

Web-Grundlagen Teil 1: Einführung in HTML und CSS

Ziel und Quellen

Ziel dieses Kurses ist es, Grundkenntnisse in HTML und CSS zu vermitteln, so dass Sie einfache, statische Webseiten erstellen können.

Die Beispiele in diesem Kurs orientieren sich am Buch «Einstieg in HTML und CSS» von Peter Müller (Rheinwerk Verlag, 2020). Vom selben Autor erschien bereits 2006 das Buch «Little Boxes – Webseiten gestalten mit HTML und CSS». Das ist nicht mehr auf dem neuesten Stand, dafür aber online frei verfügbar.¹

Browser

Für das „Surfen“ im Web werden Sie unterschiedliche Web-Browser benutzen, zum Beispiel Edge, Safari, Firefox, Chrome oder Opera. Als Webentwickler sollte man normalerweise mehrere Browser im Einsatz haben, um zu testen, ob das Ergebnis der Arbeit in den gängigen Browsern gut aussieht und verwendbar ist.

Die meisten Browser bringen heute für Web-Entwickler geeignete Werkzeuge mit, um den Aufbau von Webseiten zu untersuchen und mögliche Fehler zu lokalisieren. Diese Werkzeuge können durch Browser-Erweiterungen noch ergänzt werden.

Damit wir eine gemeinsame Basis haben, beziehen sich die folgenden Aufgaben jeweils auf den Firefox-Browser, der für Windows, Macs und Linux gleichermassen verfügbar ist. Selbstverständlich können die Aufgaben aber auch mit anderen Browsern und deren zugehörigen Entwicklertools durchgeführt werden.

Aufgabe (fakultativ): Installieren Sie die aktuelle Version des *Firefox*-Browsers. Diese bringt bereits grundlegende Entwicklertools mit. Alternativ oder zusätzlich können Sie auch die *Firefox Developer Edition* installieren. In dieser Variante werden neuere Webtechnologien testweise unterstützt. Für die Zwecke dieser Einführung reicht sicher die Standardversion. Folgende Erweiterungen (Menü Extras → Addons) sind ausserdem nützlich:

- *Web Developer*² ergänzt die Toolbar um nützliche Funktionen für die Webentwicklung.
- *uBlock Origin*³ ist ein guter Ad-Blocker. Ohne ein solches Werkzeug können viele Websites kaum noch vernünftig genutzt werden. Websites, welche Werbung weniger aufdringlich

¹ <http://little-boxes.de/little-boxes-teil1-online.html>

² <https://chrispederick.com/work/web-developer/>

³ <https://github.com/gorhill/uBlock#ublock-origin>

einsetzen, können vom Ad-Blocker ausgenommen werden, da viele Websites auf Werbeeinnahmen angewiesen sind.

Aufbau einer Webseite


Eine Webseite wird vom Browser aus verschiedenen Bestandteilen erstellt:

- Im HTML-Code wird die logische Struktur der Seite repräsentiert. HTML steht für *Hypertext Markup Language*, es handelt sich bei HTML also um eine *Auszeichnungssprache*.
- In einer oder mehreren CSS-Dateien wird die Darstellung der Seite beschrieben. CSS steht für *Cascading Style Sheets*.
- Zusätzliche können Scripts das Verhalten der Seite oder einzelner Elemente steuern. Diese werden in der Sprache *JavaScript* geschrieben.

In dieser Einführung geht es um die Grundlagen von HTML und CSS. Das Thema JavaScript wird Kurs WBE behandelt.

