

GRID

Grid Layout

- The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.

Grid Elements

- A grid layout consists of a parent element, with one or more child elements.
- ```
<div class="grid-container">
 <div class="grid-item">1</div>
 <div class="grid-item">2</div>
 <div class="grid-item">3</div>
 <div class="grid-item">4</div>
 <div class="grid-item">5</div>
 <div class="grid-item">6</div>
 <div class="grid-item">7</div>
 <div class="grid-item">8</div>
 <div class="grid-item">9</div>
</div>
```

# Display Property

- An HTML element becomes a grid container when its display property is set to grid or inline-grid.

- .grid-container

```
{
 display: grid;
}
```

- .grid-container

```
{
 display: inline-grid;
}
```

# Column gap

- The column-gap property sets the gap between the columns:

- ```
.grid-container {  
  display: grid;  
  column-gap: 50px;  
}
```

- The row-gap property sets the gap between the rows:

- ```
.grid-container {
 display: grid;
 row-gap: 50px;
}
```

# Gap property

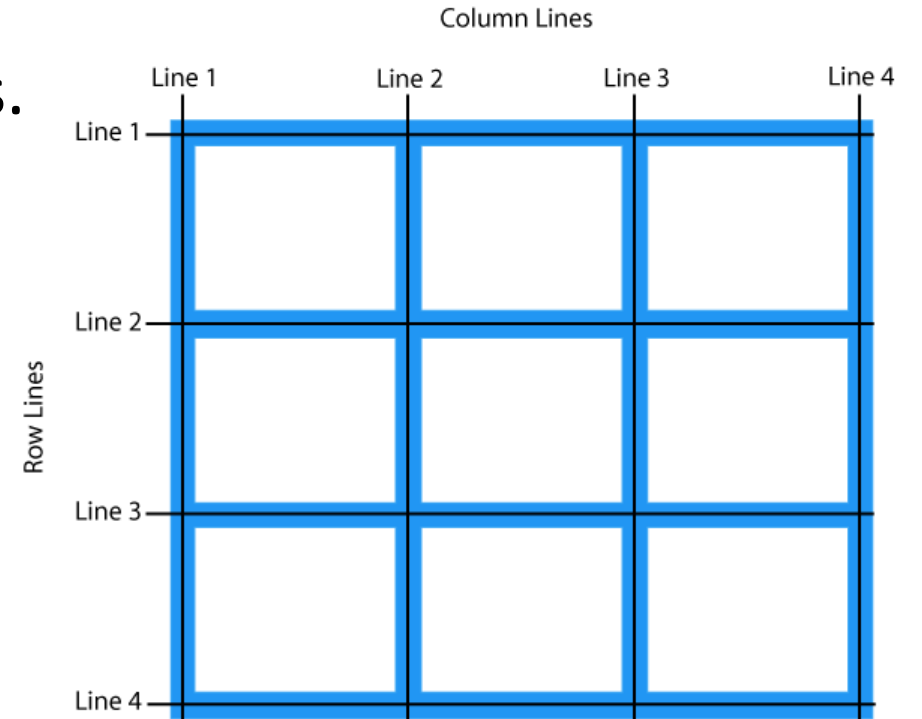
- The gap property is a shorthand property for the row-gap and the column-gap properties:
- ```
.grid-container {  
  display: grid;  
  gap: 50px 100px;  
}
```

Or

- ```
.grid-container {
 display: grid;
 gap: 50px;
}
```

# Grid Lines

- The lines between columns are called column lines.
- The lines between rows are called row lines.



- Place a grid item at column line 1, and let it end on column line 3:

```
.item1 {
 grid-column-start: 1;
 grid-column-end: 3;
}
```

- Place a grid item at row line 1, and let it end on row line 3:

```
.item1 {
 grid-row-start: 1;
 grid-row-end: 3;
}
```



- <style>
- .grid-container {
- display: grid;
- grid-template-columns: auto auto auto;
- gap: 10px;
- background-color: #2196F3;
- padding: 10px;
- }
- .grid-container > div {
- background-color: rgba(255, 255, 255, 0.8);
- text-align: center;
- padding: 20px 0;
- font-size: 30px;
- }
- .item1 {
- grid-row-start: 1;
- grid-row-end: 3;
- }
- </style>
- </head>

# CSS Grid Container

- To make an HTML element behave as a grid container, you have to set the display property to grid or inline-grid.
- Grid containers consist of grid items, placed inside columns and rows.
- .grid-container  
{  
display: grid;  
grid-template-columns: auto auto auto auto;  
}

- The `grid-template-columns` property can also be used to specify the size (width) of the columns.
- Set a size for the 4 columns:

```
.grid-container {
 display: grid;
 grid-template-columns: 80px 200px auto 40px;
}
```

# The grid-template-rows Property

- The grid-template-rows property defines the height of each row.
- The value is a space-separated-list, where each value defines the height of the respective row:

```
.grid-container {
 display: grid;
 grid-template-rows: 80px 200px;
}
```

# The justify-content Property

- The justify-content property is used to align the whole grid inside the container.

```
.grid-container {
 display: grid;
 justify-content: space-evenly;
}
```

```
.grid-container {
 display: grid;
 justify-content: space-around;
}
```

```
.grid-container {
 display: grid;
 justify-content: space-between;
}
```

```
.grid-container {
 display: grid;
 justify-content: center;
}
```

```
.grid-container {
 display: grid;
 justify-content: start;
}
```

```
.grid-container {
 display: grid;
 justify-content: end;
}
```

# The align-content Property

- The align-content property is used to vertically align the whole grid inside the container.
- ```
.grid-container {  
  display: grid;  
  height: 400px;  
  align-content: center;  
}
```
- ```
.grid-container {
 display: grid;
 height: 400px;
 align-content: space-evenly;
}
```



- `.grid-container {  
 display: grid;  
 height: 400px;  
 align-content: space-around;  
}`
- `.grid-container {  
 display: grid;  
 height: 400px;  
 align-content: space-between;  
}`

- `.grid-container {  
 display: grid;  
 height: 400px;  
 align-content: start;  
}`
- `.grid-container {  
 display: grid;  
 height: 400px;  
 align-content: end;  
}`