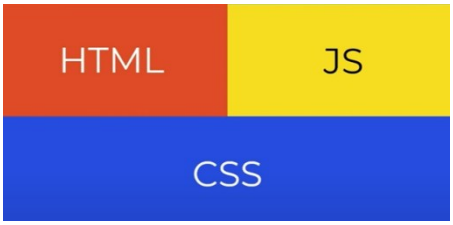
Was ist CSS?



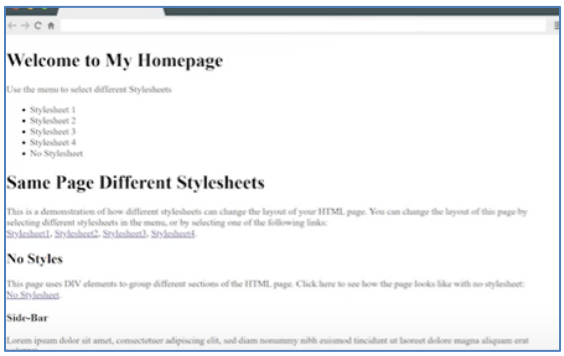
Ist eine vom W3C deklarierte Sprache zur Ausgabeformatierung von strukturierten Dokumenten wie zum Beispiel HTML.

Ist ein wichtiger Bestandteil für Websites mit HTML und JavaScript gemeinsam.

CSS steht für **Cascading Stylesheet**.

- Stylesheet bedeutet so viel wie Formatvorlage.
- CSS ist die Formatierungssprache für Webseiten. Der „Mode-Designer".
- Der Einsatz von CSS bietet viele **Vorteile**, denn CSS ermöglicht eine **strenge Trennung** von Inhalt und Layout: Damit lassen sich Webseiten **besser warten und leichter aktualisieren.** Weil alle Informationen zum Layout der HTML-Seiten zentral in einer CSS-Datei gespeichert werden, lässt sich eine Website, die z. B. aus mehreren hundert Seiten besteht, mit wenig Zeitaufwand komplett umgestalten. Der Webdesigner muss lediglich in der CSS-Datei seine Änderungen vornehmen und sobald er diese speichert, sind alle HTML-Seiten bereits umgestaltet.
- Ein Stylesheet kann Richtlinien für Position, Schriftgröße, Schriftart, Hintergrundfarbe und –bild, Höhe, Breite und vieles mehr enthalten.

Reines HTML:



Mit CSS gestylt:

Unterschied HTML und CSS

a)HTML – Beschreibung der Struktur – WAS wird angezeigt?

HTML wird oft als „Programmiersprache" bezeichnet, was allerdings nicht zutrifft. Es handelt sich um eine „Auszeichnungs- oder Markup-Sprache". Während eine Programmiersprache (wie z.B. JavaScript) Abläufe und Vorgänge beschreibt, dient eine Auszeichnungssprache dazu, die STRUKTUR von in Textform vorliegender Information inb einer semantischen (ihrer Bedeutung entsprechenden) Form zu formulieren.

b)CSS – Beschreibung der Präsentation – WIE wird es angezeigt?

Ermöglicht eine optimale Präsentation der Inhalte.
Der Begriff „cascade“ bedeutet auf Deutsch „Kaskade“. Das wiederum heißt folgendes:

Die Kaskade ist ein **Ablauf**, nach dem der Browser entscheidet, wie er eine CSS-Deklaration zu behandeln hat.
Der Begriff **Kaskade** in CSS (Cascading Style Sheets) bezeichnet das Regelwerk zur Priorisierung und Zusammenführung mehrerer CSS-Regeln, wenn mehrere Styles auf dasselbe HTML-Element angewendet werden können. Dabei entscheidet die Kaskade, welche Regel letztlich gewinnt und wirksam wird.

Eine Kaskade kann man sich gut als **mehrteiligen Brunnen** vorstellen, in dem das Wasser von der obersten Schüssel in die untere usw. fließt. Stufe für Stufe. Oder in der Chemie als hintereinandergeschaltete Gefäße.

