

Science Camp Informatik - Tag 1

HTML & CSS

Was ist HTML?

HTML ist Sprache um Webseiten zu bauen. Mit HTML kann man *beschreiben*, wie eine Webseite aussehen soll. Im engeren Sinne ist es deswegen eigentlich keine Programmiersprache, sondern lediglich eine beschreibende Sprache.

HTML Dateien enden mit dem Dateiformat `.html` und können ganz einfach vom Browser geöffnet werden.

Auf deinem USB Stick befindet sich schon eine Datei `index.html`. Öffne Sie.

Grundgerüst HTML

Das Grundgerüst von HTML sieht immer so aus:

```
1 <html>
2
3   <head>
4   </head>
5
6   <body>
7   </body>
8
9 </html>
```

Die spitzen Klammern beschreiben immer einen Bereich. `<html>` beginnt einen html Bereich oder Abschnitt. `</html>` beendet ihn.

Die Einrückungen sind eigentlich unwichtig.

Diese Schreibweise ist genauso korrekt:

```
<html><head></head><body></body></html>
```

Diese sagt auch das gleiche:

```
1 <html>
2 <head>
3 </head>
4 <body>
5 </body>
6 </html>
```

Allerdings ist es ein guter Stil, den Code richtig einzurücken. Also gewöhnt es euch am besten sofort an :) Außerdem ist es besser lesbar.

Der Body-Bereich

Im Body ist der eigentliche Inhalt der Webseite. Wenn ich zum Beispiel einen Text auf meiner Webseite haben möchte, Schreibe ich ganz einfach den Text in den Body.

```
1 <html>
2
3 <head>
4 </head>
5
6 <body>
7     Willkommen auf meiner Webseite!
8
9     Hier werde ich alles ausprobieren,
10    was ich auf dem ScienceCamp Informatik
11    gelernt habe und eine coole Webseite bauen.
12
13    Diese Seite wird in einer Woche
14    wirklich toll aussehen!
15 </body>
16
17 </html>
```

Schreibe deine erste HTML-Seite mit Inhalt.

Wie du siehst, ist die Formatierung des Textes auf der Webseite leider nicht so, wie wir es uns wünschen. Dafür müssen wir noch einige Textformatierungen lernen:

- `
` ist eine Abkürzung für break, also ein Zeilenumbruch
- `<h1>Überschriften</h1>` sind Überschriften. **h** ist die Abkürzung für Headline, also einer Überschrift. statt **h1** kann man auch **h2** oder **h3** usw. benutzen. Die Überschriften werden mit steigender Zahl kleiner.
- `<p>Paragaphen</p>` sind Paragaphen. Die sind nicht unbedingt nötig, sind aber zu empfehlen.

Versuche nochmal deine erste Seite zu schreiben indem du die neu gelernten Tags anwendest.

```
1 <html>
2
3 <head>
4 </head>
5
6 <body>
7     <h1>
8         Willkommen auf meiner Webseite!
9     </h1>
10    <p>
11        Hier werde ich alles ausprobieren,
```

```

12         was ich auf dem ScienceCamp Informatik
13         gelernt habe und eine coole Webseite bauen.
14     </p>
15     <p>
16         Diese Seite wird in einer Woche
17         wirklich toll aussehen!
18     </p>
19 </body>
20
21 </html>

```

Es gibt auch weitere Tags für Text wie zum Beispiel ` text ` oder ` text `.
Probier strong und em aus.

Der Head-Bereich

Im Head - also dem Kopfteil - sind Informationen, die nicht auf der Webseite selbst dargestellt werden. Stattdessen sind hier andere Informationen drin, die nur indirekt mit der eigentlichen Webseite zu tun haben. Man nennt solche Informationen auch Meta-Informationen.

