



Marp

# CSS Grids

# 0. Einführung in CSS Grid

- **Was ist CSS Grid?**

- Ein CSS-Layout-Modul, das ein zweidimensionales Raster-basierendes Layoutsystem bietet.
- Ermöglicht das einfache Erstellen komplexer Layouts durch das Definieren von Reihen und Spalten.

# 0. Einführung in CSS Grid

- **Vorteile von CSS Grid:**

- Flexibilität: Ermöglicht die einfache Erstellung von sowohl Zeilen- als auch Spaltenlayouts.
- Einfachheit: Reduziert den Bedarf an verschachtelten Containern.
- Kontrolle: Bietet präzise Kontrolle über die Positionierung und Ausrichtung von Elementen.

# 0. Einführung in CSS Grid

- **Wann sollte CSS Grid verwendet werden?**
  - Ideal für komplexe Layouts.
  - Gut geeignet für Responsive Designs.
  - Bietet mehr Kontrolle als Flexbox für zweidimensionale Layouts.

# 1. Grundlegende Konzepte

- **Grid Container und Grid Items:**

- Der `Grid Container` ist das Element, das das Grid Layout enthält.
- `Grid Items` sind die direkten Kinder des Grid Containers.

- **Erstellen eines Grid Containers:**

- `display: grid;` oder `display: inline-grid;`
- Beispiel:

```
.grid-container {  
  display: grid;  
}
```

## 2. Grid Gaps

- **Definition von Abständen zwischen den Grid Items:**
  - `grid-gap`: Kurzform für `row-gap` und `column-gap` (Abstand zwischen Zeilen/Spalten)
- **Beispiel:**

```
.grid-container {  
  display: grid;  
  grid-gap: 10px;  
}
```

Item 1

Item 2

Item 3

# 3. Grid Template Rows und Columns

- Definition von Reihen und Spalten
- Beispiele:

```
.grid-container {  
  display: grid;  
  grid-template-rows: 50px 100px;  
}
```



## 4. Grid Colum/Row

- `grid-column`: Bestimmt die Spalte, die ein Item belegt.
- `grid-row`: Bestimmt die Zeile, die ein Item belegt.
- Beispiel:

```
.item {  
  grid-row: 1;  
}
```

```
.item-span {  
  grid-column: span 2;  
}
```

# Übung 2

Übungsfile: `02.html`

- ToDo's:
  - die erste und letzte Reihe müssen eine height von 50px haben
  - die mittlere Reihe muss eine height von 200px haben
  - die erste und letzte Reihe müssen sich über 2 Spalten spannen

# 4. Grid Areas

- Benennung der Bereiche für einfachere Platzierung von Items.
- Beispiel:

```
.grid-container {  
  grid-template-areas:  
    "header header"  
    "sidebar main"  
    "footer footer";  
}  
.header {grid-area: header;}  
.sidebar {grid-area: sidebar;}  
.main {grid-area: main;}  
.footer {grid-area: footer;}
```

# Übung 3

Übungsfile: `03.html`

- ToDo's:
  - die Struktur mit Grid Areas nachstellen

# 6. Responsives Design mit CSS Grid

- **Die Einheit fr im CSS Grid:**

- fr steht für "Fraction" (Bruchteil) und wird verwendet, um den verfügbaren Platz im Grid zu verteilen.
- Zum Beispiel: `grid-template-columns: 1fr 2fr;` bedeutet, dass die erste Spalte 1/3 und die zweite Spalte 2/3 des verfügbaren Platzes einnimmt.
- Insgesamt können die fr-Einheiten so eingestellt werden, dass sie zusammen 1, 2, 12 usw. ergeben, je nach gewünschter Layout-Struktur.

# Übung 4

Übungsfile: `04.html`

- ToDo's:
  - make it responsive!