Die Kaskade ist eine Hilfe für den Browser. Dabei muss der **Browser** in Bruchteilen einer Sekunde die Befehle abarbeiten und sich an Regeln halten. Treffen nämlich mehrere CSS-Deklarationen für ein Element innerhalb eines oder mehrerer Stylesheets eines Dokuments aufeinander, so muss der Browser entscheiden, welche der ihm vorliegenden Deklarationen er zu beachten hat.

Die Priorität wird in folgender Reihenfolge bestimmt:

1. **Spezifität** der Selektoren (z. B. #id ist stärker als .klasse, die wiederum stärker ist als ein Element-Selektor wie div)
2. **Herkunft** des Styles (z. B. Benutzer-Styles, externe Stylesheets, Inline-Styles)
3. **Reihenfolge im Code** – bei gleicher Spezifität gilt die zuletzt definierte Regel

Beispiel: ein <p>-Element kann viele Eigenschaften haben, wie Schriftfarbe, HIntergrundfarbe, usw., eventuell eine Veränderung durch ein id-Element oder eine Klasse. Was gilt nun?

Daher gilt: Die letzte Deklaration für ein Element, welches direkt beim Element steht, hat das höchste Gewicht. Frei nach dem Motto „Der Letzte gewinnt!“. Inline-CSS über dem style-Attribut sind höherwertiger als Deklarationen die z.B. in externe CSS-Dateien stehen.

Was ist alles wichtig zu lernen in CSS?

- Korrekte Syntax verwenden
- Wie verbindet man es mit HTML?
- Farben verwenden
- Größe z.B. der Schrift, width, height eines Elements
- Rahmen – border: 4px solid #525252;
- Box-Modell (margin, padding)
- Selektoren (class, ID, Elemente)

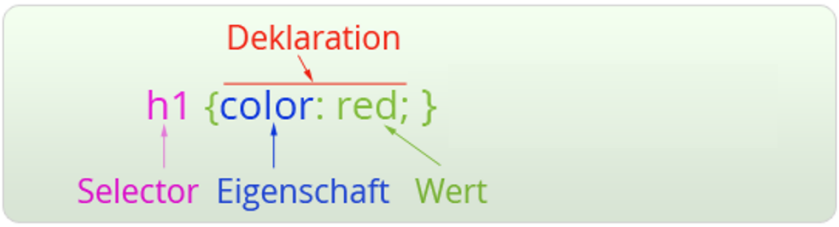
- Display
- Positionierung – position: absolut;
- Responsives Design – media queries - @media (max-width: 480px;)

Gute Zusammenfassung: https://www.youtube.com/watch?v=c26O1N3x6ug

CSS-Syntax

Für eine CSS-Formatierung sind immer drei Angaben notwendig:

- Element (Selektor)
- Eigenschaft (Attribut) – hier wird angegben, welche Eigenschaft bei dem vorher ausgewählten Element verändert werden soll, z.B. Schriftgröße, Farbe.
- Wert – hier erfolgt die tatsächliche Formatierung selbst. Hier wird z.B. die gewünschte Größe als Zahlenwert angegeben.



Element { Eigenschaft:Wert; } Beispiel: **h1 { color:black; }**

- Wichtig ist dabei, dass das zu formatierende Element kleingeschrieben vor der geschweiften Klammer steht und durch ein Leerzeichen getrennt ist.
- Zwischen den Klammern und den darin enthaltenen Elementen sollten ebenfalls Leerzeichen gesetzt werden, weil viele Browser sich damit leichter tun.
- Die gewünschte Eigenschaft wird dann direkt notiert und mit einem Doppelpunkt (:) vom zugewiesenen Wert getrennt. Hier erfolgt kein Leerzeichen.
- Der Wert wird schließlich mit einem Semikolon (;) beendet. Dadurch wird die Wertzuweisung abgeschlossen und es kann die nächste folgen.

