

Layouts

CSS Grid

ART326

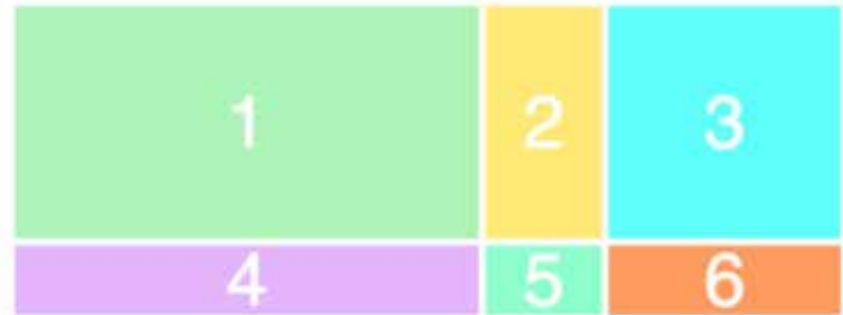
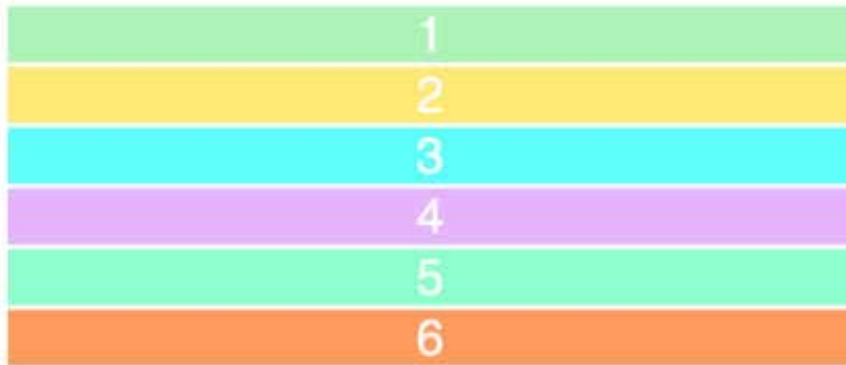
CSS Grid

Components:

The Grid Container (display: grid; on parent)

Placing and aligning items (children) **on the grid**

Columns + Rows

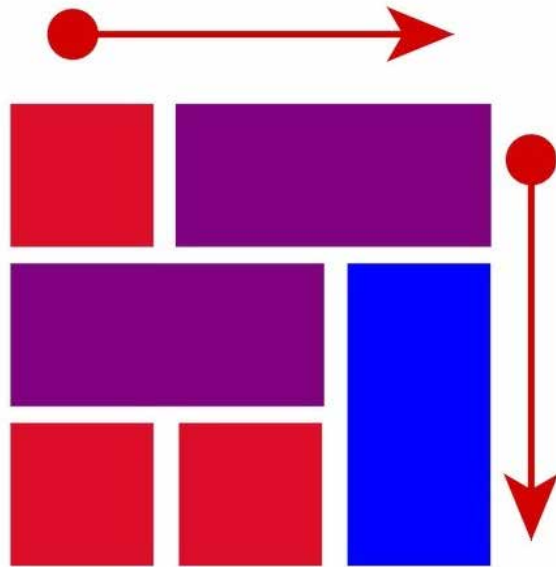


```
.wrapper {  
  display: grid;  
  grid-template-columns: 200px 50px 100px;  
  grid-template-rows: 100px 30px;  
}
```

CSS Grid vs. Flexbox

CSS Grid

two-dimensional
row AND column

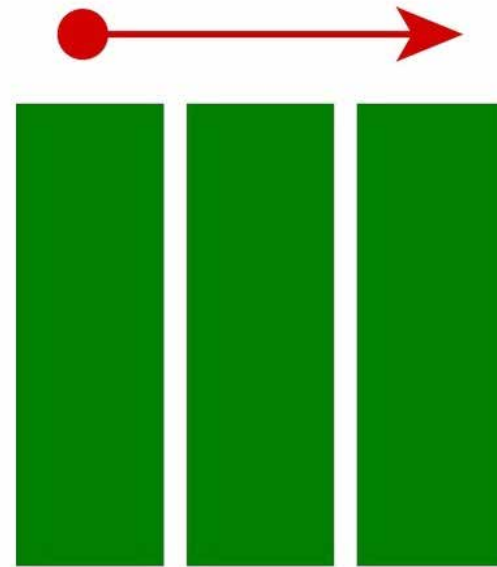


CSS Grids

Two Dimensions

Flexbox

one-dimensional
row OR column



Flexbox

One Dimensions

Grid Line

The horizontal and vertical lines that make up the grid.

