

# Grid container properties





#### Topics

B SKILLS

- Introduction
- Grid container properties



#### Introduction



A grid container is an HTML element that defines a grid layout. To control the layout of the grid items within the container, various CSS properties can be applied to the container itself.







Below is sample HTML we will use in later slides,



#### Grid container properties



#### display

display property is set to "grid" to define the container as a grid container.

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



#### grid-template-columns:



grid-template-columns property specifies the number and size of columns in the grid.

You can use values like pixels, percentages, or fractions to define the column widths.

```
.container
 display: grid;
 grid-template-columns: repeat(3, 1fr);
```

```
.container
 display: grid;
 grid-template-columns: repeat(3, 1fr);
```

```
.container
display: grid;
grid-template-columns: repeat(3, 1fr);
```

grid-template-columns: repeat(3, 1fr)		
1		3
4		6
grid-template-columns: auto auto auto		
each column will automatically size itself to fit	width depends on the content	each column will automatically size itself to fit
its content		its content
4	5	6
grid-template-columns: 100px 200px 300px		
1		
4		

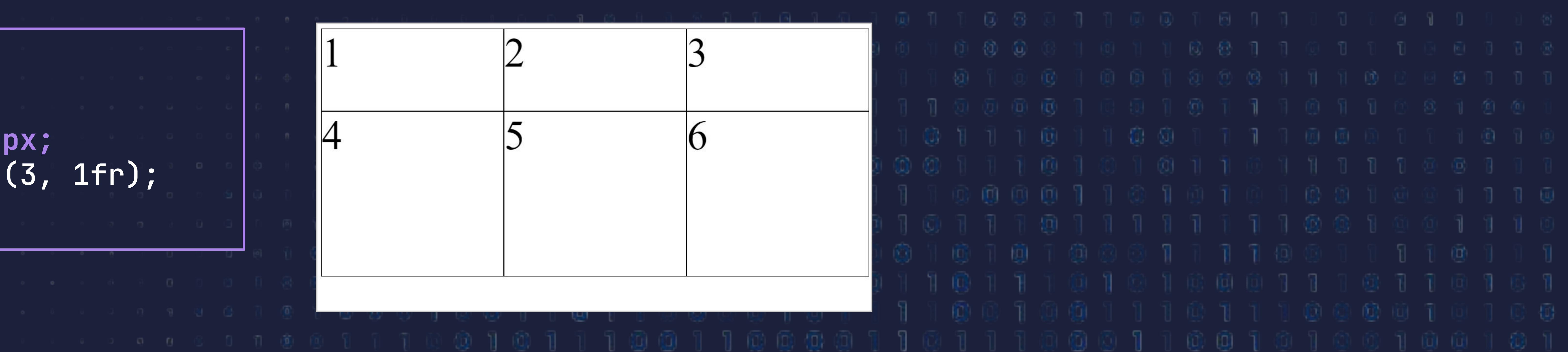


## grid-template-rows



grid-template-rows property specifies the number and size of rows in the grid. You can use values like pixels, percentages, or fractions to define the row heights.

```
.container {
 display: grid;
 grid-template-rows: 100px 200px;
 grid-template-columns: repeat(3, 1fr);
```





#### grid-template-areas



grid-template-areas property defines named grid areas within the grid container. Each area can be assigned a name and then referenced in the CSS rules for the grid items.

```
.grid-container {
 display: grid;
 grid-template-areas:
   "header header header "
   "sidebar main main"
   "footer footer footer";
 height: 90vh;
.header
 grid-area: header;
 background-color: lightblue;
.sidebar
 grid-area: sidebar;
 background-color: lightgreen;
main
 grid-area: main;
 background-color: yellow;
.footer 3
 background-color: lightpink;
```

```
Header
                                               Main Content
```



#### grid-template



grid-template is a shorthand property that combines grid-template-rows and grid-template-columns into a single property. It allows you to specify both the rows and columns of the grid in a single line.

```
.container {
  display: grid;
  grid-template:
    repeat(2, 1fr)
    / repeat(3, 1fr);
}
```