Ein Beispiel wäre zum Beispiel der Titel einer Webseite.

```

1 <html>
2
3     <head>
4         <title>Meine Webseite</title>
5     </head>
6
7     <body>
8         <h1>
9             Willkommen auf meiner Webseite!
10        </h1>
11        <p>
12            Hier werde ich alles ausprobieren,
13            was ich auf dem ScienceCamp Informatik
14            gelernt habe und eine coole Webseite bauen.
15        </p>
16        <p>
17            Diese Seite wird in einer Woche
18            wirklich toll aussehen!
19        </p>
20    </body>
21
22 </html>

```

Probier es aus und finde raus, wo der Titel auftaucht.

Mehr als nur Text

Webseiten bestehen üblicherweise aus mehr als nur Text. Moderne Webseiten haben Bilder, Links oder Listen.

Bilder

Bilder kann man wie folgt einfügen:

```

```

Der Pfad zum Link ist eine Angabe, wo sich das Bild befindet.

Beispiel: Wenn diese HTML Datei und das Bild im gleichen Ordner sind, also so:

```
1  irgendein Ordner
2      |
2      └─ index.html
3      |
3      └─ bild.jpg
```

dann wäre das Bild so aufzurufen: ``

Wenn diese HTML Datei in einem Ordner ist, in dem sich ein weiterer Ordner **bilder** befindet, in dem bilder drin sind, also so:

```
1  irgendein Ordner
2      |
2      └─ index.html
3      |
3      └─ Ordner
4          |
4          └─ bild.jpg
```

dann wäre das Bild so aufzurufen: ``

Du kannst alle möglichen Bildformate benutzen. Es funktioniert zum Beispiel auch mit **.gif** oder **.png**-Dateien.



Du kannst auch auf Bilder von anderen Webseiten zugreifen.

Auf <http://i.giphy.com/achoDiZFxZvdm.gif> ist zum Beispiel ein Bild. Um dieses anzuzeigen, kannst du den Link einfügen, also:

```

```



Falls du deine Webseite mal online stellst, solltest du das allesdings aus verschiedenen Gründen (z.B. Copyright) vermeiden. Für den kleinen persönlichen Gebrauch ist es aber erstmal ok.

Füge ein Foto von dir und Bilder in deine Webseite ein.

Links

Webseiten brauchen natürlich auch Links zu anderen Webseiten:

```
<a href="PFAD_ZUM_LINK"> Text </a>
```

Auch hier gilt wie bei Bildern: Du kannst du externen Links verlinken oder auch eine eigene HTML-Seite mit dem Pfad angeben:

```
<a href="https://de.wikipedia.org/wiki/Hyperlink"> Wikieintrag zu Hyperli  
nks </a>
```

[Wikieintrag zu Hyperlinks](#)

Der Inhalt innerhalb des a-Tags ist der Teil, der dann anklickbar ist und zum Link führt. Du kannst auch Bilder als Link benutzen:

```
1 <a href="https://de.wikipedia.org/wiki/Haushund">  
2     
3 </a>
```



Wie du merkst, öffnet sich der Link direkt im aktuellen Tab oder Fenster. Manchmal (vor allem bei Links zu externen Seiten), möchte man den Link lieber in einem neuen Tab öffnen. Setze dafür ein `target`, also ein Ziel:

```
1 <a href="https://de.wikipedia.org/wiki/Hyperlink" target="_blank">
2   Wikieintrag zu Hyperlinks
3 </a>
```

[Wikieintrag zu Hyperlinks](https://de.wikipedia.org/wiki/Hyperlink)

Erstelle drei Links zu deinen Lieblingsseiten ein.

