

## Naming Grid Lines

- In order to make complex templates we use naming conventions for every line which is known as "Grid Line Naming".
- If we want to make grid-layout then we have to name every element starting-point & ending-point whatever we want.
- Make sure that we can join two words by underscore / hyphens not with space [e.g., line1-start]

## 1- Naming Grid-Rows Lines

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    .container
    {
      background-color: silver;
      width: 700px;
      height:650px;
      margin: 70px 0 0 50px;
      display: grid;
      grid-template-columns: 1fr 1fr 1fr ;
      grid-template-rows: [header-start] 100px [header-end] 100px 100px 100px;
      grid-column-gap:20px;
      grid-row-gap: 50px;
    }
    .items1
    {
      background-color: orangered;
    }
  
```

```
.items2    {
  background-color: blue;
  grid-column: 1 / -1;
  grid-row: header-start / header-end; /* In place of (grid-row: 1 / 2;) */
}
.items3    {
  background-color: green;
  grid-row: 2 / -1;
}
.items4    {
  background-color: yellow;
  grid-column: 2 / span 2;
  grid-row: 2 / 3;
}
.items5    {
  background-color: blueviolet;
}
.items6    {
  background-color: magenta;
}
</style>
</head>
<body>

<div class="container">
  <div class="item items1">Div1 </div>
  <div class="item items2">Div2 </div>
  <div class="item items3">Div3 </div>
  <div class="item items4">Div4 </div>
  <div class="item items5">Div5 </div>
  <div class="item items6">Div6 </div>
</div>

</body>
</html>
```

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    .container {
      background-color: silver;
      width: 700px;
      height: 650px;
      margin: 70px 0 0 50px;
      display: grid;
      grid-template-columns: 1fr 1fr 1fr;
      grid-template-rows: [div2_start] 100px [div2_end div3-start] 100px [div3-end]
100px 100px;
      grid-column-gap: 20px;
      grid-row-gap: 50px;
    }
    .items1 {
      background-color: orange;
    }
    .items2 {
      background-color: blue;
      grid-column: 1 / -1;
      grid-row: div2_start / div2_end;
    }
    .items3 {
      background-color: green;
      grid-row: div3-start / div3-end;
    }
  </style>
</html>
```

```
.items4
{
  background-color: yellow;
  grid-column: 2 / span 2;
  grid-row: 2 / 3;
}
.items5
{
  background-color: blueviolet;
}
.items6
{
  background-color: magenta;
}
</style>
</head>
<body>
<div class="container">
  <div class="item items1">Div1 </div>
  <div class="item items2">Div2 </div>
  <div class="item items3">Div3 </div>
  <div class="item items4">Div4 </div>
  <div class="item items5">Div5 </div>
  <div class="item items6">Div6 </div>
</div>

</body>
</html>
```



## 2- Naming Grid-Column Lines

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    .container {
      background-color: silver;
      width: 700px;
      height: 650px;
      margin: 70px 0 0 50px;
      display: grid;
      grid-template-columns: repeat(3, [col-start] 1fr [col-end]);
      grid-template-rows: [div2_start] 100px [div2_end div3-start] 100px [div3-end]
100px 100px;
      grid-column-gap: 20px;
      grid-row-gap: 50px;
    }
    .items1 {
      background-color: orangered;
    }
    .items2 {
      background-color: blue;
      grid-column: col-start 1 / col-end 3;
      grid-row: div2_start / div2_end;
    }
    .items3
    {
      background-color: green;
      grid-row: div3-start / div3-end;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="items1">
      <div class="items2">
        <div class="items3">
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

```
.items4
{
  background-color: yellow;
  grid-column: 2 / span 2;
  grid-row: 2 / 3;
}
.items5
{
  background-color: blueviolet;
}
.items6
{
  background-color: magenta;
}
</style>
</head>
<body>
<div class="container">
  <div class="item items1">Div1 </div>
  <div class="item items2">Div2 </div>
  <div class="item items3">Div3 </div>
  <div class="item items4">Div4 </div>
  <div class="item items5">Div5 </div>
  <div class="item items6">Div6 </div>
</div>

</body>
</html>
```



## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    .container {
      background-color: silver;
      width: 700px;
      height: 650px;
      margin: 70px 0 0 50px;
      display: grid;
      grid-template-columns: repeat(3,[col-start] 1fr [col-end]) ;
      grid-template-rows: [div2_start] 100px [div2_end div3-start] 100px [div3-end]
100px 100px;
      grid-column-gap: 20px;
      grid-row-gap: 50px;
    }
    .items1 {
      background-color: orange;
    }
    .items2
    {
      background-color: blue;
      grid-column: col-start 1 / col-end; /* grid-line-name column-number / grid-line-
name column-number */
      grid-row: div2_start / div2_end;
    }
    .items3
    {
      background-color: green;
      grid-row: div3-start / div3-end;
    }
  </style>
</html>
```

```
.items4
{
  background-color: yellow;
  grid-column: 2 / span 2;
  grid-row: 2 / 3;
}
.items5
{
  background-color: blueviolet;
}
.items6
{
  background-color: magenta;
}
</style>
</head>
<body>
<div class="container">
  <div class="item items1">Div1 </div>
  <div class="item items2">Div2 </div>
  <div class="item items3">Div3 </div>
  <div class="item items4">Div4 </div>
  <div class="item items5">Div5 </div>
  <div class="item items6">Div6 </div>
</div>

</body>
</html>
```





## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    .container {
      background-color: silver;
      width: 700px;
      height: 650px;
      margin: 70px 0 0 50px;
      display: grid;
      grid-template-columns: repeat(3,[col-start] 1fr [col-end]) ;
      grid-template-rows: [div2_start] 100px [div2_end div3-start] 100px [div3-end]
100px 100px;
      grid-column-gap: 20px;
      grid-row-gap: 50px;
    }
    .items1 {
      background-color: orange;
    }
    .items2
    {
      background-color: blue;
      grid-column: col-start 1 / -1; /* grid-line-name column-number / grid-line-name
column-number */
      grid-row: div2_start / div2_end;
    }
    .items3
    {
      background-color: green;
      grid-row: div3-start / div3-end;
    }
  </style>
</html>
```

```
.items4
{
  background-color: yellow;
  grid-column: 2 / span 2;
  grid-row: 2 / 3;
}
.items5
{
  background-color: blueviolet;
}
.items6
{
  background-color: magenta;
}
</style>
</head>
<body>
<div class="container">
  <div class="item items1">Div1 </div>
  <div class="item items2">Div2 </div>
  <div class="item items3">Div3 </div>
  <div class="item items4">Div4 </div>
  <div class="item items5">Div5 </div>
  <div class="item items6">Div6 </div>
</div>

</body>
</html>
```



### 3 - Naming Grid Area Properties

- In order to naming grid-areas we use two properties :-
  - grid-template-areas
  - grid-area
- The "grid-template-areas" property should be apply on the "parent container" and the "grid-area" property should be apply on the "child container" so as to name each child element.
- For this procedure we have to decide the dimensions of every container earlier.
- This methodology is not suitable for large projects.

