

HTML CRASHKURS

Teil 2: Layouting / Outlining

Speaker notes

- HTML Elemente zur Textstrukturierung
 - h1 bis h6, listen (ol, ul, dl)
 - p, blockquote, figure mit figcaption
- HTML Elemente zur Textauszeichnung
 - a, small, strong, b, em, i, br
- CSS sind "stufenförmige Formatvorlagen", die das Design einer Website mit Hilfe von Regeln angeben.
- unterschiedliche Selektoren:
 - Element-Selektor p {...} (HTML Tag),
 - Class Selektor .my-class {...} (mit Punkt vor dem Klassennamen)
 - ID Selektor #my-id {...} (mit Hash vor dem ID Namen)
 - Pseudo Selektor a:hover {...} (mit Doppelpunkt vor dem Zustand)

BASISWISSEN

- Nenne HTML Tags zur Textstrukturierung & -auszeichnung.
- Was bedeutet CSS?
- Wie sieht eine CSS Regel aus?
- Was für CSS Selektoren gibt es?

HTML: GRUNDGERÜST

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Titel oder Name der Website</title>
    <meta name="description" content="Beschreibungstext 120-158 Zeichen">
    <!-- CSS Styles verlinken & JavaScript Anweisungen laden -->
    <link href="main.css" rel="stylesheet">
    <script src="main.js" async></script>
  </head>
  <body>
    <!-- Sichtbarer Bereich -->
    <h1>Hallo Welt</h1>
    <p>Meine erste Website</p>
  </body>
</html>
```

Speaker notes

- **html:** die gesamte Website
- **head:** die Kopfdaten der Website
 - **title:** Title der Seite (wird im Tab angezeigt, in den SERPs, in Bookmarks)
 - **link:** Verknüpft andere Dateien im gleichen Webespace
 - **meta:** enthält zusätzliche Angaben (Zeichencodierung, Darstellung, SEO etc.)
 - **style:** für interne style Angaben
 - **script:** für JavaScript Anweisungen
- **body:** der sichtbare Inhaltsbereich der Website

HTML: SEITENSTRUKTUR

```
<!DOCTYPE html>
<html lang="de">
  <head>
    ...
  </head>
</body>
<header>einleitende Inhalte, Orientierungshilfen</header>
<nav>Seitennavigation <ul>...</ul></nav>
<main>Hauptinhalt</main>
<section>thematische Gruppierung von Inhalten
  <h2>Mit Überschrift h2</h2>
  <article>einzelne in sich geschlossene Inhaltsbereiche</article>
  <article>einzelne in sich geschlossene Inhaltsbereiche</article>
  <article>einzelne in sich geschlossene Inhaltsbereiche</article>
</section>
<aside>ergänzende Hinweise, „Randbemerkungen“</aside>
<footer>informative Inhalte etwa Impressum, Copyright, Autor</footer>
</body>
</html>
```

Speaker notes

header-tag: Oberer Bereich der Website, sollte Logo, Motto und interne Links enthalten. Kann auch im article-Tag genutzt werden.

footer-tag: Unterer Bereich der Website, sollte Copyright, Impressum Links etc. enthalten. Kann auch im article-Tag genutzt werden.

main-tag: Hauptbereich der Website. Screenreader können direkt dort hinspringen.

article-tag: geschlossener Abschnitt eine Website (z.B. Blog Beitrag, News), kann wieder header, footer, section und aside enthalten

section-tag: thematische Gruppierung von Inhalten typischerweise mit einer h2

aside-tag: bereich mit indirektem Zusammenhang zum Inhalt, z.B. Links zu weiteren Webseiten, Kommentare zum Blogbeitrag, Randbemerkungen

nav-tag: In erster Linie als Haupnavigation gedacht. Kann neben Listen auch Überschriften enthalten

LAYOUTING: FLOAT

so wars früher ... man sieht es noch oft

```
<div class="row">
  <div class="col"></div>
  <div class="col"></div>
</div>
```

```
.row {
  clear: both;
}

.row:after {
  content: " ";
  display: block;
  clear: both;
  visibility: hidden;
  height: 0;
  line-height: 0;
}

.col {
  float: left;
  margin-left: 1rem;
}

.col:first-child {
  margin-left: 0;
}
```

Speaker notes

- Die "float" Eigenschaft legt fest, in welche Richtung ein Element verschoben wird (innerhalb seines Elternelements)
- Viele Website Layouts sind noch immer mit "float" umgesetzt. Das CSS Framework [Bootstrap 3](#) nutzt für rows und cols float ([Bootstrap 4](#) (aktuell) nutzt flexbox)
- weitere Links
 - [Css Column Structure Using Floats](#)

LAYOUTING: FLEXBOX

so gehts heute ... Eindimensional (Zeile oder Spalte)

```
<div class="row">
  <div class="col"></div>
  <div class="col"></div>
</div>
```

```
.row {
  display: flex;
}

.col {
  flex: 1 1 100%;
  margin-left: 1rem;
}

.col:first-child {
  margin-left: 0;
}
```

Speaker notes

- Eindimensionales Layoutmodell (entweder als Zeile oder Spalte)
- weitere Links
 - [Self HTML: Flexbox](#)
 - [MDN: Flexbox](#)
 - [Online Cheat Sheet](#)

LAYOUTING: GRID

so gehts heute noch besser... Zweidimensional (Zeilen und Spalten)

```
<div class="container">
  <div>Element 1</div>
  <div>Element 2</div>
  ...
  <div>Element 11</div>
  <div>Element 12</div>
</div>
```

```
/* 3 Zeilen mit je 4 Spalten */
.container {
  display: grid;
  grid-template-rows: 200px 1fr 100px;
  grid-template-columns: 25% 25% 25% 25%;
}
```

Speaker notes

- Zweidimensionales Layoutmodell (Zeilen und Spalten - ähnlich wie Tabellen)
- weitere Links
 - [Self HTML: Grid](#)
 - [MDN: Grid Layout](#)
 - [Online Cheat Sheet](#)

LET'S CODE ...

lokal mit Atom