

CSS PART-4

INTRODUCTION TO WEB DEVELOPMENT

R.A: Assad Ullah Khan

GRID IN CSS

- In CSS, a **grid** layout is a powerful system for creating **responsive** and flexible web page layouts. It allows you to divide the page into a **grid** of **rows** and **columns**, and then **position** and align elements within those **grid** areas. CSS **Grid** Layout is supported by most modern browsers.

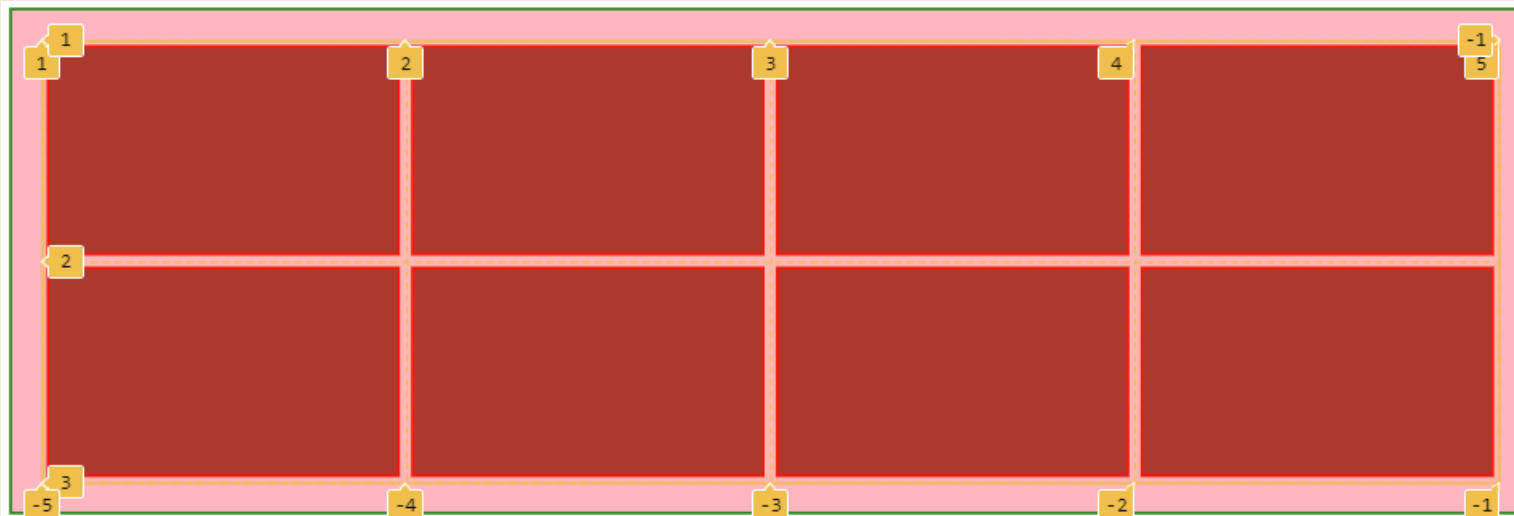
```
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
```

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

GRID-TEMPLATE-COLUMN

```
.container{  
  border: 2px solid green;  
  padding: 20px;  
  background-color: lightpink;  
  display: grid;  
  grid-template-columns: auto auto auto auto;  
}
```

