

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

[www.udemy.com/https://www.udemy.com/course/tailwindcss-deutsch](https://www.udemy.com/course/tailwindcss-deutsch)

Tailwindcss - muss man können!

Hi und Servus! Erstmal herzlich willkommen. Du hast dieses Skript in der Hand, bzw. auf dem Bildschirm, weil du entweder meinen [Kurs online gebucht hast](#)



oder weil du eine Kopie von jemandem bekommen hast.

Wie auch immer - Ich arbeite seit schon 20 Jahren als

“Full-Stack-Webentwickler” und kann Dir nur sagen:

Wenn du die nächsten Jahre irgendwie mit CSS zu tun haben wirst, dann lerne jetzt Tailwindcss!

Tailwindcss ist eine neue Art CSS zu “denken” bzw. einzusetzen, die du kennen solltest in den 2020'ern. Viel Spass! Martin

Benötigte Software

[Visual Studio Code \(VSC\) + Erweiterungen](#)

[Node.js](#)

[Git Versionierungssystem \(Optional\)](#)

[Bitbucket-Account und SourceTree Software \(Optional\)](#)

[Bitbucket-Account anlegen](#)

[SourceTree herunter laden und installieren](#)

Installation und basic Skills

[Tailwindcss installieren in 5 Schritten](#)

[Styles und Workflow zum Laufen bekommen](#)

[Hintergrund: Git mit SourceTree bedienen](#)

[Hintergrund: Die Kommandozeile bedienen](#)

[Hintergrund: Emmet](#)

[Hintergrund: VSC Shortcuts / Bedienung](#)

[Hintergrund: Die 3 Hauptbestandteile von Tailwindcss](#)

Mini-Projekt

[Container + Breakpoints](#)

[Planung der Mini-Website](#)

[Seiteninhalte grob anordnen](#)

[Überschriften formatieren und voreinstellen](#)

[Text-Formatierungen in Tailwindcss](#)

[Eigenen Überschriften-Font definieren](#)

[Eine eigene Farbe hinzufügen](#)

[Die Listen formatieren](#)

[Den Rest formatieren](#)

[Einen Button mit Icon designen](#)

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

[www.udemy.com/https://www.udemy.com/course/tailwindcss-deutsch](https://www.udemy.com/course/tailwindcss-deutsch)

[Hintergrund: Kostenlose Icons](#)
[Einführung ins Responsive Design mit Tailwindcss](#)
[Optimierung für Mittlere Bildschirme \(md\)](#)
[Optimierung für Kleine Bildschirme \(sm\)](#)
[Optimierung für ganz kleine Bildschirme \(Handy\)](#)
[Optimierung für große Bildschirme XL und 2XL](#)
[Hintergrundmuster](#)
[Zitat oben mit Anführungszeichen und anderer Farbe](#)
[Nachtmodus = "Dark Mode" anlegen](#)
[Kurzeinführung Alpine.js + Darkmode-Button](#)
[Dynamische Darstellung des Darkmode-Buttons](#)
[Button-Styles auslagern](#)
[CSS mit Purgecss verkleinern](#)
[Projektzusammenfassung](#)

[Fortsetzung des Kurses](#)

Benötigte Software

Visual Studio Code (VSC) + Erweiterungen

Visual Studio Code ist nicht der beste Editor (Ich bin absoluter PHPStorm-Fan), aber es ist ein sehr guter KOSTENLOSER Editor! :-)

Du kannst es hier für alle Betriebssysteme herunterladen:

<https://code.visualstudio.com/download>

Auch, **falls du VSC schon installiert hast**, dann installiere bitte noch folgende

Erweiterungen für VSC:

- Headwind
- Live Server
- Tailwind CSS IntelliSense
- Alpine.js IntelliSense

[Nach oben](#)

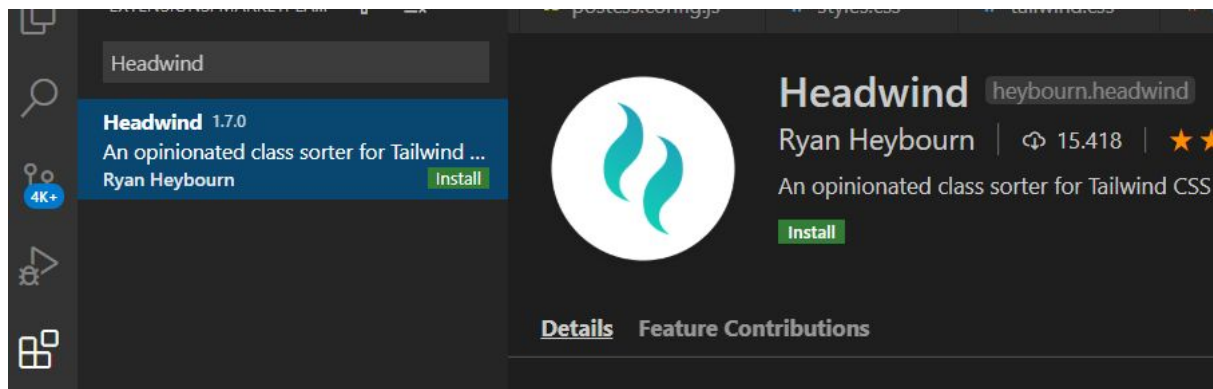
Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch



Eine Erweiterung installierst du in VSC, indem du links auf das Erweiterungssymbol gehst, dann die Erweiterung im Suchfeld suchst und dann auf das grüne "Install" klickst.

Node.js

Node.js ist für das Arbeiten mit tailwindcss unverzichtbar. Du verwendest u.a. dafür um tailwindcss zu **konfigurieren**, zu **erweitern** und **Dein Stylesheet** am Projektende zu **kompilieren** = dramatisch zu **verkleinern**.

Du kannst es für alle Betriebssysteme hier herunterladen: <https://nodejs.org/en/download/>

Du kannst während der Installation alle Voreinstellungen des Installers so lassen, wie sie sind.

Wenn du Node ordnungsgemäß installiert hast, dann kannst du es folgendermaßen **testen**:

- Öffne **Visual Studio Code**
- Wähle in der Navigation **Terminal / New Terminal**
- In dem Terminal was sich öffnet tippst du einfach `npm` (+ ENTER), und solltest in etwas folgende Meldung sehen:

```
$ npm

Usage: npm <command>

where <command> is one of:
  access, adduser, audit, bin, bugs, c, cache, ci, cit,
```

usw...

Git Versionierungssystem (Optional)

Ich empfehle grundsätzlich wenn du irgendetwas programmierst mit Git zu arbeiten. Und wenn es nur dafür genutzt wird dass du Deine Arbeit später wieder nachvollziehen kannst.

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

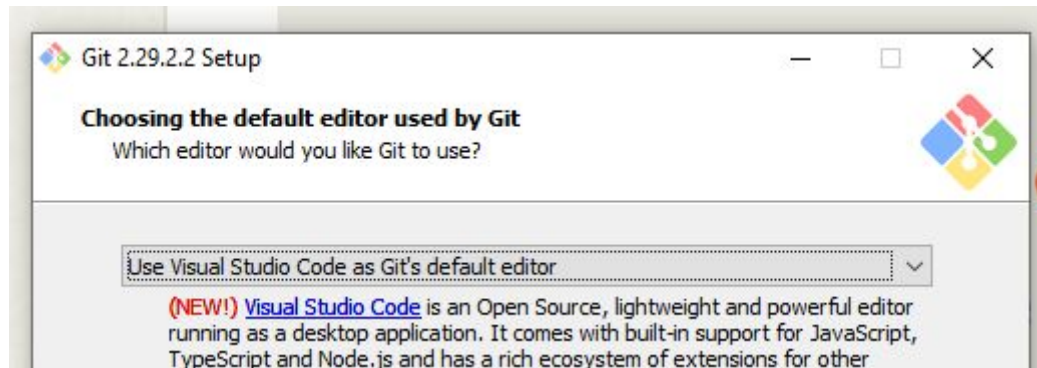
Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy**:

www.udemy.com/course/tailwindcss-deutsch

Mit Git kannst du nämlich **Zwischenstände abspeichern**, und später genau **sehen, was du an welcher Datei gemacht hast**.

Hier kannst du Git herunter laden: <https://git-scm.com/downloads>

Während der Installation kannst du auswählen, dass du **Visual Studio Code** als **“Default Editor”** verwendest:



Alle anderen Einstellungen kannst du während der Installation so lassen wie sie voreingestellt sind.

Nach der erfolgreichen Installation kannst du im Terminal einfach **git** ausführen und siehst dann das:

```
$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      <command> [<args>]
```

usw...

Bitbucket-Account und SourceTree Software (Optional)

Falls du auf Windows oder Mac arbeitest empfehle ich Dir SourceTree als optische Git-Oberfläche. Falls du auf Linux arbeitest kannst du mal diese Liste durchsehen, und Dich für eine andere Software entscheiden: <https://git-scm.com/download/gui/linux>

Bitbucket-Account anlegen

Für die Verwendung von SourceTree brauchst du allerdings einen Bitbucket-Account. Bitbucket ist etwas ähnliches git GitHub. Keine Sorge, du hast keinerlei Nachteile wenn du einen Bitbucket-Account anlegst, und du kannst trotzdem GitHub verwenden, kein Problem.

Geh also hier hin: <https://bitbucket.org/product>

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

und klicke auf den Button "Get it free". Dann registrierst du Dich. Falls du dich mit der voreingestellten Methode registrierst, also mit Deiner E-Mail-Adresse, dann wird Dir Bitbucket eine E-Mail zukommen lassen, damit du Deine **E-Mail-Adresse bestätigen** kannst.

SourceTree herunter laden und installieren

SourceTree bekommst du hier: <https://www.sourcetreeapp.com/>

Nachdem du das Programm installiert hast fragt Dich SourceTree irgendwann nach Deinen Login-Daten für BitBucket. Gib diese dann ein, und du kannst SourceTree verwenden.

Installation und basic Skills

In diesem Kapitel werden wir ein erstes ganz kleines Projekt umsetzen. Folgendes wirst du lernen:

1. Wie man Tailwindcss **installiert**
2. Was die wichtigsten **Bestandteile** von Tailwindcss sind
3. Wie man eine **Erweiterung** installiert
4. Wie Tailwindcss im groben **funktioniert**
5. Wie man eine Seite mit Tailwindcss **responsive** macht
6. Mit man Tailwindcss **erweitert bzw. konfiguriert**
7. Wie man die Styles **kompiliert bzw. minimiert**

Tailwindcss installieren in 5 Schritten

Schritt 1: Installationsort festlegen

Als erstes musst du einen Ordner auf Deinem Computer festlegen, wo du die Beispiele von diesem Kurs ablegen möchtest.

Wir werden in diesem Kurs mehrere Tailwindcss-Installationen durchführen, deswegen schlage ich vor du gibst diesem Ordner einen allgemeinen Namen, wie z.B. "udemy_tailwindcss" oder ähnliches. In diesem Ordner legst du bitte noch einen Unterordner "miniprojekt" an.

Die offizielle Anleitung zu Tailwindcss findest du übrigens hier:

<https://tailwindcss.com/docs/installation>

Aber Achtung! Wir werden leicht von der offiziellen Version abweichen, denn ich habe eine für mich praktischere Lösung gefunden.