1	2	3
4	5	6



## grid-column-gap



grid-column-gap is a CSS property that defines the size of the gap or space between the columns of a grid. It is used to create space between the columns and make the grid more visually appealing.

```
.grid-container {
   display: grid;
   grid-template-columns: auto auto
auto;
   grid-column-gap: 20px;
}
```

4		



## grid-row-gap



grid-row-gap is a CSS property that defines the size of the gap or space between the rows of a grid. It is used to create space between the rows and make the grid more visually appealing.

```
.grid-container {
  display: grid;
  grid-template-columns: auto auto
auto;
 grid-row-gap: 20px;
```

	2	
4		

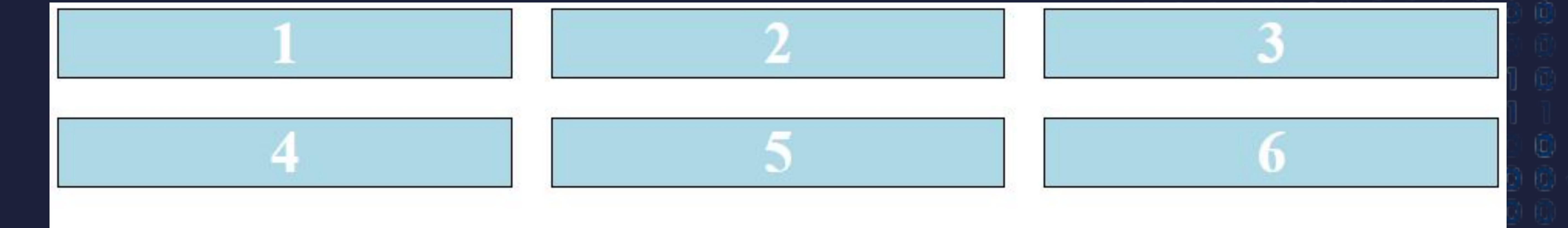


#### grid-gap



grid-gap property sets the size of the gap between grid items, both horizontally and vertically. It's a shorthand property for grid-column-gap and grid-row-gap properties.

```
.container {
  display: grid;
  grid-template:
  repeat(2, 1fr) / repeat(3, 1fr);
  grid-gap:20px;
}
```