#### Example in steps

##### Step 1

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    *
    {
      box-sizing: border-box;
    }
    html,body
    {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 800px;
      background-color: silver;
```

```
margin: 45px auto;
font-size: 40px;
color: white;
display: grid;
grid-template-columns: repeat(3, 1fr);
grid-template-rows: 120px 60px 100px 400px 50px;
grid-gap: 20px;
grid-template-areas: "header header header";
}
.item1 {
  background-color: red;
  grid-area: header;
}
.item2 {
  background-color: blue;
  grid-area: menu;
}
.item3 {
  background-color: green;
  grid-area: box1;
}
.item4 {
  background-color: yellow;
  grid-area: box2;
}
.item5 {
  background-color: blueviolet;
  grid-area: content;
}
.item6 {
  background-color: orangered;
  grid-area: sidebar;
}
.item7 {
  background-color: darkgreen;
  grid-area: footer;
}
```

```
</style>
</head>
<body>

<div class="container">
  <div class="item item1">Header</div>
  <div class="item item2">Menu</div>
  <div class="item item3">Box1</div>
  <div class="item item4">Box2</div>
  <div class="item item5">Sidebar</div>
  <div class="item item6">Content</div>
  <div class="item item7">Footer</div>
</div>

</body>
</html>
```



## Step 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    *
    {
      box-sizing: border-box;
    }
    html,body
    {
      margin: 0px;
      padding: 0px;
    }
    .container
    {
      width: 800px;
      background-color: silver;
      margin: 45px auto;
      font-size: 40px;
      color: white;
      display: grid;
      grid-template-columns: repeat(3, 1fr);
      grid-template-rows: 120px 60px 100px 400px 50px;
      grid-gap: 20px;
      grid-template-areas: "header header header"
                           "menu menu menu" ;
    }
    .item1 {
      background-color: red;
      grid-area: header;
    }
```

```
.item2    {
  background-color: blue;
  grid-area: menu;    }
.item3    {
  background-color: green;
  grid-area: box1;    }
.item4    {
  background-color: yellow;
  grid-area: box2;    }
.item5    {
  background-color: blueviolet;
  grid-area: content;  }
.item6    {
  background-color: orangered;
  grid-area: sidebar;  }
.item7    {
  background-color: darkgreen;
  grid-area: footer;   }
</style>
</head>
<body>
  <div class="container">
    <div class="item item1">Header</div>
    <div class="item item2">Menu</div>
    <div class="item item3">Box1</div>
    <div class="item item4">Box2</div>
    <div class="item item5">Sidebar</div>
    <div class="item item6">Content</div>
    <div class="item item7">Footer</div>
  </div>

</body>
</html>
```

### Step 3

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    *
    {
      box-sizing: border-box;
    }
    html,body
    {
      margin: 0px;
      padding: 0px;
    }
    .container
    {
      width: 800px;
      background-color: silver;
      margin: 45px auto;
      font-size: 40px;
      color: white;
      display: grid;
      grid-template-columns: repeat(3, 1fr);
      grid-template-rows: 120px 60px 100px 400px 50px;
      grid-gap: 20px;
      grid-template-areas: "header header header"
                          "menu menu menu"
                          "box1 box2 sidebar" ;
    }
    .item1
    {
      background-color: red;
      grid-area: header;
    }
    .item2
    {
      background-color: blue;
      grid-area: menu;
    }
```



```
.item3    {
  background-color: green;
  grid-area: box1;
}
.item4    {
  background-color: yellow;
  grid-area: box2;
}
.item5    {
  background-color: blueviolet;
  grid-area: content;
}
.item6    {
  background-color: orangered;
  grid-area: sidebar;
}
.item7    {
  background-color: darkgreen;
  grid-area: footer;
}
</style>
</head>
<body>
  <div class="container">
    <div class="item item1">Header</div>
    <div class="item item2">Menu</div>
    <div class="item item3">Box1</div>
    <div class="item item4">Box2</div>
    <div class="item item5">Content</div>
    <div class="item item6">Sidebar</div>
    <div class="item item7">Footer</div>
  </div>

</body>
</html>
```



### Step 4

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 800px;
      background-color: silver;
      margin: 45px auto;
      font-size: 40px;
      color: white;
      display: grid;
      grid-template-columns: repeat(3, 1fr);
      grid-template-rows: 120px 60px 100px 400px 50px;
      grid-gap: 20px;
      grid-template-areas: "header header header"
                           "menu menu menu"
                           "box1 box2 sidebar"
                           "content content sidebar" ;
    }
    .item1 {
      background-color: red;
      grid-area: header;
    }
```

```
.item2
{
    background-color: blue;
    grid-area: menu;    }
.item3    {
    background-color: green;
    grid-area: box1;    }
.item4    {
    background-color: yellow;
    grid-area: box2;    }
.item5    {
    background-color: blueviolet;
    grid-area: content;    }
.item6    {
    background-color: orangered;
    grid-area: sidebar;    }
.item7    {
    background-color: darkgreen;
    grid-area: footer;    }
</style>
</head>
<body>

<div class="container">
    <div class="item item1">Header</div>
    <div class="item item2">Menu</div>
    <div class="item item3">Box1</div>
    <div class="item item4">Box2</div>
    <div class="item item5">Content</div>
    <div class="item item6">Sidebar</div>
    <div class="item item7">Footer</div>
</div>

</body>
</html>
```



LABS

## Step 5

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    *
    {
      box-sizing: border-box;
    }
    html,body
    {
      margin: 0px;
      padding: 0px;
    }
    .container
    {
      width: 800px;
      background-color: silver;
      margin: 45px auto;
      font-size: 40px;
      color: white;
      display: grid;
      grid-template-columns: repeat(3, 1fr);
      grid-template-rows: 120px 60px 100px 400px 50px;
      grid-gap: 20px;
      grid-template-areas: "header header header"
                           "menu menu menu"
                           "box1 box2 sidebar"
                           "content content sidebar"
                           "footer footer footer" ;
    }
    .item1
    {
      background-color: red;
      grid-area: header;
    }
```

```
.item2    {
  background-color: blue;
  grid-area: menu;    }
.item3    {
  background-color: green;
  grid-area: box1;    }
.item4    {
  background-color: yellow;
  grid-area: box2;    }
.item5    {
  background-color: blueviolet;
  grid-area: content;  }
.item6    {
  background-color: orangered;
  grid-area: sidebar;  }
.item7    {
  background-color: darkgreen;
  grid-area: footer;   }
</style>
</head>
<body>

<div class="container">
  <div class="item item1">Header</div>
  <div class="item item2">Menu</div>
  <div class="item item3">Box1</div>
  <div class="item item4">Box2</div>
  <div class="item item5">Sidebar</div>
  <div class="item item6">Content</div>
  <div class="item item7">Footer</div>
</div>

</body>
</html>
```

#### 4- Replacing with Dot(.) Operator

When we don't want to fill the full width of container / or we don't want to pass the name of "grid-area" then we use Dot(.) Operator.

##### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 800px;
      background-color: silver;
      margin: 45px auto;
      font-size: 40px;
      color: white;
      display: grid;
      grid-template-columns: repeat(3, 1fr);
      grid-template-rows: 120px 60px 100px 400px 50px;
      grid-gap: 20px;
      grid-template-areas: ". header ."
                          "menu menu menu"
                          "box1 box2 sidebar"
                          "content content sidebar"
                          "footer footer footer" ;
    }
  </style>
</html>
```

```
.item1    {
  background-color: red;
  grid-area: header;    }
.item2    {
  background-color: blue;
  grid-area: menu;      }
.item3    {
  background-color: green;
  grid-area: box1;      }
.item4    {
  background-color: yellow;
  grid-area: box2;      }
.item5    {
  background-color: blueviolet;
  grid-area: content;    }
.item6    {
  background-color: orangered;
  grid-area: sidebar;    }
.item7    {
  background-color: darkgreen;
  grid-area: footer;     }
</style>
</head>
<body>
  <div class="container">
    <div class="item item1">Header</div>
    <div class="item item2">Menu</div>
    <div class="item item3">Box1</div>
    <div class="item item4">Box2</div>
    <div class="item item5">Sidebar</div>
    <div class="item item6">Content</div>
    <div class="item item7">Footer</div>
  </div>
</body>
</html>
```

## 5- Dynamic Grids - minmax() / min-content / max-content

### a) max-content

- max-content is a keyword representing the largest maximal content contribution of the grid items occupying the grid track.
- This property is not used generally.
- If we use the value max-content, the column will take the maximum space, according to it needs.

