

Grid Layout -Spickzettel

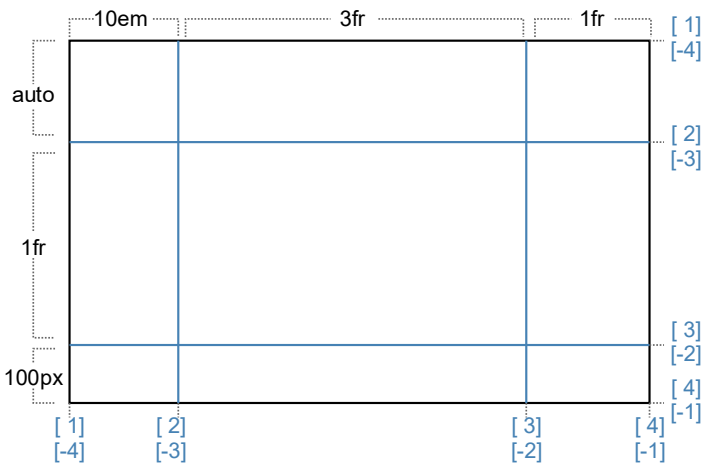
Das **CSS Grid Layout Modul** ermöglicht responsive Flexibilität mit nur wenigen Festlegungen, was spätere Änderungen sehr erleichtert.

display

```
.container {
  display: grid; |inline-grid |subgrid
}
```

Explizite Raster festlegen

```
body {
  grid-template-columns: 10em 3fr 1fr;
  grid-template-rows: auto 1fr 100px;
}
```

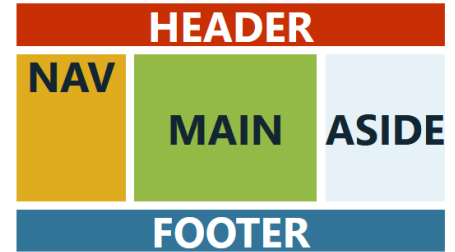


```
.container {
  display: grid;
  grid-template-columns:
    [nav-start] 10em [nav-end main-start] 3fr [main-end aside-start] 1fr [aside-end];
  grid-template-rows:
    [row1-start] auto [row1-end] 1fr [third-line] 100px [last-line];
  gap: 0;
}
```

```
body{
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  grid-template-areas: "head head"
                      "nav nav"
                      "main main"
                      "foot foot";
}
@media (min-width: 30em) {
  body{
    grid-template-columns: repeat(3, 1fr);
    grid-template-areas: "head head head"
                        "nav news aside"
                        "main main main"
                        "foot foot foot";
  }
}
```

Implizite Raster festlegen

```
.container {
  display: grid;
  grid-auto-flow: column;
  grid-auto-columns: minmax(150px, 1fr);
}
```



Grid-Container werden durch `display: grid` bzw. `inline-grid` und `subgrid` definiert und können eine Breite, bzw. Mindest- oder Maximalbreite haben.

Beachte: Dieses Raster wirkt nur auf direkte Kind-Elemente des Grid-Containers.

Die Eigenschaft `grid-template-columns` legt Anzahl, Größe und optional auch die Benennung von Spalten - `grid-template-rows` von Zeilen fest.

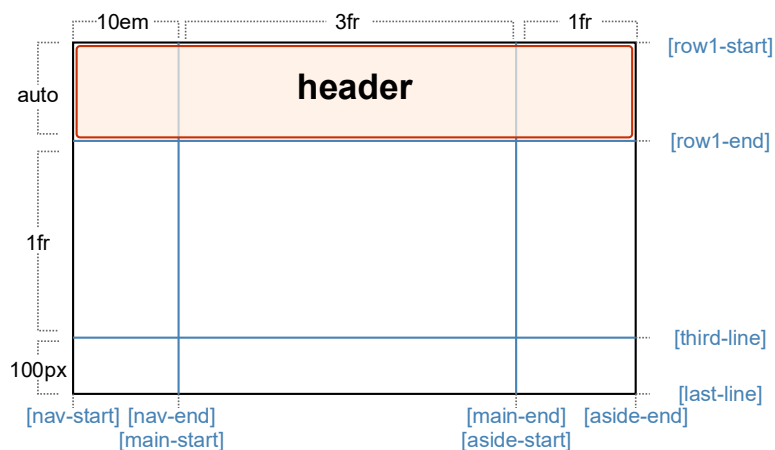
Die Werte können normale Längenangaben, Prozentwerte – oder mit `fr` Bruchteile der verfügbaren Viewport-Maße sein.