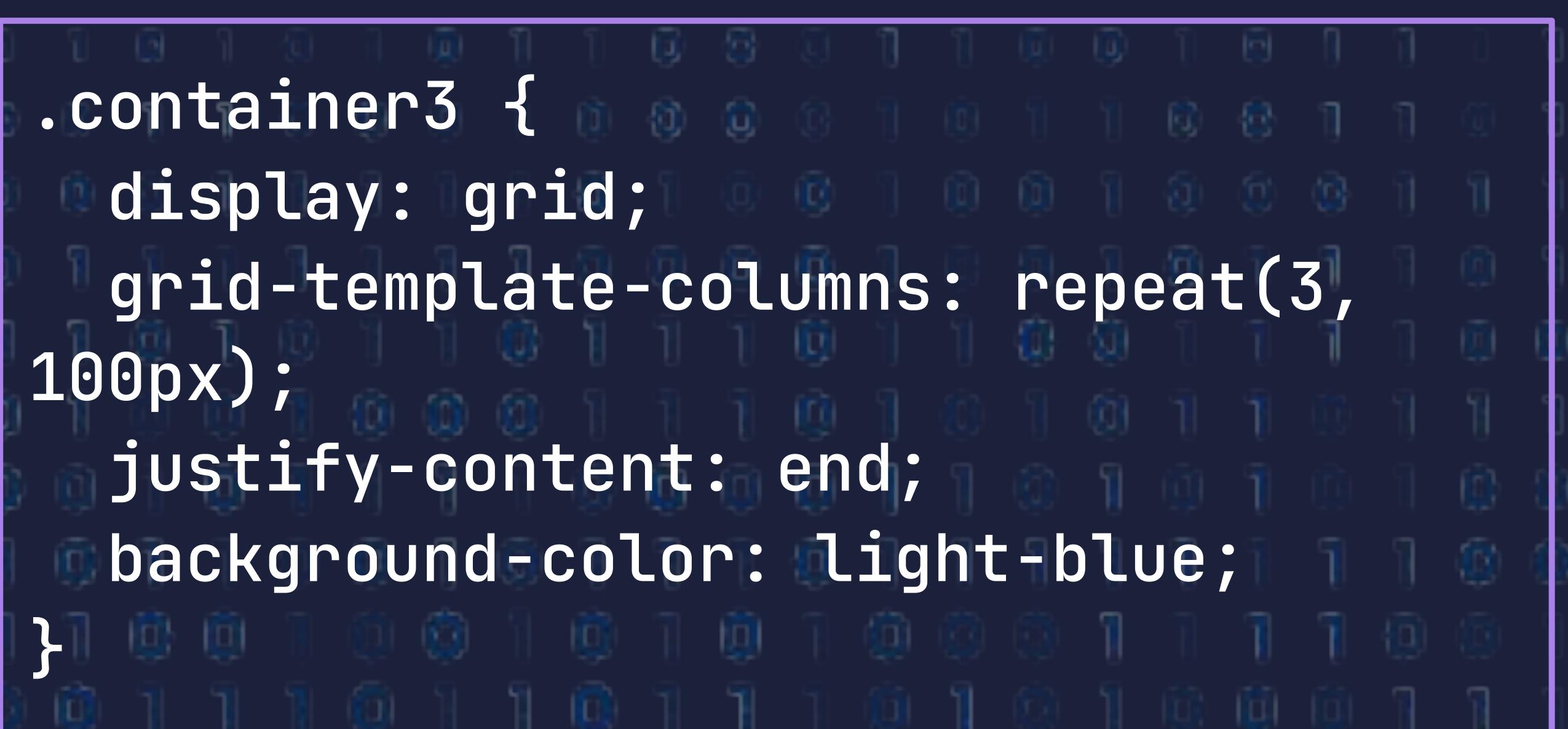
#### justify-content



The justify-content property in CSS Grid is used to horizontally align grid items within a grid container. It controls how the grid tracks (the columns or rows) are aligned along the horizontal axis of the grid container.

```
.container1 {
   display: grid;
   grid-template-columns: repeat(3,
100px);
   justify-content: center;
   background-color: light-blue;
}
```

```
.container2 {
   display: grid;
   grid-template-columns: repeat(3,
100px);
   justify-content: start;
   background-color: light-blue;
}
```



```
.container4 {
   display: grid;
   grid-template-columns: repeat(3,
100px);
   justify-content: space-between;
   background-color: light-blue;
}

.container5 {
   display: grid;
   grid-template-columns: repeat(3,
100px);
   justify-content: space-around;
   background-color: light-blue;
}
```



#### align-content



align-content is a CSS property used to set how the browser distributes the space between rows in a grid container, when there is extra space available. It only applies when there is extra space in the cross-axis.

```
.container1 {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    grid-template-rows: repeat(2, 30px);
    grid-row-gap: 10px;
    align-content: flex-start;
    height: 120px;
    border: 1px solid red;
}
```

```
.container2 {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    grid-template-rows: repeat(2, 30px);
    grid-row-gap: 10px;
    align-content: flex-end;
    height: 120px;
    border: 1px solid red;
}
```

```
.container3 {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    grid-template-rows: repeat(2, 30px);
    grid-row-gap: 10px;
    align-content: center;
    height: 120px;
    border: 1px solid red;
}
```

```
.container4 {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    grid-template-rows: repeat(2, 30px);
    grid-row-gap: 10px;
    align-content: space-between;
    height: 120px;
    border: 1px solid red;
}
```

```
.container5 {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    grid-template-rows: repeat(2, 30px);
    grid-row-gap: 10px;
    align-content: space-around;
    height: 120px;
    border: 1px solid red;
}
```

```
.container6 {
        display: grid;
        grid-template-columns: repeat(3, 1fr);
        grid-template-rows: repeat(2, 30px);
        grid-row-gap: 10px;
        align-content: stretch;
        height: 120px;
        border: 1px solid red;
}
```