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 800px;
      background-color: silver;
      margin: 45px auto;
      font-size: 40px;
      color: white;
      display: grid;
      grid-template-rows: 200px 200px;
      grid-template-columns: max-content 1fr 1fr 1fr;
      grid-gap: 20px;
    }
```



```
.item1    {
  background-color: red;
}
.item2    {
  background-color: blue;
}
.item3    {
  background-color: green;
}
.item4    {
  background-color: yellow;
}
.item5    {
  background-color: blueviolet;
}
.item6    {
  background-color: orangered;
}
.item7    {
  background-color: darkgreen;
}
.item8    {
  background-color: indigo;
}
</style>
</head>
<body>
  <div class="container">
    <div class="item item1">Lorem ipsum, dolor sit amet consectetur adipisicing elit.
Animi pariatur quibusdam, asperiores deserunt rerum quos quo! Est molestiae quis ex
quod facere quos mollitia error itaque delectus, enim odio dignissimos cupiditate
aperiam, molestias minima, suscipit natus ducimus quisquam accusantium eaque
sapiente consectetur quidem quia magni. Distinctio maiores similique aliquid animi, quo
quibusdam perspiciatis vel facilis quis inventore sequi excepturi eos nostrum expedita
amet earum soluta, corrupti officia commodi fugit hic eaque voluptas. Temporibus
autem inventore adipisci quibusdam reiciendis, doloremque omnis consequuntur atque
```

veritatis praesentium, nulla eos neque repellendus expedita mollitia facilis fugit dolores.

Quae provident ullam adipisci. Explicabo, sequi ex!</div>

```
<div class="item item2">Menu</div>
```

```
<div class="item item3">Box1</div>
```

```
<div class="item item4">Box2</div>
```

```
<div class="item item5">Sidebar</div>
```

```
<div class="item item6">Content</div>
```

```
<div class="item item7">Footer</div>
```

```
<div class="item item8">Extra</div>
```

```
</div>
```

```
</body>
```

```
</html>
```



## b) min-content

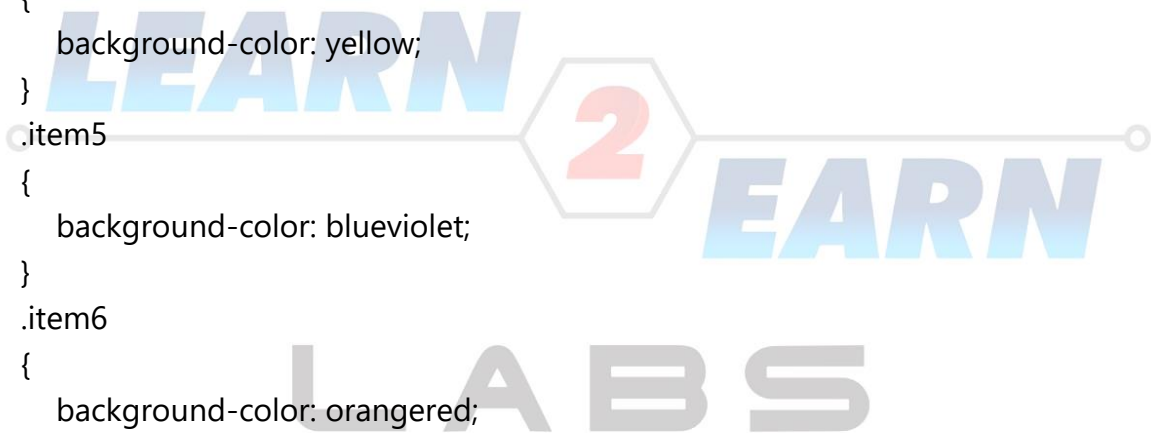
- If we use the value min-content, the column will take the minimal space according to the content.
- This property is very useful.
- This property expands the div according to the "largest character word" in case of "columns" and according to the "content" in case of expanding "rows".

Example -- In case of columns [Expand Column according to the largest word]

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 800px;
      background-color: silver;
      margin: 45px auto;
      font-size: 40px;
      color: white;
      display: grid;
      grid-template-rows: 200px 200px;
      grid-template-columns: min-content 1fr 1fr 1fr;
      grid-gap: 20px;
    }
```

```
.item1
{
  background-color: red;
}

.item2
{
  background-color: blue;
}
.item3
{
  background-color: green;
}
.item4
{
  background-color: yellow;
}
.item5
{
  background-color: blueviolet;
}
.item6
{
  background-color: orangered;
}
.item7
{
  background-color: darkgreen;
}
.item8
{
  background-color: indigo;
}
</style>
</head>
<body>
```



```
<div class="container">  
  <div class="item item1">Lorem ipsum, dolor sit amet consectetur adipisicing elit.  
  Animi pariatur quibusdam, asperiores deserunt rerum quos quo! Est molestiae quis ex  
  quod facere quos mollitia error itaque delectus, enim odio dignissimos cupiditate  
  aperiam, molestias minima, suscipit natus ducimus quisquam accusantium eaque  
  sapiente consectetur quidem quia magni. Distinctio maiores similique aliquid animi, quo  
  quibusdam perspiciatis vel facilis quis inventore sequi excepturi eos nostrum expedita  
  amet earum soluta, corrupti officia commodi fugit hic eaque voluptas. Temporibus  
  autem inventore adipisci quibusdam reiciendis, doloremque omnis consequuntur atque  
  veritatis praesentium, nulla eos neque repellendus expedita mollitia facilis fugit dolores.  
  Quae provident ullam adipisci. Explicabo, sequi ex!</div>  
    <div class="item item2">Menu</div>  
    <div class="item item3">Box1</div>  
    <div class="item item4">Box2</div>  
    <div class="item item5">Sidebar</div>  
    <div class="item item6">Content</div>  
    <div class="item item7">Footer</div>  
    <div class="item item8">Extra</div>  
</div>  
</body>  
</html>
```

LABS

Example -- In case of rows [Expand row according to the full content]

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    *
    {
      box-sizing: border-box;
    }
    html,body
    {
      margin: 0px;
      padding: 0px;
    }
    .container
    {
      width: 800px;
      background-color: silver;
      margin: 45px auto;
      font-size: 40px;
      color: white;
      display: grid;
      grid-template-rows: repeat(2,min-content);
      grid-template-columns: min-content 1fr 1fr 1fr;
      grid-gap: 20px;
    }
    .item1
    {
      background-color: red;
    }
    .item2
    {
      background-color: blue;
    }
  </style>
</html>
```

```
.item3    {
  background-color: green;    }
.item4    {
  background-color: yellow;    }
.item5    {
  background-color: blueviolet;    }
.item6    {
  background-color: orangered;    }
.item7    {
  background-color: darkgreen;    }
.item8    {
  background-color: indigo;    }
</style>
</head>
<body>
  <div class="container">
    <div class="item item1">Lorem ipsum, dolor sit amet consectetur adipisicing elit.
    Animi pariatur quibusdam, asperiores deserunt rerum quos quo! Est molestiae quis ex
    quod facere quos mollitia error itaque delectus, enim odio dignissimos cupiditate
    aperiam, molestias minima, suscipit natus ducimus quisquam accusantium eaque
    sapiente consectetur quidem quia magni. Distinctio maiores similique aliquid animi, quo
    quibusdam perspiciatis vel facilis quis inventore sequi excepturi eos nostrum expedita
    amet earum soluta, corrupti officia commodi fugit hic eaque voluptas. Temporibus
    autem inventore adipisci quibusdam reiciendis, doloremque omnis consequuntur atque
    veritatis praesentium, nulla eos neque repellendus expedita mollitia facilis fugit dolores.
    Quae provident ullam adipisci. Explicabo, sequi ex!</div>
    <div class="item item2">Menu</div>
    <div class="item item3">Box1</div>
    <div class="item item4">Box2</div>
    <div class="item item5">Sidebar</div>
    <div class="item item6">Content</div>
    <div class="item item7">Footer</div>
    <div class="item item8">Extra</div>
  </div>
</body>
</html>
```

### c) minmax() function

- CSS Grid Layout makes it easy to create grid tracks that have a size value between a certain range with the CSS minmax function.
- The minmax() CSS function defines a size range greater than or equal to min and less than or equal to max.
- This function is very useful in case of "rows".

#### Syntax

minmax(minimum\_size\_of\_container / maximum\_size\_of\_container)

#### In case of rows

#### Example -- with pixels

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 800px;
      background-color: silver;
      margin: 45px auto;
      font-size: 40px;
      color: white;
      display: grid;
      grid-template-rows: repeat(2,minmax(100px,400px));
```



```
    grid-template-columns: 1fr 1fr 1fr 1fr;
    grid-gap: 20px;
}
.item1    {
    background-color: red;    }
.item2    {
    background-color: blue;    }
.item3    {
    background-color: green;    }
.item4    {
    background-color: yellow;    }
.item5    {
    background-color: blueviolet;    }
.item6    {
    background-color: orangered;    }
.item7    {
    background-color: darkgreen;    }
.item8    {
    background-color: indigo;    }
</style>
</head>
<body>
    <div class="container">
        <div class="item item1">Lorem ipsum dolor sit amet consectetur adipisicing elit.
        Illo eius nisi excepturi aperiam voluptates laudantium ut reprehenderit iure earum
        a!</div>
        <div class="item item2">Menu</div>
        <div class="item item3">Box1</div>
        <div class="item item4">Box2</div>
        <div class="item item5">Sidebar</div>
        <div class="item item6">Content</div>
        <div class="item item7">Lorem ipsum dolor sit, amet consectetur adipisicing elit.
        Facere esse molestias ab itaque voluptatem eos consequatur. Quam nostrum possimus
        deserunt?</div>
        <div class="item item8">Extra</div>
    </div>
</body></html>
```