Listen

Listen macht man mit einem `ul`-Tag und jeder einzelne Bulletpoint wird mit einem `li`-Tag beschrieben. So sieht das dann aus:

```
1 Einkaufsliste
2 <ul>
3   <li> Eiscreme </li>
4   <li> Smoothie </li>
5   <li> Kekse </li>
6 </ul>
```

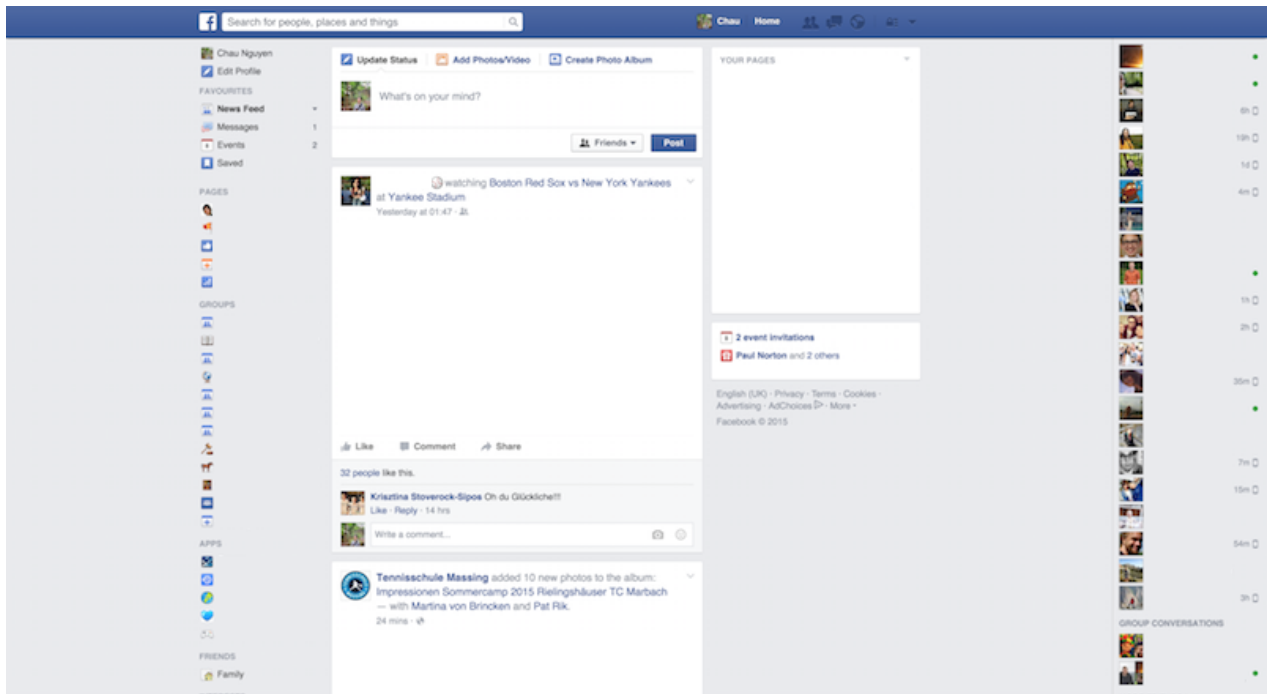
Einkaufsliste

- Eiscreme
- Smoothie
- Kekse

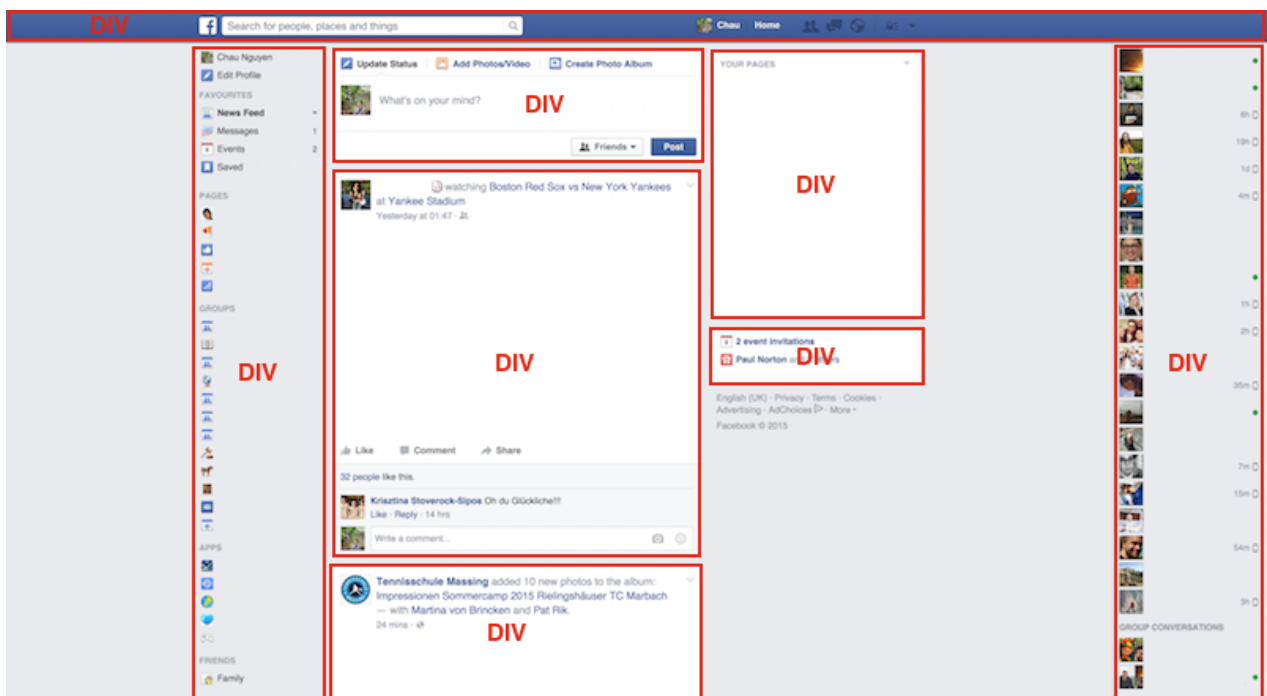
Erstelle eine Liste mit deinen Hobbys

DIVs

Kommen wir nun zum wichtigsten Konstrukt von Webseiten: Divs. Ein Div-Tag ist schwer zu erklären. Man kann mit einem Div, gewisse Inhalte gruppieren und strukturieren. Nehmen wir als Beispiel Facebook.



Die Seite hat verschiedene Inhalte. Um dies zu strukturieren, werden verschiedene Divs benutzt. Facebook könnte zum Beispiel so aussehen:



Warum genau DIVs nützlich sind und gebraucht werden, wird später nochmal klarer. Inhalte in DIVs zu unterteilen ist recht einfach:

- 1 <div>
- 2 <!-- Inhalte vom Div -->

3 </div>

Schreibe deinen aktuellen Webseiten Inhalt in ein DIV.

Glückwunsch!

Du kannst nun alle Basics für HTML <3

