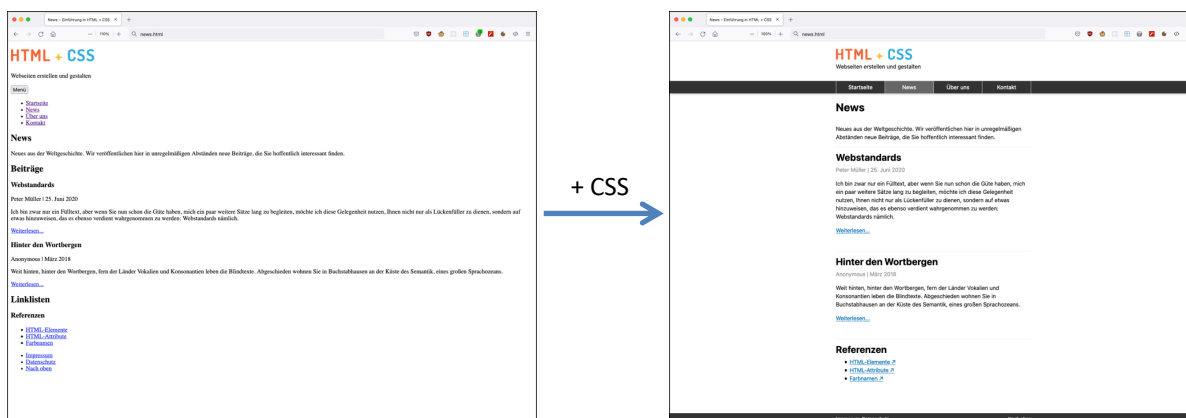


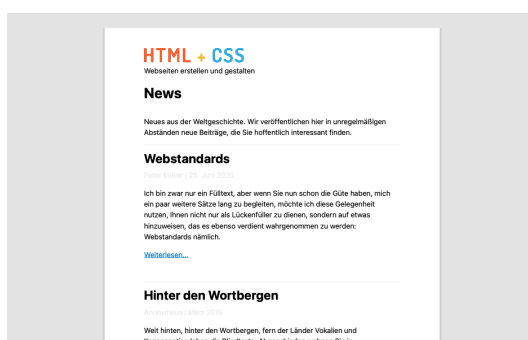
Web-Grundlagen Teil 3: Einführung in HTML und CSS

HTML und CSS

Während HTML den Inhalt einer Seite beschreibt und strukturiert, wird mit CSS (Cascading Style Sheets) die Darstellung beschrieben. Das letzte Beispiel aus der vorangehenden Übungslektion würde ohne CSS – mit der Standard-Darstellung des Browsers – beispielweise so aussehen, wie in folgendem Bild links dargestellt. Erst mit CSS ergibt sich die Darstellung auf der rechten Seite:



Das gleiche HTML-Dokument kann mit anderen CSS-Regeln komplett anders aussehen. Zudem lassen sich CSS-Regeln angeben, die nur in bestimmten Situationen angewendet werden, zum Beispiel beim Drucken des Dokuments, oder abhängig von der Grösse des Displays. Beim Drucken kann man zum Beispiel die Navigation weglassen:



Es zeigt sich, dass die Kombination von HTML und CSS mächtig und flexibel genug ist, um ein Dokument angepasst auf verschiedene Situationen auszugeben. Voraussetzung dafür ist eine Trennung von Struktur/Inhalt und Darstellung.

CSS einbinden

Es gibt verschiedene Möglichkeiten, CSS mit HTML zu verbinden. Wir werden diese nun anhand einer einfachen Beispieldatei ausprobieren.