Schritt 2: Ordner im VSC betreten und Terminal öffnen

Öffne den soeben angelegten Ordner in Visual Studio Code:

[Nach oben](#)

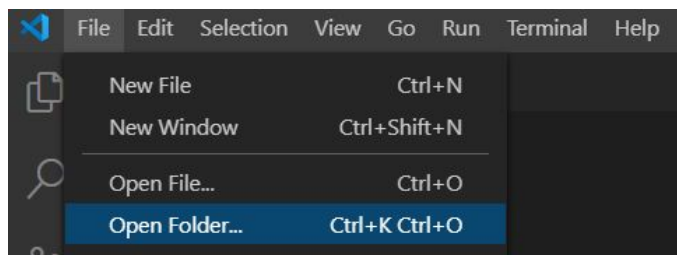
Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

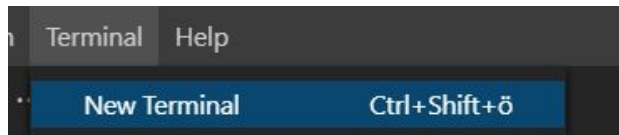
Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch



Dann öffnest du ein Terminalfenster:



Schritt 3: Git initialisieren und .gitignore erzeugen

Bevor du irgendetwas machst, erzeugst du erstmal ein Git Repository:

```
git init
```

Dann erzeugst du eine Datei **.gitignore** und öffnest sie im Editor

Dort fügst du ein:

node_modules/

Danach speicherst du und schließt die **.gitignore** wieder.

Schritt 4: Tailwindcss + Hilfsmittel installieren

Im Terminal fügst du folgende 3 Befehle aus:

```
npm init -y
```

> Das macht die package.json

```
npm install tailwindcss postcss postcss-cli autoprefixer
```

Achtung: Die postcss-cli wird in der offiziellen Doku nicht erwähnt. Brauchen wir aber! (siehe unten)

```
npx tailwindcss init -p
```

- Das erzeugt die **tailwind.config.js**
(Die Datei ist wichtig! Denn sonst geht Autocomplete im VS Code nicht)
- Das **-p** sorgt dafür dass auch gleich die Datei **postcss.config.js** erzeugt wird.

Schritt 5: Ordnerstruktur und Dateien anlegen

Anlegen:

- **tailwind.css**
- Ordner **"dist/css"** (geht alles zusammen in VSC!)
 - > In Ordner "dist" ist dann der Code zum Veröffentlichen drin

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

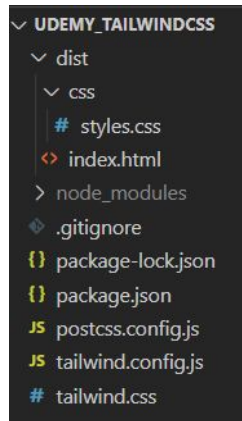
Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

- Eine Datei "styles.css" in dem css-Ordner
- Eine *index.html* in /dist/
- In der index.html mit "! + TAB" ein HTML-Gerüst erzeugen

Deine Struktur sollte jetzt so aussehen:



Styles und Workflow zum Laufen bekommen

Schritt 1: Die 3 Teile von Tailwindcss

In die tailwind.css die 3 Zeilen kopieren:

@tailwind base;

@tailwind components;

@tailwind utilities;

Schritt 2: styles.css in HTML einbinden

In der **/dist/index.html**

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

Schritt 3: Kompilierungs-Scripts einrichten

In der **/package.json**

```
"scripts": {  
  "build": "postcss tailwind.css -o dist/css/styles.css",  
  "watch": "postcss tailwind.css -o dist/css/styles.css --watch"  
},
```

Achtung: Halt die richtigen Input / Output - Files setzen, falls du eine andere Ordnerstruktur verwendest.

Anmerkung: Diese Scripte funktionieren nur mit der "postcss-cli". Das ist der Grund warum wir diese vorher - abweichend zur offiziellen Tailwindcss-Doku installiert haben!

(CLI = **C**omand **L**ine **I**nterface)

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

[www.udemy.com/https://www.udemy.com/course/tailwindcss-deutsch](https://www.udemy.com/course/tailwindcss-deutsch)

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Seite 8

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

Schritt 4: Das CSS kompilieren

Im Terminal führst du jetzt entweder `npm run build`, oder `npm run watch` aus, um deine styles.css mit Tailwind-Styles zu füllen.

Der Unterschied ist, daß `npm run build` eine einmalige Kompilierung ist, während `npm run watch` auf Änderungen im Hintergrund "wartet".

Im Ergebnis bekommen wir eine `/dist/styles.css` mit den ganzen tailwind.css - Styles und einer Dateigröße von **fast 4 MB!**

Aber keine Sorge: Wir werden später die CSS dramatisch verkleinern! :-)

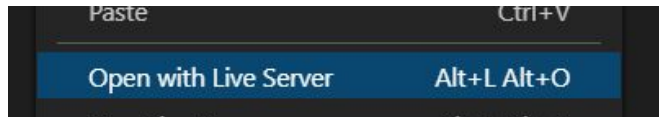
Schritt 5: Die tailwind-Styles testen

Zunächst müssen wir den Editor (VSC) neu starten, damit das Tailwind Intellisense Plugin funktioniert!

In der index.html können wir jetzt dem Body eine Hintergrundfarbe geben:

`<body class="bg-blue-400">`

Dann öffnen wir die Seite mit dem **Live-Server** (Maus rechts-klick in der index.html):



Dann müssten wir eine blaue Seite sehen!

Schritt 6: Den ersten Git Commit machen

Wir führen in der Kommandozeile aus:

git status

Das zeigt uns die geänderten Dateien (in Rot) an:

```
Untracked files:
(use "git add <file>..." to include in what will be committed)
.gitignore
dist/
package-lock.json
package.json
postcss.config.js
tailwind.config.js
tailwind.css
```

Wie wir sehen, sind die node_modules NICHT dabei, was wir mit der .gitignore erreicht haben.

git add -A

Das fügt uns alle Dateien zu Git hinzu (jetzt in Grün):

git status

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

```
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
new file:   .gitignore
new file:   dist/css/styles.css
new file:   dist/index.html
new file:   package-lock.json
new file:   package.json
new file:   postcss.config.js
new file:   tailwind.config.js
new file:   tailwind.css
```

git commit -m 'Styles und Workflow zum Laufen bekommen'

Achtung: Einfache Anführungszeichen für die Commit-Message verwenden!

git status

```
$ git status
On branch master
nothing to commit, working tree clean
```

Jetzt haben wir eine erste "Zwischenspeicherung" in Git gemacht.

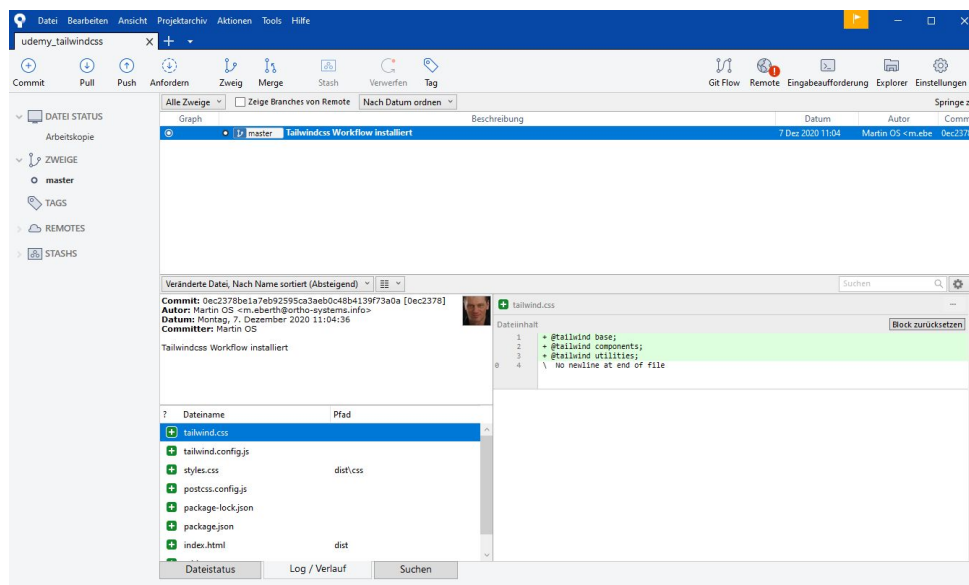
Hintergrund: Git mit SourceTree bedienen

Du kannst jetzt Dein Projekt auch mit SourceTree öffnen.

Du gehst einfach auf **Datei > Öffnen** und wählst Dein Projektverzeichnis.

Das Fenster, welches dann erscheint "**Lesezeichen erstellen?**" bestätigst du mit **Ok**.

Dann gehst du UNTEN auf **Log / Verlauf** und bekommst folgende Ansicht:



[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

Hier siehst du:

- Oben den Verlauf Deiner Commits = Zwischenspeicherungen
- Unten Links die geänderten Dateien Deines gewählten Commits
- Unten Rechts den Inhalt der Datei welche geändert oder hinzugefügt wurde

Wir ändern jetzt eine Datei, und zwar schreiben wir in der index.html im Body einfach ein "Hallo" rein:

```
<body class="bg-blue-400">
```

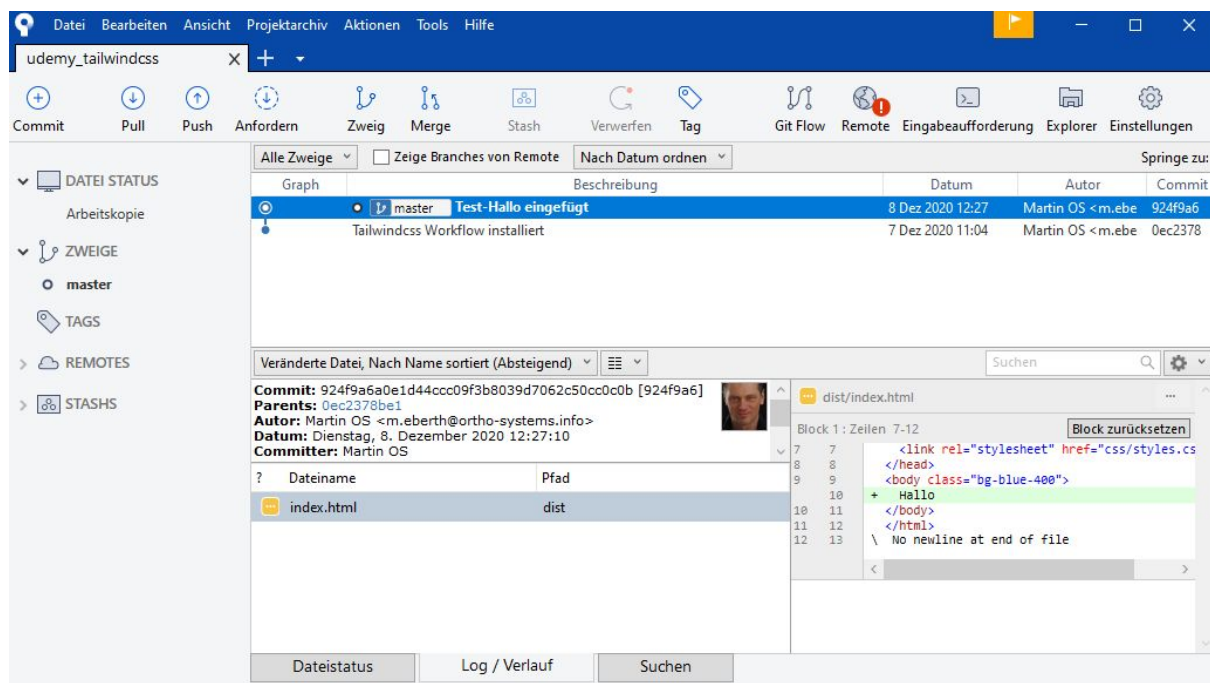
Hallo

```
</body>
```

Jetzt gehst du im **SourceTree** unten auf den **Reiter Dateistatus**, und siehst Deine Änderung. Danach klickst du links vertikal ca. in der Mitte auf **Alles vormerken**, damit Deine Änderung von links unten nach links oben, in den sogenannten "Staging-Bereich" wandert.