Aber ehrlich gesagt sieht unsere Webseite noch nicht wirklich schön aus. Wir könnten zwar auch mit HTML unser Design machen, aber es gibt einen viel besseren Ansatz, um das zu bewerkstelligen. Es nennt sich **CSS**.

CSS

Wir haben ein Webseite und sie sieht schonmal recht gut aus! Das Design lässt allerdings noch ein bisschen zu wünschen übrig. Um Webseiten zu designen kann man CSS benutzen. Eine CSS-Datei ist eine Datei, in der man Anweisen zum Design geben kann. Zum Beispiel kannst du dort angeben, welche Hintergrundfarbe oder Textfarbe die Webseite haben soll.

Einfügen der CSS Datei

Auf deinem Stick befindet sich eine leere Datei mit dem Namen `style.css`. In dieser Datei werden wir das Design festlegen.

Dann fügst du im `head`-Bereich deiner Webseite die Information ein, dass wir in `style.css` den Style der Webseite festlegen werden:

```
<link rel="stylesheet" type="text/css" href="style.css"/>
```

Aufbau von CSS

Der Aufbau eines CSS sieht so aus:

```
1 Element_das_ein_Styling_kriegt {
2     Attribut_1: Wert_für_Attribut_1;
3     Attribut_2: Wert_für_Attribut_2;
4 }
```

Eine Übersicht über alle möglichen Attribute gibt es auf

https://developer.mozilla.org/de/docs/Web/CSS/CSS_Referenz.