Spalten und Zeilen werden durch Rasterlinien getrennt. Sie sind fortlaufend nummeriert, negative Werte werden vom Ende her gezählt.

Rasterlinien können auch benannt werden. Innerhalb der eckigen Klammern sind auch mehrere Namen möglich.

`grid-template-areas` legen in einer Liste das Schema der Seite fest.

Mit *media queries* können so unterschiedliche Layouts für größere Viewports festgelegt werden.



`grid-auto-columns` legt Größe von Rasterzellen fest. Das umgebende Grid wird implizit vom Browser erzeugt



Rasterzellen anordnen

```

nav {
  grid-column-start: 1;
  grid-column-end: 2;
  grid-row-start: [row1-start];
  grid-row-end: 3;
}
header {
  grid-column-start: span 3;
}

nav {
  grid-column: 1 / 2;
  grid-row: 1 / -1;
}
header {
  grid-column: span 3;
}

section {
  grid-area: 2 / 2 / auto / span 3;
}
header {
  grid-area: head;
}

```

grid-column-start

grid-column-end

legen Beginn und Ende der Rasterzelle fest.

- Ganzzahl. Negative Werte zählen vom Ende des expliziten Grids her.
- benannter Name der Rasterlinie
- Schlüsselwort span mit Anzahl der Zellen, die überspannt werden sollen.

grid-column: Anfangs- und Endwerte für Spalte

grid-row: Anfangs- und Endwerte für Zeile

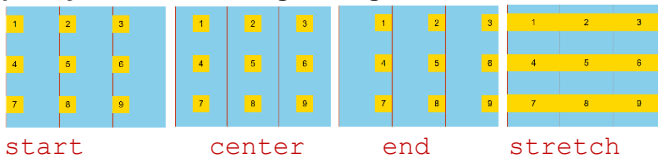
grid-area:

- 4 Werte für Anfangs- und Endlinien
- benannter Name aus **grid-template-areas**

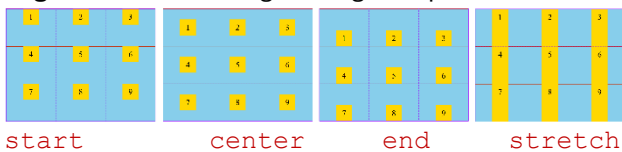
Rasterzellen ausrichten

Im Grid-Container

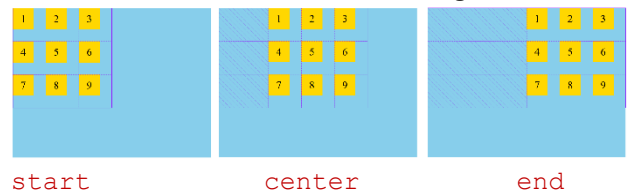
justify-items Anordnung entlang der Zeilen-Achse



align-items Anordnung entlang der Spalten-Achse



justify-content positioniert Raster entlang der Grid-Achse, wenn diese kleiner als der verfügbare Platz ist



align-content positioniert Raster quer zur Grid-Achse, wenn diese kleiner als der verfügbare Platz ist

Beachte: Normalerweise nehmen Rasterzellen (*items*) den gesamten verfügbaren Platz ein (**auto** | **stretch**).

Im Raster-Element

justify-self Anordnung eines Items entlang der Ausrichtung



align-self Anordnung eines Items quer zur Ausrichtung





SVG (Scalable Vector Graphics) ist ein vom W3C empfohlenes Grafikformat zur Beschreibung zweidimensionaler Vektorgrafiken. SVG kann als Standalone-Dokument verwendet, aber auch in HTML eingebunden werden. Es kann mit CSS formatiert und animiert werden.

Grundgerüst mit Namensraumangabe

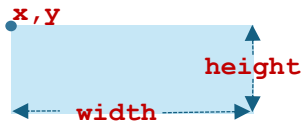
```
<svg xmlns="http://www.w3.org/2000/svg"
     xmlns:xlink="http://www.w3.org/1999/xlink">
  viewBox="x y width height">
  <title>...</title>
  <desc> ... </desc>
  <!-- Inhalt -->
</svg>
```

- svg** Wurzelement mit Namensraumangabe
- viewBox** neues Benutzer-Koordinatensystem aus dimensionslosen Größen
- title** Name, wird im Tab angezeigt
- desc** Metaangabe(n), Beschreibung

Dokumenttyp-Deklaration und Doctype sind in SVG2 nicht mehr nötig!