Aufgaben:

- Öffnen Sie im Firefox die Website <https://nuudel.digitalcourage.de/> – das ist ein Webtool zum Termine finden und Umfragen anzulegen.
- Sehen Sie sich den HTML-Code der geöffneten Seite an: *Extras* → *Browser-Werkzeuge* → *Seiten-quelltext anzeigen*. Oder CMD-U (Mac) bzw. Strg-U (Windows).
- Können Sie die Seiteninhalte im HTML-Quelltext identifizieren? Wo finden Sie den Text der Schaltfläche «Termin finden» im Quelltext? Suchen Sie auch den ersten Punkt der FAQ «Was ist das?» im Quelltext. Schliessen Sie den Quelltext.
- Wählen Sie in der Web-Developer-Erweiterung (Zahnrad-Icon) den Punkt *CSS* → *View CSS* aus. Es werden die verschiedenen CSS-Dateien angezeigt, die die Darstellung der Seite steuern. Können Sie irgendwelche Angaben zur Darstellung, zum Beispiel Farbangaben, finden? Schliessen Sie die Ausgabe der CSS-Dateien.
- Wählen Sie in der Web-Developer-Erweiterung den Punkt *Outline* → *Outline Block Level Elements*. Probieren Sie noch ein paar andere Outline-Varianten aus. Schalten Sie alle Outlines wieder aus.
- Wählen Sie in der Web-Developer-Erweiterung den Punkt *CSS* → *Disable all Styles*. So sieht die Seite ohne die geladenen Stylesheets aus. Nicht mehr so schön, aber immer noch benutzbar. Es gibt aber immer noch gewisse Unterschiede in der Darstellung, zum Beispiel werden Überschriften grösser und Listenelemente eingerückt dargestellt. Das ist die Standard-Darstellung des Browsers. Deaktivieren Sie diese auch noch (*Disable Browser Standard Styles*). Nun sind alle Texte gleich gross. Aktivieren Sie die Browser-Styles und die geladenen Stylesheets (auch Designer-Stylesheets genannt) wieder.

- Es schadet nichts, auch noch ein paar andere Möglichkeiten der Web-Developer-Erweiterung auszuprobieren. Es ist auch nicht schlimm, wenn nicht bei allen Punkten klar ist, was da ein- und ausgeschaltet oder überprüft wird. Man sieht aber bereits, dass sich diese Erweiterung ganz gut für eine schnelle Inspektion einer Seite eignet.
- Deutlich mehr Möglichkeiten bieten die in den Browser eingebauten Entwicklertools. Öffnen Sie diese über *Extras* → *Browser-Werkzeuge* → *Werkzeuge für Entwickler*. Alternativ über CMD+OPT-I (Mac) bzw. Strg-Umschalt-I (Windows). Über das Menü mit den drei Punkten können Sie die Entwickler-Tools innerhalb des Fensters platzieren oder in einem eigenen Fenster öffnen. Die Entwickler-Tools sind umfangreich und es ist sinnvoll, sich Schritt für Schritt damit vertraut zu machen. Zunächst sehen wir uns nur einen sehr kleinen Ausschnitt an.
- Wählen Sie in der Tableiste das Werkzeug *Inspektor*. Nun haben Sie auf der linken Seite den HTML-Code als hierarchische Struktur angezeigt, in der Sie den Inhalt einzelner Elemente anzeigen oder verbergen können. Beim Überfahren von Elementen mit der Maus werden diese auf der Seite hervorgehoben.
- Klicken Sie auf das Icon  links vom Tab Inspektor. Nun können Sie auf ein Element in der Seite klicken und dadurch das Element im Inspektor auswählen.
- Das soll für einen ersten Eindruck der Entwickler-Tools genügen.

Das Web ist nicht aus Papier

Wer gute Webseiten machen will, muss verstehen, dass man als Webdesigner nur eine eingeschränkte Kontrolle darüber hat, wie die Seiten beim Leser dargestellt werden:

- Unterschiedliche Bildschirmgrößen und Größen des Browser-Fensters. Pixeldichte.
- Verschiedene Gerätetypen: Mobilgerät mit Touch-Bedienung, grafischer Browser mit Mausbedienung, Textbrowser in einem (Unix-) Terminal, Braille-Ausgabe zum Tasten für Sehbehinderte, Sprachausgabe, Ausgabe auf TV-Gerät.
- Verschiedene Browser-Einstellungen: Popup-Blocker, Werbeblocker, JavaScript deaktiviert, Einstellung der Schriftart und Schriftgröße.
- Auf dem System installierte Schriftarten.

