

Grundlagen HTML und CSS

<https://d-rens.site/lecture-notes/web-dev/toc.html>

Materialien

Praesentation und “Zusammenfassung”:

<https://d-rems.site/lecture-notes/web-dev/toc.html>

Syntax

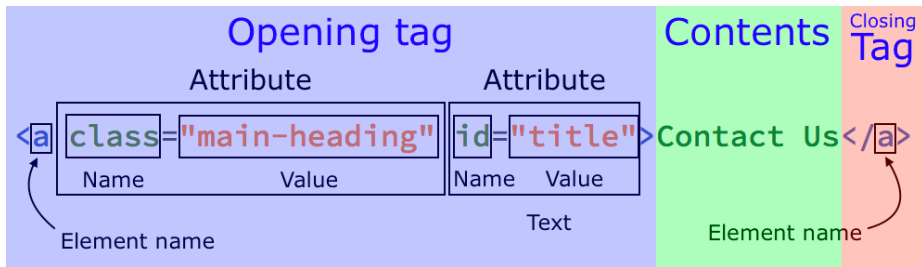


Figure 1: Syntax

Tags sind alle aus HTML, Attribute, also Klassen (class) und Id's müssen in CSS erstellt werden.

Tags

- Tags werden immer geöffnet und Geschlossen

```
<html>
```

```
</html>
```

Ausnahmen:

```
<br>, <img>, <link>, <meta> & <hr>
```

Wichtigste Tags

Fuer Struktur:

Tag	Nutzen	Wo im Dokument?
<!DOCTYPE html>	Zum indizieren das es html ist	Ganz am Anfang.
<html>	Um html schreiben zu koennen	Nach head bis unter die letzte Zeile
<head>	Fuer meta Daten.	Nach DOCTYPE, vor body
<title>	Um in dem Browser oben den Titel zu zeigen.	im head
<body>	Hier kommt der ganze inhalt rein.	Nach head
<link>	Zum verbinden zu z.B. Stylesheets	Im head

Beispiel

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <title>Beispiel</title>
    <meta charset="UTF-8">
    <link href="css/style.css" rel="stylesheet">
  </head>
  <body>
    <p>
      Lorem ipsum dolor sit amet, qui minim labore adipisicing minim sint c
      sint consectetur cupidatat.
    </p>
  </body>
</html>
```

Hier ist eine Textstruktur. Die Seite hat den Titel "Beispiel", die Sprache deutsch (*mit lang="de"*), und Inhalt der Seite ist Fuelltext.

Attribute (in CSS)

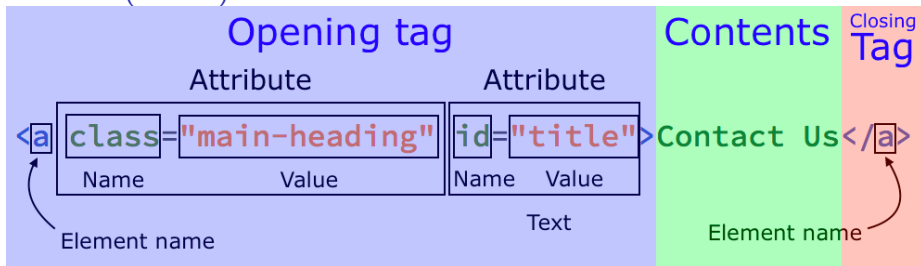


Figure 2: Syntax von vorher

class und id werden in dem Stylesheet definiert. Unterschied: Klassen koenne beliebig oft genutzt werden, Id nur ein mal.

```
.test-class {  
  color: #ffffff;  
  background-color: #555555;  
}
```

```
#test-id {  
  display: flex;  
}
```

Verwendung der Attribute

Attribute:

```
.test-class {  
    color: #ffffff;  
    background-color: #555555; }  
  
#test-id { color: navy; }
```

Verwendung

```
<body class="test-class">  
    <p>Fuelltext 1</p>  
    <p id="test-id">Fuelltext 2</p>  
</body>
```

Hier ist dann der Fuelltext 1 auf Grauem Hintergrund aus test-class mit Weiser Schriftfarbe. Fuelltext 2 hat die gleichen Attribute aus dem Body uebernommen, aber ueberschreibt die Schriftfarbe mit der eingenen Id.

mehr CSS

CSS	Bedeutung	Verwendung
color	bestimmt Textfarbe	Wenn man Textfarbe ändern will, z.B. für p, h1, usw.
background-color	bestimmt Hintergrundfarbe	Z.B. für divs, containers, usw.
font-family	bestimmt Schriftart	Z.B. für body, h1, usw.
font-size	bestimmt Schriftgröße	Z.B. für body, h1, usw.
font-weight	bestimmt Schriftstärke	Z.B. für strong, h1, usw.
font-style	bestimmt Schriftstil	Z.B. für em, i, usw.
display	bestimmt Anzeigeeigenschaften	Z.B. für div, span, usw.
width	bestimmt Breite eines Elements	Z.B. für img, div, usw.
height	bestimmt Höhe eines Elements	Z.B. für img, div, usw.
max-width	bestimmt maximale Breite eines Elements	Z.B. für img, div, usw.
max-height	bestimmt maximale Höhe eines Elements	Z.B. für img, div, usw.

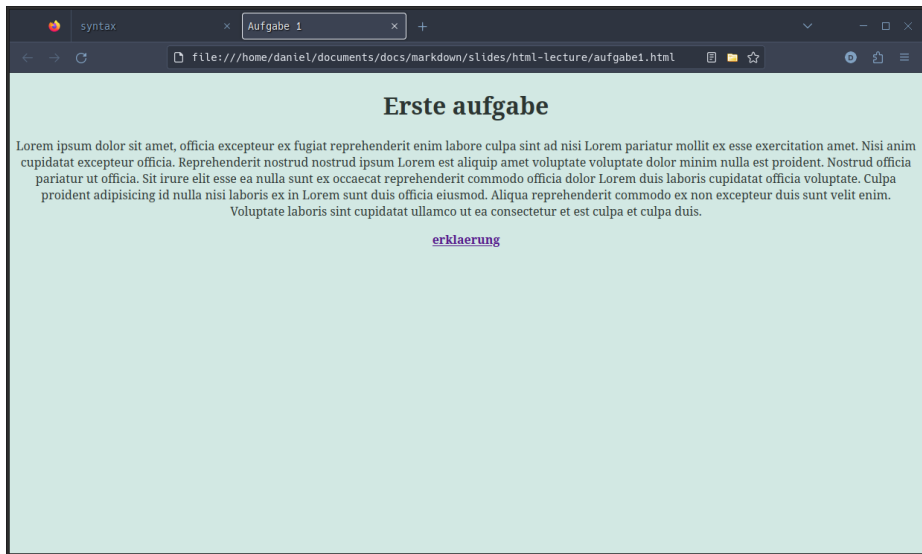
Aufgabe 1

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <title>Aufgabe 1</title>
    <meta charset="UTF-8">
    <link href="css/style.css" rel="stylesheet">
  </head>
  <body>
    <h1>Erste aufgabe</h1>
    <p>
      Fuelltext
    </p>

    <a href="https://www.w3schools.com/css/default.asp">erklaerung</a>

  </body>
</html>
```

Wie es aussehen soll:



CSS	Bedeutung	Verwendung
padding	bestimmt Innenabstand eines Elements	Z.B. für div, p, usw.
margin	bestimmt Außenabstand eines Elements	Z.B. für div, p, usw.
border	bestimmt Rahmenstil, -breite und -farbe	Z.B. für div, img, usw.
text-align	bestimmt Ausrichtung des Texts	Z.B. für p, h1, usw.
line-height	bestimmt Zeilenhöhe	Z.B. für p, h1, usw.
text-decoration	bestimmt Textdekoration	Z.B. für a, h1, usw.
text-transform	bestimmt Texttransformierung	Z.B. für p, h1, usw.
float	bestimmt Schweben des Elements	Z.B. für img, div, usw.
clear	bestimmt Behandlung von Schwebefloats	Z.B. für div, usw.
position	bestimmt Positionierung des Elements	Z.B. für div, img, usw.

Fuer Css grid

CSS Grid	Bedeutung	Verwendung
<code>display: grid</code>	erstellt ein Raster-Container-Element	Zum Erstellen eines Rasters-Containers
<code>grid-template-columns</code>	definiert die Anzahl und Breite der Raster-Spalten	Zum Definieren der Raster-Spalten
<code>grid-template-rows</code>	definiert die Anzahl und Höhe der Raster-Reihen	Zum Definieren der Raster-Reihen
<code>grid-template-areas</code>	definiert das Raster-Layout mithilfe von benannten Rasterflächen	Zum Definieren des Raster-Layouts
<code>grid-column-start</code>	definiert die Anfangsspalte eines Grid-Elements	Zur Positionierung eines Grid-Elements innerhalb des Rasters
<code>grid-column-end</code>	definiert die Endspalte eines Grid-Elements	Zur Positionierung eines Grid-Elements innerhalb des Rasters
<code>grid-row-start</code>	definiert die Anfangsreihe eines Grid-Elements	Zur Positionierung eines Grid-Elements innerhalb des Rasters