Jetzt kannst du ganz unten Deine Commit-Message einfügen, z.B. "Test-Hallo eingefügt" und klickst unten rechts auf **Commit**.

Wenn du jetzt wieder unten auf den Reiter "**Log/Verlauf**" gehst, dann siehst du Deinen Commit und Deine letzten Änderung(en) in der Git Historie:



[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udey**:

[www.udemy.com/https://www.udemy.com/course/tailwindcss-deutsch](https://www.udemy.com/course/tailwindcss-deutsch)

Hintergrund: Die Kommandozeile bedienen

Wir haben ja vorher schon die Kommandozeile zum installieren von Tailwindcss benutzt. Hierfür haben wir ein Terminal im Unix-Format verwendet, welches in VSC eingebaut ist. Es gibt natürlich auch "Standalone"-Terminalprogramme, wie z.B. [Git Bash für Windows](#) oder das "Terminal" für den Mac.

Hier die aller-aller-wichtigsten Tastatur-Shortcuts:

- Mit der **TAB-Taste** (links auf der Tastatur) kannst du Deine Eingabe **autovervollständigen**.
- Mit der **Pfeil-nach-oben-Taste** rufst du die letzten verwendeten Befehle auf und kannst diese somit wiederholen oder abgeändert wiederholen.
- Mit der **rechten Maustaste** kannst du zuvor kopierte **Inhalte einfügen**, bzw. markierten Text im Terminal selber einfügen.

Die 2 wichtigsten Befehle im Terminal:

cd = change directory wechselt in ein Verzeichnis. Wenn du nur "**cd**" cd alleine eingibst, kommst du in deinem Homeverzeichnis raus.

Mit **cd..** bewegst du Dich im Verzeichnis eine Ebene höher

Mit **ls** bzw. mit **ls -la** zeigst du Dir den aktuellen Verzeichnisinhalt an. Die Langform "ls -la" ist v.a. wichtig, wenn du versteckte Dateien, wie z.B. die .gitignore sehen willst.

Mehr brauchen wir in diesem Kurs wahrscheinlich nicht :-)

Hintergrund: Emmet

Mit Emmet kannst du Deine HTML-Coding-Geschwindigkeit erhöhen. Emmet ist schon in VSC eingebaut.

Die wichtigsten Shortcuts von Emmet:

- **! + Tab** erzeugt Dir ein HTML-Gerüst
- Einen beliebigen HTML-Tab erzeugst du einfach, indem du ihn ohne spitze Klammern hinschreibst und dann die Tab-Taste drückst, z.B.:

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udey.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

a + Tab = ``

- Eine Div mit einer **Klasse bzw. einer id** zu erzeugen, kannst du machen, indem du einfach nur einen Klassennamen angibt, so erzeugt z.B.: **.container#main + Tab**

`<div class="container" id="main"></div>`

- Du kannst auch mehrere Elemente auf einmal erzeugen, z.B **div*3 + Tab** erzeugt Dir:

`<div></div>`

`<div></div>`

`<div></div>`

Oder du kannst auch eine verschachtelte Struktur herstellen, z.B.

table>tr>td*2 + Tab erzeugt Dir:

`<table>`

`<tr>`

`<td></td>`

`<td></td>`

`</tr>`

`</table>`

Lorem

Mit dem Shortcut **Lorem** kannst du einen Blindtext erzeugen. Du kannst auch eine Länge des Blindtextes angeben, indem du z.B. sagst **Lorem 50**. Das bedeutet, daß Dein Blindtext eine Länge von 50 Wörtern hat. Der Standard ist übrigens 30 Wörter.

Du kannst auch verschachtelte Strukturen, mit Absätzen oder Listen mit Blindtext erzeugen mit **p*4>lorem**, oder **ul.generic-list>lorem10.item*4**

Mehr Infos zu Emmet und Lorem findest du hier:

<https://docs.emmet.io/abbreviations/lorem-ipsum/>

Der “+” - Operator

Beispiel: **a+p (TAB)** macht Dir:

``

`<p></p>`

Der “{}” - Operator

Damit kannst du Textinhalte in HTML einbinden.

Beispiel: **a*3{test}** macht Dir:

`test`

`test`

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

[www.udemy.com/https://www.udemy.com/course/tailwindcss-deutsch](https://www.udemy.com/course/tailwindcss-deutsch)

`test`

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

Der “**{#}**” - Operator

Beispiel: **a*3{test \$}** macht Dir:

`test 1`

`test 2`

`test 3`

`a*3{test`

Hintergrund: VSC Shortcuts / Bedienung

Ich bin wirklich kein Profi für VSC, weil ich normalerweise in PHPStorm arbeite aber hier ein paar wichtige Sachen. Viele dieser Abkürzungen funktionieren natürlich auch in anderen Editoren:

Mehrfacher Cursor

Wenn du den Cursor an mehreren Stellen im Dokument gleichzeitig setzen möchtest, dann drücke die **Alt-Taste** und klicke an die verschiedenen Stellen.

Um einen vertikalen Cursor über mehrere Zeilen zu erzeugen, verwendest du

Alt + Shift und ziehst die Maus nach unten.

Word-Wrap umschalten

Die Abkürzung um die Zeilen umbrechen zu lassen oder nach rechts aus dem Bild aus dem Bild heraus laufen zu lassen ist **ALT + Z**.

Einzelne Zeile duplizieren

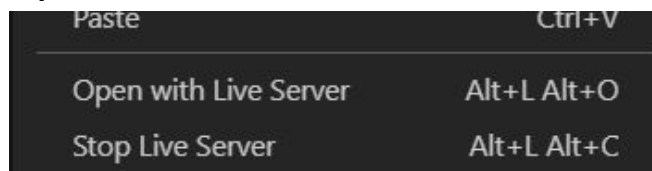
Um eine **einzelne Zeile nach unten zu duplizieren** platzierst du den Cursor an eine beliebige Stelle in dieser Zeile und drückst die **Alt + Shift + Pfeil-nach-unten-Taste**

Mehrere Zeilen duplizieren

Um eine **mehrere Zeilen nach unten zu duplizieren** markierst du diese Zeilen mit der Maus und drückst die **Alt + Shift + Pfeil-nach-unten-Taste**

Live-Server-Plugin starten

Hierfür klickst du mit der rechten Maus-Taste irgendwo in Dein Dokument und wählst “**Open with Live-Server**”



[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

[www.udemy.com/https://www.udemy.com/course/tailwindcss-deutsch](https://www.udemy.com/course/tailwindcss-deutsch)

Hintergrund: Die 3 Hauptbestandteile von Tailwindcss

Wir haben ja vorher gesehen, daß wir die 3 Bestandteile

[@tailwind base;](#)

[@tailwind components;](#)

[@tailwind utilities;](#)

in unsere **tailwind.css** importiert haben. Um jetzt zu testen, was sich hinter diesen Bestandteilen jeweils verbirgt, kann man die 3 Teile auch einzeln in die **tailwind.css** eintragen und dann jeweils `npm run build` laufen lassen, um dann das jeweilige Ergebnis in der `/dist/css/styles.css` zu sehen:

@tailwind base;

Ein "Normalisierungs-CSS" von ca. 550 Zeilen, siehe auch:

<https://github.com/sindresorhus/modern-normalize>

Diese Styles dienen dazu, ein ungestyltes HTML-Dokument auf allen Browsern gleich aussehen zu lassen.

@tailwind components;

Hier befindet sich die "Container" - Klasse, bzw. findet hier die Anwendung der Breakpoints statt. Dieser Teil hat etwas mehr als 200 Zeilen.

@tailwind utilities;

Hier finden sich alle übrigen Klassen mit denen du Deine Tailwind-Seiten aufbauen kannst. Bei den Utility-Klassen handelt es sich um sinnvolle Zusammenfassungen von Styles für gebräuchliche Use-Cases.

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy**:

www.udemy.com/course/tailwindcss-deutsch

Mini-Projekt

Container + Breakpoints

Viele Webseiten-Gerüste kannst du starten, indem du erst einmal einen **zentrierten** Container erzeugst, der über die **gesamte Höhe des Bildschirms** geht:

```
<div class="container min-h-screen mx-auto bg-gray-100">
</div>
```

Wie du siehst, verhält sich dieser Container bereits **responsive**, wenn du das Browserfenster in der Breite veränderst, kannst du beobachten, wie sich das Verhältnis von Rand und Containerbreite an bestimmten **Breakpoints** anpasst.

Die Breakpoints sichtbar machen

Um die **Breakpoints** während der Entwicklung der Website **sichtbar** zu machen, gibt es das Plugin "**Tailwind Debug Screens**". So wird es aktiviert:

Installation

```
npm install tailwindcss-debug-screens --save-dev
```

In tailwind.config.js eintragen

```
module.exports = {
  //...
  plugins: [
    require('tailwindcss-debug-screens'),
  ]
}
```

Optional - Oben links platzieren (in der tailwind.config.js)

```
theme: {
  debugScreens: {
    position: ['top', 'left'],
  },
}
```

Im HTML-Dokument anwenden

```
<body class="debug-screens">
```

Wichtig: `npm run build` laufen lassen zur Aktivierung.

Jetzt kannst du die 5 Breakpoints "**sm, md, lg, xl und 2xl**" während der Entwicklung sehen. Wenn du später Live gehst, kannst du einfach die Klasse "debug-screens" aus dem Body-Tag entfernen.

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udey**:

www.udemy.com/course/tailwindcss-deutsch

```
git commit -m 'Container + Breakpoints'
```

Hier der Link zur Github-Seite des Plugins:

<https://github.com/jorenvanhee/tailwindcss-debug-screens>

Planung der Mini-Website

Dein Nachbar ist ein freiberuflicher Personal Trainer im Fitness-Studio und hat aufgrund von Covid19 sein Einkommen verloren. Er bittet Dich eine "Web-Visitenkarte" zu machen, denn er möchte jetzt sein 1:1-Personal-Training Online über Skype anbieten.

Die Website enthält ein **Bild**, ein wenig **Text**, und einen **E-Mail-Link**, sowie einen **Telefon-Link** für die Handy-Ansicht. Ausserdem möchte er eine ganz bestimmte Hintergrundfarbe haben, welche der Farbe auf seiner Visitenkarte entspricht. Der Farbwert ist: **#212f49**.

Ich habe Dir das Bild und den Text - bereits als grobes HTML formatiert - zum Download im Anhang dieser Lektion bereit gestellt.

Bitte

- Lege einen Ordner "img" im "dist" - Verzeichnis an. Und lege dort das Bild rein.
- Kopiere das HTML in die Container-Div.

Wie du im HTML siehst, habe ich die Inhalte schon einmal grob mit Divs und Absätzen formatiert, sowie ein paar Kommentare eingefügt.

Ausserdem habe ich überall den **Hauptinhalts-Divs eine ID** gegeben, z.B.:

```
<!-- Leistungen -->
```

```
<div id="leistungen">
```

Dies ist eine **gute und sinnvolle Angewohnheit** und wird Dir später helfen Dich im Code zurecht zu finden!