Highlighted is column line 2.

	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

Grid Line

Column Lines

Line 1 Line 2 Line 3 Line 4

Row Lines

Line 1			
Line 2			
Line 3			
Line 4			

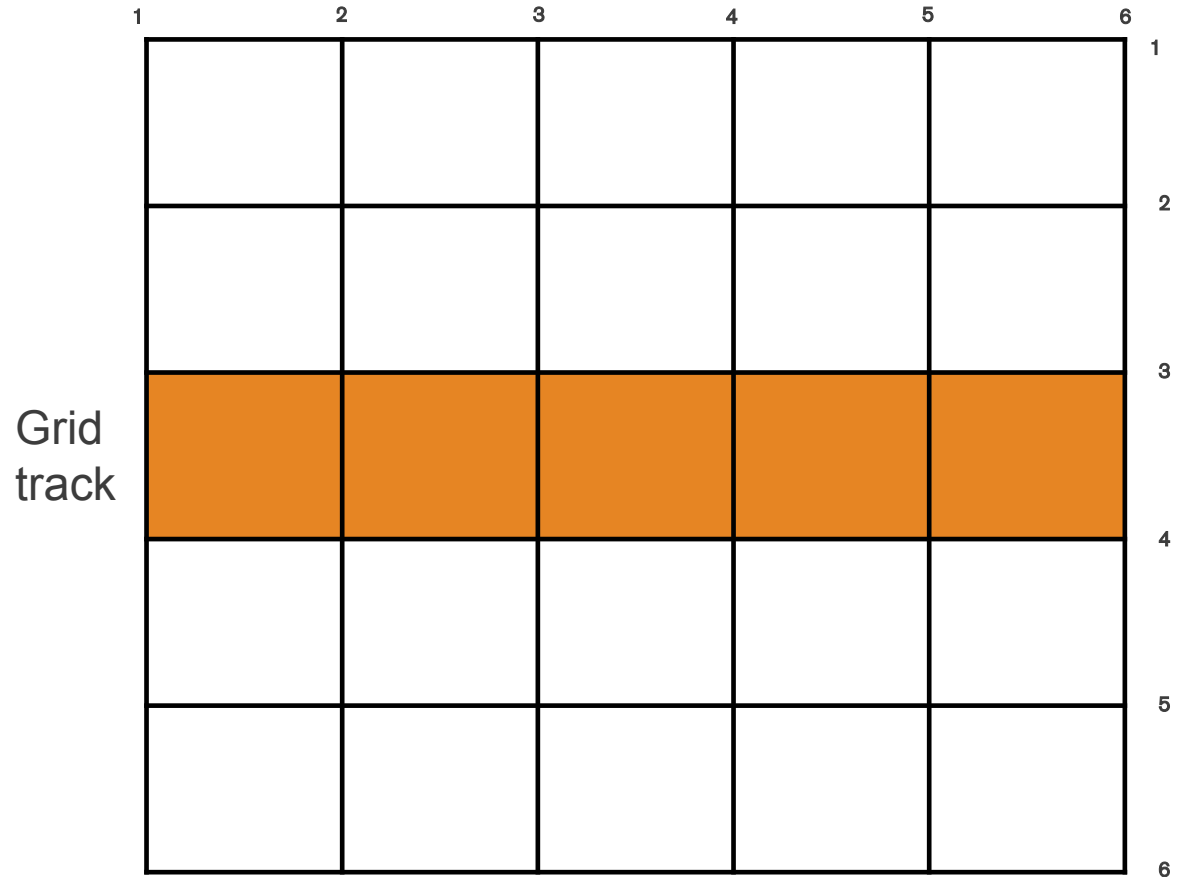
Row			
Row			
Row			

Column	Column	Column

Grid Track

The space between two grid lines, horizontal or vertical.

