

Schulung **Web-Entwicklung**

Tag 1:
Grundlagen

Michael Brüggemann
Moritz Hipper



Übersicht

01

Grundlegende Technologien

HTML, CSS, JavaScript, HTTPS

02

Lebenslauf einer Webanwendung

03

WCAG

Grundlegende Technologien

01





- **HyperText Markup Language**
- **Auszeichnungssprache**
 - Dient der Strukturierung von Texten und anderen Daten
- **„Grundgerüst“**
 - für Websites und Webapplikationen, das mit Fallrelevanten Daten befüllt wird
- Beschreibt **HTML-Objekte**, die weitere **HTML-Objekte** beinhalten
 - Baumstruktur/ DOM

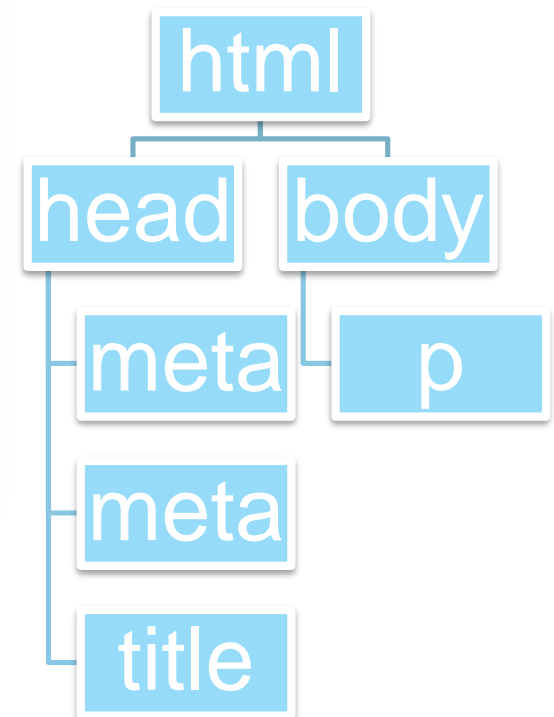


HTML – Grundgerüst Roh

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="utf-8"/>
  <meta name="viewport"/>
  <title>Titel der Seite sichtbar im Fenstertitel des Browsers </title>
</head>
<body>
  <!-- Nichtsichtbarer Kommentar -->
  <p>Sehen Sie sich den Quellcode dieser Seite an.</p>
</body>
</html>
```

HTML – Grundgerüst vs. DOM

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="utf-8"/>
  <meta name="viewport"/>
  <title>Titel der Seite sichtbar im Fenstertitel des Browsers </title>
</head>
<body>
  <!--Nicht sichtbarer Kommentar-->
  <p>Sehen Sie sich den Quellcode dieser Seite an.</p>
</body>
</html>
```



HTML – Tags

- **Singleton**

z.B.: <meta/>

```
<meta charset="utf-8"/>
```

- **Opening und closing -Tag**

z.B.: <p>Textinhalt</p>

```
<title>Titel der Seite sichtbar im Fenstertitel des Browsers </title>
```



HTML – Objekt - Attribute

- Es gibt Attribute **mit** und **ohne Wert**
 - Beispiele mit: id, name, class, charset
 - Beispiele ohne: disabled

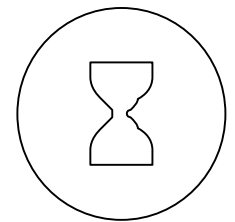
```
<meta charset="utf-8" />
```




Aufgabe 1: Wichtige HTML-Objekte

Sucht zu folgenden HTML-Tags die Definition und den Nutzen aus dem Internet zusammen. Notiert sowohl eure Erkenntnisse als auch Verständnisfragen zu den gefundenen Definitionen.

<form>, <input>, <script>, <a>, <style>, <canvas>, <svg>



15 Minuten



- **Custom Style Sheet**
- **Style** – Sprache
- Erweitert HTML-Objekte um **visuelle Eigenschaften**
- Kann: **aufhübschen, signalisieren, bewegen, anordnen**
- Kann direkt im HTML-Objekt, im der DOM im Style-Tag oder in externen Dateien stehen
- Beeinflusst zum Beispiel: Farbe, Hintergrundfarbe, Schriftgröße, Schatten, Rotation, Effekte beim Mausüberfliegen oder Anfokussieren
- Hilft:
 - Relationen zwischen Daten herzustellen
 - Bei der Nutzerführung
 - Der Visuellen Anordnung von Objekten
 - Der dynamische Inhaltsanzeige auf verschiedenen Geräten oder Fenstergrößen



