

CSS3 GRUNDLAGEN

...



Was ist CSS?

Cascading StyleSheet

Mit CSS ist es möglich HTML Elemente zu gestalten und so eine Webseite zu formen.

Arten CSS anzuwenden

Externes CSS

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

Internes CSS

```
<style type="text/css">  
  p {font-size: 20px;}  
</style>
```

Inline CSS

```
<div style="font-size: 20px;"></div>
```



Dies ist auch die Reihenfolge wie der Browser CSS aufruft
Inline CSS überschreibt alle anderen Eigenschaften.

.html

```
<h1 style='font-size: 20px;'>  
  Home  
</h1>
```

.html

```
<h1 style='font-size: 20px;'>  
  Kontakt  
</h1>
```

.html

```
<h1 style='font-size: 20px;'>  
  Impressum  
</h1>
```

Internes CSS & Inline

style.css

```
h1{  
  font-size:20px;  
}
```

.html

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

```
<h1>  
  Home  
</h1>
```

.html

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

```
<h1>  
  Kontakt  
</h1>
```

.html

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

```
<h1>  
  Impressum  
</h1>
```

Externes CSS

SELEKTOREN

4 verschiedene Arten HTML Elemente mit CSS anzusprechen

- Tag Selektoren
- ID Selektoren
- Class Selektoren
- Zusammengesetzte Selektoren

Mit Tag Selektoren kann man HTML Tags direkt ansprechen

Zum Beispiel:

```
h1 { font-size: 24px; }
```

Dadurch bekommen alle `<h1></h1>` Elemente eine Schriftgröße von 24px;

TAG Selektoren

Mit ID Selektoren kann man HTML Tags über deren ID ansprechen

Zum Beispiel:

```
#myID { min-height: 200px; }
```

Dadurch bekommen alle Elemente mit der ID Klasse eine Mindesthöhe von 200px.

`<div id="myID"></div>` oder `<p id="myID"></p>` usw ...

ID Selektoren

Mit **Class Selektoren** kann man HTML Tags über deren Klasse ansprechen

Zum Beispiel:

```
.myClass { min-height: 200px; }
```

Dadurch bekommen alle Elemente mit der Klasse eine Mindesthöhe von 200px.

`<div class="myClass"></div>` oder `<p class="myClass"></p>` usw ...

CLASS Selektoren

Unterschied zwischen ID & Class

Eine **ID** darf nur einmal pro Seite verwendet werden

Klassen dürfen so oft verwendet werden wie man möchte

Somit kann man mit Klassen mehreren Elementen die selben Eigenschaften zuweisen ohne dafür jedes eigens im CSS anzupassen

Wertigkeit der Selektoren

| Gruppe | Selektortyp | Beispiel | Wertigkeit (Spezifität) |
|--------|----------------------|-------------------------|-------------------------|
| A | Attribut style="..." | <p style="color: red;"> | 1000 |
| B | ID | #wrapper | 100 |
| C | Klasse | .info | 10 |
| D | einfacher Selektor | <div> | 1 |

Wertigkeit der Selektoren

| Gruppe | Selektortyp | Beispiel | Wertigkeit (Spezifität) |
|--------|----------------------|-------------------------|-------------------------|
| A | Attribut style="..." | <p style="color: red;"> | 1000 |
| B | ID | #wrapper | 100 |
| C | Klasse | .info | 10 |
| D | einfacher Selektor | <div> | 1 |

Beispiel-Selektor

| | A | B | C | D |
|---------------------------------------|---|---|---|---|
| <p style="color: red;">roter Text</p> | 1 | 0 | 0 | 0 |
| #wrapper #info .fehler {color:red;} | 0 | 2 | 1 | 0 |
| body #wrapper div#left p{color:red;} | 0 | 2 | 0 | 3 |
| #wrapper p.info {color:red;} | 0 | 1 | 1 | 1 |
| p.red {color:red;} | 0 | 0 | 1 | 1 |
| p {color:red;} | 0 | 0 | 0 | 1 |

Zusammengesetzte Selektoren

Hier gibt es zum einen Pseudoklassen wie z.B.:

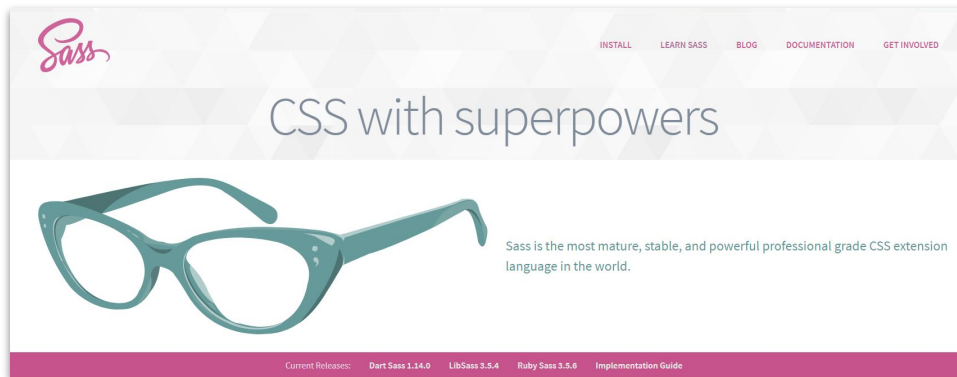
a:hover, p:first-letter, h1:after, usw ...

Weiters gibt es auch noch Verschachtelungen

nav ul li a

Stylesheet-Sprachen

Less & Sass sind Stylesheet-Sprachen mit dem Ziel, das Schreiben von CSS effizienter zu gestalten.



It's CSS, with just a little more.

Use with Node.js:

```
npm install -g less  
> lessc styles.less styles.css
```

Or the browser:

```
<link rel="stylesheet/less" type="text/css" href="styles.less" />  
<script src="//cdnjs.cloudflare.com/ajax/libs/less.js/3.7.1/less.min.js" ></script>
```

Stylesheet-Sprachen

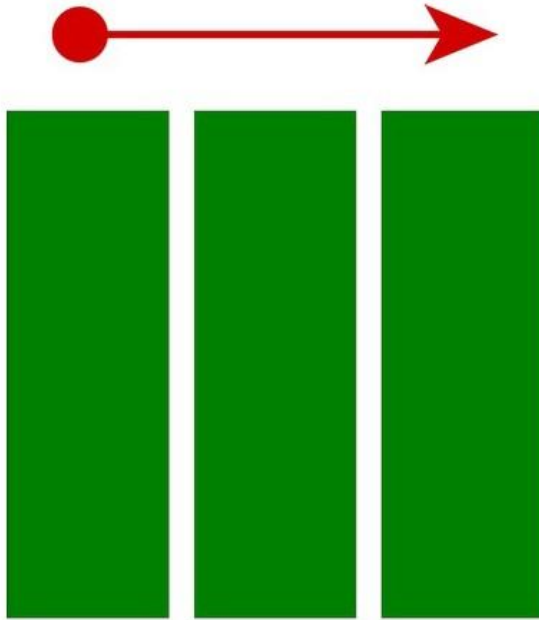
```
// LESS
```

```
#header {  
  h1 {  
    font-size: 26px;  
    font-weight: bold;  
  }  
  p { font-size: 12px;  
    a { text-decoration: none;  
      &:hover { border-width: 1px }  
    }  
  }  
}
```

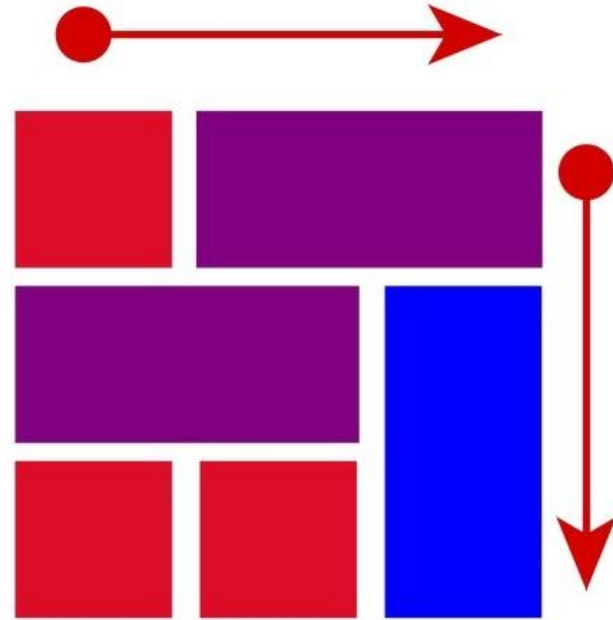
```
/* Compiled CSS */
```

```
#header h1 {  
  font-size: 26px;  
  font-weight: bold;  
}  
#header p {  
  font-size: 12px;  
}  
#header p a {  
  text-decoration: none;  
}  
#header p a:hover {  
  border-width: 1px;  
}
```