Die Website sollte jetzt in etwas so aussehen:

[Nach oben](#)

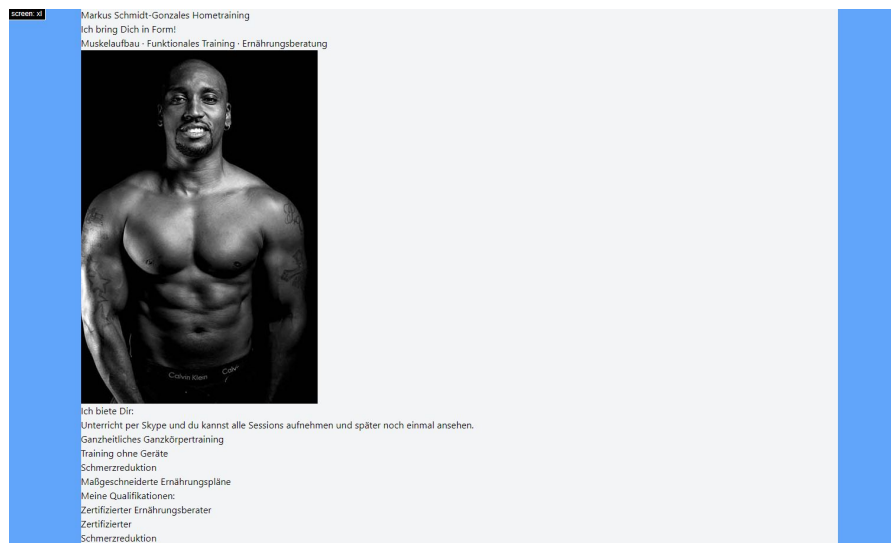
Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udey.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udey**:

www.udemy.com/course/tailwindcss-deutsch



git commit -m 'Planung der Mini-Website'

Seiteninhalte grob anordnen

Schritt 1

Zunächst fassen wir bestimmte Bereiche der Website in **Sections** zusammen. Das gibt uns noch zusätzliche Möglichkeiten später für das Ausrichten und **Einfügen von Abständen**.

Wir machen **Sections** für:

1. Die Überschriften und das Zitat oben: `<section id="top">`
2. Für das Bild und "Leistungen" welche auf normalen Bildschirmen nebeneinander erscheinen sollen: `<section id="bild_leistungen">`

Schritt 2

Jetzt geben wir dem gesamten Container ein **Padding**: `"px-6 py-8"`.

(x = horizontal, y = vertikal)

Schritt 3

Jetzt ordnen wir die Überschriften oben mittig an, indem wir der Div aussenherum die Klasse `"text-center"` geben. Außerdem machen wir unter dem Überschriftenblock einen Abstand, indem wir der "top-Section" unten ein Margin geben mit der Klasse `"mb-8"`.

Schritt 4

Jetzt wollen wir das Bild und den Block "Leistungen" **gleichmäßig nebeneinander anordnen**. Der einfachste Weg dafür, ist der übergeordneten Section die Klassen

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udey.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

[www.udemy.com/https://www.udemy.com/course/tailwindcss-deutsch](https://www.udemy.com/course/tailwindcss-deutsch)

`grid grid-cols-2 gap-4` zu geben. Damit wird der Inhalt auf **2 Spalten** verteilt, und in der Mitte ist, wenn man den Bildschirm zusammen schiebt immer ein **Lücke** von "`gap-4`", was 1rem bzw. 16px entspricht.

Schritt 5

Den Block "**Preise und Bezahlung**" wollen wir **mittig** anordnen, aber der Block soll v.a. auf großen Bildschirmen **nicht die ganze Breite** einnehmen.

Wir platzieren den Inhalt verschachtelt in einer Section und dann noch einmal in einer Div.

Der Section geben wir erstmal einen Abstand oben mit: **mt-12**.

Dann geben wir der inneren Div eine **Breite von w-8/12** (also $\frac{2}{3}$), und setzen sie auf **mx-auto** für die **mittige Anordnung**. Das Ganze sieht jetzt so aus:

```
<!-- Preise und Bezahlung -->
<section class="mt-12" id="preise_bezahlung">
  <div class="w-8/12 mx-auto">
    <h2>Preise und Bezahlung</h2>
    <p>Mein wichtigstes Ziel ist...</p>
    <p>In einer kostenlosen Anfangssession...</p>
  </div>
</section>
```

Schritt 6

Aus der Div "Kontakt" machen wir wieder eine Section und geben dieser wieder oben den Abstand von **mt-12**.

Auch hier machen wir am Besten wieder eine Div um die beiden inneren Divs, und wir machen diese Div wieder zweispaltig mit **grid grid-cols-2**. Dann wieder mittig mit **mx-auto**, bloss soll diesmal die eingenommene Breite etwas schmaler sein, deswegen verwenden wir für die Breite: **w-6/12** (also die Hälfte des Platzes).

Das Ergebnis ist dann:

```
<!-- Kontakt -->
<section class="mt-12" id="kontakt">
  <div class="grid w-6/12 grid-cols-2 mx-auto">
    <div id="email">
      <p>Schreibe mir: <span>markusgonzales@gmail.com</span></p>
      <a href="mailto:markusgonzales@gmail.com">Email senden</a>
    </div>
    <div id="whatsApp_telefon">
      <p>WhatsApp:</p>
      0275 / 555 44 33
    </div>
  </div>
</section>
```

Somit sind wir mit der groben Formatierung durch, das interessante ist, dass die Seite auch schon bei **relativ kleinen Bildschirmen gut aussieht**, sogar das **Bild schrumpft**, wenn wir den Bildschirm zusammen schieben.

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

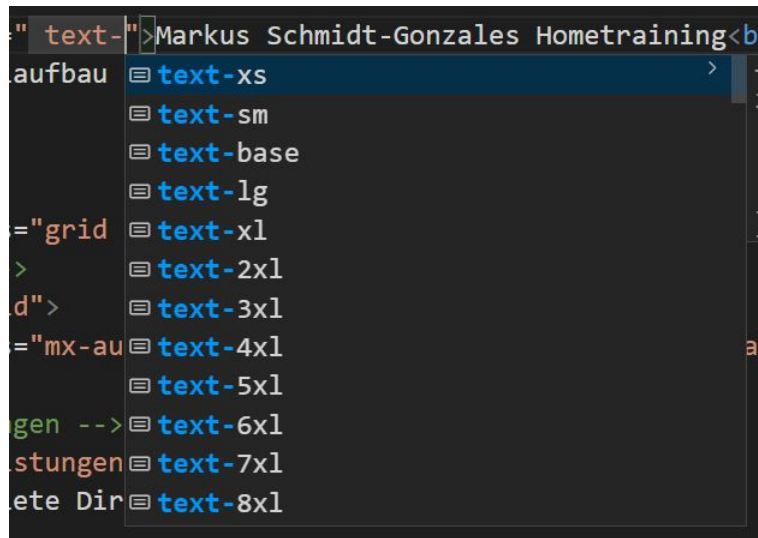
Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udey**:

www.udemy.com/course/tailwindcss-deutsch

```
git commit -m 'Seiteninhalte grob anordnen'
```

Überschriften formatieren und voreinstellen

Wir gehen jetzt mal in die ganz **obere h1** rein und passen die Größe an. Die Größen findest du indem du **im class-Attribut "text-..." tippst** und in der Auswahl etwas nach unten scrollst:



Du kannst so in Verbindung mit dem Live-Server sehr schnell verschiedene Textgrößen ausprobieren. Wir nehmen jetzt mal vorrübergehend:

- h1: text-5xl
- h2: text-3xl
- h3: text-2xl

Ausserdem geben wir der **h1** noch einen Abstand unten von **mb-4**.

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udey.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

Überschrift-Größen vordefinieren

Natürlich wollen wir nicht für alle einzelnen Überschriften auf einer großen Website die Größe angeben müssen. Deswegen können wir in der **tailwind.css** bestimmte Sachen voreinstellen.

Als erstes jedoch starten wir im Terminal `npm run watch`, damit wir nicht bei jeder Änderung jedes mal `npm run build` laufen lassen müssen:

```
@layer base {  
  h1 {  
    @apply text-5xl;  
  }  
  h2 {  
    @apply text-3xl;  
  }  
  h3 {  
    @apply text-2xl;  
  }  
}
```

Das **@layer base** bedeutet, dass unsere Definitionen in den kompilierten Stylesheets am Ende der Base-Styles eingefügt werden.

Wie du siehst kannst du mit **@apply** wiederum auf die Utility-Klassen zugreifen.

Jetzt können wir in der `index.html` wieder alle Größen-Klassen aus den Überschriften **entfernen**.

Aber, **WICHTIG!**

Die Größen können immer noch manuell im HTML überschrieben werden!

Also können wir z.B. die voreingestellte Größe der H1 immer noch mit z.B. `text-8xl` überschreiben. Dies entspricht dem Überschreiben von Styles im HTML mit dem `style="..."` -Attribut!

Mehr Infos in der Tailwindcss-Doku:

<https://tailwindcss.com/docs/adding-base-styles>

```
git commit -m 'Überschriften formatieren und voreinstellen'
```

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

Text-Formatierungen in Tailwindcss

Link zu Text-Formatierungen in der Tailwind-Doku: <https://tailwindcss.com/docs/font-family>

System-Fonts (css: font-family)

Tailwindcss hat schon 3 Fonttypen integriert, weche du mit den Klassen

- font-sans
- font-serif
- font-mono

aufrufen kannst.

Voreingestellt ist font-sans (Systemschrift ohne Serifen)

Schrift-Größen (css: font-size)

Die Schriftgrößen gehen, **von text-sm bis text-5xl**, wobei **text-base** die Standardgröße von 1rem (16px) ist.

Zeilen- und Buchstabenabstand (css: line-height bzw. letter-spacing)

Etwas gewöhnungsbedürftig sind die Angaben für den Zeilenabstand “leading-”, sowie für den Buchstabenabstand “tracking-”.

Der Begriff “leading-” kommt aus der alten Drucktechnik, wo Zeitungen und Bücher noch von Hand gesetzt wurden, und “leading” bedeutet das Einfügen von Bleistreifen, zwischen den Zeilen (lead = Blei).

Schriftstärke (font-weight)

Die Schriftstärke geht von “font-thin” bis “font-black”: <https://tailwindcss.com/docs/font-weight>

Natürlich muss die entsprechende Schriftstärke auch installiert sein, wenn du einen externen Font verwendest. Die Standard-Schriftstärke ist: font-normal (400)

Weiteres

Ansonsten stellen die Text-Utility-Klassen von Tailwindcss oft Abkürzungen vom normalen CSS dar: z.B. ist die Klasse “**capitalize**” eine Kurzform von: text-transform: capitalize

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

[www.udemy.com/https://www.udemy.com/course/tailwindcss-deutsch](https://www.udemy.com/course/tailwindcss-deutsch)

Eigenen Überschriften-Font definieren

Um einen Webaufttritt interessanter zu gestalten, kann man eigene Fonts einfügen. Wir wollen jetzt zur Übung für unsere Überschriften den Font "Oswald" [von Google herunter laden](#).

Wir laden den Font herunter weil die Rechtssprechung bzgl. Datenschutz bzgl. Google Fonts in Deutschland [nicht eindeutig geklärt ist](#). Wenn wir den Google Font per URL einbinden, dann verbindet sich Deine Website mit Google und übermittelt alle möglichen Daten...

Schritt 1: Font herunterladen und lokal speichern

Du gehst oben rechts auf den Button **Download Family** und bekommst ein Zip bzw. Ordner mit verschiedenen Schriftstärken.

