

1

Übung Web- und Multimedia-Engineering

Organisatorisches und Allgemeines
Aufgabenstellung A1 und thematische Einführung

1. Organisatorisches und Allgemeines
 - Website, Zugang und Kontakt
 - Ablauf und Abgabe
 - Überblick Aufgabenstellungen
 - Terminplan
 - Teambildung und Vorbereitung
2. Aufgabenstellung A1: HTML5, CSS3 und JS
3. Grundlagen Webentwicklung
 - HTML
 - CSS
 - JS
4. Hilfreiches, Tipps und Links

Teil 1

Organisatorisches und Allgemeines

Die Übung zur Vorlesung

■ Zentrale Webseite für die Übung

https://imld.de/study/teaching/ws_19-20/wme_19-20/ueb-wme_19-20/

- Allgemeine Informationen, Neuigkeiten und Ankündigungen, Übungsaufgaben, Materialien, Übungsergebnisse

■ Ansprechpartner und Betreuung

- Patrick Reipschläger

Büro: APB/2047

Kontakt: patrick.reipschlaeger@tu-dresden.de

- Robert Fuhrmann


Kontakt: robert.fuhrmann@mailbox.tu-dresden.de

ZUGANGSDATEN

Benutzername: mtstudent

Passwort: WME1920

Die Übung zur Vorlesung

- Versteht sich als **Ergänzung** zur Vorlesung
 - Keinesfalls Ersatz, nur zur Übung gehen = „keine gute Idee“
 - Kann nur einige Teilen des Vorlesungsinhalts abdecken
 - Nicht synchron zur Vorlesung (Format-bedingt sehr schwer )
- Verständnis durch **praktische** Aufgaben- und Problemstellungen
- Unterstützung und Hilfestellungen während des Selbststudiums
- Austausch durch Gruppen-/Teamarbeit
- Qualifikation bzw. Zulassung zur Klausur
 - Mindestens 50% der Übungspunkte

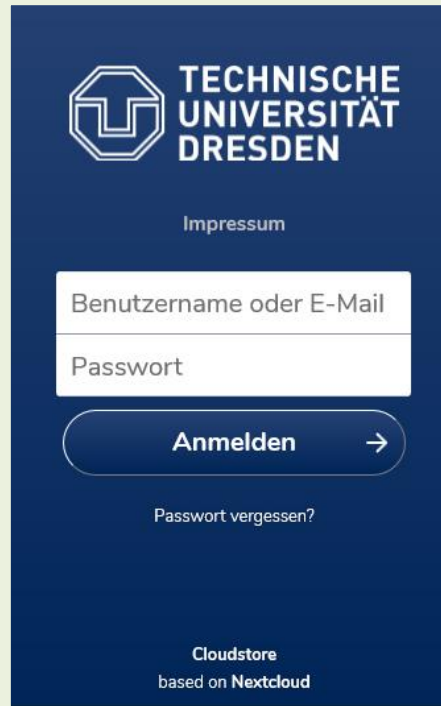
Ablauf der Übung

- Drei Übungstermine je Woche:
 - Montags 3.DS in APB/E067
 - Dienstags 4.DS und Mittwochs 4. DS in APB/E065
- Teamarbeit während des gesamten Semesters möglich
 - Zwei Studierende je Team (Studiengang, Übungstermin, o.ä. egal)
 - Teams bleiben über alle Aufgaben hinweg bestehen
 - Gleiche Bewertung für alle Teammitglieder
 - Unbedingt die Teamnummer aufschreiben
- Übungsbewertung basiert auf
 - Erfüllung konkreter Teilaufgaben/Anforderungen
 - Quellcode, Kommentare, Fehler, Aufbau und Struktur, Grafik und Design

Ablauf der Übung

- Abgabe der Ergebnisse
 - Immer Montags 9.20 Uhr über Cloudstore der TU Dresden
 - Hochladen als gepacktes ZIP-Archiv
 - Jedes Team erstellt ein **ZIP**-Archiv mit Lösung (keine anderen Formate zulässig)
 - Auf Benennung achten: `Aufgabe<NR>_Team<T_NR>.zip`
 - <NR> Nummer der Aufgabe (Übung), 1-2-3-4
 - <T_NR> Teamnummer, 1-2-3-n
 - Ergebnisabgabe umfasst immer
 - alle Quell-Dokumente, eine README.md mit Teamnummer und Namen
 - Denken Sie auch an Materialien, wie z.B. Bilder, Datensätze
 - Nachreichungen werden nicht akzeptiert!
 - 0 Punkte bei verpasster Abgabe
 - Abgabe verpassen ist „keine gute Idee“, es gibt nur 4 Aufgaben

How-To Cloudstore (=> nur 1x pro Team)



TECHNISCHE
UNIVERSITÄT
DRESDEN

Impressum

Benutzername oder E-Mail

Passwort

Anmelden →

Passwort vergessen?