Aufgaben:

- Öffnen Sie die Datei *index.html* aus den Dateien zu dieser Lektion im Code-Editor und starten Sie eine Vorschau (im VSCode zum Beispiel mit dem Live Server).
- Ergänzen Sie das Start-Tag des mittleren Absatzes um eine *style*-Angabe und überprüfen Sie die resultierende Darstellung.

```
<p style="font-weight:bold">Nulla consequat ...</p>
```

- Mit dem *style*-Attribut lässt sich die Darstellung eines Elements direkt beeinflussen. *font-weight* ist eine von vielen CSS-Eigenschaften, die es erlauben, das Aussehen der HTML-Elemente zu spezifizieren. Halten Sie es für gut oder schlecht, die Darstellung in *style*-Attributen zu beschreiben? Begründung? Denken Sie an das Ziel, Inhalt/Struktur und Darstellung getrennt zu halten.
- Entfernen Sie das *style*-Attribut wieder aus dem *p*-Element und fügen Sie die folgenden Zeilen in *head* des HTML-Dokuments nach dem Titel ein:

```
<style>
  p { font-weight: bold; }
</style>
```

- Nun werden alle Absätze **bold** ausgegeben. Können Sie sich vorstellen, warum? Welche Rolle dürfte das *p* vor der geschweiften Klammer und welche Rolle der Teil in der Klammer spielen?

Zwischen dem Start- und End-Tag des *style*-Elements befindet sich eine *CSS-Regel*. Diese besteht aus einem *Selektor* *p* und den *CSS-Eigenschaften* nebst Werten in geschweiften Klammern. Der Typ-Selektor bezieht sich auf alle Elemente von einem bestimmten Typ, zum Beispiel alle *p*-Elemente. Mit dem Typ-Selektor ist es zum Beispiel nicht möglich, nur den mittleren Absatz auszuwählen. Wir werden daher noch weitere Selektoren ansehen.

Beide Varianten, Stile mit HTML zu verbinden, also das *style*-Attribut und das *style*-Element, betten die Stile in das HTML-Dokument ein, was dem Ziel, Inhalt/Struktur und Darstellung getrennt zu halten, widerspricht. Im nächsten Schritt wird der CSS-Teil daher in eine eigene Datei ausgelagert.

Aufgaben:

- Kopieren Sie die CSS-Regel (ohne die Tags des *style*-Elements) in eine eigene Datei und nennen Sie diese *styles.css*. Löschen Sie dann das *style*-Element aus der HTML-Datei.

- Nun muss in der HTML-Datei noch die Verbindung zur CSS-Datei hergestellt werden. Dies geschieht mit Hilfe eines *link*-Elements im *head* der HTML-Datei:

```
<link rel="stylesheet" href="styles.css">
```

- Überprüfen Sie, ob die Absätze im Browser immer noch fett dargestellt werden, mit anderen Worten, ob die CSS-Datei bei der Darstellung berücksichtigt wird.
- Verschieben Sie die CSS-Datei in ein Unterverzeichnis *css* und passen Sie den Pfad im *link*-Element an. Überprüfen Sie, ob alles noch funktioniert.
- Erweitern Sie die Regel in der CSS-Datei um eine *color*-Angabe:

```
p {  
  font-weight: bold;  
  color: red;  
}
```

Der Text müsste nun rot und fett dargestellt werden.

Wir haben drei Möglichkeiten gesehen, HTML mit CSS zu verbinden:

- Mit dem *style*-Attribut direkt beim HTML-Element. Dies ist wegen der Vermischung von logischer Auszeichnung und Darstellung selten sinnvoll.
- Mit dem *style*-Element im *head* der HTML-Datei. Dieses enthält CSS-Regeln. Die Verbindung zu den HTML-Elementen wird über Selektoren festgelegt. Für seitenspezifische Ergänzungen zu einem globalen Stylesheet kann dies gelegentlich sinnvoll sein.
- In einer separaten CSS-Datei, die im HTML mit dem *link*-Element angegeben wird. Dies ist im Sinne einer Trennung von Inhalt und Darstellung die empfohlene Variante.

Weitere Vorteile der dritten Variante (externes Stylesheet) sind:

- Die Darstellung ist an einer zentralen Stelle beschrieben. So kann eine kleine Änderung in der CSS-Datei die Darstellung auf der ganzen Website beeinflussen (die vielleicht aus hunderten HTML-Dateien besteht).
- Geringere Dateigrößen: Die Angaben zum Aussehen der Seite müssen nicht auf jeder Seite wiederholt werden, was die Ladezeiten verkürzt. Beim ersten Aufruf muss zwar zusätzlich eine CSS-Datei geladen werden. Diese legt der Browser aber in seinem Cache-Speicher ab. Dadurch muss die CSS-Datei bei weiteren Aufrufen nicht erneut geladen werden.