Den Inhalt dieses Zips legen wir jetzt in einen Ordner **/dist/fonts/Oswald**, welchen wir anlegen.

Schritt 2: Den Font lokal importieren

In der **tailwind.css** innerhalb von **@layer base** und oberhalb der Überschriften fügen wir ein:

```
@font-face {  
  font-family: Oswald;  
  src: url(/dist/fonts/Oswald/Oswald-Bold.ttf) format("truetype");  
}
```

Wichtig ist die Formatangabe **"truetype"**, nicht etwas **"tff"**, wie man meinen könnte, wenn man sich nicht auskennt [und die offizielle Tailwindcss-Doku liest](#).. ;-)

Wenn wir jetzt **npm run watch** laufen haben, wird der Import gleich in die **styles.css** kompiliert, und wir bekommen dort, wenn wir nach "Oswald" suchen:

```
@font-face {  
  font-family: Oswald;  
  
  src: url(/dist/fonts/Oswald/Oswald-Bold.ttf) format("truetype");  
}
```

Tipp: Wenn irgendetwas nicht funktioniert, kannst du mit gedrückter "STRG"-Taste auf den Link in der URL klicken, um zu sehen, ob der Pfad stimmt!

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

Schritt 3: Das Tailwindcss-Theme erweitern

In der **tailwind.config.js** fügen wir unter **“theme”** und **“extend”** ein:

```
theme: {  
  extend: {  
    fontFamily: {  
      ueberschrift: ['Oswald']  
    }  
  },  
}
```

Der Name “headline” ist hier frei wählbar!

Im Ergebnis ist jetzt im HTML die Klasse “font-headline” wählbar! Wie cool ist das denn?



Schritt 4: Die neue Klasse “font-headline” automatisch zuweisen

Jetzt gehen wir aber noch einen Schritt weiter, und geben den Überschriften H1, H2 und H3 automatisch die neue Klasse “font-headline”. Deswegen geben wir in der **tailwind.css** den Überschriften über **@apply** die neue Klasse:

```
h1 {  
  @apply text-5xl font-headline;  
}  
h2 {  
  @apply text-3xl font-headline;  
}  
h3 {  
  @apply text-2xl font-headline;  
}
```

Jetzt müssen wir den Style “font-headline” nicht mehr manuell den Überschriften hinzufügen!

In der kompilierten **/dist/styles.css** kann man jetzt auch schön sehen:

```
h1 {  
  font-family: Oswald;  
  font-size: 3rem;  
  line-height: 1;  
}
```

Der Kopf-Bereich unserer Website sollte jetzt ungefähr so aussehen:

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

Markus Schmidt-Gonzales Hometraining Ich bring Dich in Form!

Muskelaufbau · Funktionales Training · Ernährungsberatung

Eine eigene Farbe hinzufügen

Wie wir ja am Anfang in der Projektbeschreibung gesehen haben, möchte der Kunde eine eigen Farbe dem Webaufttritt hinzufügen, nämlich: **#212f49**.

Nichts einfacher als das:

Schritt 1: Farbe ins tailwind-Theme einfügen

In der **tailwind.config.js** fügen wir unter **“theme”** und **“extend”** ein:

```
colors: {  
  hauptfarbe: '#212f49'  
}
```

Der Name **“hauptfarbe”** ist natürlich frei wählbar.

Jetzt können wir z.B. im Body-Tag der index.html sagen:

```
class="bg-hauptfarbe debug-screens"
```

Auch wenn wir in der kompilierten **dist/styles.css** nachsehen können wir die **“hauptfarbe”** in allen möglichen Varianten finden, z.B. :

```
.divide-hauptfarbe > :not([hidden]) ~ :not([hidden]) {  
  --tw-divide-opacity: 1;  
  border-color: rgba(33, 47, 73, var(--tw-divide-opacity));  
}
```

Schritt 2: Die neue Hauptfarbe den Überschriften zuweisen

In der **/tailwind.css** können wir jetzt auch wieder den Überschriften **voreingestellt** die Hauptfarbe zuweisen:

```
h1 {  
  @apply text-5xl font-headline text-hauptfarbe;  
}  
h2 {  
  @apply text-3xl font-headline text-hauptfarbe;  
}  
h3 {  
  @apply text-2xl font-headline text-hauptfarbe;  
}
```

Jetzt haben - wie erwartet - unsere Überschriften die Hauptfarbe, ohne das wir sie noch extra zuweisen müssen.

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Die Listen formatieren

Wie man sieht, sind bei Listen in Tailwindcss alle Styles zurück gesetzt.

Zuerst können wir der oberen Liste (bei "Ich biete Dir") mal den Style "**list-disc**" geben, damit werden die üblichen Bulletpoints sichtbar.

Allerdings sehen wir dass die Liste etwas nach Links versetzt ist. Das hat mit den Styles **list-outside** bzw. **list-inside** zu tun. Der Style **list-outside** ist **voreingestellt**.

List-Inside und List-Outside

List-Outside

Ich biete Dir:

- Unterricht per Skype und du kannst alle Sessions aufnehmen und später noch einmal ansehen.

wenn wir das auf **list-inside** umstellen, dann rücken die Punkte zwar unter die Überschriften, aber mehrzeilige Listenpunkte funktionieren nicht mehr richtig:

List-Inside

Ich biete Dir:

- Unterricht per Skype und du kannst alle Sessions aufnehmen und später noch einmal ansehen.

Deswegen Merkregel: Wir lassen die Voreinstellung "list-outside" und geben der Liste einfach ein linkes Margin, z.B. **ml-6**.

Danach geben wir den Listen und den Listenpunkten noch ein bisschen Margin hier und dort. Wenn du den Listenpunkten unten (ausser dem Letzten) etwas Margin geben möchtest, dann kannst du im VSC den cursor mit Alt + Shift bzw. nur mit Alt für die unteren 3 Listenpunkte an mehreren Stellen gleichzeitig platzieren, dann brauchst du z.B. `class="mb-2"` nur einmal zu tippen.

[Nach oben](#)

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

[www.udemy.com/https://www.udemy.com/course/tailwindcss-deutsch](https://www.udemy.com/course/tailwindcss-deutsch)

Den Rest formatieren

Unten auf der Seite bei "Preise und Bezahlung" passen wir nur etwas die Abstände an und setzen den Text ebenfalls auf "text-lg" so wie die Listenelemente.

Den unteren Teil "**Kontakt**" setzen wir auch auf eine Breite von `w-8/12 mx-auto`, und was wichtig ist, ist, daß die Überschrift "**Kontakt**" ebenfalls in eine div mit `w-8/12 mx-auto` gesetzt wird.

Da wir annehmen, daß auf dieser Seite jeglicher Text innerhalb von `<p></p>` immer die Klasse text-lg hat, legen wir diese Definition in der **tailwind.css** an:

```
p {  
  @apply text-lg  
}
```

Den **Zeilenabstand** der **zweizeiligen Überschrift** erhöhen wir mit: `leading-snug`

Ausserdem geben wir ganz **unten** noch einen **kleinen Abstand** hinzu, indem wir der **section#kontakt** ein `mb-5` geben.

Einen Button mit Icon designen

Wir wollen jetzt den Mail-Link "Email senden" designen.

Mail-Icon verwenden

Zunächst suchen wir uns ersmal ein Icon. Wir gehen auf die Seite <https://feathericons.com> und suchen im Suchfeld nach "mail".

Dann stellen wir rechts bei "**Customize**" eine **Größe von 16px** ein, und klicken dann auf das E-Mail-Symbol um das Icon herunter zu laden.

Wir **öffnen** das heruntergeladene Icon im **Browser** und lassen uns den **Quelltext** anzeigen (rechte Maustaste).

Den Quelltext kopieren wir und setzen ihn innerhalb des Email-Links vor den Text "Email senden".

Wir löschen das vorhandene **class-Attribut** im SVG, dann verwenden wir im VSC die Tastenkombination **Alt + Z** um das Wordwrap auszuschalten, und setzen im SVG ein eigenes Klassenattribut ganz vorne ein.

In das Klassenattribut setzen wir als erstes **h-5** ein, um dem Icon eine Höhe von

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

[www.udemy.com/https://www.udemy.com/course/tailwindcss-deutsch](https://www.udemy.com/course/tailwindcss-deutsch)

1,25 rem = 20px zu geben, somit können wir auch das `width="16" height="16"` aus dem Icon löschen.

Um das Icon vor die Schrift zu setzen verwenden wir im umgebenden a-Tag die Klasse **flex**, und um das Icon vertikal an dem Text dahinter auszurichten, verwenden wir - ebenfalls im a-Tag: **items-center**.

Jetzt geben wir dem SVG noch einen Abstand rechts: **mr-1** und eine Linienstärke von 1, mit der Klasse **stroke-1**.

Das `stroke-width="2"` aus dem Original-SVG können wir jetzt **löschen**.

Den Link als Button stylen

Was macht einen Button aus?

- Hintergrundfarbe und Padding
- Randline und Abrundung
- Hover- und evtl. Fokus-Zustände

Alle folgenden Klassen werden auf das a-Tag angewendet.

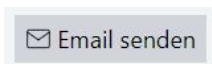
Hintergrundfarbe und Padding

`px-2 py-1 bg-gray-300`

Jetzt haben wir nur noch das Problem, dass sich der Button über die gesamte zur Verfügung stehende Breite erstreckt, dies lösen wir mit der Klasse: **w-max**

Ohne w-max:

Mit w-max:



Rundung und Schatten

Mit **rounded** machen wir die Ecken soft, und mit **shadow-sm** setzen wir den Button etwas vom Untergrund ab.

Hover und Transition

Bei Hover wollen wir den Button etwas **abdunkeln**, also geben wir ihm einen dunkleren Farbton mit **hover: bg-gray-400**.

Außerdem entfernen wir den Schatten beim Hover, damit der leichte Eindruck entsteht, dass der Button sich auf den Hintergrund zu bewegt: **hover:shadow-none**

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**
www.udemy.com/course/tailwindcss-deutsch

Als letztes bauen wir noch einen sanften Übergang zum Hover ein, indem wir die Klasse **transition** hinzufügen.

Focus

Um den User etwas Feedback zu geben, wenn wir den Button drücken, können wir ihm noch einen "Ring" geben mit: **focus:ring-2 focus:ring-yellow-500**
git commit -m 'Einen Button mit Icon stylen'

Hintergrund: Kostenlose Icons

Tolle Auswahl in verschiedenen Styles:

<https://material.io/resources/icons>

Größe und Linienstärke einstellbar beim Download:

<https://feathericons.com/>

Auch **sehr schön!** Empfohlen von Tailwindcss

<http://www.zondicons.com/icons.html>

In 2 verschiedenen Größen= Ausführungen erhältlich - sehr praktisch!

<https://heroicons.com>

Klasse Hintergrund SVG-Patterns! In Farbe einstellbar. Super!!!

<https://www.heropatterns.com/>

Einführung ins Responsive Design mit Tailwindcss

Wir haben in Tailwind folgende Breakpoints:

sm	md	lg	xl	2xl
ab 640px	ab 768px	ab 1024px	ab 1280px	ab 1536px

Die oben genannten Kürzel können wir allen Utility-Klassen mit Doppelpunkten voranstellen, z.B.: **lg:bg-blue-500**

würde bedeuten, dass ab "lg" also ab einer Viewport-Breite von 1024px und mehr wir einen Hintergrund der Farbe *bg-blue-500* haben.