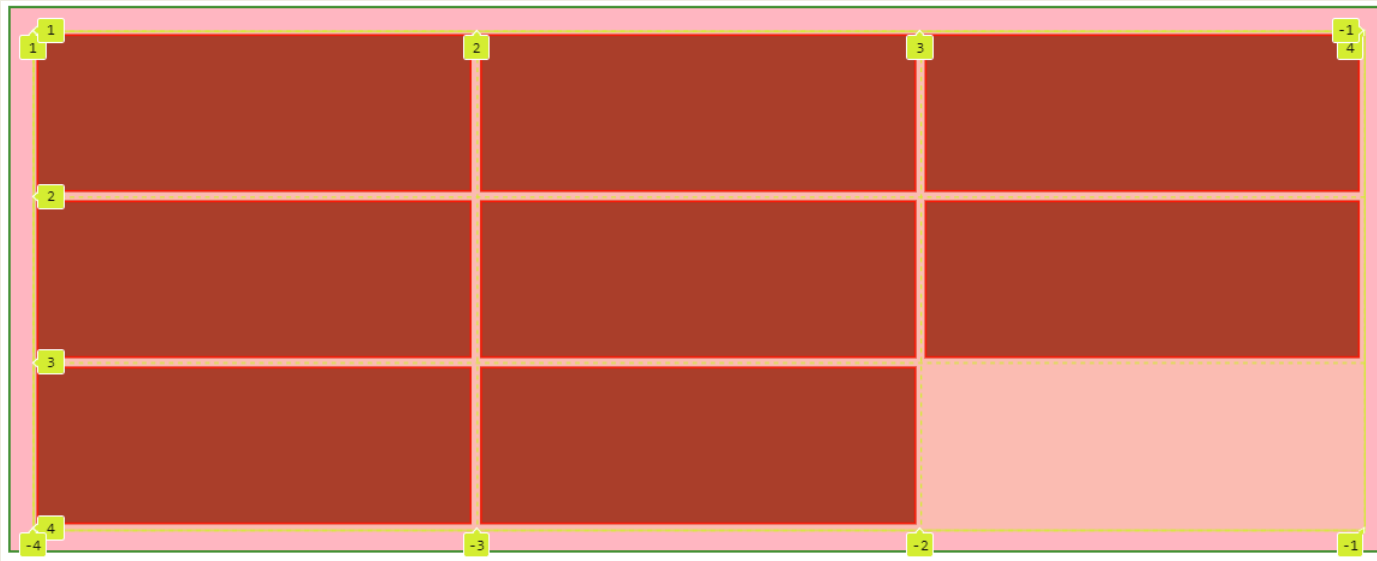
Divide Column in 4
equal parts total div is 8.



GRID-TEMPLATE-COLUMN

```
.container{  
  border: 2px solid green;  
  padding: 20px;  
  background-color: lightpink;  
  display: grid;  
  grid-template-columns: auto auto auto;  
}
```

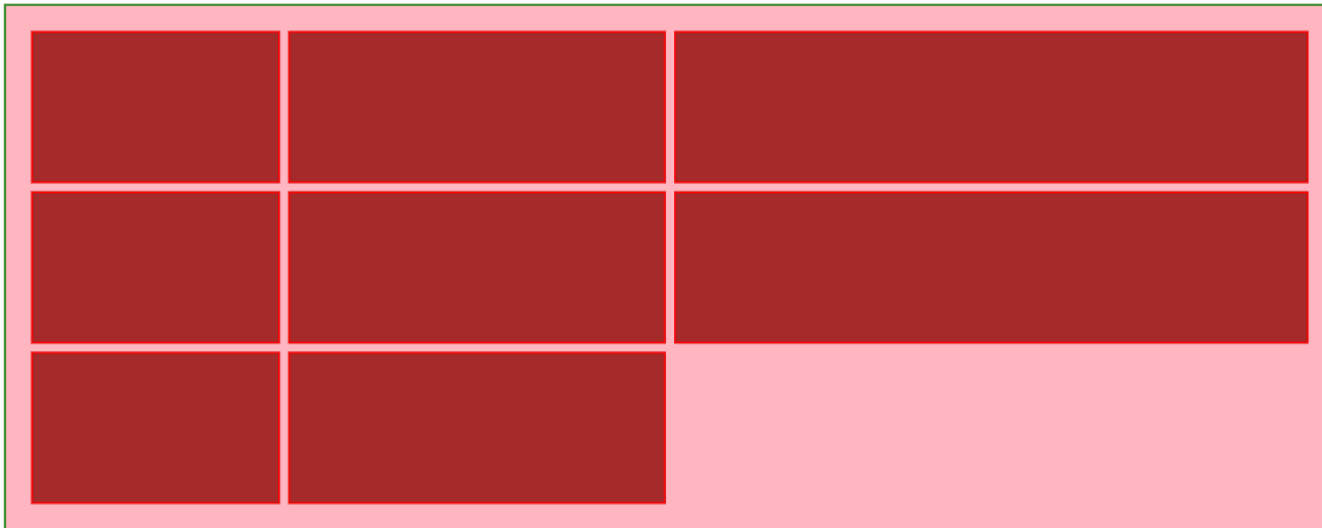
Divide Column in 3
equal parts total div is 8.



% USED

```
.container{  
  border: 2px solid green;  
  padding: 20px;  
  background-color: lightpink;  
  display: grid;  
  grid-template-columns: 20% 30% 50%;  
}
```

Divide Column in 3
different parts total div is
8.