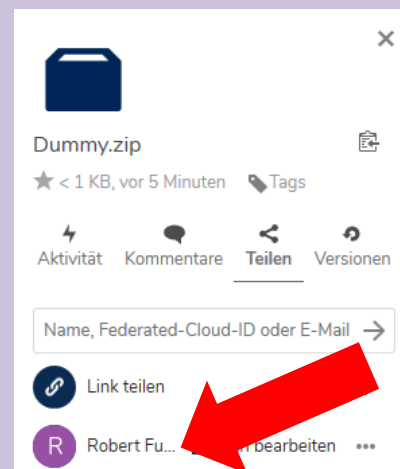
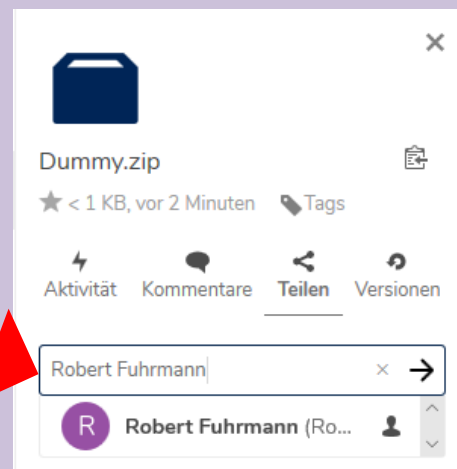
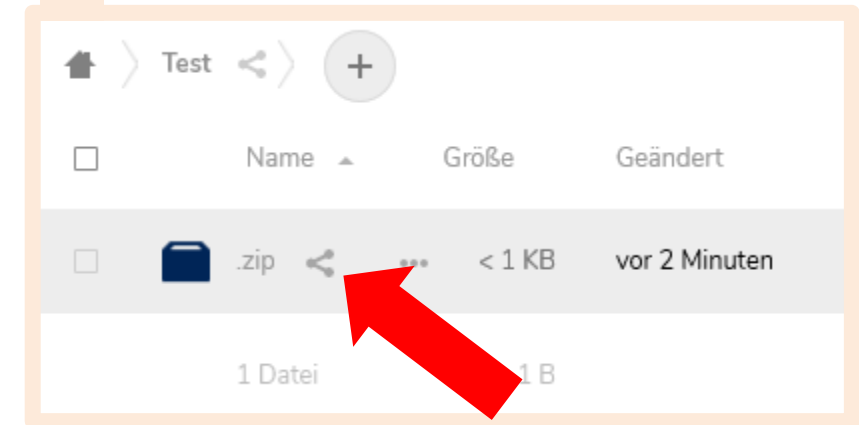
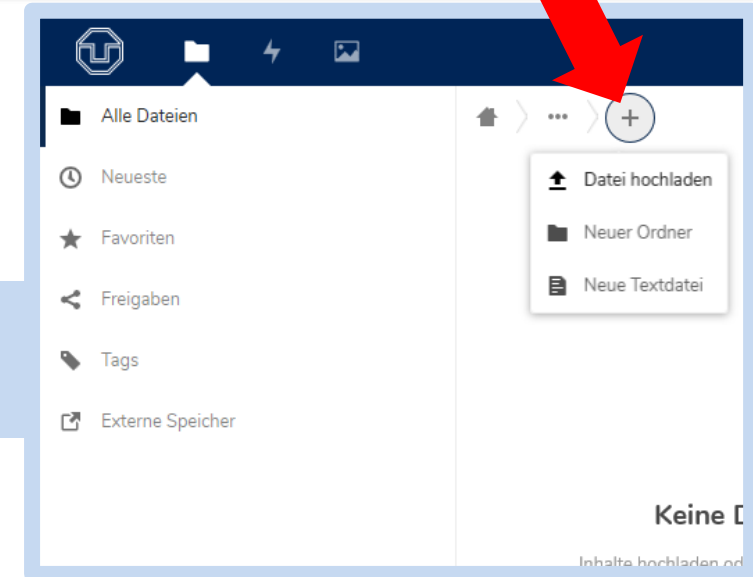
Cloudstore
based on Nextcloud

1. <https://cloudstore.zih.tu-dresden.de/>
aufrufen, mit ZIH Login (s-Nummer)
anmelden

2. Zip Archiv hochladen
(ggf. neuen Ordner erstellen)

3. Auf das Share Icon klicken

4. Patrick Reipschläger und Robert
Fuhrmann suchen und auswählen




Ablauf der Übung

- Präsentation der Musterlösung
 - Optional immer eine Woche nach der Abgabe
 - Je nach Bedarf und Interesse: Vorbereiten (Fragen/Probleme)
- Diskussion und Austausch
 - Forum im OPAL nutzen

Ablauf der Übung

- Ein Themenkomplex, fünf Aufgabenstellungen
 - Jeweils unterschiedlich lange Bearbeitungszeit
 - **A1:** Grundlagen client-seitige Technologien: HTML5, CSS3 & JS
 - Grundgerüst für eine Website als Interface für world_data
 - **A2:** Grundlagen server-seitige Technologien: XML und PHP
 - CSV sowie XML Transformation, Grundlagen PHP Serverkomponente
 - **A3:** Erweiterung server-seitige Technologien: Node.js & AJAX
 - Erstellung eines REST-Services, Abfragen durch den Client via AJAX
 - **A4:** Anwendungsfall – Visualisierung mit D3.js & Leaflet
 - Visualisierung von Daten mittels interaktiver Bar Charts und Karten

ZEITPLAN

Woche	Datum	Übungsthema /-inhalt	Materialien
KW 42	14/15/16.10.	keine Übung (erste Vorlesungswoche)	
KW 43	21/22/23.10.	Einführung Aufgabenstellung 1 „HTML + CSS + JS“ (WME-A1) 	<ul style="list-style-type: none"> ■ Screenshots ■ Logo ■ Daten (CSV)
KW 44	28/29/30.10.	Konsultation, freies Arbeiten	
KW 45	04/05/06.11.	Konsultation, freies Arbeiten	
KW 46	11.11., 09:20	Abgabe WME-A1 (bis 9:20)	
KW 46	11/12/13.11.	Einführung Aufgabenstellung 2 „PHP + XML“ (WME-A2)	
KW 47	18/19/20.11.	Konsultation, freies Arbeiten	
KW 48	25/26/27.11.	Konsultation, freies Arbeiten	
KW 49	02.12., 9:20	Abgabe WME-A2 (bis 9:20)	
KW 49	02/03/04.12.	Einführung Aufgabenstellung 3 „Node.js + REST + AJAX“ (WME-A3)	
KW 50	09/10/11.12.	Konsultation, freies Arbeiten	
KW 51	16/17/18.12.	Konsultation, freies Arbeiten	
KW 52	23/24/25.12.	keine Übung (Weihnachten)	
KW 1	30/31.12. /01.01.	keine Übung (Jahreswechsel)	
KW 2	06.01., 9:20	Abgabe WME-A3 und WME-A4 (bis 9:20)	
KW 2	06/07/08.01.	Einführung Aufgabenstellung 4 „SVG + D3 + Leaflet“ (WME-A4)	
KW 3	13/14/15.01.	Konsultation, freies Arbeiten	
KW 4	20/21/22.01.	Konsultation, freies Arbeiten	
KW 5	27.01., 9:20	Abgabe WME-A4 (bis 9:20)	
KW 5	27/28/29.01.	Abschluss, Feedback und Fragen	
KW 6	03/04/05.02.	keine Übung (selbstständige Prüfungsvorbereitung)	
Woche	Datum	Übungsthema /-inhalt	Materialien