Highlighted is the Grid Track between row lines 3 and 4.



Grid Cell

The space between four grid lines.

Highlighted is the Grid Cell between row lines 3 and 4, and column lines 2 and 3.

1	2	3	4	5	6
					1
					2
					3
					4
					5
					6

Grid Area

The area between four grid lines.

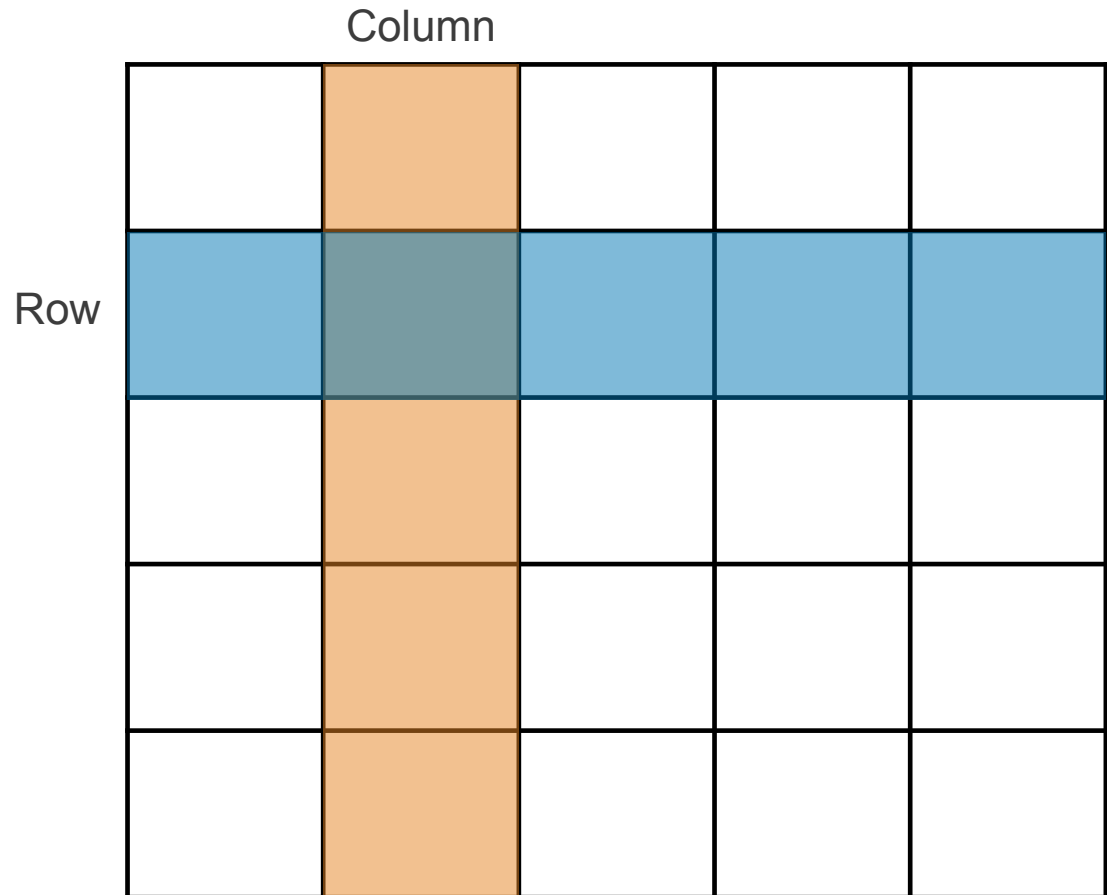
Highlighted is the Grid Area between row lines 2 and 4, column lines 3 and 6.

Grid Area

Grid

Any element with its display property set to grid.

Grids have Rows and Columns.