Wenn wir jetzt z.B. eine Hintergrundfarbe von **bg-gray-100 lg:bg-blue-500** definieren, dann bedeutet das, daß für kleine Bildschirme bis einer maximalen Breite von 1023px ein leicht grauer Hintergrund gilt, und dann ab lg = 1024px ein blauer Hintergrund zu sehen ist.

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udey**:

[www.udemy.com/https://www.udemy.com/course/tailwindcss-deutsch](https://www.udemy.com/course/tailwindcss-deutsch)

Das läßt sich auch noch feiner einteilen, es ginge z.B. auch:

bg-gray-100 lg:bg-blue-500 xl:bg-red-500

Vorgehensweise um die Website anzupassen

Für eine Website von der man annimmt, daß sie auf allen Bildschirmgrößen von ganz klein bis ganz groß angesehen werden wird, würde ich erst einmal die Website grob für die mittlere Größe **lg** optimieren.

Danach würde ich mit den kleinen Größen weiter machen und wie folgt vorgehen:

1. Du **schiebst** die Website langsam von **lg** aus **zusammen**, bis du den Punkt heraus findest, wo alles gerade noch gut aussieht.
2. Du identifizierst die Elemente bzw. Klassen, die bei einer kleineren Auflösung **NICHT mehr gut** aussehen, und setzt bei Ihnen das Kürzel voran bei welcher es gerade noch gut aussieht.
3. Dann **änderst** du die entsprechende Klassen so **ab**, bzw. duplizierst sie OHNE KÜRZEL, das heisst du definierst die Klassen von ganz klein bis zu Deinem kritischem Breakpoint.
4. Dann machst du noch eventuell nötige Anpassungen für die Breakpoints dazwischen, also **sm** und **md**.
5. Wenn alles bis **lg** soweit passt kümmerst du dich **am Schluss** um die grossen Auflösungen **xl** und **2xl**.

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udey.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

Optimierung für Mittlere Bildschirme (md)

Unsere Site sieht momentan **bei "lg"** so aus:



Wenn wir jetzt aber die Website auf md zusammenschieben bekommen wir bereits 2 Probleme:

Zu einen sind die Überschriften oben zu groß und haben auch gefühlt etwas zu viel Abstand vom Seitenkopf:



[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udey**:

www.udemy.com/course/tailwindcss-deutsch

Zum anderen rückt unten bei "Kontakt" die E-Mail-Adresse und das "WhatsApp" etwas eng zusammen, hier könnten wir also ruhig auf die gesamte Breite gehen:



Was die große **Überschrift (H1)** betrifft, haben wir ja in der tailwind.css eine Größe von `text-5xl` voreingestellt. Dies überschreiben wir jetzt, indem wir direkt beim H1-Tag angeben: `text-4xl lg:text-5xl`

Das heisst erst ab "lg" haben wir ab jetzt "5xl", und davor ist es "4xl".

Das selbe machen wir mit der **Unterüberschrift (H2)**.

Dort ist `text-3xl` voreingestellt und wir überschreiben das mit:

`text-2xl lg:text-3xl`

Der Abstand zum Seitenrand oben (und unten) ist in der Container-Klasse geregelt, mit derzeit: `py-8`.

Dies ändern wir in: `py-4 lg:py-8`

Was die **Breite des Abschnitts "Preise und Bezahlung"**, betrifft, so wird dort derzeit 3 Mal die Klasse `w-8/12` verwendet.

Dies ändern wir schlicht in `lg:w-8/12`. Die anderen Klassen können wir momentan unberührt lassen.

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udey.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

Umbau der Section "Kontakt" in Flexbox

Bei Kontakt wollen wir eine gleichmäßige horizontale Verteilung der beiden Einheiten "Email" und "WhatsApp" erreichen.

Wenn etwas gleichmäßig verteilt werden sollte dann sollte man immer an Flexbox denken.

Wir werden also alle Angaben der Spalten und Raumaufteilung für die Größe md **komplett aufheben**, indem wir überall ein "lg:" davorsetzen:

lg:flex-none lg:grid lg:w-8/12 lg:grid-cols-2 lg:mx-auto

Dann wenden wir auf den selben Container die Flexbox an mit "**flex**" und die gleichmäßige Verteilung erreichen wir mit "**justify-evenly**". Das Gesamtergebnis ist dann:

flex justify-evenly lg:flex-none lg:grid lg:w-8/12 lg:grid-cols-2 lg:mx-auto

Als letztes wollen wir nur noch bei der Breite md, den Abschnitt ab "Preise und Bezahlung" minimal näher an das Bild heran rücken. Die section#preise_bezahlung hat ein derzeit ein mt-10, das ändern wir ganz einfach in: **mt-6 lg:mt-10**.

Optimierung für Kleine Bildschirme (sm)

Als nächstes schieben wir den Bildschirm bis zur Stufe "sm" zusammen.



Was hier sofort auffällt, ist, dass der Abschnitt "Meine Qualifikationen" unten neben dem Bild hinausläuft. Es wäre doch schön, wenn dieser Abschnitt unter dem Bild wäre!

Hier gehe ich pragmatisch vor:

Ich fasse die Überschrift "Meine Qualifikationen" und die Liste darunter in einer **section#qualifikationen_rechts** zusammen, und geben dieser Section die Klassen: **hidden md:block**

Achtung: Hier ist es wichtig wirklich die Kurzformen für die CSS-Eigenschaft "display" zu nehmen (also display:hidden, bzw. display:block) und NICHT die Tailwind-Shortcuts: *visible* bzw. *invisible*. Würde man zum Ausblenden "invisible" nehmen, was eine Abkürzung für

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile - ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

“visibility:hidden” wäre, dann wäre zwar der Section-Inhalt unsichtbar, aber Es wäre immer noch der Abstand der Section vorhanden und würde die übrigen Inhalte nach unten drücken!

Als nächstes werden wir also die ausgeblendete Section kopieren und über “Preise und Bezahlung” einfügen.

Die einzige Klasse die wir hier benötigen ist: **md:hidden**

Wie cool und einfach ist das denn? :-)

Als letztes können wir jetzt noch den Abstand der Listeneinträge rechts neben dem Bild etwas erhöhen, denn wir haben ja den Platz.

Wir ersetzen also bei den Listeneinträgen bei “Ich biete Dir” die Klasse **mb-2** mit:
mb-6 md:mb-2

Jetzt hätten wir wieder eine Stufe geschafft und das Ganze sieht so aus:

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udey**:

www.udemy.com/course/tailwindcss-deutsch

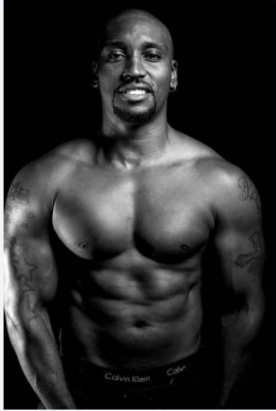
screen: sm

Markus Schmidt-Gonzales

Hometraining

Ich bring Dich in Form!

Muskelaufbau · Funktionales Training · Ernährungsberatung



Ich biete Dir:

- Unterricht per Skype und du kannst alle Sessions aufnehmen und später noch einmal ansehen.
- Ganzheitliches Ganzkörpertraining
- Training ohne Geräte
- Schmerzreduktion
- Maßgeschneiderte Ernährungspläne

Meine Qualifikationen:

- Zertifizierter Ernährungsberater
- Zertifizierter Fitnesstrainer
- 5 Jahre Erfahrung im 1:1 - Training

Preise und Bezahlung

Mein wichtigstes Ziel ist, daß du mit Deinen Zielen voran kommst. In einer kostenlosen Anfangssession besprechen wir, was du erreichen willst und wieviel du maximal ausgeben möchtest oder kannst. Dann finden wir garantiert eine Lösung, die zu Dir passt.

Kontakt

markusgonzales@gmail.com **WhatsApp:** 0275 / 555 44 33

Email senden


[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udey.com - siehe Kopfzeile- ist untersagt.

Seite 36

Optimierung für ganz kleine Bildschirme (Handy)



Ich biete Dir:

- Unterricht per Skype und du kannst alle Sessions aufnehmen und später noch einmal ansehen.
- Ganzheitliches Ganzkörpertraining
- Training ohne Geräte
- Schmerzreduktion
- Maßgeschneiderte Ernährungspläne

Kontakt

markusgonzales@gmail.com **WhatsApp:** 0275 / 555 44 33

Wenn wir das Browserfenster ganz eng zusammen schieben, dann sehen wir, daß das Bild mit der Liste rechts daneben nicht mehr funktioniert. Es wäre besser wenn das Bild und die Liste untereinander wären.

Ein weiteres Problem gibt es ganz unten auf der Seite, wo sich jetzt die E-Mail-Adresse und das "Whatsapp" überlappen. Auch hier wollen wir beides lieber untereinander haben.

Nichts einfacher als das! Alles was wir machen müssen, ist bei beiden Fällen das "grid-cols-2" erst ab einer Größe von "sm" gelten zu lassen. Das heißt wir machen beides Mal aus "grid-cols-2" : "**sm:grid-cols-2**"

Jetzt müssen wir noch einige Kleinigkeiten korrigieren, zunächst sollen die **Abstände zwischen den Listenelementen im Bereich "Ich biete Dir"** bei ganz kleinen Bildschirmen jetzt doch wieder kleiner werden, wir ersetzen also **m-6** mit **mb-3 sm:mb-6**

Auch ist der Abstand über "Qualifikationen zu hoch", (bei sm übrigens auch), dies korrigieren, wir indem wir bei der **H3 bei "Meine Qualifikationen"** den Abstand oben von **mt-8** auf **md:mt-6** korrigieren.

[Nach oben](#)

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

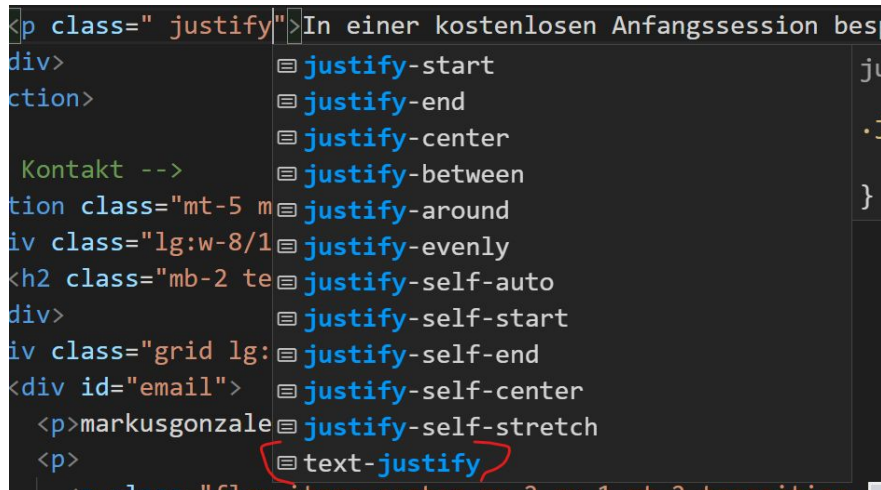
Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

Blocksatz

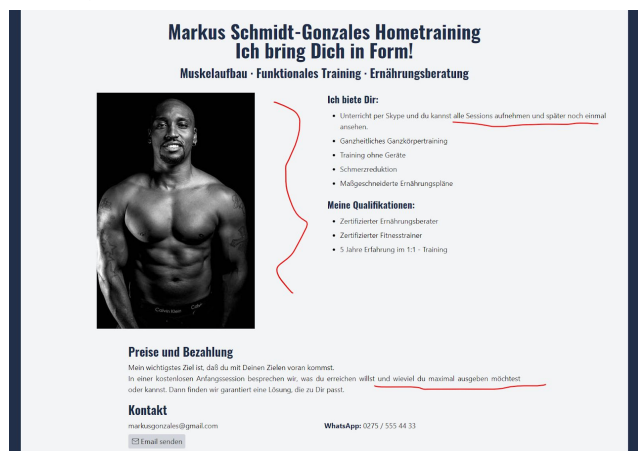
Der zweite Absatz bei "Preise und Bezahlung" kann auf **text-justify** gesetzt werden. Dies sieht auf allen Bildschirmgrößen gut aus, deswegen braucht es keine weitere Angabe.