(unter Vorbehalt, Änderungen möglich)

Teambildung

(je 2 Studierende)


Notieren Sie ihre Teamnummer 

Teil 2

Aufgabenstellung A1

HTML5, CSS3 und JS


Aufgabenstellung A1

- Einfaches Webinterface zur Anzeige der Daten, fünf Komponenten:
 - Navigationsleiste (Menü im Header), einfache Text-Box, Tabellenansicht der Daten, Anzeigoptionen, Fußbereich (Footer)
- Statische Webseite ohne spezielle Funktionen: Grundgerüst
- Dummy-Daten, grundlegend übertragen (oder einlesen)
- Testen: Firefox (aktuelle Version)
- Abgabe: Montag, 11.11. um 9.20 Uhr 

Aufgabenstellung A1

■ Erstellen (Rekonstruieren) Sie das folgende Webinterface

- Materialien: Screenshots, Logo, Daten (CSV-Datei)

 world_data

A1 - TableA2 - ParseA2 - SaveA2 - PrintA3 - RESTA4 - Vis

World Data Overview ...

Mavericks reality distortion gradients attenuation. Thought through and through notifications transparency game center multitasking aluminum advanced desktop operating system genius bar. This changes everything designed by Apple in California passbook. Control center photos all-new design SDK technology clock. Simplicity is actually quite complicated. Functional layers 9:41am partly cloudy minimalism. Dock airdrop slide to answer music. Video multitouch iTunes compass. Harmonious finder grid system retina animation compressor experience keynote.Redesign services API notes system preferences. Features siri flat buttons airplane mode calculator. Missed call cover flow compare models. Wi-Fi apple care volume reminder controls. My stations folders mac power ultimate upgrade. Shop online quicktime trackpad server aperture rumors education safari one to one. Remote desktop motion business. Backlit keyboard chess phone airport extreme support iPad. Accessories magsafe terminal final cut pro. Featured TV shows downloads digital color meter. Glossy tech specs bluetooth manuals. OpenGL products FaceTime free shipping recycling mission control applications. from: [jony_ipsum](#)

Show/Hide: birth rate | cellphones | children / woman | electric usage | internet usage

ID	Country ^ v	birth rate / 1000	cellphones / 100	children / woman	electric usage	internet usage
001	Brazil	16.405	90.01936334	1.862	2,201.808724	39.22
002	Canada	10.625	70.70997244	1.668	15,119.76414	80.17086651
003	Chile	15.04	97.01862561	1.873	3,276.06449	38.8
004	China	13.536	55.97490921	1.642	2,632.724637	28.97659939
005	Colombia	20.605	92.34584564	2.405	1,041.994137	30
006	Ecuador	20.989	92.84925653	2.69	1,078.039961	24.6
007	Egypt	24.83	69.43661504	2.919	1,607.918763	24.28
008	Finland	11.127	144.1530224	1.86	15,241.61194	82.53133098
009	France	12.21	95.44434226	1.978	7,399.946832	69.0633593
010	Germany	8.136	127.4188883	1.376	6,753.05764	79.48523153
011	Iceland	8.136	127.4188883	1.376	6,753.05764	79.48523153
012	Iraq	8.136	127.4188883	1.376	6,753.05764	79.48523153
013	Japan	8.136	127.4188883	1.376	6,753.05764	79.48523153
014	Kazakhstan	8.136	127.4188883	1.376	6,753.05764	79.48523153
015	Mexico	8.136	127.4188883	1.376	6,753.05764	79.48523153
016	New Zealand	8.136	127.4188883	1.376	6,753.05764	79.48523153
017	Nigeria	8.136	127.4188883	1.376	6,753.05764	79.48523153
018	Peru	8.136	127.4188883	1.376	6,753.05764	79.48523153
019	Russia	8.136	127.4188883	1.376	6,753.05764	79.48523153
020	Saudi Arabia	8.136	127.4188883	1.376	6,753.05764	79.48523153
021	South Africa	8.136	127.4188883	1.376	6,753.05764	79.48523153
022	Sweden	8.136	127.4188883	1.376	6,753.05764	79.48523153
023	United Arab Emirates	8.136	127.4188883	1.376	6,753.05764	79.48523153
024	United Kingdom	8.136	127.4188883	1.376	6,753.05764	79.48523153
025	United States	8.136	127.4188883	1.376	6,753.05764	79.48523153

Show/Hide: birth rate | cellphones | children / woman | electric usage | internet usage

Copyright © 2018 world_data
First course exercise HTML, CSS and JS of the lecture Web and Multimedia Engineering.

This solution has been created by:
Team 5

 world_data

A1 - TableA2 - ParseA3 - RESTA4 - Vis


World Data Overview ...

Mavericks reality distortion gradients attenuation. Thought through and through notifications transparency game center multitasking aluminum advanced desktop operating system genius bar. This changes everything designed by Apple in California passbook. Control center photos all-new design SDK technology clock. Simplicity is actually quite complicated. Functional layers 9:41am partly cloudy minimalism. Dock airdrop slide to answer music. Video multitouch iTunes compass. Harmonious finder grid system retina animation compressor experience keynote.Redesign services API notes system preferences. Features siri flat buttons airplane mode calculator. Missed call cover flow compare models. Wi-Fi apple care volume reminder controls. My stations folders mac power ultimate upgrade. Shop online quicktime trackpad server aperture rumors education safari one to one. Remote desktop motion business. Backlit keyboard chess phone airport extreme support iPad. Accessories magsafe terminal final cut pro. Featured TV shows downloads digital color meter. Glossy tech specs bluetooth manuals. OpenGL products FaceTime free shipping recycling mission control applications. from: [jony_ipsum](#)

Show/Hide: birth rate | cellphones | children / woman | electric usage | internet usage

