

Übung 01

1 Schnelleinstieg HTML

1.1 Tags

Die Auszeichnung von Seiteninhalten wird in HTML durch Tags gesteuert. Mit Ausnahme weniger spezieller Tags (etwa
) werden stets öffnende und schließende Tags verwendet um den auszuzeichnenden Inhalt zu strukturieren:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Seitentitel</title>
5    </head>

    <body>
      <h1>Eine Überschrift</h1>
      </body>
10 </html>
```

1.2 head und body

Der head-Tag beinhaltet (Meta-)Informationen über den Inhalt der Webseite. Dies sind beispielsweise Titel der Seite, Autor des Inhalts, genutzte Sprache, Encoding und weiteres (siehe Dokumentation). Normalerweise findet sich der Abschnitt head wie der Name schon suggeriert direkt zu Beginn. Das ganze kann dann beispielsweise so aussehen:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
```

```

5      <link rel="stylesheet" type="text/css" href="styles.css" media="all">
      <title>Ein Seitentitel</title>
      </head>
      <body>
        <!-- content here -->
10     </body>
     </html>

```

1.3 img

Der `img`-Tag bindet Bilddaten in die Webseite ein. `img` ist, wie `br`, ein Standalone-Tag ohne umschlossenen Inhalt und ohne schließendes Tag. Erforderlich sind jedoch die Attribute `src` und wünschenswert `alt`, mit denen die **Bildquelle** und ein **Ersatztext** definiert werden. Der Ersatztext wird angezeigt, falls das Bild nicht geladen oder dargestellt werden kann, oder falls das Anzeigemedium rein textbasiert ist. Weitere Informationen zu einzelnen Elementen und möglichen Attributen finden sich auch in den Dokumentationen.

```

```

1.4 table

Tabellarische Inhalte auf Webseiten können mit dem `table`-Tag dargestellt werden. Die Tabellenstruktur wird mithilfe der `table`, `tr`, `table head`- und `table data cell`-Tags (`tr`, `th`, `td`) erstellt. **Durch Angabe der Attribute `colspan` und `rowspan` können Zellen einer Tabelle mehrere Zeilen und Spalten zusammenfassen.** Eine minimale Tabellenstruktur kann wie folgt aussehen:

```

<table>
  <tr>
    <th>Überschrift Spalte 1</th>
    <th>Überschrift Spalte 2</th>
5   </tr>
    <tr>
      <td>Zelle 1</td>
      <td rowspan="2">Zelle 2 und 4</td>
    </tr>
10   <tr>
      <td>Zelle 3</td>
    </tr>
  </table>

```

Es empfiehlt sich, Tabellen in einen `thead`, mindestens einen `tbody` und optional einen `tfooter` aufzuteilen. So ist es beispielsweise möglich, mittels CSS die Kopfzeile einer Tabelle im Browser zu fixieren, während der Benutzer durch längere Tabellen scrollt.

1.5 HTML5 spezifische Strukturelemente

Mit der Einführung von HTML5 sind eine Vielzahl neuer Tags eingeführt worden¹, die es erlauben das eigene Dokument deutlicher zu strukturieren. So können beispielsweise die Elemente `header`, `main` und `footer` verwendet werden, um den Kontext innerhalb eines `body` zu separieren:

```
<body>
  <header>
    <!-- head-line content here -->
  </header>
5  <main>
    <!-- main content here -->
  </main>
  <footer>
    <!-- bottom-line content here -->
10 </footer>
</body>
```

1.6 nav

Navigationsleisten und Menüs können in HTML5 mit dem `nav`-Element dargestellt werden. Ein `nav`-Tag umschließt hierbei in der Regel eine ungeordnete Liste von Links, was in Kombination folgendermaßen aussieht:

```
<nav>
  <ul>
    <li><a href="#">Startseite</a></li>
    <li><a href="#">Über</a></li>
5   <li><a href="#">Kontakt</a></li>
  </ul>
</nav>
```

1.7 article

Mit dem `article`-Element wird eine geschlossene Komposition ausgezeichnet, die als solche Inhalt kapselt und wiederverwendbar macht. Dabei können `article`-Element auch verschachtelt werden, wenn es der Kontext des Dokuments so vorsieht. Bei einem Online-Zeitungsartikel könnte das folgendermaßen aussehen:

```
<article>
  <h1>Artikel Überschrift Hier</h1>
  <p>Text Text Text ...</p>
</article>
```

¹https://www.tutorialspoint.com/html5/html5_new_tags.htm

5 <h2>Zitat Max Mustermann</h2>
 <p>Text Text Text ...</p>
 </article>
 </article>