### Example -- with "min-content" property

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 800px;
      background-color: silver;
      margin: 45px auto;
      font-size: 40px;
      color: white;
      display: grid;
      grid-template-rows: repeat(2,minmax(100px,min-content));
      grid-template-columns: 1fr 1fr 1fr 1fr;
      grid-gap: 20px;
    }
    .item1 {
      background-color: red;
    }
    .item2 {
      background-color: blue;
    }
    .item3 {
      background-color: green;
    }
```

```
.item4    {
  background-color: yellow;
}
.item5    {
  background-color: blueviolet;
}
.item6    {
  background-color: orangered;
}
.item7    {
  background-color: darkgreen;
}
.item8    {
  background-color: indigo;
}
</style>
</head>
<body>
  <div class="container">
    <div class="item item1">Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Illo eius nisi excepturi aperiam voluptates laudantium ut reprehenderit iure earum
    a!</div>
    <div class="item item2">Menu</div>
    <div class="item item3">Box1</div>
    <div class="item item4">Box2</div>
    <div class="item item5">Sidebar</div>
    <div class="item item6">Content</div>
    <div class="item item7">Lorem ipsum dolor sit, amet consectetur adipisicing elit.
    Facere esse molestias ab itaque voluptatem eos consequatur. Quam nostrum possimus
    deserunt?</div>
    <div class="item item8">Extra</div>
  </div>

</body>
</html>
```

## In Case Of Columns

Example -- Non-Responsive cause parent element width static

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 1000px;
      background-color: silver;
      margin: 45px auto;
      font-size: 40px;
      color: white;
      display: grid;
      grid-template-rows: repeat(2,minmax(100px,400px));
      grid-template-columns: minmax(100px,500px) 1fr 1fr 1fr;
      grid-gap: 20px;
    }
    .item1 {
      background-color: red;
    }
    .item2 {
      background-color: blue;
    }
  </style>
</html>
```

```
.item3    {
  background-color: green;    }
.item4    {
  background-color: yellow;    }
.item5    {
  background-color: blueviolet;    }
.item6    {
  background-color: orangered;    }
.item7    {
  background-color: darkgreen;    }
.item8    {
  background-color: indigo;    }
</style>
</head>
<body>
  <div class="container">
    <div class="item item1">Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Illo eius nisi excepturi aperiam voluptates laudantium ut reprehenderit iure earum
    a!</div>
    <div class="item item2">Menu</div>
    <div class="item item3">Box1</div>
    <div class="item item4">Box2</div>
    <div class="item item5">Sidebar</div>
    <div class="item item6">Content</div>
    <div class="item item7">Lorem ipsum dolor sit, amet consectetur adipisicing elit.
    Facere esse molestias ab itaque voluptatem eos consequatur. Quam nostrum possimus
    deserunt?</div>
    <div class="item item8">Extra</div>
  </div>

</body>
</html>
```

### Example -- Responsive cause parent element width dynamic

In this example the child element(item1) respond according to the parent container but it squeezes upto the minimum range and expands upto the maximum range according to minmax() function.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 40px;
      color: white;
      display: grid;
      grid-template-rows: repeat(2,minmax(100px,400px));
      grid-template-columns: minmax(150px,500px) 1fr 1fr 1fr;
      grid-gap: 20px;
    }
    .item1 {
      background-color: red;
    }
    .item2 {
      background-color: blue;
    }
  </style>
</html>
```

```
.item3    {
  background-color: green;    }
.item4    {
  background-color: yellow;    }
.item5    {
  background-color: blueviolet;    }
.item6    {
  background-color: orangered;    }
.item7    {
  background-color: darkgreen;    }
.item8    {
  background-color: indigo;    }
</style>
</head>
<body>

<div class="container">
  <div class="item item1">Lorem ipsum dolor sit amet consectetur adipisicing elit.
  Illo eius nisi excepturi aperiam voluptates laudantium ut reprehenderit iure earum
  a!</div>
  <div class="item item2">Menu</div>
  <div class="item item3">Box1</div>
  <div class="item item4">Box2</div>
  <div class="item item5">Sidebar</div>
  <div class="item item6">Content</div>
  <div class="item item7">Lorem ipsum dolor sit, amet consectetur adipisicing elit.
  Facere esse molestias ab itaque voluptatem eos consequatur. Quam nostrum possimus
  deserunt?</div>
  <div class="item item8">Extra</div>
</div>

</body>
</html>
```

### Example -- With Fractional Unit

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    *
    {
      box-sizing: border-box;
    }
    html,body
    {
      margin: 0px;
      padding: 0px;
    }
    .container
    {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 40px;
      color: white;
      display: grid;
      grid-template-rows: repeat(2,minmax(100px,400px));
      grid-template-columns: minmax(150px,1fr) 1fr 1fr 1fr;
      grid-gap: 20px;
    }
    .item1    {
      background-color: red;
    }
    .item2    {
      background-color: blue;
    }
```



```
.item3    {
  background-color: green;    }
.item4    {
  background-color: yellow;    }
.item5    {
  background-color: blueviolet;    }
.item6    {
  background-color: orangered;    }
.item7    {
  background-color: darkgreen;    }
.item8    {
  background-color: indigo;    }
</style>
</head>
<body>

<div class="container">
  <div class="item item1">Lorem ipsum dolor sit amet consectetur adipisicing elit.
  Illo eius nisi excepturi aperiam voluptates laudantium ut reprehenderit iure earum
  a!</div>
  <div class="item item2">Menu</div>
  <div class="item item3">Box1</div>
  <div class="item item4">Box2</div>
  <div class="item item5">Sidebar</div>
  <div class="item item6">Content</div>
  <div class="item item7">Lorem ipsum dolor sit, amet consectetur adipisicing elit.
  Facere esse molestias ab itaque voluptatem eos consequatur. Quam nostrum possimus
  deserunt?</div>
  <div class="item item8">Extra</div>
</div>

</body>
</html>
```

## 6- Implicit Vs Explicit Grids

### Explicit Grid Elements --

- 1) Those grid elements whose dimensions are defined by user.
- 2) At the end of explicit grid items we have a "dark" line.

### Implicit Grid Elements --

- 1) Those grid elements whose dimensions are defined automatically.
- 2) At the end of Implicit grid items we have a "dotted" line.