Grundformen

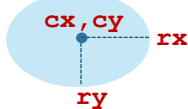
```
<rect x="0" y="0" width="20" height="10" />
```



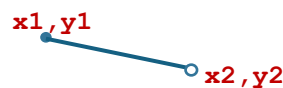
```
<circle cx="0" cy="0" r="0" />
```



```
<ellipse cx="0" cy="0" rx="0" ry="0" />
```



```
<line x1="0" y1="0" x2="0" y2="0" />
```



```
<polyline points="x1,y1 x2,y2 x3,y3..." />
```

```
<polygon points="x1,y1 x2,y2 x3,y3..." />
```



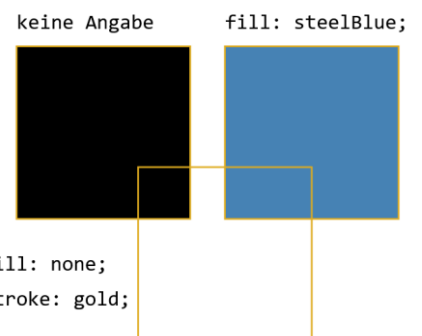
```
<path d="m x1,y1 ..." />
```



[→ Pfad-Kommandos \(Übersicht\)](#)

Füllungen und Randlinien

<code>fill</code>	Füllfarbe	Farbangabe (Standardwert: black)
<code>fill-opacity</code>	Deckkraft	0 - 1
<code>fill-rule</code>	Füllmethode	<code>evenodd</code> nonzero
<code>stroke</code>	Randfarbe	Farbangabe none
<code>stroke-width</code>	Randstärke	Längenangabe (Standardwert: 1)
<code>stroke-linecap</code>	End-Form	butt <code>round</code> <code>square</code>
<code>stroke-linejoin</code>	Spitzen-Form	miter <code>round</code> <code>bevel</code>
<code>stroke-dasharray</code>	Strichelung	none Liste aus Längenangaben



Gruppierungen

```
<defs>
  <symbol id="wheel">
    <circle cx="0" cy="0" r="8" />
  </symbol>
</defs>

<use href="#wheel" x="100" y="100" />
```

- defs** **symbol** Definitionsabschnitt
Vorlage, die dann referenziert werden kann
- use** **href** *hyper reference* Verwendung der Vorlage
Referenz auf ein Bild, URL des Bildes
- g** *group* Gruppierung (≙ div)



Pfad-Kommandos

Kommando Name	Beschreibung
M, m MoveTo	setzt den aktuellen Punkt fest, von dem der Pfad starten soll.
L LineTo	zeichnet eine Linie vom aktuellen zum angegebenen Punkt.
H Horizontal LineTo	zeichnet vom aktuellen Punkt aus eine horizontale Linie .
V Vertical LineTo	zeichnet vom aktuellen Punkt aus eine vertikale Linie .
Z ClosePath	erstellt eine geschlossene Form .
A Elliptical Arc	zeichnet vom aktuellen Punkt aus einen elliptischen Bogen .
C CurveTo	zeichnet vom aktuellen Punkt aus eine Bézier-Kurve .
S Smooth CurveTo	ähnlich dem Kommando C, jedoch wird hier als erster Kontrollpunkt der letzte aus der direkt zuvor festgelegten Bézier-Kurve verwendet.
Q Quadratic Bézier CurveTo	zeichnet eine quadratische Bézier-Kurve .
T Smooth Quadratic Bézier CurveTo	ähnlich dem Kommando Q, jedoch wird hier als erster Kontrollpunkt der letzte aus der direkt zuvor festgelegten quadratischen Bézier-Kurve verwendet.

Großschreibung bewirkt absolute Kommandos vom Ursprung (0|0) aus
Kleinschreibung relative Kommandos vom aktuellen Punkt des Stifts.

Text

```
<text x="100" y="100" >
  Klartext
  <tspan dx="10" dy="10" >
    Klartext
  </tspan >
</text>
```

text **Text**

selektier- und suchbar

tspan

bedeutungsleeres Element, das abweichend formatiert und verschoben werden kann.

Wichtige Attribute

x	x-Position	(Standardwert: 0)
y	y-Position	(0) wäre außerhalb der viewBox!
dx, dy	Verschiebung	Längenangabe (0)
text-anchor	Ausrichtung	start middle end

... und alle „normalen“ CSS-Eigenschaften wie font, ...