Hintergrund

Wenn wir also zum Beispiel die Hintergrundfarbe der gesamten Webseite ändern wollen, schreiben wir folgendes in die CSS Datei:

```
1 body {
2     background: red;
3 }
```

Wenn du die Farbe genauer spezifizieren möchtest, brauchst du einen bestimmten Code für die Farben. Auf <http://www.flatuicolorpicker.com/> findest du eine Menge Farben und deren Codes. Die Codes bestehen immer aus sechs Zeichen und müssen mit einem # anfangen. Zum Beispiel:

```
1 body {
2     background: #FDE3A7;
```



```
3 }
```

Statt einer Farbe kannst du auch Bilder benutzen:

```
1 body {  
2     background-image: url(bilder/subtlepatterns/pink_rice.png);  
3 }
```

Eine tolle Seite zum Suchen nach Hintergrundbildern ist <http://subtlepatterns.com/>.

Such dir einen Hintergrund für deine Webseite aus.

Schrift

Schriftart, -farbe, -größe und -ausrichtung bestimmt man mit

```
1 color: #111111;  
2 font-family: Helvetica;  
3 font-size: 14px;  
4 text-align: left;
```

Die Schriftgröße kannst du mit einer festen Anzahl an Pixeln setzen. Du kannst sie aber auch relativ zu der Standardgröße (die Größe, die im Body genutzt wird) setzen. Wenn du zum Beispiel die Größe im **body** auf **14px** setzt und in der Überschrift eine Schrift haben willst, die vier mal so groß ist, schreib:

```
font-size: 4em;
```

Style die Schrift im Body und für die h1 Überschrift

Google bietet übrigens auch Schriften zum Einbinden auf eigene Webseiten an. Die Schriftarten findest du hier <https://fonts.google.com/>

Design des Inhalts

Wir wollen unsere Webseite nun auch noch zentrieren und den Hintergrund ändern. Wir wollen also allem, was innerhalb des Divs ist

- einen anderen Hintergrund geben
- die Breite ein wenig kleiner machen
- zentrieren

Dazu geben wir dem Div erstmal einen Namen. Dafür gehen wir zurück in die HTML-Datei und ändern

```
1 <div>  
2     ...  
3 </div>
```

um in

```
1 <div class="inhalt">  
2     ...  
3 </div>
```

Jetzt ist unser DIV ein DIV der Klasse "inhalt" und wir können die Breite und den Hintergrund wie folgt festlegen:

```
1 .inhalt {  
2     background: white;  
3     width: 600px;
```