PX USED

```
.container{  
  border: 2px solid green;  
  padding: 20px;  
  background-color: lightpink;  
  display: grid;  
  grid-template-columns: 300px 600px 240px;  
}
```

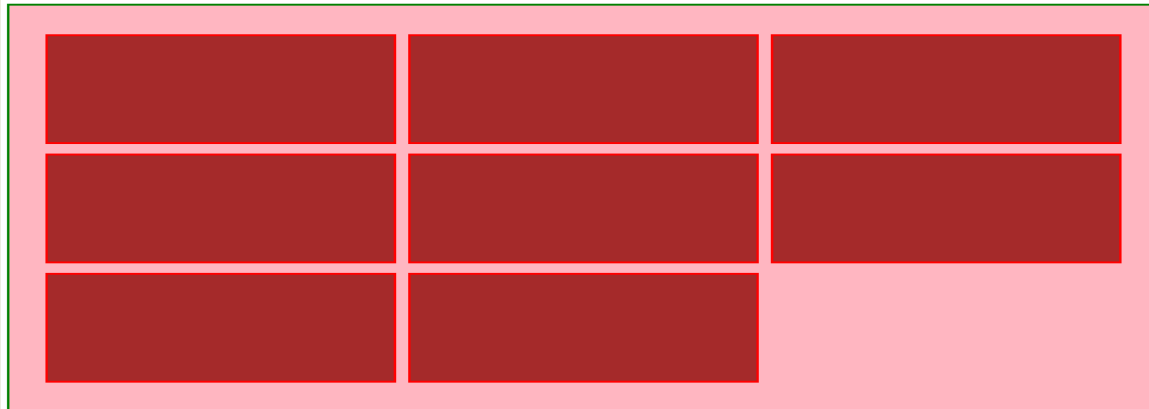


- Divide Column in 3 different parts total div is 8.
- Not recommended used % for responsiveness not px.

GRID-TEMPLATE-ROWS

- The **grid-template-rows** property in CSS Grid Layout is used to define the height of the rows in a grid container. Each value specified represents the height of a row.

```
.container{  
  border: 2px solid green;  
  background-color: lightpink;  
  grid-template-columns: auto auto auto;  
  grid-template-rows: 100px 100px 100px;  
}
```



- Every row height is 300px. You can also add a different.
- The value is also used in px, % or cm.

FR FRACTION IN GRID

- The **grid-template-columns** property in CSS Grid Layout is used to define the width of the columns in a grid container. Each value specified represents the width of a column.
- For example, if you set **grid-template-columns** to **1fr 1fr 1fr;**, it means that the grid container will have three columns, each with an equal width of one fraction of the available space.



FRACTION EXAMPLE

```
.container{  
  border: 2px solid green;  
  padding: 20px;  
  background-color: lightpink;  
  display: grid;  
  grid-template-columns: 1fr 2fr 1fr;  
}
```

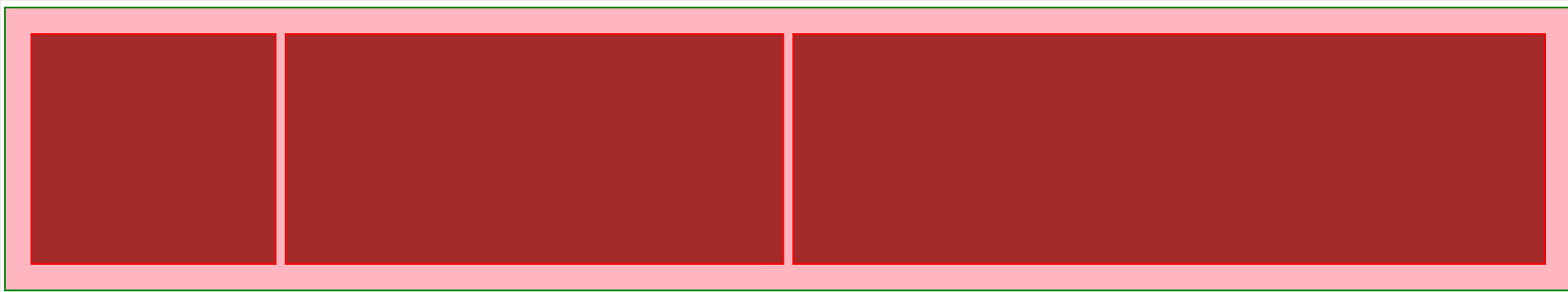
- Divide section in different parts.



FRACTION EXAMPLE

```
.container{  
  border: 2px solid green;  
  padding: 20px;  
  background-color: lightpink;  
  display: grid;  
  grid-template-columns: 1fr 2fr 3fr;  
}
```

- Divide section in different parts.