Man kann auch mehrere Werte zuweisen, wenn sie mit einem Semikolon getrennt werden:

```
h1 { color:black; font-size:30px; }
```

oder untereinander:

```
h1 {
        color:black;
        font-size:30px;
}
```

Somit lässt sich jedes HTML-Element formatieren.

Einen vollständige Aufbau wie oben nennt man eine **REGEL**. In jeder Regel können beliebig viele Deklarationen enthalten sein, die wiederum aus Eigenschaftsbezeichnungen und einem Wert bestehen.

Als **Deklaration (Anweisung)** bezeichnet man immer die Kombination aus einer Eigenschaft und einem Wert:
Beispiel: ***color: red;***

STYLE – CSS-Formatierung einleiten

Damit die Formatierung per CSS in ein HTML-Dokument integriert werden kann, sind einige zusätzliche Befehle nöig. Der **grundlegende Befehl lautet <STYLE>.** Er muss geöfnet und geschlossen werden. Meist ist es üblich, dem Befehl über das Attribut TYPE noch mitzuteilen, dann nun CSS-Befehle in Textform erfolgen.

3 Möglichkeiten der Einbindung von CSS-Code wobei man „style“ verwenden muss:

1) Intern im <head>
   <style>
   CSS Code
   </style>

```
<style>
h1 {
    color: #525252;
    margin-top: 120px;
}
</style>
```

2) ODER intern wenn man Elemente direkt formatiert (im body beim Element – hier wird <h1> eine rote Farbe erhalten)

   <h1 style=“color:red;“>

3) Wird **jedoch eine externe CSS-Datei** in eine HTML-Datei eingebaut, muss dies über einen einfachen Link im Bereich <head> erfolgen, wobei der Befehl nicht mehr <style> sondern <stylesheet> lautet:

   <head>
   <link rel=“stylesheet“ href=“mycss.css“>
   </head>

   In diesem Beispiel heißt die externe CSS-Datei „mycss.css“.

Externe CSS-Datei: Regeln

- Die Endung der Datei muss **zwingend .css lauten**, sonst wird sie von vielen Browsern nicht akzeptiert. Wie die Datei selbst heißt ist egal, z.B. layout.css, format.css.

- In der Datei darf **<u>außer CSS-Befehlen nicht stehen</u>**. Die einzige Ausnahme stellen Kommentare dar, welche anders als in HTML dargestellt werden.
  Der Kommentar muss von einem Slash und einem Stern eingeschlossen werden:

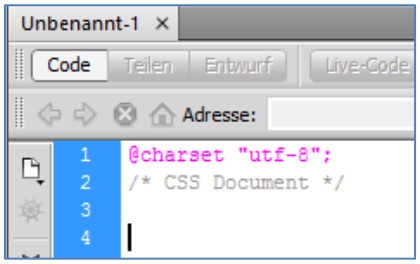
  /* Kommentar xyz */

UTF-8

Ein CSS-Datei beginnt immer mit der Definition des vorliegenden Zeichensatzes. Dieser legt fest, welche Schriftzeichen erlaubt sind und wie diese ausgeschrieben werden. Auf Nummer sicher geht man mit dem Zeichensatz „utf-8“, da dieser alle gängigen lateinischen Zeichen und Symbole beihaltet. Den Zeichensatz definiert man mit einer einfachen Zeile Code am Anfang der CSS-Datei:

@charset „utf-8“;

Startet man eine neue CSS-Datei ist dies automatisch vorhanden.



3 Möglichkeiten um HTML mit CSS zu stylen:

- direkt das Element ansprechen (generic tag)
- mit einer Klasse
- oder einer ID

Elemente werden bei ihrem **Tag-Namen** (z.B.: body, h1, h2, header) genannt, Elemente sind daher bereits vorhandene HTML Namen.

Klassen werden per „class“+ freien Namen beim HTML-Bereich erstellt und dann im <style>-Bereich einem **vorangestellten Punkt** (z.B.: .class) genau mit CSS-Befehlen gestylt.