Css grid

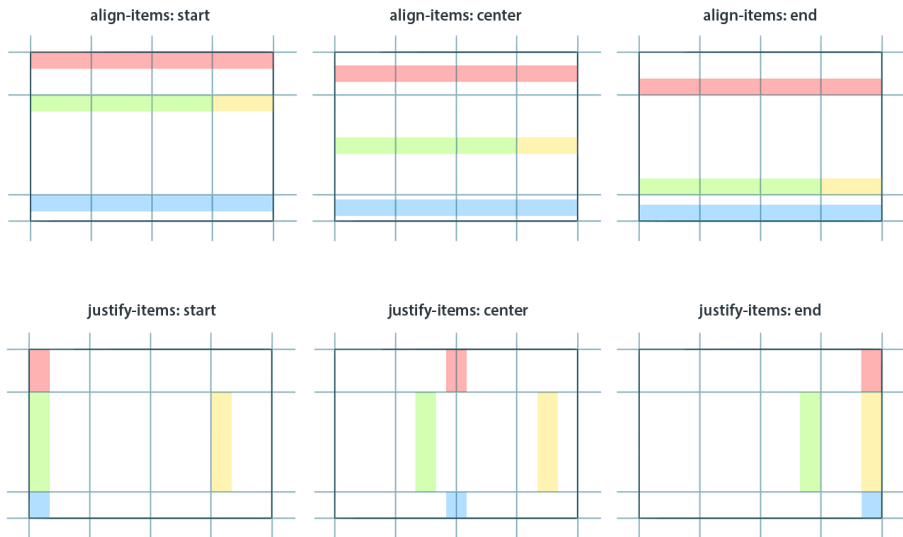


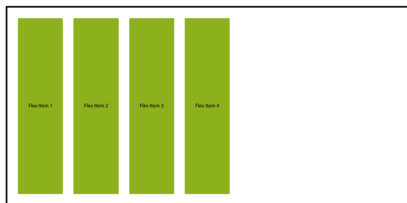
Figure 3: Css Grid

Fuer Flexbox

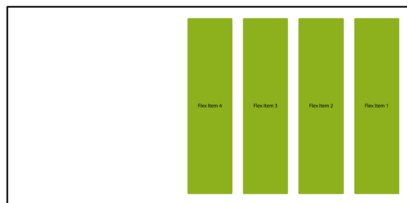
CSS Flexbox	Bedeutung	Verwendung
<code>display: flex</code>	erstellt ein Flex-Container-Element	Zum Erstellen eines Flex-Containers
<code>flex-direction</code>	definiert die Hauptachse des Flex-Containers	Zur Festlegung der Ausrichtung der Elemente im Flex-Container
<code>justify-content</code>	definiert die horizontale Ausrichtung der Elemente im Flex-Container	Zur Ausrichtung der Elemente entlang der Hauptachse
<code>align-items</code>	definiert die vertikale Ausrichtung der Elemente im Flex-Container	Zur Ausrichtung der Elemente entlang der Querachse
<code>align-self</code>	definiert die vertikale Ausrichtung eines einzelnen Elements	Zur Ausrichtung eines einzelnen Elements innerhalb des Flex-Containers

Flexbox Grid Darstellung

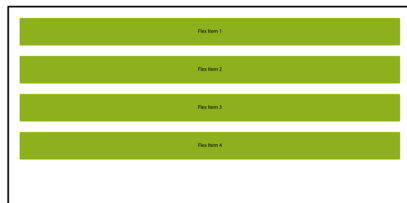
flex-direction: row



flex-direction: row-reverse



flex-direction: column

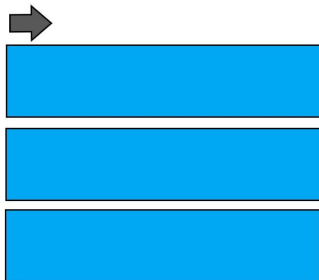


flex-direction: column-reverse

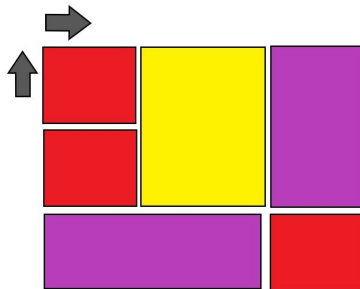


Figure 4: Flexbox Grid

Unterschied Flexbox/CSS Grid



Flexbox is one-dimensional



CSS Grid is two-dimensional

Mehr Ressourcen zum lernen

HTML

w3, html

CSS

w3, css