4 }

Zentrieren geht mit

```
margin: 0 auto;
```

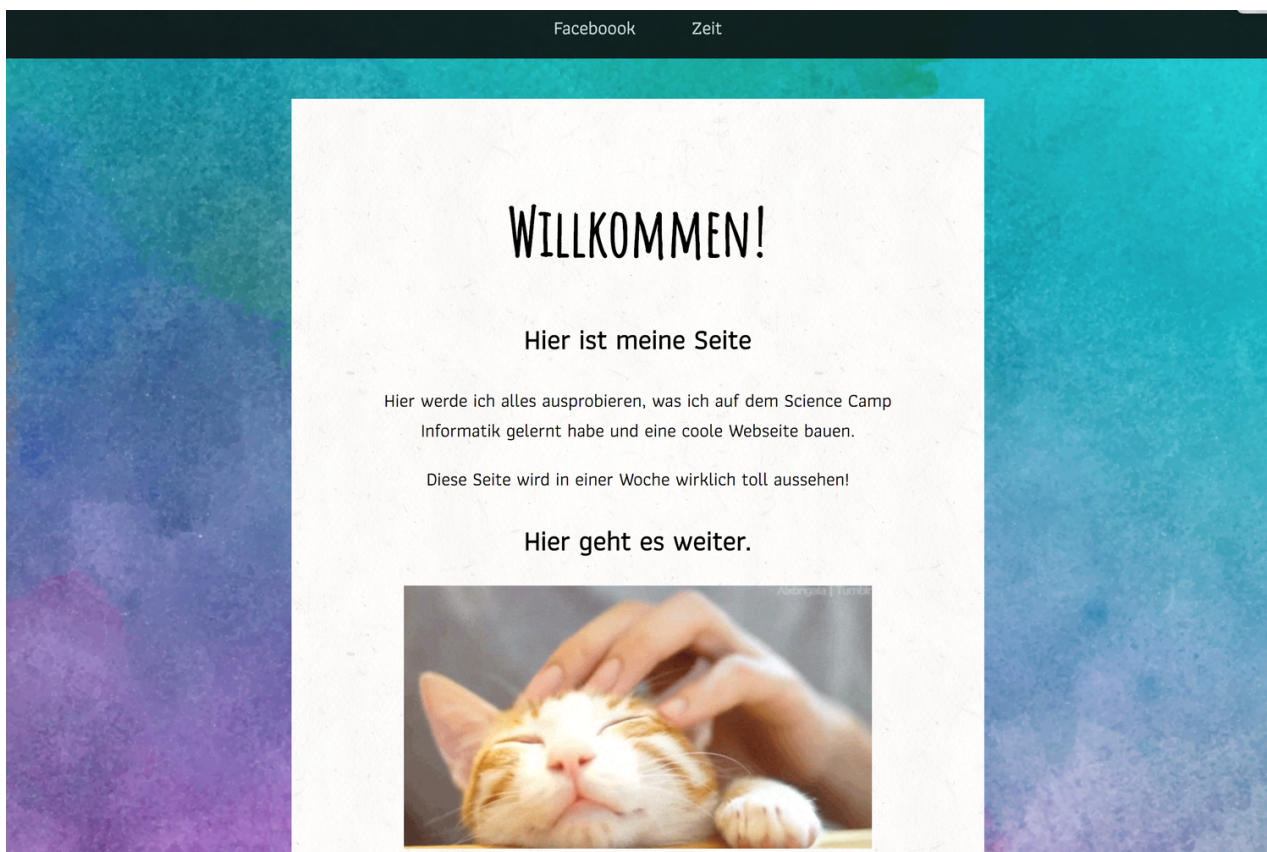
Leider ist es etwas unschön, dass es keinen Rand zwischen dem DIV und dem Text gibt. Das kann man mit einem sogenannten **padding** ändern. Ein Padding wird mit 4 Werten angegeben: Der Abstand nach oben, nach rechts, nach unten und nach links (also im Uhrzeigersinn). Dein Padding kann z.B. so aussehen:

```
padding: 40px 30px 60px 30px;
```

Style deinen Inhalt, sodass er dir gefällt.

Menüleiste

Was jetzt noch fehlt, ist eine schöne Menüleiste, wie in unserer Demo:



Die Menüleiste sieht sehr verdächtig nach einem neuen DIV aus!

Füge ein neues DIV mit einer Liste vor dem restlichen Inhalt in die HTML-Seite ein. Gib dem DIV die Klasse **menu und setze ein padding.**

Jetzt wollen wir das DIV stylen.

Gib dem Menu eine dunkle Hintergrundfarbe und setze die Schrift auf eine helle Farbe.

Nun wollen wir die Listensymbole verschwinden lassen. Außerdem möchten wir den Text nicht untereinander, sondern nebeneinander. Dafür kannst du folgende Attribute setzen:

```
1 .menu ul {  
2     display: inline-block;  
3     list-style: none;
```

```
4 }
```

und

```
1 .menu ul li {  
2     float: left;  
3 }
```

Jetzt ist der Text zwar nebeneinander, aber es ist viel zu nah und klebt sofort am vorherigen Text. Dafür benutzen wir wieder ein padding!

```
1 .menu ul li {  
2     padding: 8px 25px;  
3 }
```

Glückwunsch! Deine erste Webseite ist fertig!