

Modul 293 - Aufgabenblatt 01

von Lukas Meier

Unterricht vom 02.12.2025

1 Aufgaben

Um die soeben behandelte Theorie zu festigen und um einen ersten Eindruck von der Anwendungsweise von HTML zu erhalten, sollten Sie die folgenden Aufgaben der Reihe nach abschliessen.

1.1 Einrichten der Entwicklungsumgebung

Da es sich bei HTML nur um eine deskriptivve Sprache handelt, welche nativ von jedem modernen Browser verstanden wird, ist keine Installation einer Runtime notwendig. Um den Ihre Dateien optimal zu bearbeiten, sollten Sie jedoch eine IDE oder einen simplen Code-Editor installiert haben.

Beliebte Lösungen hierfür sind: Visual Studio Code, Sublime Text oder NetBeans. Oder, für die fanatischsten unter euch, NeoVim.

Bitte beachten Sie, dass Sie jeden Editor verwenden möchten, den Sie wollen, ich jedoch nur Support für Visual Studio Code, Sublime Text und NeoVim (ausserhalb der Unterrichtszeiten) anbiete. Sollten Sie sich für einen anderen Editor entscheiden, sind Sie selber für dessen Funktionalität verantwortlich.

1.2 Grundlegende HTML-Datei

Laden Sie die Datei `index.html` herunter, welche Sie auf dem Samba-Server finden können.

Öffnen Sie die Datei in Ihrem Editor und studieren Sie die grundlegende Struktur der Datei. Öffnen Sie im Anschluss die gleiche Datei in einem Webbrowser Ihrer Wahl. Was für eine Ausgabe erwarten Sie und wieso? Vergleichen Sie Ihre Erwartung mit dem eigentlichen Output. Lagen Sie korrekt?

1.3 Hello World

Als Informatiker sind wir verpflichtet, als erste Aufgabe den Output 'Hello World' auf unserem Bildschirm ausgeben zu lassen. Erstellen Sie einen Tag vom Typ **h1** mit dem Inhalt 'Hello World'. Öffnen (oder reloaden) sie die Datei im Browser. Wenn Sie den Tag korrekt platziert haben, sollten Sie den Text **Hello World** in grosser Schrift auf dem Screen sehen.

1.4 Paragraf

Erweitern Sie ihre Webseite mit einem Paragrafen. Erstellen Sie einen HTML-Tag unterhalb des **h1**-Tags und befüllen Sie Ihn mit einem Text ihrer Wahl.

1.5 Verschachteln

Unsere Seite beinhaltet nun einen Titel und einen kleinen Einleitungstext. Diese beiden Elemente sind aktuell direkt im **body**-Eingepflegt. Um unsere Seite besser strukturieren zu können, sollten wir diese Elemente in einen Container packen. Erstellen Sie einen neuen Tag vom Typ **header** und packen Sie ihren **h1**- und **p**-Tag in den neuen **header**-Tag.

1.6 Mehr Elemente

Ein guter Header benötigt natürlich eine Navigation, damit sich der User unserer Webseite navigieren kann. Erstellen Sie einen neuen Tag vom Typ **nav** über Ihrem **h1**. Als nächstes befüllen wir unsere Navigation mit Zwei Links. Erstellen Sie in Ihrem **nav**-Tag zwei neue Tags vom Typ **a**. Der Inhalt des ersten Tags soll **Home** sein, den zweiten Tag befüllen Sie mit dem Inhalt **Kontakt**.

Was erwarten Sie passiert, wenn Sie Ihre Webseite im Browser neu laden? Kontrollieren Sie Ihre Erwartungen? Lagen Sie korrekt? Was passiert, wenn Sie die Links in der Navigation anklicken und wieso?

1.7 Die ersten Attribute

Einige Elemente, wie zum Beispiel die vorher angelegten Links funktionieren nicht ordnungsgemäss, wenn gewisse Attribute nicht definiert sind. Links benötigen natürlich ein Ziel worauf sie verlinken sollen.

Erweitern Sie Ihre **a**-Tags mit jeweils einem Attribut 'href'. Der Wert dieses Attributes soll für den Home-Link **/index.html** sein und für den Kontakt-Link **/contact.html**.

1.8 Kontaktseite anlegen

Kopieren Sie die Datei index.html an den selben Ort und benennen Sie diese 'contact.html'. Öffnen Sie die Datei ebenfalls in Ihrem Editor und passen Sie den Inhalt Ihres **h1**-Tags zu 'Kontakt' an. Wenn Sie möchten, können Sie den Inhalt des Paragraphen ebenfalls anpassen.

1.9 Testing

Öffnen Sie die Datei index.html in Ihrem Browser. Wenn Sie alles korrekt gemacht haben, sollten Sie nun mithilfe der Navigation zwischen Ihren beiden Seiten hin- und herwechseln können.

1.10 Externe Links

Erweitern Sie die Navigation in Ihrer Datei index.html um einen weiteren Link, dieser soll als **href** Attribut den Wert <https://google.com> haben. Erfassen Sie zusätzlich ein weiteres Attribut mit dem Namen **target**. Der Wert dieses Attributes setzen Sie auf `_blank` (_ nicht vergessen). Was passiert, wenn Sie diesen Link in Ihrem Webbrowser anklicken? Was macht das Attribut **target**?

1.11 Eine Observation

In Ihrem Browser, klicken Sie nun in der Datei index.html auf den Kontakt-Link. Was fällt Ihnen bezüglich der Navigation auf? Wie beheben Sie diese Inkonsistenz?

1.12 Einbinden einer Bild-Ressource

Nehmen Sie ein Bild/Foto Ihrer Wahl und speichern Sie es im selben Ordner in welchem auch die Dateien index.html und contact.html abgelegt sind. Erstellen Sie nun in der Datei contact.html unterhalb des Paragraphen einen neuen Tag vom Typ **img**. Beachten Sie, dass es sich hierbei um einen selbstschliessenden Tag handelt.

Fügen Sie dem Tag nun ein neues Attribut mit dem Namen **src** (source) hinzu. Der Wert dieses Attributes ist der Name Ihrer Bild-Datei. Also zum Beispiel `mein-hund.jpg`.

1.13 Styling ohne CSS

In der nächsten Lektion werden wir uns mit CSS beschäftigen, damit können Elemente ausführlich gestyled werden. Doch für gewisse Elemente (wie zum Beispiel das Bild) existieren einige simple Attribute welche das Erscheinungsbild des Tags bereits verändern können.

Fügen Sie Ihrem Bild das Attribut `width` hinzu und geben Sie ihm den Wert 200. Was erwarten Sie, hat das für eine Auswirkung auf das Bild?

1.14 Simples Styling

Um Ihnen einen Vorgeschmack auf die kommende Lektion zu geben, werden wir nun zum Abschluss den Paragraf in der index.html-Datei rot färben. Fügen Sie dazu beim Paragraf ein neues Attribut hinzu. Das Attribut heisst: `style`. Der Wert des Attributes setzen sie auf: `color: red;`. Laden Sie die Seite danach in Ihrem Browser neu.

1.15 Bonus: Styling

Können Sie die Farbe des Paragrafen auch auf etwas anderes als rot setzen? Können Sie die Farbe auch ohne Worte, sondern mit Farbcodes definieren? Wie können Sie anstelle der Textfarbe, die Hintergrundfarbe des Paragrafen anpassen? Recherchieren Sie im Internet.