

CSS Grid

Basics

Grid = "Raster", eine Layout Alternative zu Flexbox. Die Anzahl der Spalten sowie der Zeilen wird zu Beginn festgelegt, und auch deren Breiten. Es ist oft nicht notwendig, die Anzahl der Zeilen im Vorhinein festzulegen. Beispiel:

```
.grid-container {  
  display: grid;  
  grid-template-columns: 2fr 3fr 1fr;  
  grid-template-rows: 1fr 2fr 1fr;  
  grid-gap: 5px;  
}
```

Man braucht jetzt nichts weiter zu tun, als innerhalb des Containers HTML-Elemente anzulegen.

Benannte Bereiche

Bereich = "Area". Gewünschte Bereiche des Grids können benannt werden und im Container Element definiert werden, wir bleiben in obigem Beispiel:

```
.grid-container {  
  [...]  
  grid-template-areas:  
    "header    header side"  
    "main      main  side"  
    "impressum footer footer";  
}  
.header {  
  grid-area: header;  
}
```

Wichtig hier ist, daß jede **area** wirklich ein Rechteck ist, und nicht "um die Ecke" geht. Diesbezügliche Fehler sieht man dann in den Developer Tools (F-12).

Numerierte Bereiche

Es ist mit **grid-row-start**, **grid-row-end**, **grid-column-start** und **grid-column-end** der gleiche Effekt möglich, wie mit den benannten Areas. Möchte man z.B. ein Schachspiel simulieren, wird man eher diese Technik verwenden, und mittels Javascript die berechneten Werte setzen. Beispiel:

```
#bauer {  
  grid-row-start: 2;  
  grid-column-start: 5;  
}
```

Weiterführend

- [Dokumentation MDN](#)
- [Kevin Powell](#)
- [Kevin Grid Areas](#)