ID	Country ^ v	birth rate / 1000	cellphones / 100
001	Brazil	16.405	90.01936334
002	Canada	10.625	70.70997244
003	Chile	15.04	97.01862561

Aufgabenstellung A1: Bewertungskriterien

- Allgemeine Kriterien und Verwendung
 - ☐ Nur HTML, CSS und JavaScript
 - ☐ **Nur** ausgewählte Bibliotheken, Frameworks o.ä. einbinden:
 - ☐ Font Awesome Free (≥ 5.4), <https://fontawesome.com/free>
 - ☐ CSS-Reset, <https://github.com/murtaugh/HTML5-Reset/blob/master/assets/css/reset.css>
 - ☐ Webfont Roboto, <https://fonts.google.com/specimen/Roboto>
 - ☐ Dateikodierung/Encoding: **UTF-8** (*und Zeilenende Unix-LF*)
 - ☐ Valides HTML5 und CSS3, Validator (<https://validator.w3.org/>) nutzen 
 - ☐ HTML Header-Angaben: mind. title, description, author, keywords
 - ☐ HTML5-Elemente: header, footer, nav
 - ☐ Fußzeile (Footer): Teamnummer
 - ☐ Hinweis: *Readme nicht vergessen*

Aufgabenstellung A1: Bewertungskriterien

■ Design

- ☐ Basisschrift: Roboto, 14px, Zeilenabstand: 145%, Style: Normal
- ☐ Iconschrift "Font Awesome" für Nav und Spaltensortierung
- ☐ Hover für Standard-Links: Farbe und Unterstrich
- ☐ Responsive Design: „mobile-first“ Ansatz mit drei expliziten, sinnvollen Schwellwerten:
 - ☐ ≥ 320 Phones
 - ☐ ≥ 768 Tablets
 - ☐ ≥ 1200 Desktops: feste Container-Breite (Seiten-Wrapper) 1170px, links-rechts-zentriert
- ☐ Responsive Design: auf Basis der o.g. Schwellwerte Anpassung der Spaltenanzahl für Text-Box über der Tabelle (1/2/3-spaltig)

Aufgabenstellung A1: Bewertungskriterien

■ Design

- ☐ Header mit Logo und sechs Menüelemente/Links
 - ☐ Farbgradient im Header (*Hinweis: geht auch ohne Bild*)
 - ☐ „Klickbares“ Logo mit Hover (farblos/grau)
 - ☐ Liste (Unordered List, ul) für Menüelemente
 - ☐ „Font Awesome“ Icon für Menüelemente
 - ☐ Hover der Menüelemente: Kurze Animation für den Wechsel der Hintergrundfarbe im Header (CSS-Transition, **kein** abrupter/sofortiger Farbwechsel), siehe Live-Demo
 - ☐ Bleibt beim Scrollen am oberen Bildschirmrand (Sticky Navigation)

Aufgabenstellung A1: Bewertungskriterien

■ Design

☐ Tabelle für Datenanzeige

- ☐ Elemente thead und tbody einsetzen
- ☐ Mind. folgende Spalten: ID und Country, sowie fünf Attribute
- ☐ Mind. 25 Länder (ID und Country) mit beliebigen Daten (Zufall, CSV, ...)
- ☐ „Show/Hide“ Funktionalität für fünf Attributspalten
- ☐ Sortierung der Tabelle (auf- und absteigend) nach Country, „Font Awesome“ Icons in Spalte Country
- ☐ Wechselnde Hintergrundfarbe der Zeilen (bspw. #d8d8d8 für gerade Zeilen)
- ☐ Responsive Design: Auf Basis der CSS-Schwellwerte (mindestens drei) automatisch die jeweils letzte Spalte ausblenden

■ Empfohlene Zusatzaufgaben (ohne Bewertung)

☐ Daten dynamisch aus CSV einlesen

☐ Responsive Design für Header/Navigation:

- ☐ Menüelemente zusammenfassen/einklappen
(Hinweise u.a. <http://www.hongkiat.com/blog/responsive-web-nav/>)

Aufgabenstellung A1: Übersicht bisheriger Fehlercodes

A00	Keine Abgabe	A20	Stylevorgaben falsch/fehlerhaft umgesetzt
A01	Abgabe Falsch benannt / falsches Format	A21	Containerbreite (1170px) falsch, nicht zentriert
A02	Abgabe nicht gezippt	A22	Farbhover Links/Hover Logo fehlt
A03	Keine ReadMe Datei / ReadMe unvollständig	A23	Gradient im Header fehlt
A04	Kein valides HTML5	A24	Font Style Fehlerhaft
A05	Kein valides CSS3	A25	CSS Transition im Header fehlt (bei Hover)
A06	Falsche Dateikodierung (Encoding nicht UTF-8)	A26	Header nicht klickbar
A07	Falsche Zeilenendekodierung (ist nicht UNIX LF)	A27	Standard-Links nicht richtig angepasst
A08	Keine eigene CSS-Datei, keine separaten CSS-Files		geringe Stil-Abweichungen im Header (Handwerklich, techn. Verständnis)
A09	nur 3 Attribute dargestellt	A28	
A10	keine eigene JS-Datei	A29	JS Tabellen Sortierung funktioniert nicht richtig
A11	zusätzliche Frameworks genutzt	A30	JS Tabellen Sortierung funktioniert nicht
A12	HTML Reset nicht eingebunden	A31	Tabelle nicht responsive
A13	Google Web Fonts/FontAwesome nicht eingebunden	A32	Show/Hide funktioniert nicht
	Unzureichende Metadaten im HTML-Head (Title, Description, Author, Keywords)	A33	Tabelle nicht richtig responsive
A14		A34	CSS LoremIpsum falsche Spaltenanzahl
A15	Header nicht sticky	A35	CSS Responsiveness Fehler
A16	Keine semantischen HTML5-Tags für Header, Footer oder Navigation	A36	Kein Lorem Ipsum Text
A17	Fehler beim Hovern des Logos	A37	Menü verschwindet bei kleinster Breite
A18	Fehlende Tabellen-Tags (thead,tbody)	A38	kein font-awesome bei Tabellen Sortierung
A19	Footer unvollständig	A39	LoremIpsum keine Responsiveness
		A40	Zu wenig Tabellen Elemente

Teil 3

Grundlagen Webentwicklung: Eine thematische Einführung

■ HTML5 – Grundstruktur

```
– <!DOCTYPE html>
  <html>
    <head>
      <!-- ... -->
    </head>
    <body>
      <!-- ... -->
    </body>
  </html>
```

- **head** enthält Seitentitel, CSS- und JS-Referenzen, etc.
- **body** enthält sichtbare Seitenelemente: Überschriften, Texte, Bilder, Tabellen, etc.