Aufgabe:

Lesen Sie die Abschnitte 1.1 und 1.2 von „Das Web ist nicht aus Papier“ aus dem Buch „Little Boxes“ durch: <http://little-boxes.de/lb1/1-das-web-ist-nicht-aus-papier.html>. Wie erwähnt ist das Buch schon etwas älter, diese Abschnitte sind aber nach wie vor aktuell.

Wer die eingeschränkte Kontrolle über das Aussehen der Seiten als zu starke Einschränkung empfindet, sollte eher PDF-Dateien anstelle von Webseiten erstellen. **Das Web funktioniert nicht nach dem WYSIWYG-Prinzip** (WYSIWYG: What you see is what you get).

Web-Editor

Nun gehen wir daran, eigene – zunächst sehr einfache – Webseiten zu erstellen. Dazu benötigen wir als Werkzeug einen Web-Editor. Das Ziel sollte sein, die drei Aspekte Inhalt und logische Struktur, Darstellung, Verhalten möglichst klar zu trennen. Ein WYSIWYG-Editor ist daher nicht zu empfehlen.

Prinzipiell kann zum Bearbeiten von HTML- und CSS-Dateien jeder Texteditor benutzt werden. Es ist allerdings von Vorteil, einen Editor zu verwenden, der das Bearbeiten von HTML- und CSS-Code speziell unterstützt:

- Syntax von HTML oder CSS wird durch verschiedene Farben für Elemente, Attribute etc. hervorgehoben. Auf diese Weise behält man leichter den Überblick und findet Fehler schneller.
- Unterstützung beim Eingeben von HTML-Elementen (Vorschlag passender Attribute) oder CSS-Eigenschaften (Vorschlag passender Eigenschaften). Ausserdem eine einfache Möglichkeit, das aktuell offene Tag zu schliessen.
- Eine einfache Möglichkeit zur Vorschau der aktuell bearbeiteten Seite in einem Editor-Fenster oder einem Web-Browser.

Die folgenden Eigenschaften sind nicht ganz so wichtig, erweisen sich aber ebenfalls als hilfreich für das Erstellen von Webseiten:

- Anlegen und Abrufen von häufig benötigten „Textfragmenten“ (Snippets).
- Zusammenfassen von Dateien und Verzeichnissen zu Webprojekten und Funktionen zum Upload auf den Server.

Es gibt zahlreiche Editoren, die mindestens die ersten drei Anforderungen erfüllen, viele davon sind kostenlos verfügbar. Falls Sie bereits mit einem geeigneten Editor arbeiten, können Sie diesen selbstverständlich verwenden. Ansonsten geben wir hier ein paar Empfehlungen:

Visual Studio Code: Der VSCode ist für Windows, Mac und Linux kostenlos erhältlich und heute weit verbreitet. Download: <https://code.visualstudio.com>.

Alternative:

Atom <https://atom.io>

Alternativen für Windows

Notepad++ <https://notepad-plus-plus.org>

PSPad <https://www.pspad.com>

Alternative für Mac

Whisk <https://tumult.com/whisk/> (relativ einfach, \$30)

Alternative für Linux

Bluefish <http://bluefish.openoffice.nl/>

Erste Testseite

Whitespace

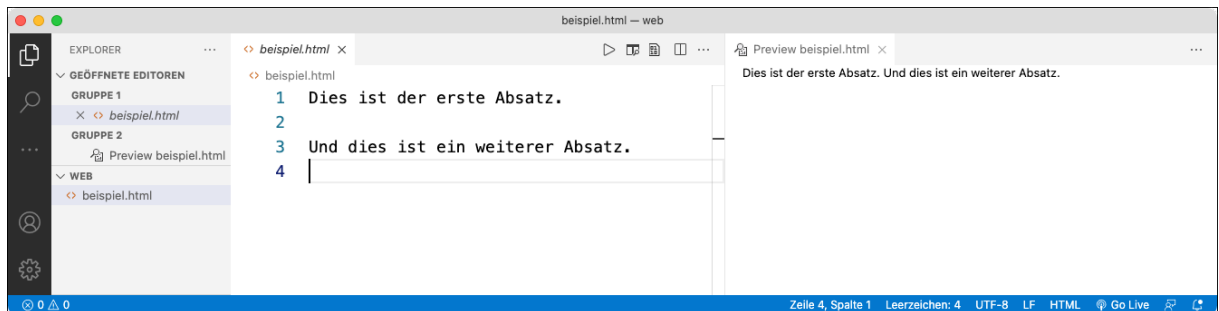
Die erste Variante einer HTML-Seite soll nur zwei Zeilen Text enthalten, mit einer Leerzeile dazwischen:

```
Dies ist der erste Absatz.  
  
Und dies ist ein weiterer Absatz.
```