In order to make Implicit grids, the CSS grid system provides us the following properties:-

- a) grid-auto-columns
- b) grid-auto-rows
- c) grid-auto-flows



## Example -- Explicit Grid Element

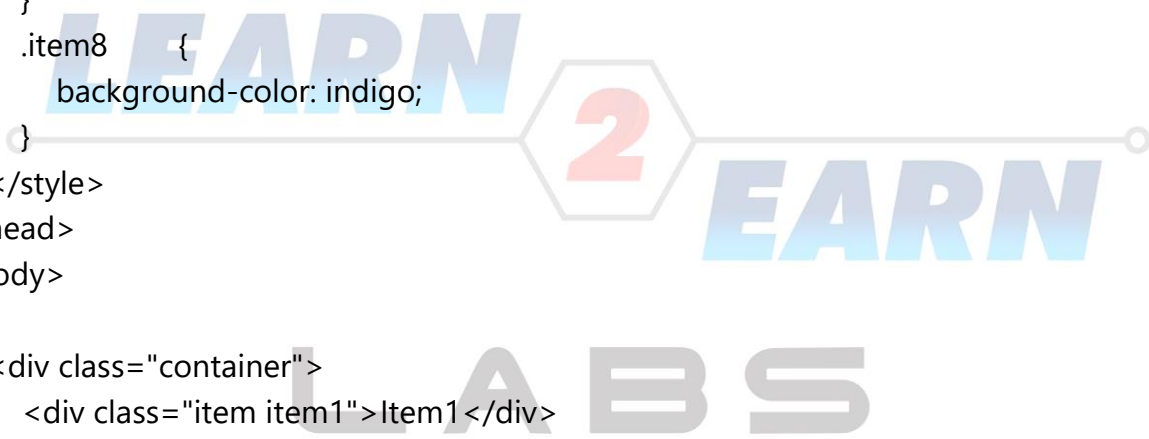
In this example Item1,Item2,Item3,Item4 following explicit grid system.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,1fr);
      grid-template-rows: repeat(2,120px);
    }
    .item1
    {
      background-color: red;
    }
    .item2
    {
      background-color: blue;
    }
```

```
.item3    {
  background-color: green;
}
.item4    {
  background-color: yellow;
}
.item5    {
  background-color: blueviolet;
}
.item6    {
  background-color: orangered;
}
.item7    {
  background-color: darkgreen;
}
.item8    {
  background-color: indigo;
}
</style>
</head>
<body>

<div class="container">
  <div class="item item1">Item1</div>
  <div class="item item2">Item2</div>
  <div class="item item3">Item3</div>
  <div class="item item4">Item4</div>
  <div class="item item5">Item5</div>
  <div class="item item6">Item6</div>
  <div class="item item7">Item7</div>
  <div class="item item8">Item8</div>
</div>

</body>
</html>
```

The logo for 'Learn 2 Earn Labs' is centered in the background. It features the word 'LEARN' in blue, a large red number '2' inside a hexagon, and the word 'EARN' in blue. Below this, the word 'LABS' is written in a large, grey, sans-serif font.

## Example -- Implicit Grid Elements

### a) grid-auto-rows

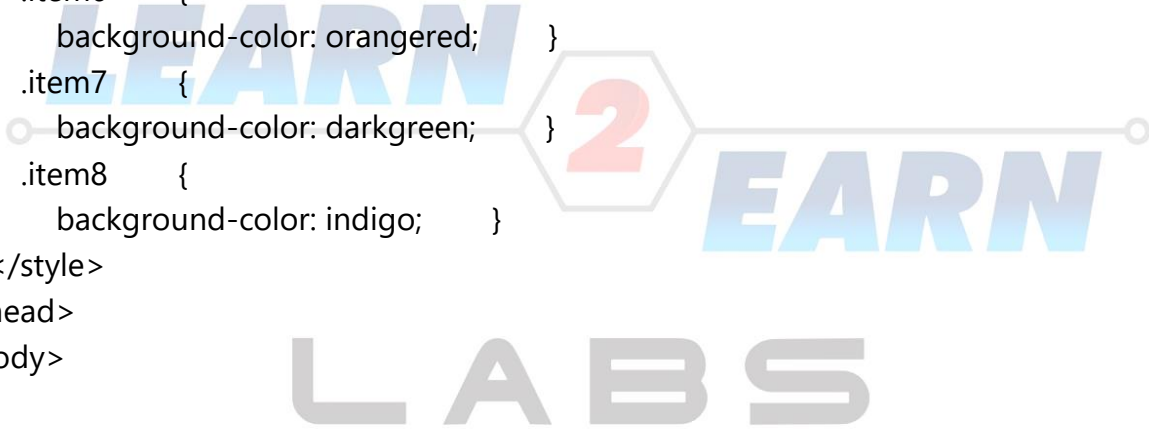
- Defines the size of grid rows that were created implicitly, it means that grid-auto-rows targets the rows that were not defined with grid-template-rows or grid-template-areas.
- The grid-auto-rows CSS property specifies the size of an implicitly-created grid row track or pattern of tracks.
- The properties come-in really handy when you want to be able to add items that will spill onto new rows/columns, but want to retain some control over the sizing of the implicitly created rows/columns.
- The default value of grid-auto-rows is "auto".

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,1fr);
```

```
grid-template-rows: repeat(2,120px);
grid-gap: 10px;
grid-auto-rows: 100px;
}
.item1    {
    background-color: red;    }
.item2    {
    background-color: blue;    }
.item3    {
    background-color: green;    }
.item4    {
    background-color: yellow;    }
.item5    {
    background-color: blueviolet;    }
.item6    {
    background-color: orangered;    }
.item7    {
    background-color: darkgreen;    }
.item8    {
    background-color: indigo;    }
</style>
</head>
<body>

<div class="container">
    <div class="item item1">Item1 </div>
    <div class="item item2">Item2 </div>
    <div class="item item3">Item3 </div>
    <div class="item item4">Item4 </div>
    <div class="item item5">Item5 </div>
    <div class="item item6">Item6 </div>
    <div class="item item7">Item7 </div>
    <div class="item item8">Item8 </div>
</div>
</body>
</html>
```

The logo for 'Learn 2 Earn Labs' is positioned in the background. It features the word 'LEARN' in blue, a large red number '2' inside a hexagon, the word 'EARN' in blue, and the word 'LABS' in large grey letters below. A horizontal line with circles at both ends passes through the '2'.

## b) grid-auto-flows

- The grid-auto-flow CSS property controls how the auto-placement algorithm works, specifying exactly how auto-placed items(rows,columns) get flowed into the grid.
- The grid-auto-flow property controls how auto-placed items get inserted in the grid.
- The default value of grid-auto-flows is "row".

### Example -- For Row

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,1fr);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 100px;
      grid-auto-flow: row;
    }
  </style>
</head>
<body>
  <div class="container">
    <div></div>
    <div></div>
    <div></div>
    <div></div>
  </div>
</body>
</html>
```

```
.item1    {
  background-color: red;    }
.item2    {
  background-color: blue;   }
.item3    {
  background-color: green;  }
.item4    {
  background-color: yellow; }
.item5    {
  background-color: blueviolet;  }
.item6    {
  background-color: orangered;    }
.item7    {
  background-color: darkgreen;    }
.item8    {
  background-color: indigo;    }
</style>
</head>
<body>

<div class="container">
  <div class="item item1">Item1 </div>
  <div class="item item2">Item2 </div>
  <div class="item item3">Item3 </div>
  <div class="item item4">Item4 </div>
  <div class="item item5">Item5 </div>
  <div class="item item6">Item6 </div>
  <div class="item item7">Item7 </div>
  <div class="item item8">Item8 </div>
</div>
</body>
</html>
```



### Example -- For Column

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,1fr);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 100px;
      grid-auto-flow: column;
    }
    .item1 {
      background-color: red;
    }
    .item2 {
      background-color: blue;
    }
    .item3 {
      background-color: green;
    }
```

```
.item4    {
    background-color: yellow;    }
.item5    {
    background-color: blueviolet;    }
.item6    {
    background-color: orangered;    }
.item7    {
    background-color: darkgreen;    }
.item8    {
    background-color: indigo;    }
</style>
</head>
<body>

<div class="container">
    <div class="item item1">Item1</div>
    <div class="item item2">Item2</div>
    <div class="item item3">Item3</div>
    <div class="item item4">Item4</div>
    <div class="item item5">Item5</div>
    <div class="item item6">Item6</div>
    <div class="item item7">Item7</div>
    <div class="item item8">Item8</div>
</div>

</body>
</html>
```

## The "dense" parameter

- This property will allow you to fill those empty spaces with grid items of the right size.
- The dense keyword places items on blank spaces of the grid, according to their size.

