

Übung 01: HTML

- △ Ziel ist mit HTML und CSS *vertraut zu werden* und *wichtige Konzepte* dahinter zu verstehen. **Richtig eingesetzt** kann Generative KI dafür sehr hilfreich sein!

Aufgabe 1 - Ein einfaches HTML-Dokument

Erstelle ein einfaches HTML-Dokument (ohne CSS!), dass

- eine semantisch korrekte HTML-Dokumentenstruktur aufweist.
- mithilfe von semantischen HTML-Elementen strukturiert ist.
- Seitentitel mit Logo im header enthält.
- eine Navigation mit Links enthält.
- Überschriften, Absätze und Listen enthält.

- ☒ Um schnell effizient zu werden, solltest du einige HTML-Tags auswendig lernen. Eine gute Übung dazu ist, wenn du versuchst das Dokument aus dem Kopf aufzuschreiben, sobald du diese Aufgabe einmal gelöst hast.

Aufgabe 2 - Ein Einstieg in CSS

2.1. Nutze die Entwicklungstools deines Browsers, um die Struktur der Informationsseite für diese Veranstaltung unter <https://webdev.pages.gwdg.de/info/> HTML-Dokuments zu analysieren.

Die Seite beinhaltet folgendes CSS¹:

```
1 body{
2   color: #444;
3   background-color: #EEE;
4   margin: 40px auto;
5   max-width: 650px;
6   line-height: 1.6em;
7   font-size: 18px;
8   padding: 0;
9 }
```

2.2. Verwende die Entwicklungstools und die CSS-Dokumentation unter <https://developer.mozilla.org/>, um herauszufinden, welchen Effekt die einzelnen CSS-Attribute haben. Du kannst in den Entwicklungstools die CSS-Attribute der Seite (lokal = innerhalb deines Browsers) anpassen und die Änderungen sofort sehen. Deaktiviere nacheinander Attribute, um nachzuvollziehen, wie die Seite ohne CSS aussieht.

2.3. Vergleiche die Website (mit CSS) in Firefox und Chromium-basierten Browsern. Gibt es Unterschiede? Finde heraus, warum das so ist.

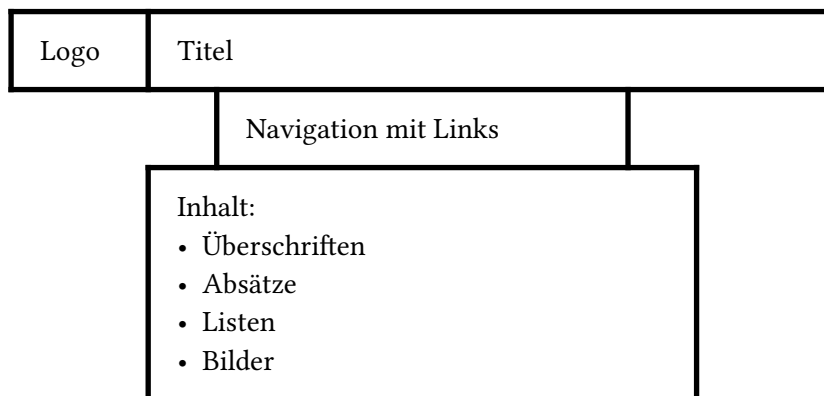
Passes das CSS so an, dass du keine Unterschiede mehr siehst. Speichere dir die Website als HTML-Datei dafür lokal ab und öffne sie z.B. mit dem Live-Server in VSCode.

¹<https://github.com/Zachtreis/bettermotherfuckingwebsite>

Aufgabe 3 - Ein CSS-Dokument

Verwende CSS um das Dokument aus Aufgabe 1 zu stylen. Erstelle dazu eine CSS-Datei `style.css` und binde sie in deinem HTML-Dokument ein. Ergänze deine Styles schrittweise wie folgend:

1. Gib verschiedenen Seitenabschnitten eine (hellere) Hintergrundfarbe, um das Layout „sichtbar“ zu machen.
2. Verwende ein grid-Layout für das globale d.h. grobe Layout der Seite, dass etwa so aussehen könnte:



3. Verwende ein flex-Layout für die Links in der Navigation. Die Links sollten:
 - auf der horizontalen Achse angeordnet sein.
 - horizontal zentriert sein.
 - einen Abstand von 1em zueinander haben.
4. Ändere die Schriftart, -größe und -farbe des gesamten Dokuments.
5. Füge einen Hover-Effekt für die Links in der Navigation hinzu.
6. Füge einen Darkmode hinzu, der sich automatisch an die Systemeinstellungen anpasst.
7. Passe die Seite so an, dass sie auf mobilen Geräten gut aussieht. Dabei sollte sich das Layout an die Bildschirmgröße anpassen, so dass:
 - Logo, Titel und Navigation untereinander angezeigt werden.
 - die Links in der Navigation untereinander angezeigt werden.

Aufgabe 4 - Bulma

Probiere Aufgabe 3 mit dem CSS-Framework [Bulma](<https://bulma.io/>) aus. Du kannst Bulma in dein HTML-Dokument einbinden, indem du den folgenden Link in den head-Bereich deines HTML-Dokuments einfügst:

```
1 <link
2   rel="stylesheet"
3   href="https://cdn.jsdelivr.net/npm/bulma@1.0.2/css/bulma.min.css"
4 >
```



Das solltest du in der Praxis vermeiden, da dadurch dein Dokument von einem externen Server abhängt und personenbezogene Daten (IP-Adresse) an fremde übertragen werden.

Aufgabe 5 - Tailwind und DaisyUI

Probiere Aufgabe 3 mit dem CSS-Framework [Tailwind](<https://tailwindcss.com/>) und [DaisyUI](<https://daisyui.com/>) aus. Du kannst Tailwind genauso wie Bulma in Aufgabe 4 einbinden, allerdings ist das mit DaisyUI etwas komplizierter. Eine einfache Möglichkeit beides auszuprobieren ist <https://daisyui.com/tailwindplay/>.