Aufgaben:

- Erstellen Sie ein neues, leeres Dokument *beispiel.html*. mit diesem Inhalt. Am besten legen Sie für jede Übung ein Verzeichnis an, in dem Sie neben der Aufgabenstellung auch Ihre Ergebnisse ablegen.
- Öffnen Sie die Datei in einem Web-Browser. Was stellen Sie fest?
- Fügen Sie zusätzliche Leerzeilen und Leerschläge zwischen einzelnen Wörtern hinzu. Ändert sich die Darstellung? Wie ist es mit Tabulatorzeichen?

Hinweis 1: Im *VSCode* können Sie eine Erweiterung *Preview* installieren, die es erlaubt, eine Vorschau des aktuell bearbeiteten Dokuments anzuzeigen:



Hinweis 2: Alternativ (oder zusätzlich) können Sie im *VSCode* eine Erweiterung *Live Server* installieren. Durch einen Klick auf «Go Live» unten rechts in der Statusleiste (Bild) wird ein Webserver gestartet und die Seite im Browser angezeigt. Beim Speichern des Dokuments wird die Vorschau im Browser aktualisiert. Voraussetzung dafür ist allerdings, dass das HTML-Dokument in einem Verzeichnis abgelegt wird und dieses als Projektverzeichnis in den *VSCode* geöffnet wird. Ausserdem muss fürs *Live Reloading* eine korrekte HTML-Seite bearbeitet werden. Die zwei Textzeilen im Beispiel sind dafür noch nicht ausreichend.

Absätze: p für paragraph

Aufgabe: Betten Sie die beiden Absätze in p-Tags ein:

```
<p>Dies ist der erste Absatz.</p>  
  
<p>Und dies ist ein weiterer Absatz.</p>
```

Wie ist die Darstellung jetzt? Ändert sich etwas, wenn man im HTML-Code alles in eine Zeile schreibt? Was schliessen Sie aus den bisherigen Ergebnissen?

Komplettes Dokument

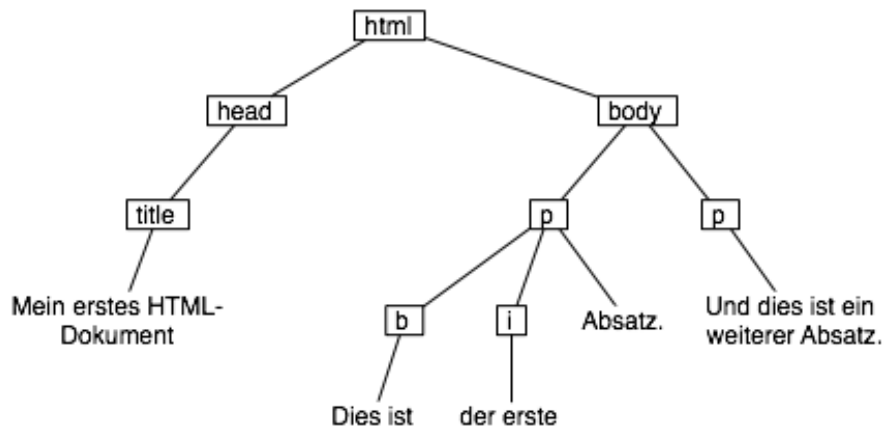
Jetzt wird das Dokument zu einem (fast) kompletten HTML-Dokument erweitert:

```
<html>  
  <head>  
    <title>Mein erstes HTML-Dokument</title>  
  </head>  
  <body>  
    <p>Dies ist der erste Absatz.</p>  
    <p>Und dies ist ein weiterer Absatz.</p>  
  </body>  
</html>
```

Aufgaben:

- Erweitern Sie Ihr Dokument entsprechend der Vorlage.
- Taucht der Titel des Dokuments ebenfalls in der Ausgabe auf?