### Example -- Before "dense" property

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html, body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 700px;
      height: 700px;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2, 250px);
      grid-template-rows: repeat(2, 120px);
      grid-gap: 10px;
      grid-auto-rows: 70px;
      justify-content: center;
      align-content: center;
    }
```

```
.item1    {
  background-color: red;    }
.item2    {
  background-color: blue;   }
.item3    {
  background-color: green;  }
.item4    {
  background-color: yellow; }
.item5    {
  background-color: blueviolet;  }
.item6    {
  background-color: orangered;
  grid-column: span 2;    }
.item7    {
  background-color: darkgreen;  }
.item8    {
  background-color: indigo;    }
</style>
</head>
<body>

<div class="container">
  <div class="item item1">Item1</div>
  <div class="item item2">Item2</div>
  <div class="item item3">Item3</div>
  <div class="item item4">Item4</div>
  <div class="item item5">Item5</div>
  <div class="item item6">Item6</div>
  <div class="item item7">Item7</div>
  <div class="item item8">Item8</div>
</div>
</body>
</html>
```



LABS

### Example -- after "dense" property

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 700px;
      height: 700px;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,250px);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 70px;
      grid-auto-flow: row dense;
      justify-content: center;
      align-content: center;
    }
    .item1 {
      background-color: red;
    }
    .item2 {
      background-color: blue;
    }
  </style>
</html>
```

```
.item3    {
    background-color: green;    }
.item4    {
    background-color: yellow;    }
.item5    {
    background-color: blueviolet;    }
.item6    {
    background-color: orangered;
    grid-column: span 2;    }
.item7    {
    background-color: darkgreen;    }
.item8    {
    background-color: indigo;    }
</style>
</head>
<body>
    <div class="container">
        <div class="item item1">Item1</div>
        <div class="item item2">Item2</div>
        <div class="item item3">Item3</div>
        <div class="item item4">Item4</div>
        <div class="item item5">Item5</div>
        <div class="item item6">Item6</div>
        <div class="item item7">Item7</div>
        <div class="item item8">Item8</div>
    </div>
</body>
</html>
```

### c) grid-auto-column

- The grid-auto-columns property sets a size for the columns in a grid container. This property affects only columns with the size not set.
- The grid-auto-columns CSS property specifies the size of an implicitly-created grid column track or pattern of tracks.
- The grid-auto-columns property only deals with columns that are created implicitly.

#### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,1fr);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 100px;
      grid-auto-flow: column;
      grid-auto-columns: .5fr;
    }
```

```
.item1    {
  background-color: red;    }
.item2    {
  background-color: blue;   }
.item3    {
  background-color: green;  }
.item4    {
  background-color: yellow; }
.item5    {
  background-color: blueviolet;  }
.item6    {
  background-color: orangered;    }
.item7    {
  background-color: darkgreen;    }
.item8    {
  background-color: indigo;    }
</style>
</head>
<body>

<div class="container">
  <div class="item item1">Item1 </div>
  <div class="item item2">Item2 </div>
  <div class="item item3">Item3 </div>
  <div class="item item4">Item4 </div>
  <div class="item item5">Item5 </div>
  <div class="item item6">Item6 </div>
  <div class="item item7">Item7 </div>
  <div class="item item8">Item8 </div>
</div>
</body>
</html>
```



## 7- CSS Grid Items Alignment Properties

- In order to align grid items we have to follow CSS Grids "alignment properties".
- In Grid Layout, these properties controls the alignment of items on the Block Axis within their grid area.

We have six properties for grid items alignment that includes :-

- a) align-items
- b) justify-items
- c) place-items
- d) align-self
- e) justify-self
- f) place-self

### a) align-items & align-self

- The align-items & align-self property controls alignment of grid elements(in case of align-items)/element(in case of align-self) on the block axis(Vertically).
- This property provides vertical alignment to grid elements (in case of align-items)/element (in case of align-self).
- This property apply on the parent container(in case of align-items)/child container(in case of align-self) and have effect on child/individual container.

These property have following values :-

auto, normal, start, end, center, stretch, baseline.

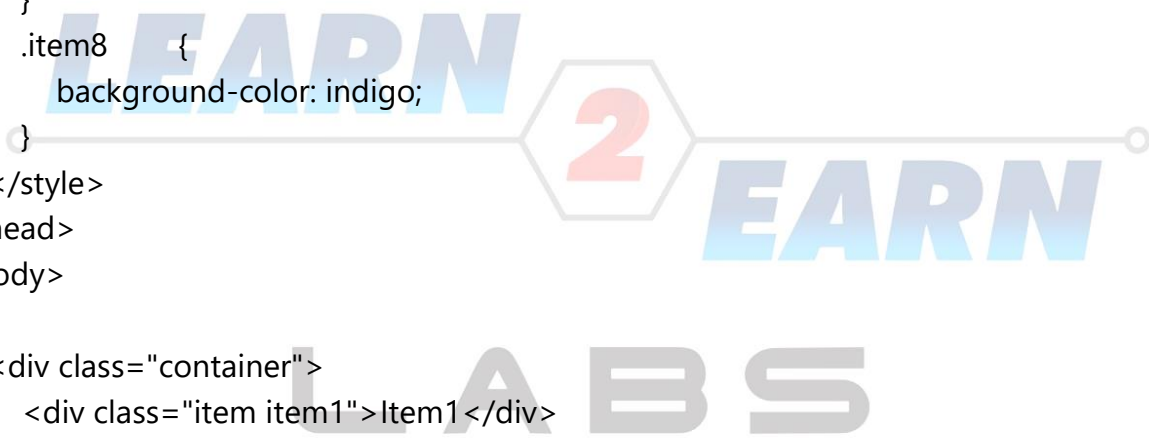
### Example -- align-items

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,1fr);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 70px;
      align-items: center;
    }
    .item1 {
      background-color: red;
    }
    .item2 {
      background-color: blue;
    }
  </style>
</html>
```

```
.item3    {
  background-color: green;
}
.item4    {
  background-color: yellow;
}
.item5    {
  background-color: blueviolet;
}
.item6    {
  background-color: orangered;
}
.item7    {
  background-color: darkgreen;
}
.item8    {
  background-color: indigo;
}
</style>
</head>
<body>

<div class="container">
  <div class="item item1">Item1</div>
  <div class="item item2">Item2</div>
  <div class="item item3">Item3</div>
  <div class="item item4">Item4</div>
  <div class="item item5">Item5</div>
  <div class="item item6">Item6</div>
  <div class="item item7">Item7</div>
  <div class="item item8">Item8</div>
</div>

</body>
</html>
```

The logo for 'Learn 2 Earn Labs' is centered in the background. It features the word 'LEARN' in blue, a large red number '2' inside a hexagon, and the word 'EARN' in blue. Below this, the word 'LABS' is written in a large, grey, sans-serif font.

### Example -- align-self

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    *
    {
      box-sizing: border-box;
    }
    html,body
    {
      margin: 0px;
      padding: 0px;
    }
    .container
    {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,1fr);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 70px;
    }
    .item1
    {
      background-color: red;
    }
    .item2
    {
      background-color: blue;
    }
  </style>
</html>
```

```
.item3    {
    background-color: green;
    align-self: center;
}
.item4    {
    background-color: yellow;
}
.item5    {
    background-color: blueviolet;
}
.item6    {
    background-color: orangered;
}
.item7    {
    background-color: darkgreen;
}
.item8    {
    background-color: indigo;
}
</style>
</head>
<body>
    <div class="container">
        <div class="item item1">Item1</div>
        <div class="item item2">Item2</div>
        <div class="item item3">Item3</div>
        <div class="item item4">Item4</div>
        <div class="item item5">Item5</div>
        <div class="item item6">Item6</div>
        <div class="item item7">Item7</div>
        <div class="item item8">Item8</div>
    </div>
</body>
</html>
```

## b) justify-items & justify-self

- The justify-items & justify-self property controls alignment of grid elements(in case of justify-items)/element(in case of justify-self) on the inline axis(Horizontally).
- This property provides horizontal alignment to grid elements(in case of justify-items)/element(in case of justify-self).
- This property apply on the parent container(in case of justify-items)/child container(in case of justify-self) and have effect on child/individual container.

These property have following values :-

auto, normal, start, end, center, stretch, baseline.



### Example -- justify-items

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,1fr);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 70px;
      justify-items: start;
    }
    .item1 {
      background-color: red;
    }
    .item2 {
      background-color: blue;
    }
  </style>
</html>
```

```
.item3    {
  background-color: green;
}
.item4    {
  background-color: yellow;
}
.item5    {
  background-color: blueviolet;
}
.item6    {
  background-color: orangered;
}
.item7    {
  background-color: darkgreen;
}
.item8    {
  background-color: indigo;
}
</style>
</head>
<body>
  <div class="container">
    <div class="item item1">Item1 </div>
    <div class="item item2">Item2 </div>
    <div class="item item3">Item3 </div>
    <div class="item item4">Item4 </div>
    <div class="item item5">Item5 </div>
    <div class="item item6">Item6 </div>
    <div class="item item7">Item7 </div>
    <div class="item item8">Item8 </div>
  </div>
</body>
</html>
```