■ HTML – Basics

- `<h1>Überschrift ersten Grades</h1>`
- ``
- `<p>Ich bin ein Absatz.</p>`
- Bilder, Skripte etc. können **absolut** oder **relativ** adressiert werden
 - Start ohne Slash, z.B. „`bilder/logo.jpg`“ ist relativ zum aktuellen Verzeichnis
 - Start mit Slash, z.B. „`/css/layout.css`“ ist relativ zum Wurzelordner des Servers
 - Start mit `http://`, z.B. „`http://mt.inf.tu-dresden.de/lehre/wme`“ ist absolut
- **Semantische Tags** in HTML5 für SEO
 - Struktur: `<section>`, `<article>`, `<nav>`, ...
 - Textsemantik: `<time>`, `<meter>`, `<progress>`, ...





■ HTML5 – Strukturtags

- Statt wie bisher mit <div>-Tags werden Dokumente nun mittels neuer Tags strukturiert → verbessert die Maschinenlesbarkeit und macht den Code übersichtlicher

- `<!DOCTYPE html>`

- `<html>`

- `<head> ... </head>`

- `<body>`

- `<header><h1>HTML5 Demo</h1>`

- `<nav> ... </nav>`

- `</header>`

- `<article>`

- `<h2>Semantische Tags</h2>`

- `<section> ... </section>`

- `</article>`

- `<footer> ... </footer>`

- `</body>`

- `</html>`

Kopfbereich eines
Dokumenteninhaltes

Navigationselemente

Textuell relevante
Inhalte

Untergliederung von
Textinhalten

Fußbereich eines
Dokumenteninhaltes

■ HTML – Spezielle Elemente



- Listen können durch `` (unordered) oder `` (ordered) definiert werden. Die Kind-Elemente werden durch `` definiert.
- Tabellen werden durch `<table>` definiert. Sie bestehen aus `<thead>` und `<tbody>` sowie Reihen:

```
<tr>  
  <td></td>  
</tr>
```

- Alle Elemente können zusätzliche Informationen über die HTML5 data-* Attribute erhalten:

```
<ul>  
  <li data-animal-type="bird">Owl</li>  
  <li data-animal-type="fish"></li>  
</ul>
```

■ Cascading Style Sheets (CSS)

- Regeln zur Darstellung eines HTML-Dokuments
- Farben, Rahmen, Abstände, Schriften, ...
- bestehend aus
 - Selektor
 - Eigenschaft-Wert-Paaren
- Syntax
 - `Selektor {Eigenschaft:Wert; Eigenschaft:Wert;...}`
 - z.B.: `p{font-family:"Times New Roman";font-size:20px;}`
passt Schriftart und -größe aller Absätze an
- *class* und *id* um HTML-Elementen Eigenschaften zuzuordnen
 - class als Selektor für Vielzahl von Elementen zb: `<p class='excercise'>`
 - id für einzelnes selektieren von Elementen `<nav id='main_menu'>`



■ Cascading Style Sheets (CSS)

– Einbinden von CSS



■ **direkt im Quellcode**

```
<h1 style=„color:red;“>Webdesign für Anfänger</h1>
```

■ **am Anfang der HTML-Datei**

```
<style>
    p.subtitle    {font-size:12pt;color:#0000FF;}
    a:hover       {text-decoration:none;}
</style>
```

■ **als externe Datei**

```
<link rel=„stylesheet“ href=„/css/layout.css“>
```

■ Cascading Style Sheets (CSS)

– Selektor gibt zu formatierende Elemente an:

- Global: * (alle Elemente des Dokuments)
- Elementtyp: `h1` (alle Überschriften 1. Grades)
- Klasse: `.subtitle` (alle Elemente mit class-Attribut „subtitle“)
- ID: `#navigation` (Element mit der ID „navigation“)
- Pseudo-Klasse: `a:hover` (beim Überfahren des Elements mit der Maus)
- Liste: `h1, img, p` (alle 1. Überschriften, Bilder und Absätze)
- Unterelemente: `p a` (alle Links innerhalb von Absätzen)

– nahezu beliebig kombinierbar

– spezifischere Regeln überschreiben allgemeinere

- Klasse ist spezifischer als Element
- ID ist spezifischer als Klasse
- betrifft jedoch nur widersprüchliche Attribut-Wert-Paare!



■ CSS – Weitere Selektoren

- CSS-Versionen definieren neue oder erweiterte Selektortypen: Attribut-Selektoren, Pseudo-Klassen, Kombinatoren
- **Attribut-Selektoren:** erlauben Selektion von Elementen unter Prämissen der id-Benennung

<code><div id="nav-primary"/></code>	<code>div[id^="nav"] { background:#ff0; }</code>	Alle Elemente die mit „nav“ beginnen
<code><div id=„second-nav"/></code>	<code>div[id\$="nav"] { background:#ff0; }</code>	Alle Elemente die mit „nav“ enden
<code><div id=„supernavone"/></code>	<code>div[id*="nav"] { background:#ff0; }</code>	Alle Elemente die „nav“ enthalten

- **Strukturelle Pseudo-Klassen:** erlaubt die Selektion von Elementen abhängig von ihrer Position im DOM-Baum:
 - root, nth-child, first-child, last-child, only-child, only-of-type, empty, ...