Tipp: Wenn du übrigens nicht genau weißt, wie eine Klasse heisst, dann tippe einfach einen Wortbestandteil, den du kennst (in diesem Fall "justify" und dann findest du beim durchscrollen der Liste of die Lösung als Vorschlag in der Tailwind-Extension von VSC:



Optimierung für große Bildschirme XL und 2XL

Bei sehr großen Bildschirmauflösungen kann es passieren, daß die Elemente zu weit auseinander gezogen wirken. Dies gilt natürlich insbesondere, wenn wir nur wenig Inhalt haben, wie hier:



Das wird dann generell schwer für das Auge zu erfassen, weil das Auge zu sehr hin- und herwandern muss, um zusammenhängende Inhalte ans Gehirn weiter zu leiten.

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemey:**

www.udemy.com/course/tailwindcss-deutsch

Anpassung an XL

Bei XL genügt es wenn wir den Bereich oben mit dem Bild und den Listen rechts etwas zusammenrücken mit: **xl:w-8/12 xl:mx-auto**

Ausserdem würde ich jetzt doch ab der Stufe “lg” die Listenelemente rechts weiter auseinander ziehen mit jeweils: **lg:mb-6**

Man kann auch ruhig ab “md” über “Meine Qualifikationen” etwas Abstand reintun, indem wir in der Überschrift sagen: **md:mt-12**

Anpassung an 2XL

Bei 2XL würde ich nur noch unten die Bereiche “*Preise und Bezahlung*”, sowie “Kontakt” etwas enger fassen, in dem wir immer dort wo die Klasse **lg:w-8/12** steht, dies mit **2xl:w-7/12** ergänzen.

Noch ein letztes: Ab der Größe “lg” würde ich dringend den Zeilenabstand oben in H1 erhöhen, das schaffen wir mit: **lg:leading-snug**

Warum mussten wir das machen - denn eigentlich hatten wir doch schon ein “leading-snug” als Klasse?

Der Grund ist der, dass wir ab “lg” auch die Schriftgröße geändert haben. Deswegen müssen wir auch den Zeilenabstand sozusagen “refreshen”

Hintergrundmuster

Wir werden jetzt sowohl für den blauen Hintergrund, als auch für den hellgrauen Hintergrund im Container ein Hintergrundmuster verwenden.

Dazu gehen wir auf die Seite: <https://www.heropatterns.com/>

Dort tragen wir als Hintergrundfarbe den Farbwert **#212f49** ein, welchen wir ja als “hauptfarbe” definiert hatten (siehe *tailwind.config.js*)

Dann stellen wir evtl. noch die Foreground Color ein und spielen mit dem Schieberegler für die Opacity, und klicken dann auf das gewünschte Muster.

Im Fenster erhalten wir einen Code, welchen wir direkt auf das Body-Element anwenden können. Wir kopieren von dort nur den Bereich “background-image...” und setzen diesen Style dann als “body” in die **tailwind.css** ein:

```
body {  
  background-image: url("data:image/svg+xml,  
}
```

Achtung! **npm run build** bzw. **npm run watch** nicht vergessen!

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemey.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

[www.udemy.com/https://www.udemy.com/course/tailwindcss-deutsch](https://www.udemy.com/course/tailwindcss-deutsch)

Das selbe machen wir jetzt noch mit der **.container** - Klasse. (Muster auswählen + in der tailwind.css einsetzen)

Ich habe für den Hintergrund "Jigsaw" verwendet und für den Container "Topography"

Zitat oben mit Anführungszeichen und anderer Farbe

Für die zweite Zeile der Überschrift wollen wir eine andere Farbe (Indigo) nehmen, wir wollen den Text kursiv und leicht transparent setzen und als Zitat darstellen (<q>)

Interessant dabei ist, dass das HTML-Element <q>Text</q> je nach Länderkennung im HTML-Tag die Anführungszeichen anders darstellt. [Siehe hier](#).

Insgesamt sieht unser neuer Block für den zweiten Teil der Überschrift jetzt so aus:

```
<span class="text-indigo-900 italic text-4xl sm:text-5xl lg:text-7xl opacity-80">  
  <q>&nbsp;Ich bring Dich in Form!&nbsp;</q>  
</span>
```

Wie man sieht, haben wir die Textgrößen der zweiten Überschriftszeile je nach Auflösung angepasst.

Nachtmodus = "Dark Mode" anlegen

Der sog. Dark Mode ist dafür da ein Blenden bei dunkler Hintergrundbeleuchtung zu vermeiden, z.B. wenn man nachts bei ausgeschaltetem Licht im Internet surft.

Aktivierung des Dark Mode

Wir wählen für unser Beispiel den sogenannten **"Manual Mode"** welchen wir wie folgt in der **tailwind.config.js** aktivieren:

```
module.exports = {  
  darkMode: 'class', // or 'media' or 'class'  
  // ...  
}
```

Bitte **achte darauf**, dass die Eigenschaft 'darkMode' **schon vorhanden** ist, und mit **false** **vorbelegt** ist!

Dunkelmodus im Dokument aktivieren

Hierfür müssen wir einfach im HTML-Tag die Klasse "dark" einfügen:

```
<html class="dark" lang="de">
```

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

[www.udemy.com/https://www.udemy.com/course/tailwindcss-deutsch](https://www.udemy.com/course/tailwindcss-deutsch)

Alternative Dark-Klassen im Dokument unterbringen

Noch sehen wir keinen Unterschied, aber wenn wir jetzt irgendwo im Dokument eine Klasse mit dem **Prefix "dark:"** versehen, dann wird diese aktiviert solange eben die Klasse "dark" im HTML-Tag ist.

Es ist dann sogar so, dass eine alternative Klasse überschrieben wird.

Beispiel:

Wir haben ja in der **tailwind.css** für die Überschriften H1 die Farbklasse "text-hauptfarbe" vergeben, welches die H1 in einem sehr dunklem blau einfärbt.

Wir können das jetzt **direkt im Dokument überschreiben**, indem wir z.B. bei der ersten H1 die Klasse "**dark:text-yellow-200**" verwenden.

Jetzt ist die Überschrift hellgelb.

Wir können das aber auch ganz zentral machen, indem wir in der tailwind.css direkt bei H1, H2 und H3 hinzufügen: **dark:text-yellow-200**

Jetzt wird die Klasse bei allen Überschriften angewendet.

Jetzt machen wir noch folgendes:

> In der **tailwind.css** bei "p" fügen wir ein **dark:text-gray-200**

> Dann fügen wir ebenfalls in der tailwind.css einen neuen Selektor für die Listenelemente ein:

```
li {  
  @apply dark:text-gray-200  
}
```

> Bei der Span: "Ich bring Dich in Form!" setzen wir direkt im Dokument ein:

dark:text-indigo-200

> Und ganz unten bei der Span "WhatsApp" fügen wir hinzu: **dark:text-gray-200**

> Jetzt können wir den Hintergrund abdunkeln indem wir in der in der tailwind.css beim container sagen: **dark:bg-gray-900**

> Um die Lesbarkeit des weissen Textes zu erhöhen setzen wir beim Hintergrund-SVG für die **.container-Klasse** die Deckkraft etwas herunter: **fill-opacity='0.38'**

Das ist ein guter Kompromiss für den hellen und den dunklen Modus.

> Zu guter Letzt machen wir beim Dunkelmodus den **Email-Button** etwas **kontrastreicher**, indem wir sagen:

dark:bg-gray-400 dark:hover:bg-gray-500

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**
www.udemy.com/course/tailwindcss-deutsch

Wie du siehst können wir auch **“dark:”** und **“hover:”** miteinander **kombinieren!**
Wie cool ist das denn!

Hier der offizielle Link zum Dark-Mode: <https://tailwindcss.com/docs/dark-mode>

Noch **2 kurze Anmerkungen** zum Dark-Mode:

1. Der Dark-Mode muss nicht unbedingt “dark” sein, sondern du könntest Deine Seite z.B. auch für **Rot / Grün** - Blindheit optimieren etc.
Schau mal hier: <https://www.augsburg.de/>
Hier kannst du z.B. zwischen einer “grünen”, “roten” und “schwarzen” Darstellung hin und her-wechseln:



2. Du kannst den **Dark-Mode** auch **nur für Teile** Deiner Website verwenden. Wenn du in unserem Beispiel die Klasse “dark” nur in der Section für den Seitenkopf anwenden würdest, dann würdest du das hier bekommen:



Jetzt wollen wir aber erreichen, dass du **dynamisch** zwischen “dark” und “nicht-dark” wechseln kannst, und das machen wir mit **Java-Script**.

Denkbar wäre hier die Verwendung von “reinem” = “Vanilla” - JavaScript, jquery, oder ein Framework wie z.B. Vue.js.

Wir wollen aber alpine.js verwenden. **Alpine.js ist genial!**

Es folgt eine Kurzeinführung.

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

Kurzeinführung Alpine.js + Darkmode-Button

Alpine.js findest du auf GitHub: <https://github.com/alpinejs/alpine>

Einbindung

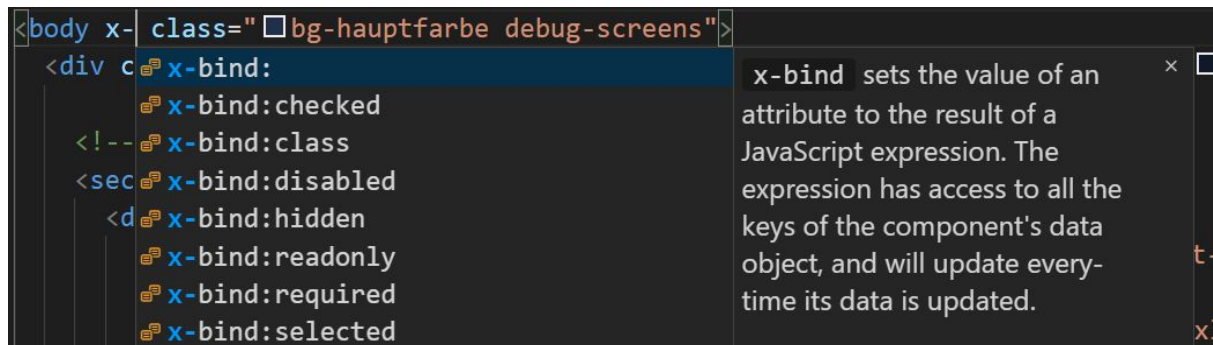
Der Einfachheit halber werden wir Alpine.js direkt über eine CDN in die **index.html** einbinden:

```
<script src="https://cdn.jsdelivr.net/gh/alpinejs/alpine@v2.x.x/dist/alpine.min.js"
defer></script>
```

Extension: Alpine.js Intellisense

Bitte installiere diese Extension in Deinem Visual Studio Code.

VSC macht dir dann Vorschläge und gibt Dir Hinweise zur Bedeutung verschiedener Direktiven:



Variablen anlegen mit x-data

Du kannst jetzt eine Variable an ein HTML-Element binden.