## Example -- justify-self

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,1fr);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 70px;
    }
    .item1 {
      background-color: red;
    }
    .item2 {
      background-color: blue;
      justify-self: center;
    }
  </style>

```

```
.item3    {
  background-color: green;
}
.item4    {
  background-color: yellow;
}
.item5    {
  background-color: blueviolet;
}
.item6    {
  background-color: orangered;
}
.item7    {
  background-color: darkgreen;
}
.item8    {
  background-color: indigo;
}
</style>
</head>
<body>
  <div class="container">
    <div class="item item1">Item1 </div>
    <div class="item item2">Item2 </div>
    <div class="item item3">Item3 </div>
    <div class="item item4">Item4 </div>
    <div class="item item5">Item5 </div>
    <div class="item item6">Item6 </div>
    <div class="item item7">Item7 </div>
    <div class="item item8">Item8 </div>
  </div>
</body>
</html>
```



### c) place-items / place-self property

- The place-items property is shorthand for the the align-items and justify-items properties. Also this property should be applied on the parent container.
- The place-self property is shorthand for the the align-self and justify-self properties. Also this property should be applied on the child container.

#### Syntax -- place-items

place-items : <align-items> <justify-items>

#### Syntax -- place-self

place-self : <align-self> <justify-self>



### Example -- place-items

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,1fr);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 70px;
      place-items: start center;
    }
    .item1 {
      background-color: red;
    }
    .item2 {
      background-color: blue;
    }
    .item3 {
      background-color: green;
    }
```

```
.item4    {
    background-color: yellow;
}
.item5    {
    background-color: blueviolet;
}
.item6    {
    background-color: orangered;
}
.item7    {
    background-color: darkgreen;
}
.item8    {
    background-color: indigo;
}
</style>
</head>
<body>

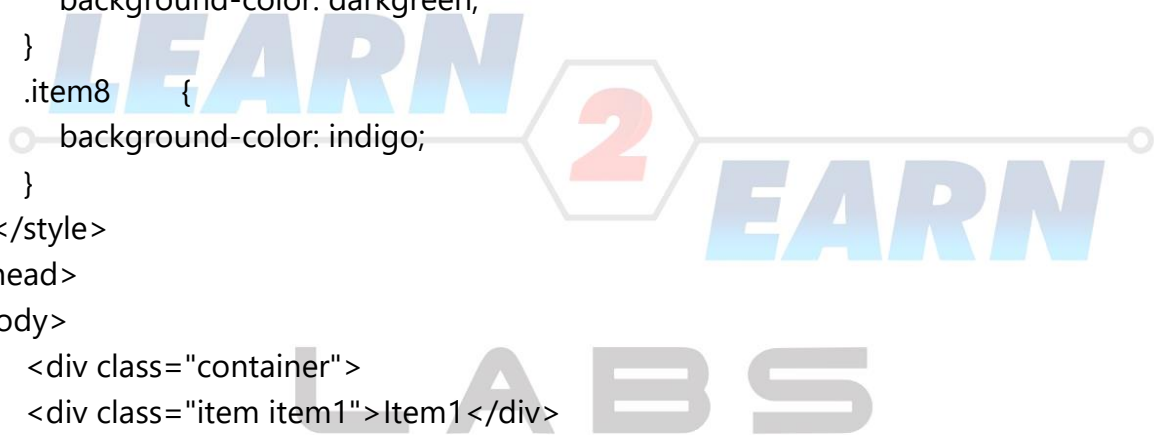
<div class="container">
    <div class="item item1">Item1</div>
    <div class="item item2">Item2</div>
    <div class="item item3">Item3</div>
    <div class="item item4">Item4</div>
    <div class="item item5">Item5</div>
    <div class="item item6">Item6</div>
    <div class="item item7">Item7</div>
    <div class="item item8">Item8</div>
</div>

</body>
</html>
```

### Example -- place-self

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    *
    {
      box-sizing: border-box;
    }
    html,body
    {
      margin: 0px;
      padding: 0px;
    }
    .container
    {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,1fr);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 70px;
    }
    .item1
    {
      background-color: red;
    }
    .item2
    {
      background-color: blue;
    }
  </style>
</html>
```

```
.item3    {
  background-color: green;
}
.item4    {
  background-color: yellow;
  place-self:center start;
}
.item5    {
  background-color: blueviolet;
}
.item6    {
  background-color: orangered;
}
.item7    {
  background-color: darkgreen;
}
.item8    {
  background-color: indigo;
}
</style>
</head>
<body>
  <div class="container">
    <div class="item item1">Item1</div>
    <div class="item item2">Item2</div>
    <div class="item item3">Item3</div>
    <div class="item item4">Item4</div>
    <div class="item item5">Item5</div>
    <div class="item item6">Item6</div>
    <div class="item item7">Item7</div>
    <div class="item item8">Item8</div>
  </div>
</body>
</html>
```



## 8 - Grids Track Alignment Properties

We have three properties for grid tracks alignment that includes :-

### a) align-content

- This property works for vertical alignment.
- This property have values start, end, center, stretch, space-around, space-between, space-evenly.

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      height: 700px;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,150px);
      grid-template-rows: repeat(2,120px);
```



```
    grid-gap: 10px;
    grid-auto-rows: 70px;
    align-content: center;    }
.item1    {
    background-color: red;    }
.item2    {
    background-color: blue;    }
.item3    {
    background-color: green;    }
.item4    {
    background-color: yellow;    }
.item5    {
    background-color: blueviolet;    }
.item6    {
    background-color: orangered;    }
.item7    {
    background-color: darkgreen;    }
.item8    {
    background-color: indigo;    }
</style>
</head>
<body>
    <div class="container">
    <div class="item item1">Item1</div>
    <div class="item item2">Item2</div>
    <div class="item item3">Item3</div>
    <div class="item item4">Item4</div>
    <div class="item item5">Item5</div>
    <div class="item item6">Item6</div>
    <div class="item item7">Item7</div>
    <div class="item item8">Item8</div>
    </div>
</body>
</html>
```

### Example -- space-evenly

This property provides equal space between & around grid elements.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      height: 700px;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,150px);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 70px;
      align-content: space-evenly;
    }
    .item1
    {
      background-color: red;
    }
```