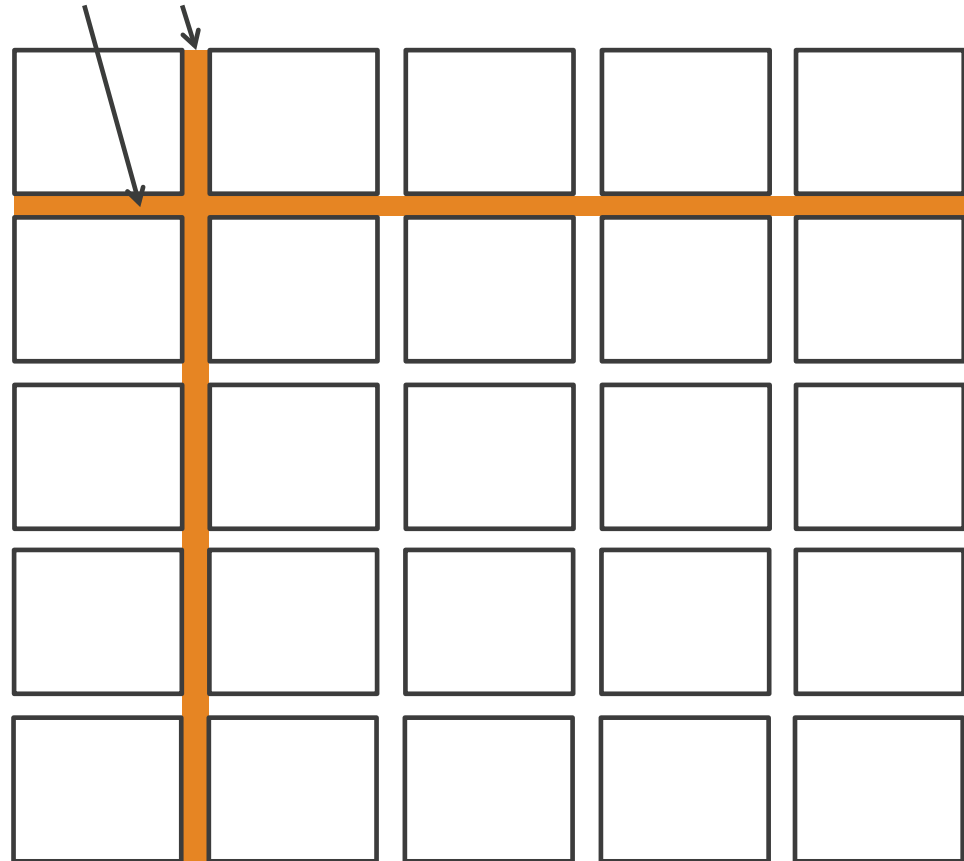


Gutters

Grid Gutters

The defined space between rows and columns.