#### align-content



1	2	3
4	5	6

1	2	3
4	5	6

itent: space-aro	und	
1	2	3
4	5	6

align-content: flex-end		
1	2	3
4	5	6

ntent: space-bet	2	3
1	2	3

align-content: stretch									
1	2	3							
4	5	6							
						0			



#### justify-items



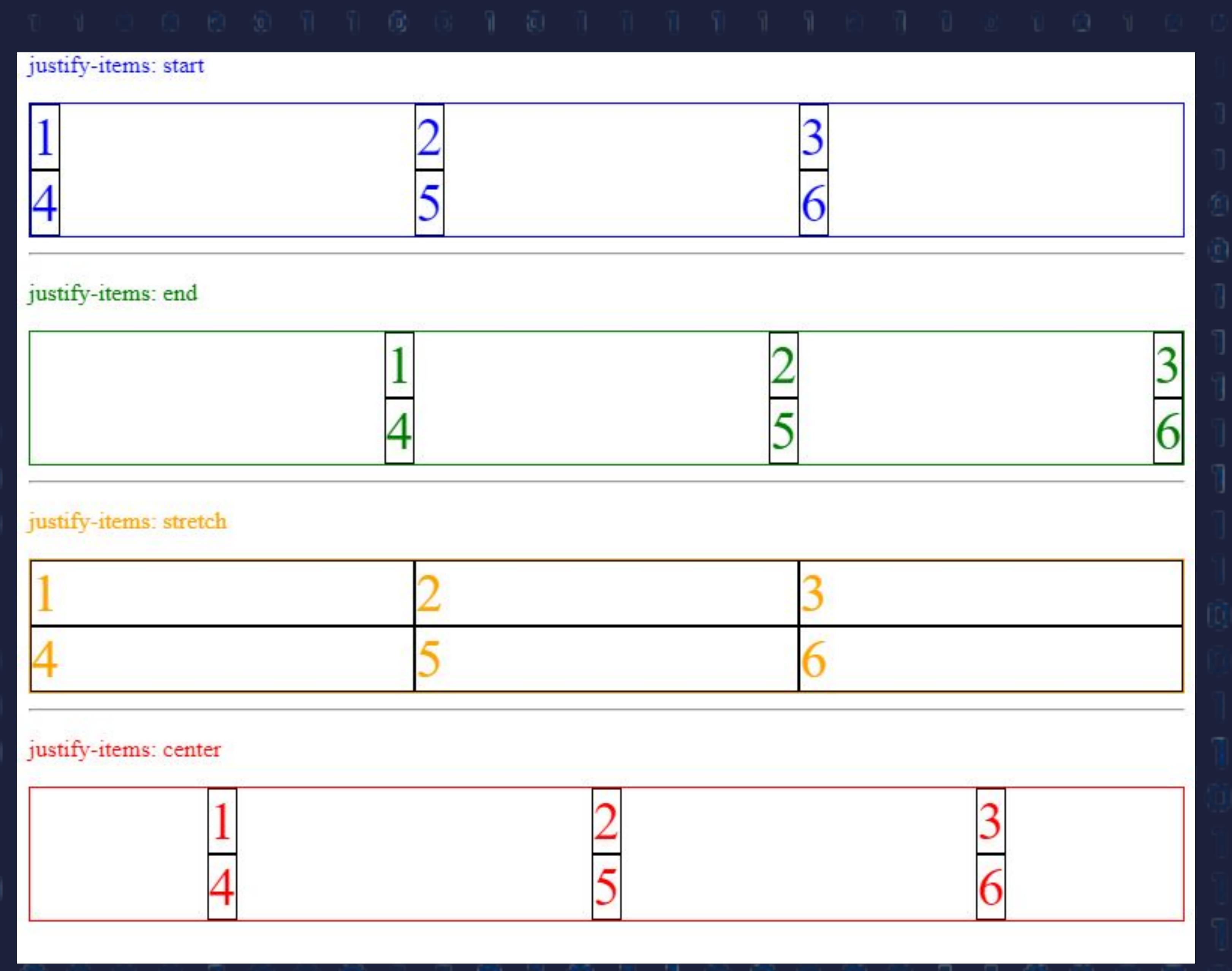
justify-items property aligns grid items horizontally within their grid cell. The values are start, end, center, and stretch

```
.container1 {
   display: grid;
   grid-template-columns: repeat(3, 1fr);
   justify-items: start;
   font-size: 36px;
   margin-bottom: 10px;
   border: 1px solid blue;
   color: blue;
}
```

```
.container2 {
   display: grid;
   grid-template-columns: repeat(3, 1fr);
   justify-items: end;
   font-size: 36px;
   margin-bottom: 10px;
   border: 1px solid blue;
   color: blue;
}
```

```
.container1 {
   display: grid;
   grid-template-columns: repeat(3, 1fr);
   justify-items: stretch;
   font-size: 36px;
   margin-bottom: 10px;
   border: 1px solid blue;
   color: blue;
}
```

```
.container1 {
   display: grid;
   grid-template-columns: repeat(3, 1fr);
   justify-items: center;
   font-size: 36px;
   margin-bottom: 10px;
   border: 1px solid blue;
   color: blue;
}
```