Layouthilfen im CSS



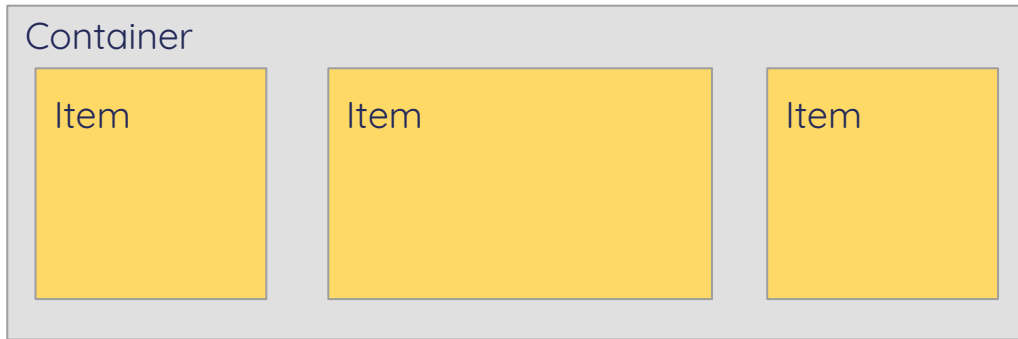
FLEXBOX
(one-dimensional)



CSS GRID
(two-dimensional)

FLEXBOX

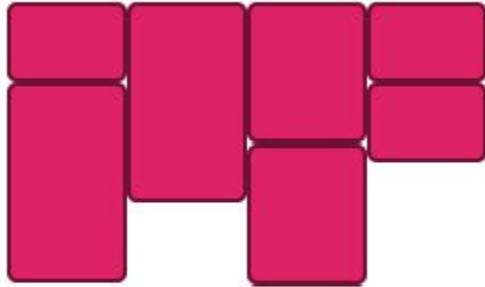
Flexbox hilft dabei Inhalte (Items) innerhalb eines Containers zu platzieren.
Es wird die Ausrichtung der Items innerhalb einer Zeile oder einer Spalte angesprochen.



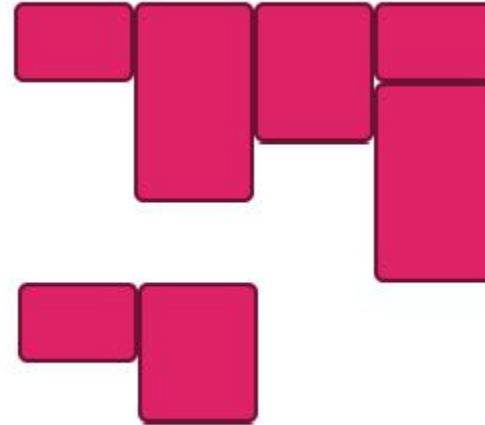
FLEXBOX

Bei mehr Inhalt oder verzweigten Layouts können Probleme durch die eindimensionale Aufteilung entstehen.