■ CSS – Weitere Selektoren

- nth-child() – Funktion kann nach vielfältigen Formeln Kindelemente selektieren

<code>:nth-child(3) { color:#f00; }</code>	Selektiert das dritte Kindelement
<code>:nth-child(odd) { color:#f00; }</code>	Selektiert alle ungeraden Kindelemente (even für gerade)
<code>:nth-child(2n+7) {color:#f00; }</code>	Selektiert alle Elemente die der Formel $an+b$ entsprechen (a - Zyklusgröße, n - Zählvariable (ab 0), b - Offset)

- **Beispiel:** abwechselnde Farben der Tabelle hier?
- `tr:nth-child(2n) { background-color: #D0D8E8; }`
`tr:nth-child(2n+1) { background-color: #E9EDF4; }`



■ CSS – Kombination von Selektoren [8]



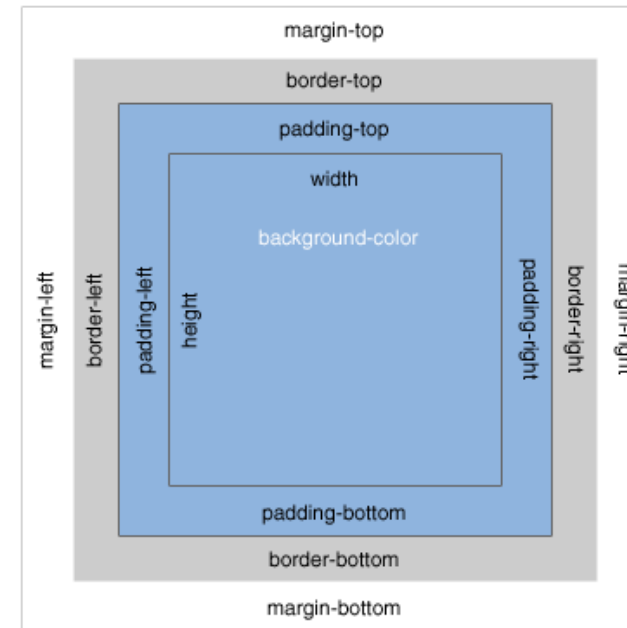
X > Y	Selektiert <u>nur</u> die <u>direkten</u> Kinder Y von X	div#container > ul { border: 1px solid blue; }
X + Y	Selektiert <u>nur</u> das Y-Element nach einem X	ul + p { color: blue; }
X ~ Y	Selektiert <u>alle</u> Y nach einem X (weniger strikt als X + Y)	ul ~ p { color: blue; }

– Beispiel:

```
<div id="container">
  <ul>
    <li> List Item
      <ul>
        <li> Child </li>
      </ul>
    </li>
    <li> List Item </li>
  </ul>
  <p> Paragraph </p>
  <p> Paragraph </p>
</div>
```

■ Cascading Style Sheets (CSS)

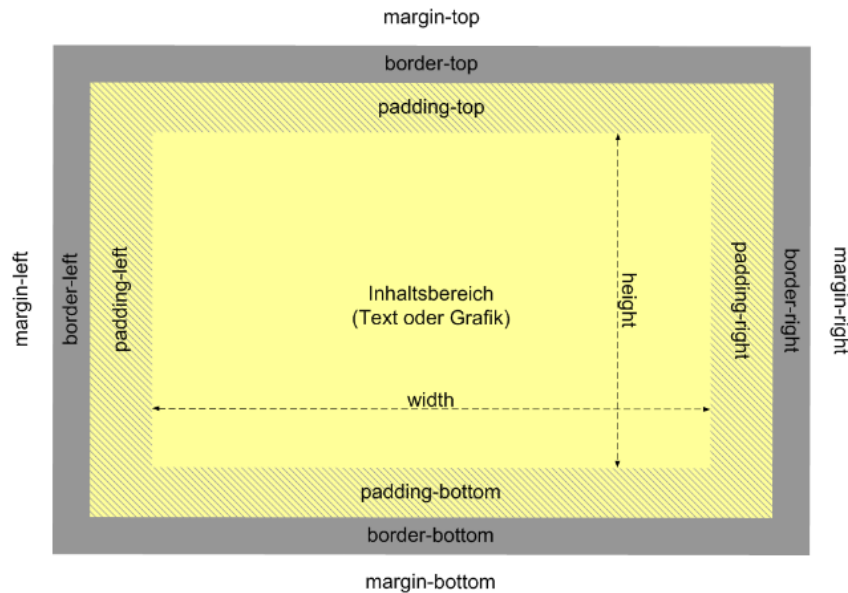
- Stil gibt an, wie Elemente formatiert werden sollen
 - Rahmen und Abstände: **CSS Box Model**
 - Positionierung: immer relativ zum Elternelement
 - absolute, relative, inline, fixed
 - Farben
 - als RGB-Hex-String: #FFFFFF
 - als RGB-Tupel: rgb(0,0,0)
 - Weitere: rgba, hsl, hsla
 - Transparenzen: opacity: 0.2;
 - Schrift
 - Größe: font-size: 12pt;
 - Stil: font-style: italic;
 - Stärke: font-weight: bold;
 - Extras: text-decoration: underline;



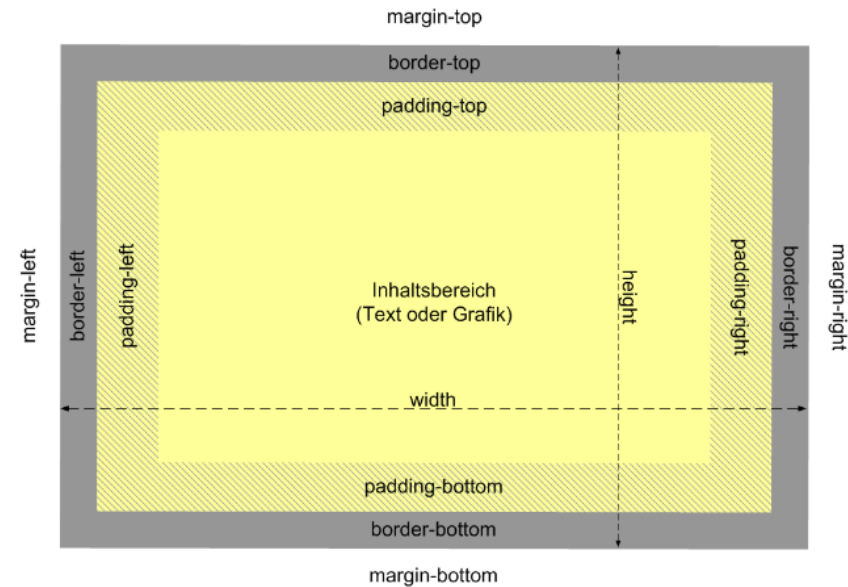
Quelle: [<http://www.mandalatv.net/itp/drivebys/css/>]

