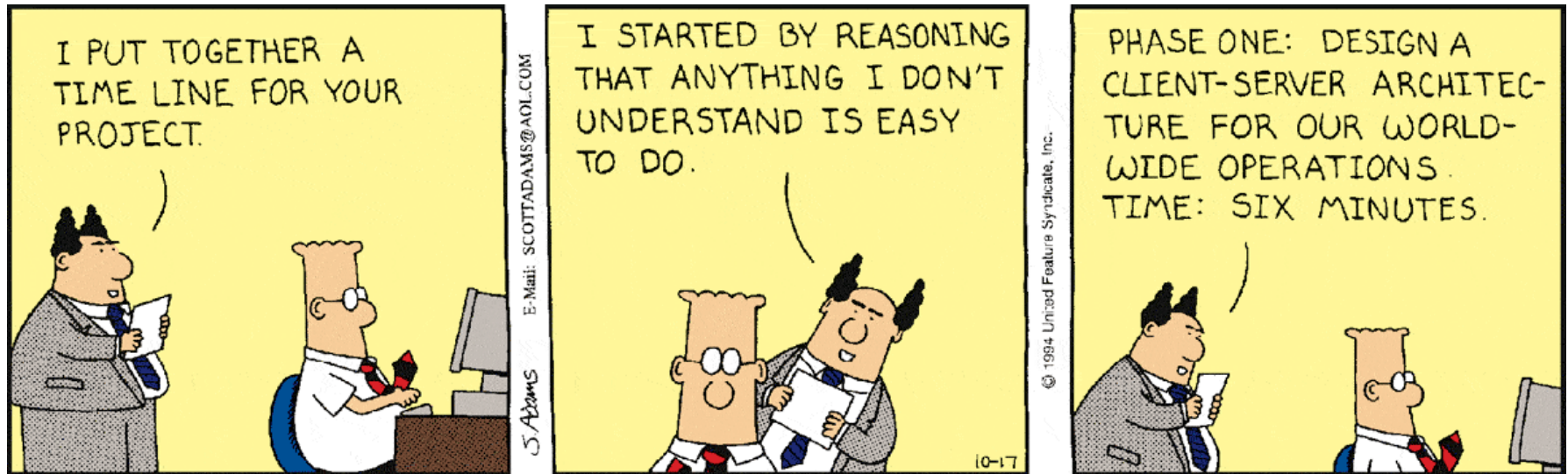
- ME2 besteht aus 70% Übungsnote (6 Übungen) und 30% Klausur
- Vorausgesetzt werden Kenntnisse in Softwaretechnik, HTML/CSS und insbesondere ECMAScript5

Bearbeitung der Übung in Gruppen

- 2-3 Personen
- **Bis Freitag abend**, 15.04.2015 in Moodle selbst Ihre Übungsgruppen anlegen, ggf. PW weitergeben.
(oder mir eine Email schicken, ich teile dann zu).

Nächste Woche

■ Client-Server Architekturen (Stacks)



**Vielen Dank
und
bis zum nächsten Mal**