What you want

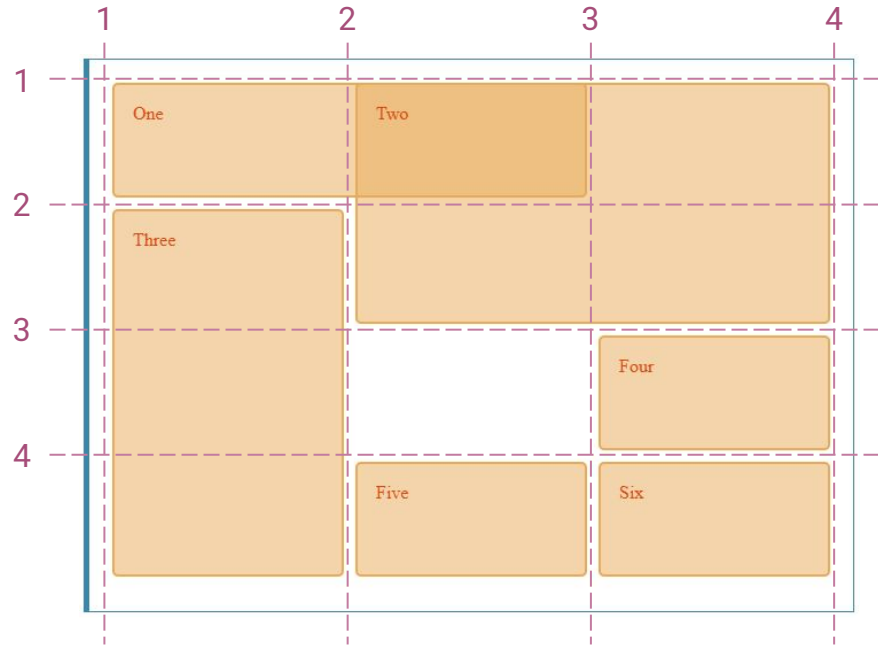


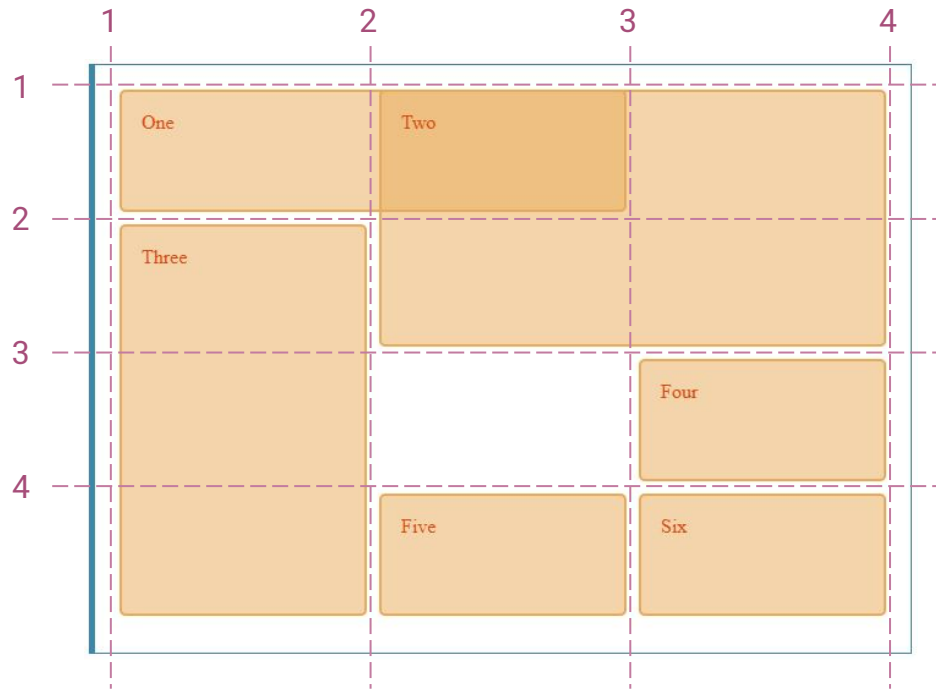
What you most likely get



CSS GRID

CSS Grid spricht im Gegensatz zu Flexbox die Ausrichtung innerhalb mehrerer Zeilen (Rows) und Spalten (Columns) an.





```
1  .wrapper {
2    display: grid;
3    grid-template-columns: repeat(3, 1fr);
4    grid-gap: 10px;
5    grid-auto-rows: minmax(100px, auto);
6  }
7  .one {
8    grid-column: 1 / 3;
9    grid-row: 1;
10 }
11 .two {
12   grid-column: 2 / 4;
13   grid-row: 1 / 3;
14 }
15 .three {
16   grid-column: 1;
17   grid-row: 2 / 5;
18 }
19 .four {
20   grid-column: 3;
21   grid-row: 3;
22 }
23 .five {
24   grid-column: 2;
25   grid-row: 4;
26 }
27 .six {
28   grid-column: 3;
29   grid-row: 4;
30 }
```

Mehr dazu auf YT

FLEXBOX:

YouTube: **Warum jeder #css #FLEXBOX nutzen sollte! Float ist tot! [Tutorial]**
by Unleashed Design

<https://www.youtube.com/watch?v=IT4c1x8hQ5I&t=633s>

CSS GRID:

YouTube: **CSS Grid Changes EVERYTHING - Amazing Presentation**
by Coding Tech

<https://www.youtube.com/watch?v=7kVeCqQCxIk&list=PLFjq8z-aGyQ6UpssdSYOtSreZuisB2iLU>



WEBBROWSER