■ CSS3 – Neues Box-Modell [9]

- Bisher: {box-sizing: content-box} → Höhe und Breite definieren Inhaltsbereich; Padding, Border, Margin werden hinzugefügt
- Neu: {box-sizing: border-box} → alle Außenelemente sind in Höhe und Breite enthalten



Klassisches Box-Modell für Block-Elemente – { box-sizing: content-box } (Standard-Einstellung) – <http://title-boxes.de/>



CSS3 Box-Modell für Block-Elemente – { box-sizing: border-box } – <http://title-boxes.de/>

■ Cascading Style Sheets (CSS)

- **Media Types** geben Zielmedien für Regeln an

- @media print, handheld, projection, tv, speech, braille

- z.B.:

```
@media handheld, print {  
    p.subtitle { color: #000000; }  
}
```

- **CSS 3 Media Queries** als Erweiterung

- ermöglichen Adaptivität der Darstellung an Bildschirmauflösung etc.
- UND-verknüpft
- (max-/min-)width, color, aspect-ratio, ...

- z.B.:

```
@media screen and (max-width: 500px) {  
    p.subtitle { font-size: 14pt; }  
}
```



■ Java Script (JS)

- Scriptsprache für Einsatz im Browser und Server
- Typische **Anwendungsbereiche** von JavaScript:
 - **Überprüfung** von Formularen vor dem Absenden
 - Dynamische **Interaktion**, z. B. Buttons, Darstellungsgröße, ...
 - Dynamische **Inhalte**, z. B.: AJAX, WebSockets, ...
 - **Veränderung** bzw. Animation von Eigenschaften der HTML-Elemente
- **Leistungsmerkmale:**
 - Zugriff auf Bestandteile des geladenen Dokumentes über DOM
 - Dynamische Änderung des geladenen Dokumentes
 - Steuerung externer Komponenten
 - Reaktion auf Benutzerinteraktionen, Fenstermanagement
 - Bei Serverseitiger Anwendung: Erstellen von komplexen Programmen, Schnittstellen etc. ...



■ Einbindung

- Innerhalb eines HTML-Elements

```

```

- Innerhalb eines HTML-Dokuments in einem `<script>`-Teil im Header

```
<script language="javascript">  
    //...Hier kommt der Code ...  
</script>
```

- Innerhalb einer externen Datei im Header

```
<script src="menue.js" type="text/javascript"></script>
```





■ Funktionen

`function Fkt.name (Parameterliste) {Funktionsrumpf}`

- Rückgabewert kann hinter Schlüsselwort `return` angegeben werden
 - Beispiel: `function square(i) {... return i*i; }`

■ Operatoren entsprechen im Wesentlichen denen von Java [15]

- Zuweisung: `=`
- Vergleichsoperatoren: `==, !=, <, <=, >, >=, ===`
- Rechenoperatoren: `+, -, *, /`
- logische Operatoren `&&, ||`

■ Verzweigungen und Schleifen ebenso (Beispiele)

- | | |
|---|---|
| <ul style="list-style-type: none">– <code>if (...) {</code>
 <code>...}</code>
 <code>else {</code>
 <code>...}</code> | <ul style="list-style-type: none">– <code>while (...) {</code>
 <code>...}</code>
– <code>for (...) {</code>
 <code>...}</code> |
|---|---|

■ Objekte

- zum Zugriff auf Elemente des HTML-Dokuments
- Zugriff auf Objekt-Attribute & –Methoden über den **Auswahloperator** `"."`
z. B.: `navigator.geolocation.getCurrentPosition(succ, err);`
zum Zugriff auf den aktuellen Standort des Benutzers

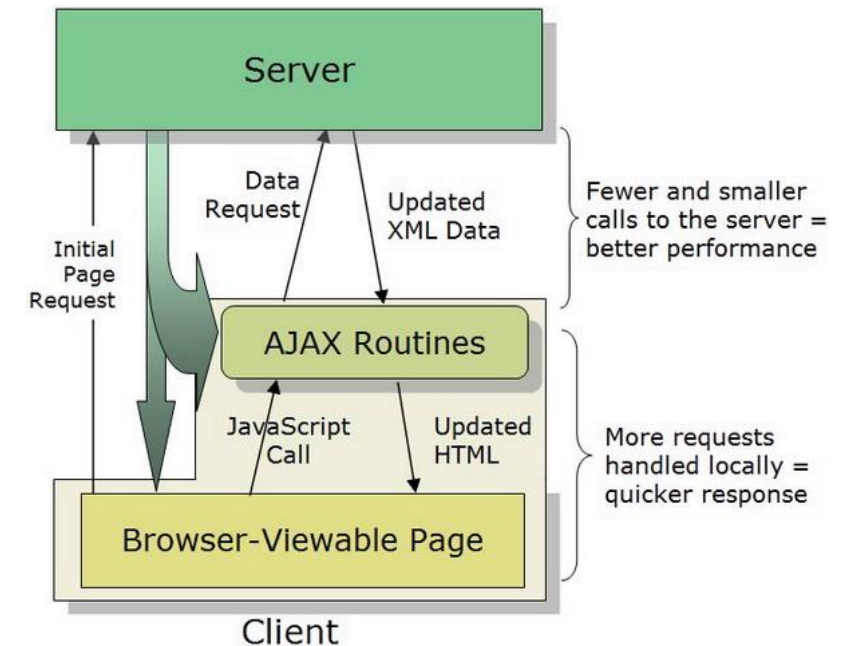
```
var cup = { color: `green` };  
Console.log(cup.color) //green
```