GRID GAP

- The **grid-gap** property in CSS Grid Layout is used to define the gap, or spacing, between the rows and columns in a grid container. It sets the distance between adjacent grid lines, creating space between grid cells.
- or example, if you set **grid-gap** to **30px;**, it means that there will be a gap of 30 pixels between each row and column in the grid.



EXAMPLE

```
.container{  
  background-color: lightpink;  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
  grid-gap: 30px;  
}
```

30 pixel between each row and column.



COLUMNS GAP

```
.container{  
  background-color: lightpink;  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
  grid-column-gap: 60px;  
}
```

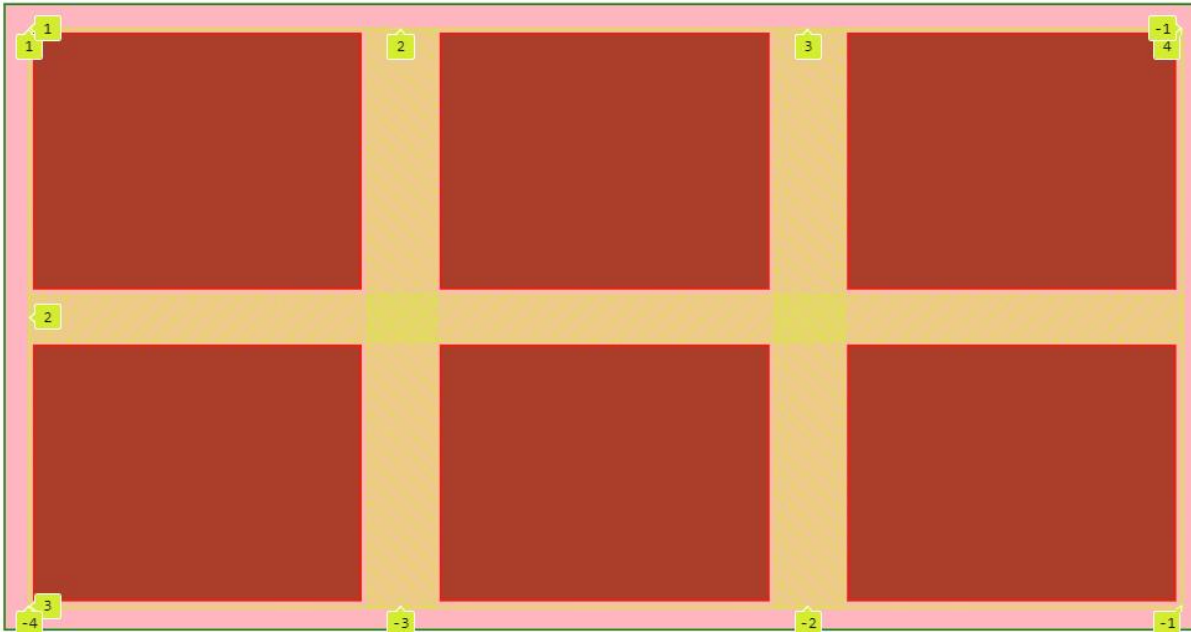
60 pixel space between columns.



ROWS GAP

```
.container{  
  background-color: lightpink;  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
  grid-row-gap: 40px;  
}
```

40 pixel space between rows.