2 Praxishinweise Tools

2.1 Editor

Um die benötigten Dateien anzulegen und zu bearbeiten empfiehlt sich ein kostenloser Texteditor wie Notepad++², welcher einige Vorteile gegenüber dem Standardeditor unter Windows hat. Unter Linux kann beispielsweise mit vim oder nano gearbeitet werden. Ein anderer guter Texteditor ist Sublime³, dieser bietet eine Testversion an. Natürlich bietet sich auch die Nutzung eines anderen modernen Editors oder einer IDE an. Hier mag als Beispiel für eine IDE die JetBrains⁴ Suite genannt werden, die (Stand 2022) für Studenten der RUB kostenlos ist. WebStorm und PhpStorm sind Leistungsstarke WebDev IDEs. Auch Visual Studio Code⁵ bietet sich mit entsprechenden Erweiterungen an. Es empfiehlt sich auch besonders Tools zu nutzen, mit denen Bereits Erfahrungen gesammelt wurden. Nach der Wahl der Entwicklungsumgebung bietet es sich an die folgenden Punkte zu testen beziehungsweise zu beachten:

- Files am besten mit UTF-8 Encoding speichern
- <meta charset="UTF-8"> im head stellt richtige Interpretation des Browsers sicher.
- Zum Ausprobieren der Auswirkungen:
 - einfaches Dokument mit Umlauten und Sonderzeichen anlegen
 - Mit eingebauten Konvertierfunktionen in Notepad++ verschiedene Kombinationen von Encodings der Datei mit und ohne die Definition des character encodings im Browser betrachten.
- Syntax Highlighting und Hervorhebung von Tag-Paaren sollte automatisch geschehen.
- Character escapes in HTML5

²<https://notepad-plus-plus.org/downloads/> ↗

³<https://www.sublimetext.com/> ↗

⁴<https://www.jetbrains.com/de-de/> ↗

⁵<https://code.visualstudio.com/> ↗

- Browser sind prinzipiell recht fehlertolerant, aber in unterschiedlichen Ausprägungen. Spezifikationen sollten eingehalten werden, da insbesondere bei komplexen Projekten und dynamischen Inhalten ein ungewolltes Verhalten oder fehlerhafte Darstellung riskiert wird.
- `<` muss mit der Escape Sequence `<` ersetzt werden.
- `&` wird laut Spezifikation nur in uneindeutigen Fällen verboten, jedoch besser immer mit `&` escapen.

2.2 Browser Dev-Tools (z.B. Firefox)

Moderne Browser haben Werkzeuge, die es dem Webdeveloper einfacher machen Inhalte zu erstellen. Beispielsweise der Inspektor⁶ zur Überprüfung/Manipulation von Seitenstruktur und Gestaltung. Diese Tools finden sich bei allen gängigen Browsern in den Developer-Dokumentationen.

- Mouseover über Tags zur Hervorhebung der Elemente in der Seitenstruktur
- Direktes, temporäres Bearbeiten von Tags, Attributen und Styles
- Debugger als wichtiges Tool zur Arbeit mit dynamischen Inhalten in Webseiten
- Aktivierung von Events wie MouseOver

2.3 Ressourcen

Es gibt unzählige Quellen für Webentwickler, hier sind einige Quellen genannt, die den Einstieg möglicherweise erleichtern:

- w3schools⁷ - Einstieg/Tutorial
- SelfHtml⁸ - Einstieg/Tutorial
- Mozilla Developer Network⁹ - Dokumentation

⁶https://developer.mozilla.org/de/docs/Tools/Page_Inspector ↗

⁷<https://www.w3schools.com/html/default.asp> ↗

⁸<https://wiki.selfhtml.org/wiki/Startseite> ↗

⁹<https://developer.mozilla.org/de/docs/mdn> ↗

- HTML Living Standard¹⁰ - Dokumentation

3 Aufgaben

Für einen sehr einfach gehaltenen Blog soll eine HTML-Webseite mit dem Titel “Katzenblog” und der Überschrift “Informationen über Katzen” aufgesetzt werden. Der `body`-Tag des Dokuments soll HTML5 konform in die Bereiche `header`, `main` und `footer` strukturiert sein.

Aufgabe 1

Erstellen Sie den `header`-Block, indem Sie dort eine Tabelle mit zwei Zeilen und zwei Spalten anlegen. Die erste Spalte enthält je Zeile den Namen und das Aktualisierungsdatum. Die zweite Spalte ist durch das Attribut `rowspan` zu einer Zelle zusammengefasst und enthält ein Profilbild oder Logo des Benutzers/Bloggers als `img`-Element. Beschränken Sie die Breite des Bildes in der Anzeige auf 240 Pixel.

Aufgabe 2

Erstellen Sie nun den `main`-Block, indem Sie dort drei `article`-Elemente hinzufügen. Jeder `article` besitzt eine Sub-Überschrift und einen Textblock. Befüllen Sie die `article`-Elemente mit beispielhaften Inhalten.

Aufgabe 3

Erstellen Sie eine Verzeichnis- und Dateistruktur im Wurzelverzeichnis, bestehend aus den Ordnern `other` und `contact`, die jeweils eine `index.html` enthalten. Die beiden HTML-Dateien sollen jeweils Links beinhalten, die zurück auf die Startseite verweisen. Verwenden Sie relative Pfade. Erweitern Sie nun den `footer`-Block um ein `nav`-Element, welches wahlweise den Kontext der Seite zu `contact` oder zu `other` wechselt.

¹⁰<https://html.spec.whatwg.org/> ↗