IDs werden per „id“ + Namen im HTML-Bereich erstellt und mit einer **beginnenden Raute** (z.B.: #digbiz) im <style>-Bereich formatiert.

Beispiele

a)class

Eine Klassen-Regel (englisch „class“) beginnt im CSS-Dokument immer mit einem **Punkt**. Damit können Mehrfachbenutzungen durchgeführt werden, da „class“-Namen öfter im HTML-Quelltext vergeben werde können (im Gegensatz zu ID´s) und in der CSS-Datei damit gemeinsam angesprochen werden können.

- Der DIV-Bereich bekommt eine Klasse mit dem Namen class=“nav“

```
78      <div class="nav">
79          <a href="">Home</a>
80          <a href="">Kontakt</a>
81          <a href="">Impressum</a>
82      </div>
```

- Der dazupassende CSS-Code wird mit einem Punkt gestartet

```
.nav {
    margin-top: 10px;
}
```

b)ID-Selektor

Der ID-Selektor wirkt immer genau auf ein Element, das ein id-Attribut mit dem übergebenenen Bezeichner besitzt. Hierfür wird der ID-Wert mit vorangestelltem Hash-Zeichen eingesetzt.

Beachte: Man kann Klassen beliebig oft einsetzen und ansprechen. IDs hingegen dürfen genau einmal vorkommen.

Übung: Erstelle in der layout.css eine ID namens #impressum:

- Statt eines Punktes setzte eine **RAUTE**:

```
5
6  #impressum {
7      color: white;
8      background-color: black;
9  }
10
```

Die Einbindung erfolgt in der einstieg.html für den zweiten Paragraf <p>:

```
11          <br>
12          <p id="impressum">Mit IDs kann man genau einmal etwas formatieren.</p>
13      </body>
```

ID ist die **Abkürzung für „Identität“** und im englischen auch für „Personalausweis“.
Der Wert einer ID darf in der HTML-Datei nur einmal pro Seite vorkommen

Bestimmte häufig verwendete Eigenschaften (Attribute):

- font-size - gibt die Größe der Schrift an
- font-family - legt die Schriftart fest; ist eine Schriftart nicht installiert, wird automatisch die zweite oder dritte verwendet.
  Beachte: der Name muss direkt hinter dem Doppelpunkt stehen.
  Beispiel: p { font-family:Calibri,'Arial Black'; }
  Man kann auch generel diese schreibweise nutzen:
  font: Helvetica, sans-serif;
- font-weight - steht für Fettdruck, der sich stufenlos variieren lässt
- background-color - Hintergrundfarbe
- background-image:url(...jpg)
- text-align:center - legt man die Ausrichtung des Textes oder des jeweils deklarierten Objektes fest, hier „zentriert"
- margin - äußerer Abstand von Objekten zu anderen Objekten, erstellt einen Abstand zum Browserfensterrand
  Beispiel: margin:25px
- padding - innerer Abstand des Objekts zu seinem eigenen Rand
- width - Breite eines Objektes
- height - Höhe eines Objektes
  Beispiel: soll ein großes Bild auf eine bestimmte Breite gebracht werden, aber die Höhe soll angepasst und nicht falsch skaliert werden, verwende für height:auto
  img { width:250px; height:auto; }
- border erwartet drei Werte: die Rahmenstärke, die Rahmenart und die Rahmenfarbe.
  Beispiel: border: 1px solid red;
- background-image
  damit definiert man einen Dateipfad, der auf das Hintergrundbild verweist. Mit „background-repeat" definiert man ob und wie der Hintergrund wiederholt werden soll. Möglich ist auch „no-repeat", falls der Hintergrund nur einmal angezeigt werden soll.
  Beispiel: background-image: url(img/bg.jpg) repeat center;
- background-size
  Damit kann man die größe des Bildes in Pixel definieren.
  z.B. cover: automatische Anpassung, dass das komplette Element davon bedeckt (cover) wird

Groß- und Kleinschreibung

Wird in CSS nicht beachtet. Einfachheitshalber werden alle Wörter kleingeschrieben.