Neben den Möglichkeiten, HTML und CSS zu verbinden, geben die ersten Beispiele auch Hinweise zu CSS selbst. Die Darstellung wird durch Deklarationen beschrieben, welche Eigenschaften (*color*) und Werte (*red*) enthalten. Wenn die Deklarationen nicht direkt mit einem *style*-Attribut mit einem HTML-Element verbunden werden, muss noch angegeben werden, für welche Elemente im Dokument die Deklarationen gelten sollen. Dazu dienen die so genannten *Selektoren*.

Selektoren

Sehen Sie sich noch einmal die CSS-Regel in der Datei *screen.css* an. Der Selektor dieser Regel besteht nur aus einem *p*. Dieser so genannte *Typselektor* wählt alle *p*-Elemente der HTML-Datei aus. Nehmen wir ein weiteres Element dazu.

Aufgaben:

- Umschliessen Sie die Wörter "dolor sit amet" des ersten Absatzes im HTML-Dokument mit:
`...`
- Wenn Sie nicht mehr wissen, was das *span*-Element bewirkt, schauen Sie noch einmal in Teil 2 dieses Tutorials. Durch das Hinzufügen des *span*-Elements sollte sich in der Darstellung unserer Seite zunächst nichts ändern.
- Fügen Sie der CSS-Datei eine weitere Regel hinzu, die den durch *span* umschlossenen Bereich blau darstellt.

Durch die neue Regel werden alle *span*-Elemente im Dokument blau dargestellt. Das ist in diesem einfachen Beispiel noch akzeptabel, aber es ist bereits klar, dass man in der Regel präzisere Angaben der auszuwählenden Elemente benötigt. Dazu gibt es zahlreiche weitere Selektoren, zum Beispiel den *Klassenselektor*.

Aufgaben:

- Ergänzen Sie das Stylesheet um eine dritte Regel (beachten Sie den Punkt in der ersten Zeile vor *markiert*, dieser gibt an, dass es sich um eine Klasse handelt):

```
.markiert {  
    background-color: #ff9;  
}
```

Dieser Selektor wählt alle Elemente mit der Klasse *markiert* aus. Die Eigenschaft *background-color* dürfte selbsterklärend sein.

- Nun müssen die betreffenden Elemente aber auch im HTML-Dokument mit der Klasse ausgezeichnet werden. Dies geschieht mit dem *class*-Attribut. Fügen Sie dem dritten Absatz (drittes *p*-Element) die Klasse *markiert* hinzu und überprüfen Sie das Ergebnis in der Browser-Vorschau. Die Farbangabe *#ff9* beschreibt den roten, grünen und blauen Anteil der Farbe: viel rot und grün, weniger blau ergibt einen gelben Farbton.
- Mehrere Elemente können die gleiche Klasse haben. Fügen Sie auch der Überschrift die Klasse *markiert* hinzu. Nun sollte sowohl die Überschrift als auch der dritte Absatz mit der Hintergrundfarbe dargestellt werden.
- Der Klassenselektor kann auch mit dem Typ kombiniert werden, um die Auswahl weiter einzuschränken. Ändern Sie den Selektor in der CSS-Regel von *.markiert* zu *p.markiert*. Jetzt werden durch den Selektor alle *p* mit der Klasse *markiert*, aber zum Beispiel nicht das *h1* mit der Klasse *markiert* ausgewählt.

Mit der Möglichkeit, HTML-Elemente über ihre Klasse auszuwählen, haben wir bereits viele Situationen abgedeckt. Mit dem *class*-Attribut kann man übrigens auch mehrere Klassen kombinieren, indem man sie mit Leerzeichen getrennt schreibt. Im folgenden Beispiel hat der Absatz die Klassen *intro*, *wichtig* und *neu*:

```
<p class="intro wichtig neu">...</p>
```

Auch in CSS-Regeln können mehrere Klassen gefordert werden:

```
p.intro.wichtig {  
  font-size: 120%;  
}
```

Zwischen den Klassenangaben im CSS-Selektor darf dabei kein Leerzeichen stehen. Das ist als Selektor zwar auch zulässig, hat aber eine andere Bedeutung.