- **Scriptsprache**

- simple bis sehr komplexe **Programmierlogik abbilden**
- **was andere Sprachen auch können** (if-else, for-loops, while-loops, etc.)
- den **DOM-Baum manipulieren**
- auf **Nutzereingaben reagieren**
- einige **Browser** und **Hardwarespezifika auslesen**
- **Browserpopups**
- **Externe Ressourcen anbinden**
- Funktionsumfang definiert in **EcmaScript – Standard** (aktuell EcmaScript 2019)

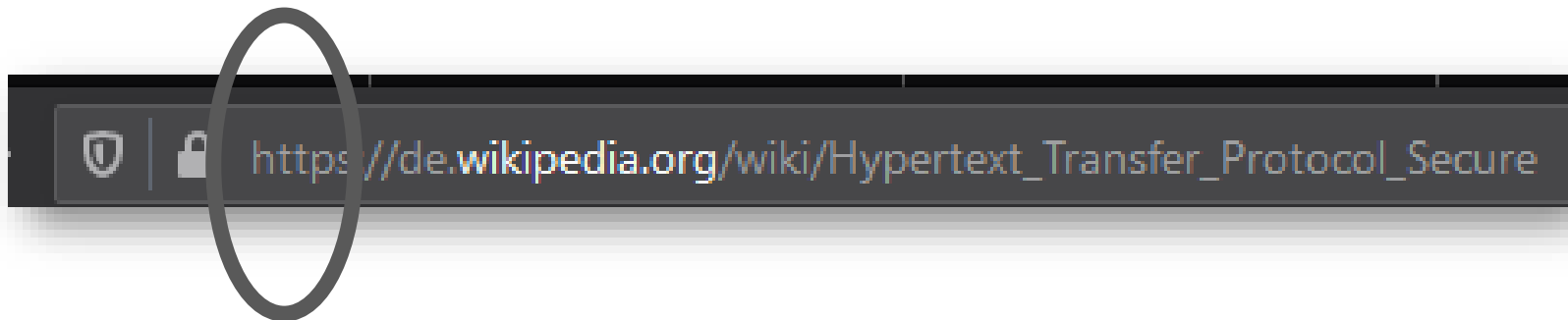


Zusammenspiel HTML, CSS, JS

- **HTML** bildet **Grundgerüst**
- **CSS** erweitert das Grundgerüst um **Layout, Design** und **Nutzerführung**
- **JavaScript** erweitert das **HTML** um die Ebene **Nutzerinteraktion**



- **HyperText Transfer Protocoll**
- **Protokoll-Standard für die Übertragung von Dateien im Internet**
 - darunter Websites/Dateien des Formats .html oder deren assets wie z.B. *.jpg und *.mp4
- Statuscodes für HTTP-Anfragen
 - z.B.: 200, 404, 401





- **HyperText Transfer Protocoll *Secure***
- **Erweitert das HTTP-Protokoll um die Verschlüsselung per TLS-Protokoll**
 - Transport Layer Security





Wie kommt die Website zum Benutzer?

1. **Nutzer fragt Netzwerkressource** im Browser per URL an
2. **Netzwerkressource antwortet** mit dem hinter der angefragten Route hinterlegten Dokument (z.B.: HTML, MP4, XML, ...)
3. Browser **verarbeitet Netzwerkressource** nach erfolgreichem Laden
 - Wenn HTML mit CSS → Visualisieren
 - Wenn HTML mit CSS und JavaScript → Visualisiere, Script ausführen
 - Wenn anderer Dokumenttyp → nach Funktionsumfang des Browsers anzeigen
 - Wenn weitere Netzwerkressourcen im Dokument verwiesen → Diese Imports auflösen, beginn bei Schritt 1.

WCAG

02





- **Web Content Accessibility Guidelines**
 - **Barrierefreie Gestaltung der Softwarewelt**
 - Verbesserung des **Pageranks** bei Suchmaschinen
- Verbesserung der **Usability** für alle Nutzer
- Praktische Umsetzung mitunter in Form von **HTML-Attributen** (aria), logischeren **Anordnung der Elemente** und **eindeutigem CSS-Style**
- Guidelines zusammengefasst in **vier Prinzipien**:
 1. **Wahrnebarkeit / perceivable**
 2. **Benutzbarkeit / operable**
 3. **Verständlichkeit / understandable**
 4. **Robustheit / robust**



Aufgabe2: WCAG

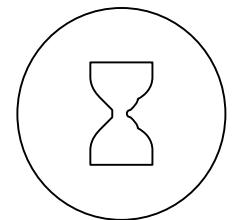
Fasst die vier Prinzipien des WCAG zusammen. Notiert sowohl eure Erkenntnisse als auch Verständnisfragen zu den gefundenen Inhalten.

**Wahrnebarkeit /
percievable**

**Benutzbarkeit /
operable**

**Verständlichkeit
/
understandable**

**Robustheit /
robust**



20 Minuten