Durch das Klammern von Teilen des Dokuments mit Start- und End-Tags entsteht eine hierarchische Struktur. Die Einrückung der Zeilen dient nur zur besseren Lesbarkeit und hat keine Auswirkungen auf die Darstellung. Die hierarchische Struktur ist in dem folgenden Baumdiagramm veranschaulicht:



Die Baumstruktur in der Abbildung enthält zwei kleine Abweichungen gegenüber dem oben angegebenen Dokument: Es wurde ein b- und ein i-Element hinzugefügt.

Aufgaben:

- Versuchen Sie, diese Änderung im Ihrem HTML-Dokument nachzuführen. Was erreicht man mit `` und `<i>`?
- Fügen Sie dem Dokument schliesslich noch zwei Überschriften hinzu, die erste mit `<h1>` ausgezeichnet und die zweite mit `<h2>`.

Umlaute und Sonderzeichen

Aufgaben:

- Ändern Sie den Text so ab, dass er Umlaute ä, ö, oder ü enthält. Werden diese im Browser richtig angezeigt?
- Normalerweise wird Ihr Editor den HTML-Quellcode in UTF-8-Zeichencodierung speichern. Durch Klick auf «UTF-8» in der Statusleiste des VSCode können Sie das ändern. Achten Sie aber darauf, dass schliesslich wieder UTF-8 eingestellt ist.
- Für die korrekte Anzeige des Dokuments teilen wir dem Browser die Zeichencodierung mit. Probieren wir einmal eine von UTF-8 abweichende Codierung. Fügen Sie im *head* vor dem *title*-Element folgendes ein:
`<meta charset="iso-8859-1">`
Das ist die falsche Zeichencodierung. Die Umlaute werden nicht korrekt angezeigt.
- Ersetzen Sie zum Test einen der Umlaute durch die folgende umständliche Schreibweise (inklusive & und dem Semikolon am Ende): statt ä neu `ä` oder statt ö neu `ö` oder statt ü neu `ü` – diese als *Entitäten* bezeichneten Sequenzen erlauben es, Umlaute und Sonderzeichen in ASCII-Zeichencodierung darzustellen. Das ist heute aber kaum noch nötig, das heisst wir verwenden einfach UTF-8.

- Geben Sie nun die korrekte Zeichencodierung an:
`<meta charset="utf-8">`
Alle Umlaute und Sonderzeichen sollten nun korrekt angezeigt werden.

Wichtig: Die im Editor eingestellte Zeichencodierung muss mit der im HTML-Dokument angegebenen Zeichencodierung übereinstimmen.

Bild einfügen

Jetzt soll noch ein Bild auf der Seite eingefügt werden. Sie können dafür eine beliebige Bilddatei in den Formaten PNG, JPEG oder GIF verwenden. Oder die Datei *frackwoche.jpg* aus den Unterlagen. Legen Sie die Datei in das gleiche Verzeichnis wie Ihre HTML-Datei und fügen Sie das Bild folgendermassen auf der Seite ein (natürlich mit dem richtigen Namen der Bilddatei):

```
<p>  
    
</p>
```

Mit dem *img*-Element werden Bilder auf der Website eingefügt:

