

- 9:00 - 10:30 Zusammenfassung HTML, CSS, JavaScript, Ausblick: Bibliotheken, Frameworks, Templates, CMS und Weiterführung, Fragen
- 10:30 - 10:45 Pause
- 10:45 - 11:00 Aktivierungseinheit
- 11:00 - 12:30 Eigenarbeit, Fortsetzung/Finalisierung Übungsprojekt
- 12:30 - 12:45 Pause
- 12:45 - 13:30 feedback-Runde, Aufgabe

HTML, CSS und JavaScript sind Kernsprachen des Webs

- **HTML** - Auszeichnungssprache für **Strukturierung** von Inhalten
- **CSS** - Stylesheet-Sprache für **Gestaltung** von Inhalten
- **JavaScript** - Skriptsprache für **Programmierung der Interaktionen** mit Inhalten

Recap HTML 1

- HTML ist eine Auszeichnungssprache, die zur Strukturierung von Dokumenten zur Darstellung im Browser dient.
- HTML-Dokumente folgen immer demselben Aufbau:

`<html>`

“Wurzelement”: Signalisiert den Beginn eines HTML-Dokuments

`<head>`

“Kopf”:

`<title></title>`

Enthält Metadaten, dh. Informationen über das Dokument selbst (zB auch `<style>`- oder `<script>`-Informationen).

`</head>`

`<body>`

`<!-- Inhalt des Dokuments -->`

“Körper”:

Inhalt des Dokuments. Teil, der vom Webbrowser dargestellt wird.

`</body>`

`</html>`

Recap HTML 2

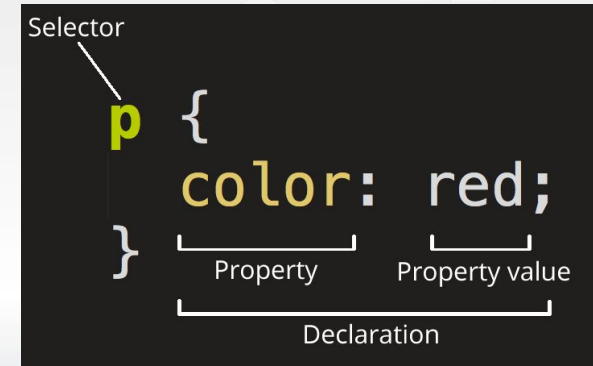
- Jedes HTML-Dokument muss **zwei Bedingungen erfüllen**:
 - **Validität**: Es muss syntaktisch korrekt sein,
 - z.B. muss dem Element `<a>` zwingend ein Attribut "href" mitgegeben werden.
`<a>Dies ist ein Link.` Validität ist nicht gegeben.
 - **Wohlgeformtheit**: Es muss eine korrekt geschachtelte Struktur aufweisen, z.B. dürfen sich Element nicht überlappen.
`<div><p></div></p>` Wohlgeformtheit wird verletzt.

Recap HTML 3

- HTML-**Elemente** werden **tags** genannt und in spitzen Klammern geschrieben: `<html>`, `<div>`, `<p>`
- Die meisten Elemente werden als **tag-Paar** geschrieben, das schließende tag wird mit einem **zusätzlichen / (Slash)** versehen:
 - `<p>`Hier steht der zu strukturierende Text, der angezeigt werden soll.`</p>`
- Elemente können **Attribute** besitzen, z.B. Klassennamen oder eine ID:
 - `<div class="Attribut-Wert">`, `<p id="gold_medal">`
- tags können **block** (zB `<p>`, `<div>`, ``) **oder inline** (zB ``, ``, `<a>`) geschrieben werden:
Dies hat Einfluss auf die Darstellung und den Umgang mit Elementen.
- Das **class-Attribut** dient dazu, mehrere Tags zu gruppieren. So können alle Elemente einer Klasse im CSS angesprochen werden (zB im CSS oder JS).
- Im Gegensatz zur `class` sind **ids eindeutige Attribute**, sie zeichnen lediglich ein einziges Element aus.