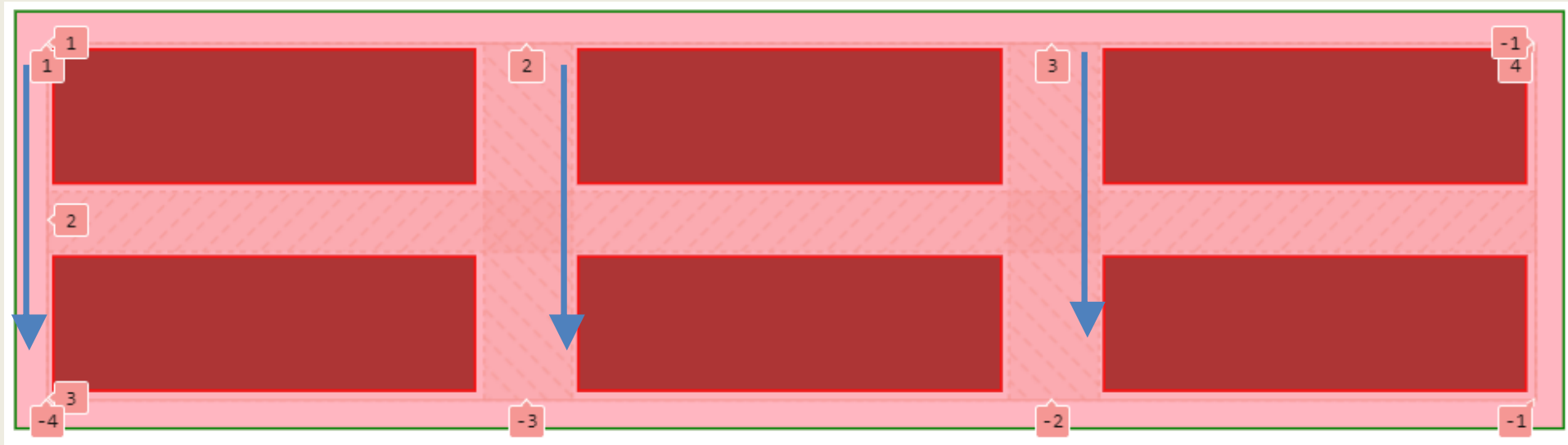
GRID AUTO ROWS

- The **grid-auto-rows** property in CSS Grid Layout is used to define the height of rows that are automatically generated to accommodate content that exceeds the explicitly defined rows.
- For example, if you set **grid-auto-rows** to **100px;**, it means that any additional rows created by the grid layout to accommodate content will have a height of 100 pixels.

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
  grid-auto-rows: 100px;  
}
```

GRID-ROW-START & END

- The **grid-row-start** property in CSS Grid Layout is used to specify the starting position of a grid item along the vertical axis (rows). It determines which grid line the item should start from.



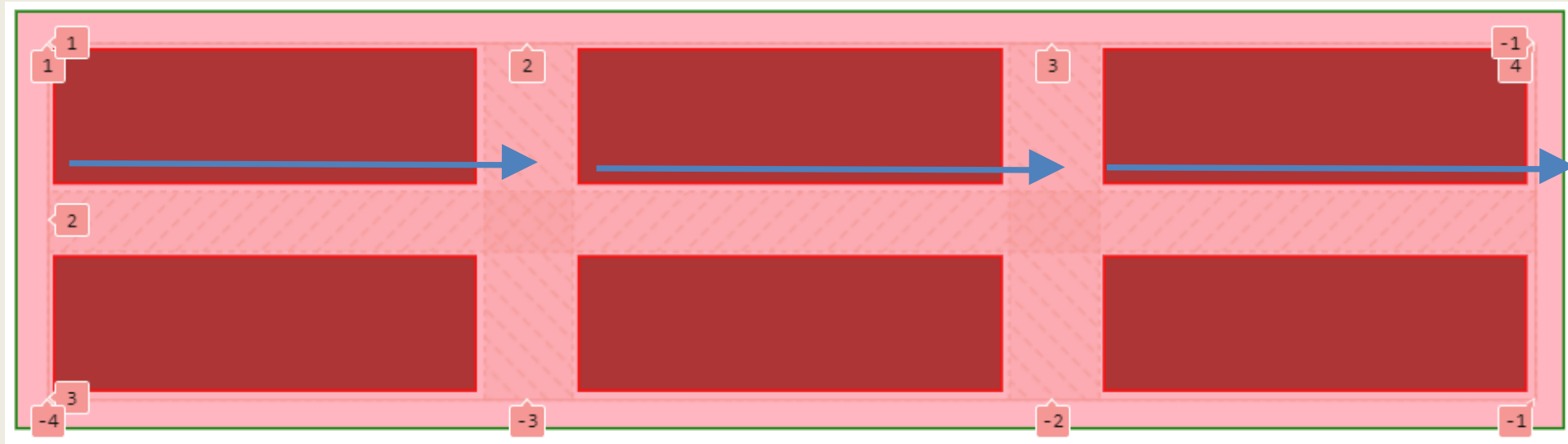
Example

```
#item1{  
  background-color: yellow;  
  grid-row-start: 1;  
  grid-row-end: 3;  
}
```



GRID-COLUMN-START & END

The **grid-column-start** property in CSS Grid Layout is used to specify the starting position of a grid item along the horizontal axis (columns). It determines which grid line the item should start from.



Example

```
#item1{  
  background-color: yellow;  
  grid-column-start: 1;  
  grid-column-end: 3;  
}
```





End

I hope this slide will help you to clear your idea