- JavaScript enthält **vorgefertigter Objekttypen**: Sprach-Objekte und Client-Objekte
 - **Sprach-Objekte**: Repräsentation häufig genutzter Datentypen,
z. B. **Array**, **Math** (math. Grundfkt.), **Date**, **Function** oder **RegExp** (reguläre Ausdrücke)
 - **Client-Objekte**: Zugriff auf Teile des **HTML-Dokumentes** und Zustand dieser



■ AJAX – Asynchronous JavaScript and XML

- Hauptproblem bei komfortablen Web-Anwendungen ist träge Reaktion
 - Je Benutzerinteraktion eine Serveranfrage, auf Ergebnis muss gewartet werden („blockierende Interaktion“)
- **Kernidee AJAX**
 - Webseite pro Interaktion nicht neu geladen, sondern dynamisch modifiziert
 - Server-Anfragen asynchron im Hintergrund, Warten meist nicht notwendig
- **Bausteine: HTML, JavaScript**
 - Dynamische Modifikation des Dokumentenbaums über DOM (Document Object Model) des (X)HTML-Dokuments
 - Asynchrone Datenübertragung mittels **XMLHttpRequest**



Bildquelle:

[http://www.codeproject.com/KB/showcase/FarPointAJAX/AJAX_process.jpg]

■ Browser-Support

- Nicht alle HTML5, CSS3 & JS Elemente werden von allen Browsern unterstützt
 - Empfehlung: [\[http://caniuse.com/\]](http://caniuse.com/)

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			63						
			67						
		61	68	11.1	11.2				4
11	17	62	69	12	11.4	all	69	11.8	7.2
	18	63	70	TP	12				
		64	71						
			72						

Aktueller Support von HTML5 [16]

Teil 4

Hilfreiches, Tipps und Links

- Klassisches Thema für Selbststudium:
„Step-by-Step learning“
- Erstellung eines Farb- und Designschemas mittels Paper-Protoyping [1, 2]
- Hilfreich bei der Gestaltung sind Rastervorlagen [3]
- Umsetzung des Entwurfs in einem Editor der Wahl
 - Liste von HTML-Editoren [10] und Universelle Texteditoren [11]
- Validierung des kompletten Markups
 - HTML5 [4] und CSS3 [5]

- Responsive Webdesign [6]
 - Optimale Ansicht einer Website auf allen Arten von Geräten
 - Früher drei Versionen vs. Heute eine flexible Version
 - Einfache Navigation, Informationsfindung mit möglichst wenig Veränderungen an der Seite selbst (Resizing, Panning, Scrolling)
 - Hauptbestandteile: flexible Schrift, flexibles Grid, Media Queries



Quelle:
http://en.wikipedia.org/wiki/File:Boston_Globe_responsive_website.jpg

- JavaScript
 - Einfacher Einstieg und Überblick [12]
 - Neue Features [13]
 - Vanilla JS ;) [14]

Quellen, Referenzen

- [1] „Paper Prototyping“, [\[http://de.wikipedia.org/wiki/Paper_Prototyping\]](http://de.wikipedia.org/wiki/Paper_Prototyping)
- [2] „Six Signs That You Should Use Paper Prototyping“, [\[http://today.java.net/pub/a/today/2003/12/23/sixSigns.html\]](http://today.java.net/pub/a/today/2003/12/23/sixSigns.html)
- [3] „Free Printable Sketching, Wireframing and Note-Taking PDF Templates“, [\[http://www.smashingmagazine.com/2010/03/29/free-printable-sketching-wireframing-and-note-taking-pdf-templates/\]](http://www.smashingmagazine.com/2010/03/29/free-printable-sketching-wireframing-and-note-taking-pdf-templates/)
- [4] „W3C Markup Validation Service“, [\[http://validator.w3.org/\]](http://validator.w3.org/)
- [5] „W3C CSS Validation Service“, [\[http://jigsaw.w3.org/css-validator/\]](http://jigsaw.w3.org/css-validator/)
- [6] „Responsive Webdesign mit HTML5 und CSS3 – Grundlagen“, [\[http://t3n.de/news/responsive-webdesign-html5-css3-grundlagen-335305/\]](http://t3n.de/news/responsive-webdesign-html5-css3-grundlagen-335305/)
- [7] „Media Queries: Very Good Examples“, [\[http://mediaqueri.es/\]](http://mediaqueri.es/)
- [8] „The 30 CSS Selectors you Must Memorize“, [\[http://net.tutsplus.com/tutorials/html-css-techniques/the-30-css-selectors-you-must-memorize/\]](http://net.tutsplus.com/tutorials/html-css-techniques/the-30-css-selectors-you-must-memorize/)
- [9] CSS3: Das etwas andere Box-Modell mit {box-sizing:border-box}, [\[http://www.drweb.de/magazin/css3-das-etwas-andere-box-modell-mit-box-sizingborder-box-39078/\]](http://www.drweb.de/magazin/css3-das-etwas-andere-box-modell-mit-box-sizingborder-box-39078/)

Quellen, Referenzen (2)

- [10] Liste von HTML-Editoren, [http://de.wikipedia.org/wiki/Liste_von_HTML-Editoren]
- [11] Liste von Texteditoren, [http://de.wikipedia.org/wiki/Liste_von_Texteditoren]
- [12] Einfacher Einstieg & Überblick JS, [<http://www.w3schools.com/js/>]
- [13] Neue Features JS, [<http://es6-features.org/>]
- [14] Vanilla JS, [<http://vanilla-js.com/>]
- [15] JavaScript Operatoren, [https://developer.mozilla.org/de/docs/Web/JavaScript/Vergleiche_auf_Gleichheit_und_deren_Verwendung]
- [16] Support von HTML5, [<https://caniuse.com/#search=HTML5>]

Fragen?



Interactive Media Lab Dresden
Professur für Multimedia-Technologie

Kontakt:

Patrick Reipschläger (patrick.reipschlaeger@tu-dresden.de)

Robert Fuhrmann (robert.fuhrmann@mailbox.tu-dresden.de)

Changelog

Datum / Zeit	Beschreibung
2018-10-15 8:40	▪ Initiale Downloadversion