Recap CSS 1

- CSS ist eine Stylesheet Language, welche die Präsentation eines html-Dokumentes beschreibt
- CSS-Declarations werden über Selektoren angesprochen
- Die property und property value geben darüber Auskunft, wie das ausgewählte Element/Klasse/ID aussehen soll



[Frieden](#)

Die Friedensnobelpreisträger:innen

1901 Henry Dunant (1828–1910) Schweiz

2022 - Das ist eine auf Wikipedia-Daten basierende Seite mit Nobelpreisträger:innen



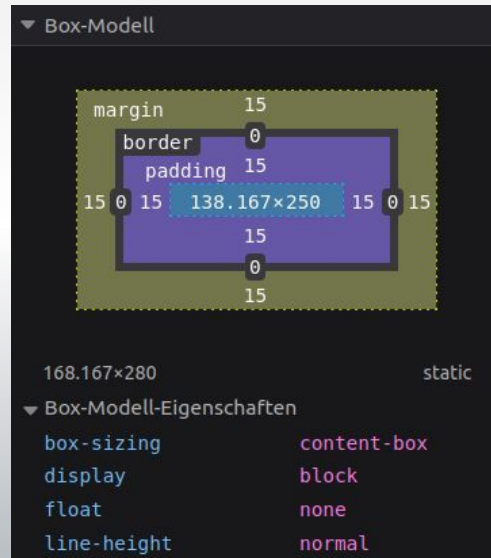
```
body{  
  color:red;  
}
```

Recap CSS 2

- Es gibt unterschiedliche Selectoren, mit denen man html-Inhalte ansprechen kann
- Einige haben eine höhere **Gewichtung** als andere
- **CSS element selector** : basierend auf dem html-Elementnamen
 - p -> Alle Elemente mit dem Namen <p>
- **CSS id selector** : nutzt ein id-Attribut eines html-Elementes
 - #headerStyle -> Das/Die Element/e mit der id="headerStyle"
- **CSS class selector** : html mit einem spezifischen class-Attribut
 - .headerStyle -> Das/Die Element/e mit der class="headerStyle"
- **universal selector** : spricht alle html-Elemente auf der Seite an "*"
- **CSS grouping selector** : Selectoren können mit "," getrennt angegeben werden

Recap CSS 3

- Um alle Elemente bestehen eigentlich **Boxen**
- Boxen bestehen aus **content**, **padding**, **border**, **margin**
- Mit dem Inspektor des Browsers kann man die Zusammensetzung der Boxen untersuchen und testen



1901
Henry Dunant
(1828–1910)
Schweiz



1902
Élie Ducommun
(1833–1906)
Schweiz



Frédéric Passy
(1822–1912)
Frankreich



Albert Gobat
(1843–1914)
Schweiz



Recap JS

- Konzepte: Variablen, Datentypen, Operatoren, Kommentare, Kontrollstrukturen wie Schleifen, Anweisungen
- Debugging, Einbindung, Document Object Model (DOM)
- Function declaration
- Javascript Engine

- Außer HTML, CSS und JavaScript gibt es andere Sprachen und Technologien, die für Webentwicklung relevant sind, zum Beispiel PHP, SQL, XML, XQuery usw.
- Je nach **Ziel und Ressourcen** wird entschieden, welche Technologien umgesetzt werden sollen.

- **Nicht alle Webseiten müssen von null programmiert werden!**
- Es gibt **Programmbibliotheken**. Sie bieten bestimmte Funktionalitäten, wie zum Beispiel Suche, die weiterverwendet werden können.
- Außerdem gibt es **Frameworks**, die eine Sammlung von Konzepten und Bausteinen bieten, die bei der Entwicklung genutzt werden können.
- Es gibt auch die kompletten Lösungen, von einfachen **Webseite-Templates** bis zu komplexen **Content Management Systemen**

Im Internet gibt es sehr viele Tutorials zu Webtechnologien und hilfreichen Seiten. Einige davon sind:

- <https://www.w3schools.com/>
- <https://wiki.selfhtml.org/>
- <https://www.freecodecamp.org/news/tag/web-development/>
- <https://stackoverflow.com/>
- <https://www.youtube.com/>

Fragen



Sprachen des Webs: HTML, CSS und JS

Sviat Drach, Pia Geißel, Lena-Luise Stahn, Nadine Sutor

Feedback

Es gibt eine Evaluation der Organisatoren der Sommer Schule:

<https://umfrage.uni-wuppertal.de/index.php/111425?lang=de>