Angabe der Schriftgrößeneinheit:

Bei der Angabe von Werten und dem entsprechenden Maß ist es wichtig, dass zwischen diesen **kein Leerzeichen** vorhanden ist, sondern die Werte wie folgt angegeben werden:

16px oder 160% - falsch: 16 px oder 160 %

```
font-family:  "Open Sans", sans-serif;
font-size: 26px;
```

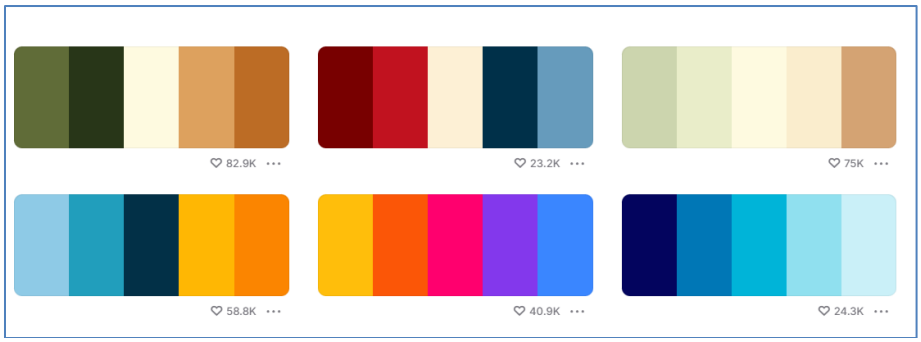
Farben mit CSS für die Schrift und für den Hintergrund

Farben werden in CSS als Hexadezimalwert definiert. Der Hex-Wert repräsentiert die Werte für Rot, Grün und Blau im RGB-Modell als Hexadezimalwerte zwischen 00 und FF.

CSS erlaubt das Arbeiten mit Farbnamen **(color keywords)** Beispiel: p { color: red; }

Farbname	Farbton	Hexadezimal normal	Hexadezimal kurz
white	Weiß	#ffffff	#fff
black	Schwarz	#000000	#000
red	Knallrot	#ff0000	#f00
maroon	Dunkelrot	#800000	-
lime	Knallgrün	#00ff00	#0f0
green	Dunkelgrün	#008000	-
blue	Knallblau	#0000ff	#00f
navy	Dunkelblau	#000080	-
gray	Dunkelgrau	#808080	-
silver	Hellgrau	#c0c0c0	-
yellow	Knallgelb	#ffff00	#ff0
orange	Orange	#ffa500	-
olive	Oliv	#808000	-
purple	Dunkellila	#800080	-
fuchsia	Helllila	#ff00ff	#f0f
aqua	Türkis	#00ffff	#0ff
teal	Aquamarin	#008080	-

Besser: weil in Farbfamilien geordnet: https://coolors.co/palettes/popular auf Farbe klicken und dann ist der Code bereits kopiert



Background

background-color:

Mit CSS wird die Farbe in 3 Arten definiert:

- HEX-Wert: #ff0000
- RGB-Wert: rgb(255,0,0)
- Ein gültiger Farbenname: red

background-image:

```
body {
    background-image: url("bg.jpg");
}
```

audio.gif

bg.jpg

Das Bild wird sooft wiederholt, bis das betreffende Element (z.B. body) bedeckt ist. Ist es zu klein, kann schnell eine unerwünschte Situation entstehen.

Überschriften:

Seitens des World Wide Web Consortiums (W3C) wurden für Überschriften 6 unterschiedliche Stufen der Gewichtung festgelegt. Diese werden durch den h-Tag ausgezeichnet und erhalten mit einer dazugehörigen Ziffer ihre geweilige Gewichtung:

z.B. <h1>Überschrift 1. Ordnung </h1>

Überschriften sind „Blockelemente“ und weisen daher Außen- und Innenabstände auf.