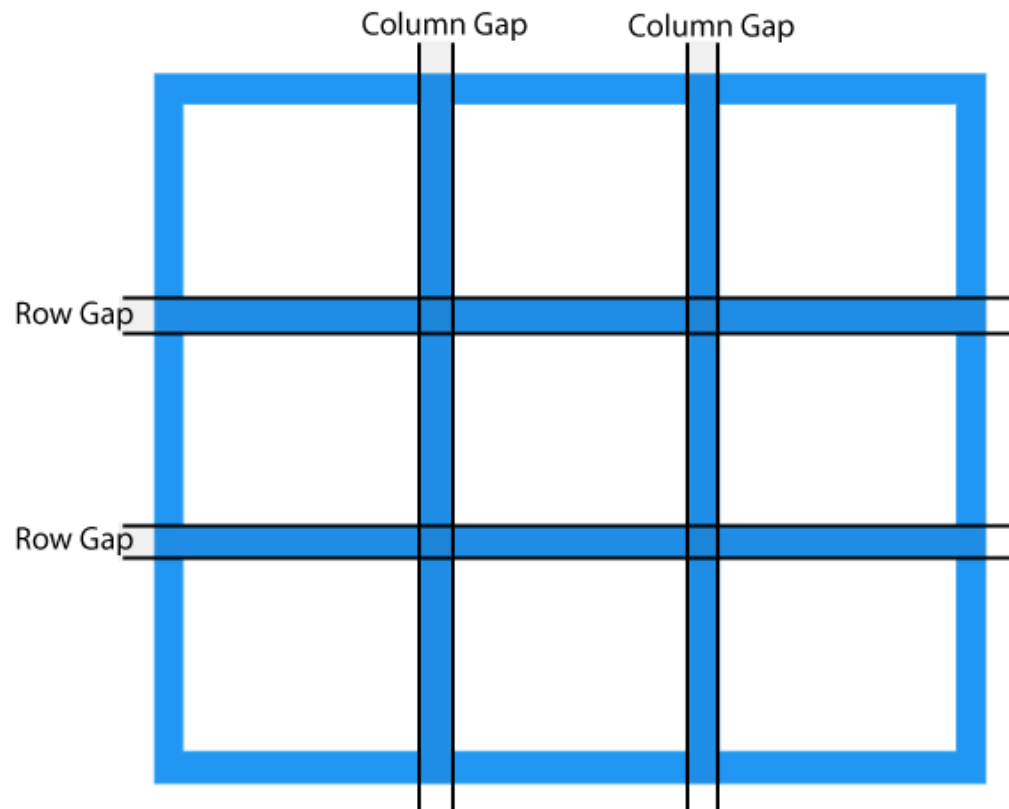
Grid Gutters



Gutters

```
grid-column-gap: 10px;  
grid-row-gap: 10px;
```

Column Gaps



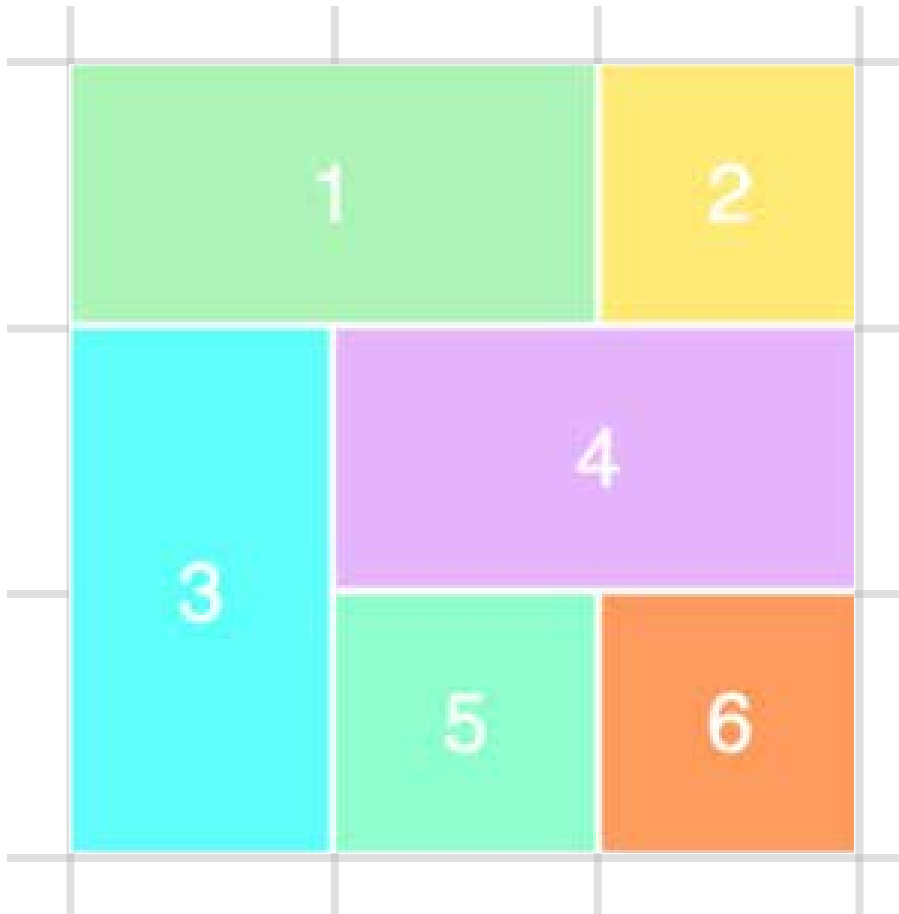
`grid-column-gap`

`grid-row-gap`

`grid-gap`

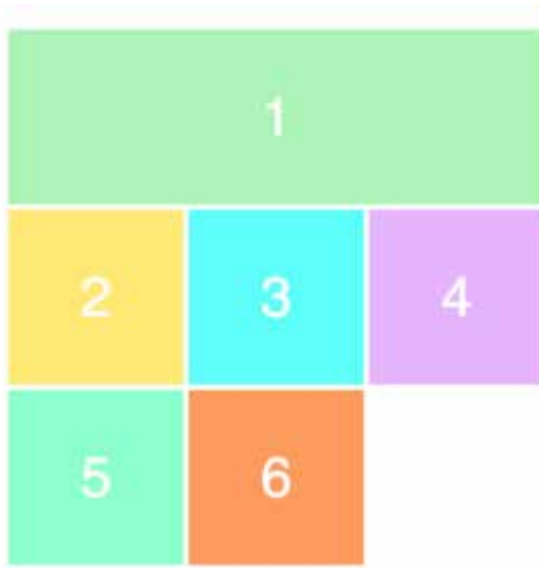
Positioning Content

Columns + Rows

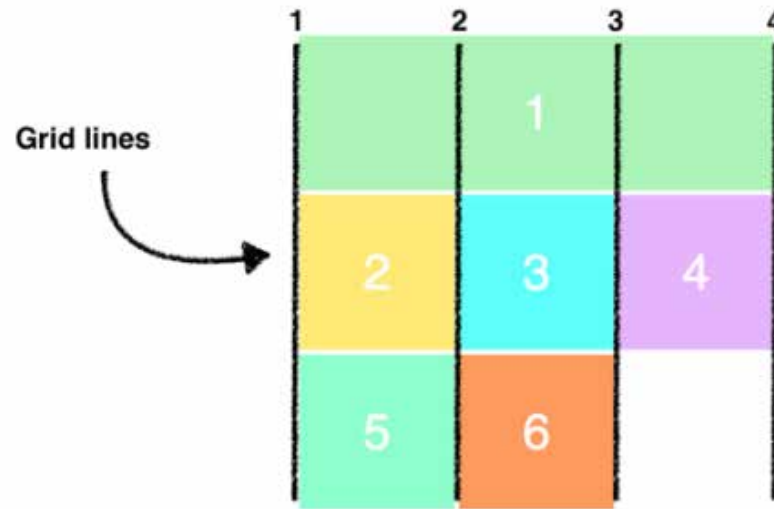


```
.item1 {  
  grid-column-start: 1;  
  grid-column-end: 3;  
}  
.item3 {  
  grid-row-start: 2;  
  grid-row-end: 4;  
}  
.item4 {  
  grid-column-start: 2;  
  grid-column-end: 4;  
}
```

Placing Items



```
.item1 {  
  grid-column-start: 1;  
  grid-column-end: 4;  
}
```



```
.item1 {  
  grid-column: 1 / 4;  
}
```

Start at column line 2,
end at column line 3.

Start at row line 3
and end at row line 4.

1	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

grid-column-start: 2;
grid-column-end: 3;

grid-row-start: 3;
grid-row-end: 4;

1	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