#### align-items



align-items property aligns grid items vertically within their grid cell.

```
.container1 {
   display: grid;
   grid-template-rows: 50px 50px;
   grid-template-columns: repeat(3, 1fr);
   align-items: start;
   border: 1px solid blue;
   color: blue;
}
```

```
.container1 {
  display: grid;
  grid-template-rows: 50px 50px;
  grid-template-columns: repeat(3, 1fr);
  align-items: end;
  border: 1px solid blue;
  color: blue;
}
```

```
.container1 {
   display: grid;
   grid-template-rows: 50px 50px;
   grid-template-columns: repeat(3, 1fr);
   align-items: stretch;
   border: 1px solid blue;
   color: blue;
}
```

```
.container1 {
    display: grid;
    grid-template-rows: 50px 50px;
    grid-template-columns: repeat(3, 1fr);
    align-items: center;
    border: 1px solid blue;
    color: blue;
}
```

align-items: start			
1	2	3	
	MED.		
4	5	6	
align-items: end			
9.	Z		.00
4		6	
align-items: stretch			
	2	3	
4	5	6	
align-items: center			
		)	
4		6	



#### grid-auto-rows



grid-auto-rows property specifies the height of rows that are automatically created when there is no explicit row definition. grid-auto-rows property specifies the height of rows that are automatically created when there is no explicit row definition.

```
.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-template-rows: 100px;
  grid-auto-rows: 250px;
}
```

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



#### grid-auto-columns



grid-auto-columns property specifies the width of columns that are automatically created when there is no explicit column definition.

```
.container {
   display: grid;
   grid-template-areas: "name email phone";
   grid-template-rows: 100px 100px;
   grid-template-columns: 200px 200px;
   grid-auto-columns: 350px;
}
```

```
5 6
```



#### grid-auto-flow



grid-auto-flow property controls the placement of grid items when they do not fit into the explicit grid.

```
.container1 {
  display: grid;
    grid-template: repeat(4, 1fr) / repeat(2, 1fr);
    grid-auto-flow: column;
}
.container2 {
  display: grid;
    grid-template: repeat(4, 1fr) / repeat(2, 1fr);
    grid-auto-flow: row;
}
```

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





grid is the shorthand property can be used to set a combination of grid properties, such as grid-template-rows, grid-template-columns, and grid-gap.

```
.container {
  display: grid;
  grid: 1fr 2fr / repeat(3, 1fr);
}
```

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