- Das *src*-Attribut spezifiziert die zu ladende Bilddatei.
- Mit *alt* wird ein Alternativtext angegeben, für den Fall, dass die Seite ohne Bilder dargestellt wird (Textbrowser, Braille-Ausgabe, Sprachausgabe, ...). Manche Browser zeigen den *alt*-Text als Tooltip an, wenn man mit dem Mauszeiger über das Bild fährt. Das ist nach der HTML-Spezifikation eigentlich nicht vorgesehen, denn dafür ist eigentlich das *title*-Attribut vorgesehen.
- Die Breiten- und Höhenangabe dient dazu, dass der Browser den für das Bild nötigen Platz bereits reservieren kann, bevor die Bilddatei geladen ist. Die Angabe ist in Pixel, die Einheit wird aber nicht dazugeschrieben.
- Das *img*-Element ist ein leeres Element, es besteht nur aus einem Tag und nicht wie beispielsweise *p* aus einem Start- und einem End-Tag. Leere Elemente werden gelegentlich mit einem Schrägstrich am Ende geschrieben:
``
Diese Schreibweise für leere Elemente stammt von XML (Extensible Markup Language) und ist auch in HTML zulässig aber nicht vorgeschrieben.
- Das *img*-Element ist im Beispiel in ein *p* eingebettet, kommt hier also in einem eigenen Absatz vor.

Aufgaben

- Können Sie die Anzeigegrösse des Bilds reduzieren, indem Sie die Höhe und Breite im HTML-Code modifizieren? Die Alternative, um zu einem kleineren Bild zu kommen, ist, das Bild zunächst in einem Bildverarbeitungsprogramm zu verkleinern und dann die neue Variante des Bilds im *img*-Element zu referenzieren. Welche Variante ist vorzuziehen und warum?
- Verschieben Sie das Bild in ein spezielles Verzeichnis *images* und passen Sie die Angabe im *img*-Element an. Es ist sinnvoll, für eine Website eine übersichtliche Ordnerstruktur anzulegen. Spätestens wenn die Website aus vielen Dateien besteht, verliert man sonst leicht die Übersicht.
- Öffnen Sie das Bild in einem Bildverarbeitungsprogramm (zum Beispiel *Paint* unter Windows oder *Vorschau* auf dem Mac). Speichern Sie es in verschiedenen Formaten. Browser können normalerweise die Formate JPG, PNG und GIF darstellen. Welche Einstellungen gibt es jeweils beim Abspeichern? Vergleichen Sie Dateigrössen und Bildqualitäten.

Hinweis: Manche Bildbearbeitungsprogramme (zum Beispiel Photoshop) haben eine Funktion "Für Web speichern". Diese sollten Sie nutzen, da die Bilder so in möglichst kleinen Dateien gespeichert werden und überflüssige Bestandteile (wie ein Icon mit einer Bildvorschau) weggelassen werden. Kleinere Bilder führen zu kürzeren Ladezeiten von Webseiten.

Element, Tag, Attribut

Es wurden nun bereits Begriffe wie Element, Attribut, Start- und End-Tag verwendet. **Elemente** in HTML dienen unter anderem zur Strukturierung des Inhalts oder dem Einbetten von Bildern oder anderen Dingen in die Seite. Das *p*-Element markiert Absätze, das *title*-Element beschreibt den Titel des Dokuments, *h1* und *h2* (es geht tatsächlich bis *h6*) dienen zur Auszeichnung von Überschriften, und mit dem *img*-Element werden Bilder in die Seite eingefügt.

Im Dokument kommen die Elemente in Form von **Tags** vor: `<p>` ist das Start-Tag eines *p*-Elements und `</p>` das End-Tag. Start- und End-Tag umschliessen den Inhalt, der als Absatz ausgezeichnet werden soll.

Um ein Element mit zusätzlichen Informationen ausstatten zu können, gibt es die **Attribute**. Das *src*-Attribut des *img*-Elements gibt den Namen der Bilddatei an, das *alt*-Attribut einen alternativen Text. Das *img*-Element ist zudem ein leeres Element, das keinen Inhalt umschliesst. Es wird ohne End-Tag geschrieben:

```

```

Aufgabe:

Lesen Sie den Abschnitt 2.6 „Wissenswertes zu HTML-Elementen“ im Buch „Little Boxes“:

<https://little-boxes.de/lb1/2.6-wissenswertes-zu-html-elementen.html>