grid-column: 2 / 3;

grid-row: 3 / 4;

1	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

justify-items

Sets how the grid items will align inside of the container.

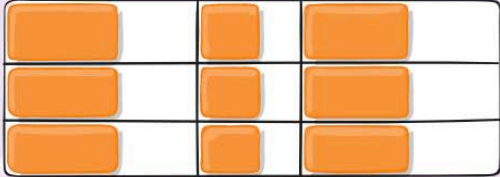
center: aligns items in the middle of cell

start: aligns items with left of cell

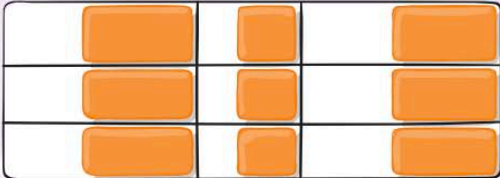
end: aligns items with end edge of cell

stretch: (default) fills the whole width of cell - *doesn't stretch items*

```
.container {  
  justify-items: start;  
}
```

A 3x3 grid where each orange item is aligned to the left side of its cell.

```
.container {  
  justify-items: end;  
}
```

A 3x3 grid where each orange item is aligned to the right side of its cell.

```
.container {  
  justify-items: center;  
}
```

A 3x3 grid where each orange item is centered horizontally within its cell.

```
.container {  
  justify-items: stretch;  
}
```

A 3x3 grid where each orange item is stretched to fill the width of its cell.