Neben *class* ist *id* ein weiteres Universalattribut, das in fast allen HTML-Elementen vorkommen kann. Mit *id* wird ein bestimmtes Element eindeutig identifiziert, jede *id* darf daher nur einmal im Dokument vorkommen.

Aufgaben:

- Ergänzen Sie das Stylesheet um die folgende Regel:

```
#zusammenfassung {  
  border: 1px solid #999;  
}
```

Dieser Selektor wählt alle Elemente mit der Klasse *markiert* aus. Die Eigenschaft *background-color* dürfte selbsterklärend sein.

- Fügen Sie dem ersten Absatz die *id* *zusammenfassung* hinzu und überprüfen Sie das Ergebnis in der Browser-Vorschau.

Nun kommen wir so langsam an die Grenze dessen, was wir mit dem einfachen Beispieldokument ausprobieren können. Es sieht mittlerweile auch schlimm genug aus... ☺

Textformatierung

Wir sind jetzt bereits einigen CSS-Eigenschaften begegnet: *color*, *background-color*, *font-weight*, *font-size*, *border*. Das ist aber nur eine sehr kleine Teilmenge der verfügbaren CSS-Eigenschaften. Eine grössere Liste finden Sie hier:

<https://cssreference.io>

Konzentrieren wir uns zunächst auf eine Reihe von Eigenschaften zur Textformatierung.

Aufgaben:

- Die Schriftart kann mit der Eigenschaft *font-family* ausgewählt werden. Eine kurze Beschreibung mit Beispielen finden Sie hier: https://www.w3schools.com/css/css_font.asp.
Warum werden in der *font-family*-Eigenschaft normalerweise mehrere Schriftarten angegeben? Was versteht man unter einer generischen Schriftartenangabe und warum wird diese in der Liste der Schriftarten am Ende angegeben?
- Sehen Sie sich unter *cssreference.io* die Dokumentation zu den folgenden Eigenschaften an und probieren Sie diese im Beispieldokument aus. Am besten setzen Sie das Beispieldokument dazu wieder auf den Ausgangszustand zurück.
 - *text-align*
 - *text-decoration*
 - *text-indent*
 - *text-transform*
 - *line-height*
 - *letter-spacing*
- Können Sie mit Hilfe der *text-decoration*-Eigenschaft die Unterstreichung von Verweisen entfernen? Achten Sie in jedem Fall darauf, dass Verweise immer als solche erkennbar sind.

Vererbung

Viele CSS-Eigenschaften werden in der hierarchischen HTML-Struktur an untergeordnete Elemente vererbt. Hier ein Beispiel:

```
<p>Dieses Beispiel ist <span class="warnung">Vorsicht</span> zu geniessen</p>
```

Angenommen, für den ganzen Absatz ist die Schriftfarbe auf rot gesetzt. Selbstverständlich wird dann auch das Wort „Vorsicht“ im *span*-Element rot dargestellt. Der Wert der Eigenschaft *color* wird also vom *p*-Element zum in der HTML-Struktur untergeordneten *span*-Element vererbt. Das *span*-Element kann die geerbte Farbe aber durch eine eigene *color*-Definition überschreiben.

Aufgaben:

- Sehen Sie sich die Definition der Eigenschaft *color* an: <https://devdocs.io/css/color>
Können Sie der Beschreibung irgendwo entnehmen, dass die Eigenschaft vererbt wird?
- Testen Sie das Vererben und Überschreiben der Eigenschaft *color* im Beispieldokument.
- Wenn in einem Absatz, dessen Textfarbe auf rot gesetzt ist, ein Verweis vorkommt, erbt dieser dann die rote Farbe oder wird die Farbe für den Verweis überschrieben? Können Sie erreichen, dass der Verweis zum Beispiel grün dargestellt wird?