Dies erfolgt mit der Direktive **x-data="{ foo: 'bar' }"**

Wir legen jetzt eine Variable für das gesamte HTML-Dokument an, welche besagt, dass der Darkmode standardmäßig ausgeschaltet ist, indem wir im HTML-Tag sagen:

```
<html x-data="{ dark: false }" class="dark" lang="de">
```

Diese Variable ist jetzt im gesamten HTML-Dokument gültig! **Bitte beachte**, dass eine Variable immer nur innerhalb des Elementes gültig ist, in welchem sie deklariert wurde!

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile - ist untersagt.

Variablen anlegen mit x-bind

Jetzt können wir -ebenfalls im HTML-Dokument- auswerten, ob die Variable "dark" entweder true oder false ist, und dementsprechend die Klasse "dark" setzen oder nicht setzen:

```
<html x-data="{ dark: false }" x-bind:class=" dark ? 'dark' : " " lang="de">
```

Jetzt können wir bei x-data entweder true oder false einsetzen und dann wirkt sich das direkt auf die Klasse und somit auf den Dark-Mode aus.

Erklärung:

Bei `dark ? 'dark' : " "` handelt es sich um einen sogenannten ternären Operator. (gibt es in verschiedenen Programmiersprachen).

Diese Art von Ausdruck ist eine Kurzform eines If-Else-Blocks.

In unserem Fall steht dort: Gibt es eine Variable "dark" bzw. ist "dark == true" ?

Falls ja, dann schreibe die Klasse "dark" hin, falls nicht (hinter dem Doppelpunkt), dann schreibe **nichts** hin, das sind dann die beiden einfachen Anführungszeichen.

Man kann übrigens beim Alpine.js - Operator x-bind: auch eine **Kurzform** verwenden, welche ein **simpler Doppelpunkt (:)** ist, also funktioniert es auch, wenn man "x-bind" weglässt:

```
<html x-data="{ dark: false }" :class=" dark ? 'dark' : " " lang="de">
```

Jetzt könnte man übrigens zum Testen mal den `x-data="{ dark: false }"` vom HTML-Dokument in das `<body>` - Tag verlegen, dann sehen wir, dass das Ganze nicht mehr funktioniert, weil eben jetzt die Variable "dark" nur noch im Body gültig ist, und deswegen im HTML-Tag nicht mehr ausgewertet werden kann.

Einfacher Ein-Ausschalter mit @click

Wir bauen jetzt oben rechts im Dokument einen einfach Button ein:

```
<div id="darkButton" class="text-right mb-4">
  <button class="bg-gray-300 px-3 py-2 hover:bg-gray-400 rounded shadow">Dark
  Mode umschalten</button>
</div>
```

Dort bauen wir jetzt das @click - Event ein:

```
<button @click="dark = !dark"
```

[Nach oben](#)

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

[www.udemy.com/https://www.udemy.com/course/tailwindcss-deutsch](https://www.udemy.com/course/tailwindcss-deutsch)

Und jetzt lässt sich mit dem Button mit Dark-Modus ein- und ausschalten.

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Dynamische Darstellung des Darkmode-Buttons

Button dynamisch machen und Style anpassen

Zunächst einmal passen wir das Design des Buttons an das Design des unteren (Email) Buttons an, indem wir folgende Klassen hinzufügen:

`transition hover:shadow-none dark:bg-gray-400 dark:hover:bg-gray-500 dark:text-white`

Dein Text im Button ändern wir ab in "Nachtmodus".

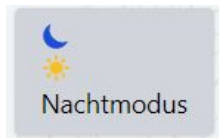
Dann gehen wir auf <https://heroicons.com/>

und suchen jeweils nach "Sun" und "Moon" und holen uns davon jeweils die "Solid Version" und setzen den Quelltext von beiden vor das Wort "Nachtmodus".

Dem Mond geben wir die Klassen: `h-5 mr-1 text-indigo-600`

Der Sonne geben wir die Klassen: `h-5 mr-1 text-yellow-400`

Jetzt haben wir das Problem dass die Icons umbrechen:



Das fixen wir indem wir dem Button die Klassen `flex items-center` geben.

(Zur Erinnerung "items-center" ist für die vertikale Ausrichtung!)

Jetzt haben wir das Problem, daß der Button nach links gerutscht ist, das können wir beheben mit der Klasse `ml-auto` auf dem Button.

Das "**text-right**" auf der übergeordneten Div können wir jetzt **löschen**!

Jetzt werden wir nur das jeweilig richtige Icon einblenden und das falsche ausblenden, dafür nehmen wir die alpinejs-Direktive **x-show**:

Beim Mond-Icon sagen wir: `x-show="!dark"`

Bei der Sonne sagen wir: `x-show="dark"`

[Nach oben](#)

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

Dynamischer Button-Text (x-text)

Jetzt müssen wir nur noch den Text des Buttons dynamisch machen, und dafür verwenden wir **x-text** von Alpine.js. Wir ersetzen jetzt das Wort "Nachtmodus" einfach mit:

```
<span x-text="dark ? 'Tagmodus': 'Nachtmodus' "></span>
```

Zusammenfassung Alpine.js

Wir haben jetzt schon folgende Befehle verwendet:

- **x-data** > zum Variablen definieren
- **x-bind**, bzw. Kurzform ":" > um Attribute dynamisch zu machen, z.B. die Klassen
- **@click** > um Variablen auf Klick zu ändern
- **x-show** > um Elemente dynamisch ein- bzw. auszublenden
- **x-text** > um Text dynamisch zu verändern

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Button-Styles auslagern

Jetzt haben wir ja 2 graue Buttons, der eine ist als Link realisiert, und der andere als Button. Die Buttons enthalten sehr viele Styles, die Frage ist jetzt ob wir das auslagern können, um Wiederholungen zu vermeiden? Versuchen wir es. Auslagern können wir:

`rounded shadow transition hover:shadow-none`

Wir löschen diese Eigenschaften bzw. Klassen also bei beiden Buttons raus und legen sie zentral in der **tailwind.css** an:

```
.btn {  
  @apply rounded shadow-sm transition hover:shadow-none px-3 py-2 dark:text-white  
}
```

Jetzt können wir tatsächlich einfach die Klasse `btn` zu beiden Buttons hinzufügen, und beide Buttons (egal ob als Link oder Button definiert) sehen wieder aus wie vorher.

Achtung: Sollte es nicht funktionieren, überprüfe bitte ob Dein `npm run watch` noch läuft, oder lasse `npm run build` laufen...

Wir können aber noch weitere Auslagerungsschritte wagen. Warum versuchen wir nicht die Eigenschaft, dass bzw. ob ein Button ein Icon enthält auszulagern?

Für ein Icon brauchen wir ja die Angaben `flex items-center`

> **flex** bedeutet, daß das Icon neben dem Text angeordnet wird (ähnlich wie "inline")

> **items-center** zentriert Icon und Schrift vertikal

Also machen wir in der **tailwind.css**

```
.btn-icon {  
  @apply flex items-center  
}
```

Damit können wir die Angaben `flex items-center` bei beiden Buttons mit `btn-icon` ersetzen und es funktioniert immer noch.

[Nach oben](#)

Tailwindcss - Ein super CSS-System daß du in den 2020'ern können musst!

Dieses Dokument gehört zum **Online-Tailwind-CSS-Kurs auf Udemy:**

www.udemy.com/course/tailwindcss-deutsch

Farben auslagern

Jetzt können wir ähnlich wie im Bootstrap-Framework auch noch die Farben auslagern, wir haben ja graue Buttons mit einem Rollovereffekt:

`bg-gray-300 hover:bg-gray-400 dark:bg-gray-400 dark:hover:bg-gray-500 dark:text-white`

Daraus wird dann in der **tailwind.css**

```
.btn-gray {  
  @apply bg-gray-300 hover:bg-gray-400 dark:bg-gray-400 dark:hover:bg-gray-500  
}
```

Auch jetzt funktioniert noch alles wenn die ganzen Ausdrücke mit **btn-gray** ersetzen.

Weitere Button Farbe erstellen

Warum machen wir nicht auch noch einen Button in "Indigo" = Lila, wenn wir schon dabei sind - wir haben die Farbe Indigo ja eh schon verwendet:

```
.btn-indigo {  
  @apply bg-indigo-300 hover:bg-indigo-400 dark:bg-indigo-400 dark:hover:bg-indigo-500  
}
```

CSS mit Purgecss verkleinern

Offizielle Seite:

<https://purgecss.com/>

Installation:

```
npm install @fullhuman/postcss-purgecss
```

Im Endeffekt bekommst du purgecss mit folgenden Zeilen in der **tailwind.config.js** zum Laufen:

```
purge: {  
  enabled: true,  
  content: ['./dist/**/*.html'],  
},
```

Wenn man das **"enabled: true"** raus nimmt, dann geht es nicht mehr. Man sollte das erst rein tun, wenn man das Projekt veröffentlichen will.

Hintergrund-Info:

<https://tailwindcss.com/docs/optimizing-for-production>

[Nach oben](#)

Copyright: Martin Krebs Eberth, masterclassudemy@gmail.com

Kopie und Weiterverbreitung per Original-PDF ist erlaubt, Abdruck oder Weiterverbreitung auch in Auszügen ohne Nennung des Tailwindcss-Kurses bei Udemy.com - siehe Kopfzeile- ist untersagt.

Projektzusammenfassung

1. Zunächst haben wir einen [Container erstellt und ein Plugin installiert](#), was uns die Breakpoints sichtbar macht.
2. Dann habe ich Euch ein wenig [HTML zur Verfügung gestellt](#) um loszulegen.
3. Dann haben wir mit der Hilfe von Sections, dem Grid-Systems und sonstigen kleinen Formatierungen den [Seiteninhalt grob angeordnet und strukturiert](#).
4. Dann haben wir die [Überschriften in der tailwind.css voreingestellt](#).
5. Dann haben wir uns die [Textformatierungs-Optionen](#) in Tailwindcss angesehen und schlussendlich unseren [eigen Font von Google](#) heruntergeladen und installiert.
6. Als nächstes haben wir eine [eigen Farbe](#) in Tailwindcss "dazu kompiliert", und anschliessend [die Listen](#), und dann grob den [Rest des Dokuments](#) formatiert.
7. Nachdem wir dann einen [Button mit Icon](#) erstellt haben, und ich Euch ein paar [kostenlose Icon-Ressourcen](#) vorgestellt habe, gab ich Euch eine kleine [Einführung in das responsive Design](#) mit Tailwindcss.
8. Dann haben wir konsequent und systematisch unsere Website erst [für mittlere](#), dann [für kleine](#) und dann [für Handys](#) optimiert. Ganz am Schluss kam dann die [Optimierung für die ganz großen](#) Auflösungen.
9. Als nächstes haben wir [Hintergrundmuster](#) verwendet und oben das [Zitat als solches gekennzeichnet und anders eingefärbt](#).
10. Dann habe ich euch den "[Darkmode](#)" vorgestellt, und anschließend [Alpine.js](#), mit dem man die Umschaltung zum Darkmode dynamisch machen kann.
11. Wir haben dann sogar den [Darkmode-Button noch dynamischer gemacht](#) (mit Icons und alternativen Text) um weitere Möglichkeiten mit Alpine.js einzuüben.
12. Dann habe ich Euch gezeigt, wie man ein ["kaskadierendes" Button-System](#) auslagern kann,
13. Und am Ende haben wir unser [CSS mit PurgeCss verkleinert](#), so dass wir unseren Webauftritt online stellen können.

[Nach oben](#)