Grössenangaben

In einem der Beispiele oben wurde die Schriftgrösse prozentual vergrössert:

```
p.intro.wichtig {  
  font-size: 120%;  
}
```

Grössenangaben bestehen in der Regel aus einer Zahl und einer Einheit, wobei zwischen Zahl und Einheit kein Leerzeichen stehen darf. Wenn die Zahl 0 ist, lässt man die Einheit weg: 0px oder 0cm sind einfach: 0.

Bei verschiedenen Eigenschaften werden Grössen- bzw. Längenangaben benötigt. Dazu gehören Abstände, die Breite und Höhe von Boxen, die Dicke von Rahmen, oder eben die Schriftgrösse. Es gibt eine ganze Reihe von Möglichkeiten, Grössen zu beschreiben. Hier werden wir nur eine Auswahl davon ansehen.

- 1em entspricht der für ein Element berechneten Schriftgrösse, ermittelt aus der vom Browser für das Wurzelement festgelegten Schriftgrösse und geerbten *font-size*-Angaben von übergeordneten Elementen.
- 1rem entspricht der Schriftgrösse der Wurzelements.
- 1px steht für ein Pixel. Wie gross dieses genau ist, hängt von verschiedenen Faktoren ab. Natürlich denkt man zunächst an einen Bildpunkt eines Displays. Dessen Grösse kann aber stark variieren und auf heutigen hochauflösenden Displays extrem klein sein. Daher ist die Grösse eines Pixels auf Bildschirmen anhand eines Referenzpixels definiert, so dass es für das Auge immer ungefähr gleich gross erscheint. Ein Pixel auf einem hochauflösenden Smartphone-Display dürfte also kleiner sein als auf einer elektronischen Reklametafel am Strassenrand. Bei Medien mit absoluten Grössen (zum Beispiel beim Ausdrucken auf Papier) gilt: 1px entspricht 1/96 von einem Inch.
- 1in (Inch), 1pt (entspricht 1/72in), 1cm, 1mm entsprechen den physikalischen Grössen und sind eigentlich nur sinnvoll auf Medien mit absoluten Grössen, zum Beispiel auf Papier.
- 100vw ist die Breite des *Viewports*, also dem Bereich des Browserfensters, in dem die Webseite angezeigt wird, 100vh die Höhe des *Viewports*.

Das ist nur ein Teil der möglichen Grössenangaben. Alle Möglichkeiten sind mittlerweile in einem eigenen CSS-Standard «CSS Values and Units Module Level 3» beschrieben:

<https://www.w3.org/TR/css-values-3/>

Die nächste Version (Level 4) ist ebenfalls bereits in Arbeit.

Aufgaben:

- Ergänzen Sie das HTML-Beispieldokument um ein leeres *div*-Element mit der Klasse *box*:

```
<div class="box"></div>
```

- Ergänzen Sie die CSS-Datei um folgende Regel:

```
.box {  
  width: 100px;  
  height: 100px;  
  background-color: #fc9;  
}
```

- Sie sollten nun ein gelbes Quadrat auf der Seite sehen. Machen Sie ein paar Versuche mit den oben aufgeführten Grössenangaben, indem Sie die Breite und Höhe des *div*-Elements modifizieren. Alternativ können Sie auch mit der Schriftgrösse experimentieren.

Farbangaben

Farben können in CSS auf verschiedene Art und Weise angegeben werden, etwa hexadezimal mit Rot-, Grün- und Blauanteil: *#fc9* steht für einen gelben Farbton. Auch Farbnamen wie *red*, *yellow* oder *MediumSeaGreen* können verwendet werden.

Aufgabe:

- Lesen Sie die Abschnitte Colors, RGB und HEX im CSS_Tutorial von W3Schools:
https://www.w3schools.com/css/css_colors.asp

CSS-Referenzen

- <https://cssreference.io>
Liste von CSS-Eigenschaften mit Beschreibung
- <https://devdocs.io/>
Verschiedene Referenzen, unter anderem auch für CSS
- <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference#index>
CSS-Referenz des Mozilla Developer Network
- <https://www.w3.org/Style/CSS/current-work.en.html>
Überblick über die CSS-Standards