```
.item2    {
  background-color: blue;
}
.item3    {
  background-color: green;
}
.item4    {
  background-color: yellow;
}
.item5    {
  background-color: blueviolet;
}
.item6    {
  background-color: orangered;
}
.item7    {
  background-color: darkgreen;
}
.item8    {
  background-color: indigo;
}
</style>
</head>
<body>
  <div class="container">
    <div class="item item1">Item1 </div>
    <div class="item item2">Item2 </div>
    <div class="item item3">Item3 </div>
    <div class="item item4">Item4 </div>
    <div class="item item5">Item5 </div>
    <div class="item item6">Item6 </div>
    <div class="item item7">Item7 </div>
    <div class="item item8">Item8 </div>
  </div>
</body>
</html>
```



## b) justify-content

- This property works for horizontal alignment.
- This property has values start, end, center, stretch, space-around, space-between, space-evenly.

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html, body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      height: 700px;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2, 150px);
      grid-template-rows: repeat(2, 120px);
      grid-gap: 10px;
      grid-auto-rows: 70px;
      justify-content: center;
    }
  </style>
</html>
```

```
.item1    {
  background-color: red;    }
.item2    {
  background-color: blue;   }
.item3    {
  background-color: green;  }
.item4    {
  background-color: yellow; }
.item5    {
  background-color: blueviolet;  }
.item6    {
  background-color: orangered;   }
.item7    {
  background-color: darkgreen;   }
.item8    {
  background-color: indigo;      }
</style>
</head>
<body>

<div class="container">
  <div class="item item1">Item1</div>
  <div class="item item2">Item2</div>
  <div class="item item3">Item3</div>
  <div class="item item4">Item4</div>
  <div class="item item5">Item5</div>
  <div class="item item6">Item6</div>
  <div class="item item7">Item7</div>
  <div class="item item8">Item8</div>
</div>

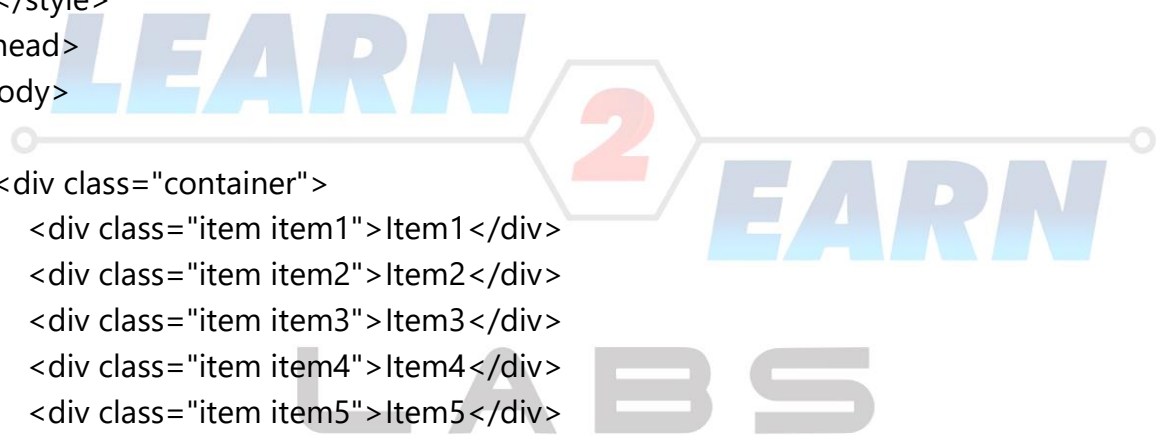
</body>
</html>
```

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    *
    {
      box-sizing: border-box;
    }
    html,body
    {
      margin: 0px;
      padding: 0px;
    }
    .container
    {
      width: 80%;
      height: 700px;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,150px);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 70px;
      justify-content: space-between;
    }
    .item1
    {
      background-color: red;
    }
  </style>
</html>
```

```
.item2    {
    background-color: blue;    }
.item3    {
    background-color: green;    }
.item4    {
    background-color: yellow;    }
.item5    {
    background-color: blueviolet;    }
.item6    {
    background-color: orangered;    }
.item7    {
    background-color: darkgreen;    }
.item8    {
    background-color: indigo;    }
</style>
</head>
<body>
<div class="container">
    <div class="item item1">Item1</div>
    <div class="item item2">Item2</div>
    <div class="item item3">Item3</div>
    <div class="item item4">Item4</div>
    <div class="item item5">Item5</div>
    <div class="item item6">Item6</div>
    <div class="item item7">Item7</div>
    <div class="item item8">Item8</div>
</div>

</body>
</html>
```



## Example

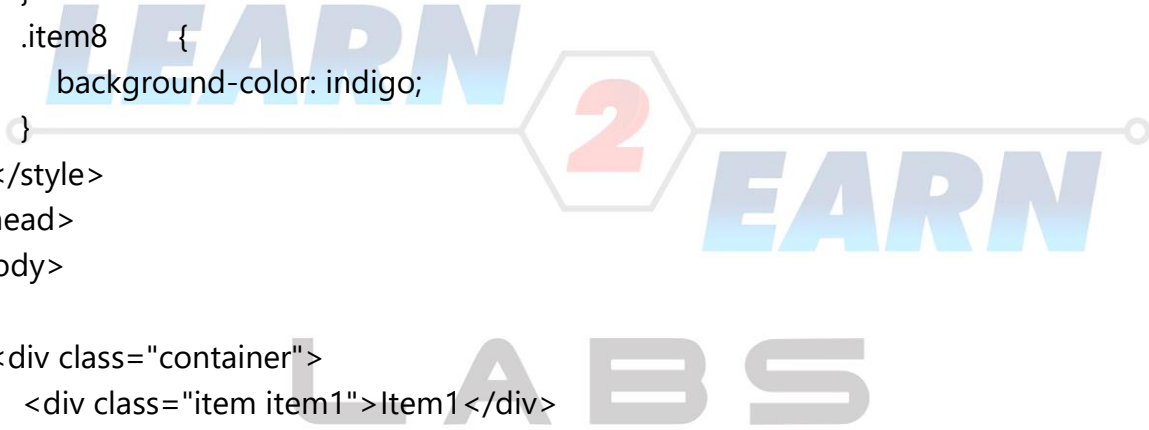
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      height: 700px;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,150px);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 70px;
      justify-content: space-around;
    }
    .item1 {
      background-color: red;
    }
    .item2 {
      background-color: blue;
    }
  </style>
</html>
```



```
.item3    {
  background-color: green;
}
.item4    {
  background-color: yellow;
}
.item5    {
  background-color: blueviolet;
}
.item6    {
  background-color: orangered;
}
.item7    {
  background-color: darkgreen;
}
.item8    {
  background-color: indigo;
}
</style>
</head>
<body>

<div class="container">
  <div class="item item1">Item1</div>
  <div class="item item2">Item2</div>
  <div class="item item3">Item3</div>
  <div class="item item4">Item4</div>
  <div class="item item5">Item5</div>
  <div class="item item6">Item6</div>
  <div class="item item7">Item7</div>
  <div class="item item8">Item8</div>
</div>

</body>
</html>
```

The logo for Learn 2 Earn Labs is centered in the background. It features the word "LEARN" in blue, a large red number "2" inside a hexagon, and the word "EARN" in blue. Below this, the word "LABS" is written in a large, grey, sans-serif font.

### Example -- space-evenly

This property provides equal space between & around grid elements.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      height: 700px;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,150px);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 70px;
      justify-content: space-evenly;
    }
    .item1
    {
      background-color: red;
    }
```

```
.item2    {
    background-color: blue;    }
.item3    {
    background-color: green;    }
.item4    {
    background-color: yellow;    }
.item5    {
    background-color: blueviolet;    }
.item6    {
    background-color: orangered;    }
.item7    {
    background-color: darkgreen;    }
.item8    {
    background-color: indigo;    }
</style>
</head>
<body>
<div class="container">
    <div class="item item1">Item1</div>
    <div class="item item2">Item2</div>
    <div class="item item3">Item3</div>
    <div class="item item4">Item4</div>
    <div class="item item5">Item5</div>
    <div class="item item6">Item6</div>
    <div class="item item7">Item7</div>
    <div class="item item8">Item8</div>
</div>

</body>
</html>
```

### c) place-content

This property is the shorthand property of <align-content>/<justify-content>.

#### Syntax

place-content : <align-content>/<justify-content>

#### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 700px;
      height: 700px;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,250px);
      grid-template-rows: repeat(2,120px);
      grid-gap: 10px;
      grid-auto-rows: 70px;
      place-content: center center;
    }
```

```
.item1    {
  background-color: red;    }
.item2    {
  background-color: blue;   }
.item3    {
  background-color: green;  }
.item4    {
  background-color: yellow; }
.item5    {
  background-color: blueviolet;  }
.item6    {
  background-color: orangered;   }
.item7    {
  background-color: darkgreen;   }
.item8    {
  background-color: indigo;      }
</style>
</head>
<body>

<div class="container">
  <div class="item item1">Item1</div>
  <div class="item item2">Item2</div>
  <div class="item item3">Item3</div>
  <div class="item item4">Item4</div>
  <div class="item item5">Item5</div>
  <div class="item item6">Item6</div>
  <div class="item item7">Item7</div>
  <div class="item item8">Item8</div>
</div>

</body>
</html>
```

## 9 - CSS Grid "auto-fill" & "auto-fit" Values

- These keywords tell the browser to handle the column sizing and element wrapping for us, so that the elements will wrap into rows when the width is not large enough to fit them in without any overflow.
- To achieve wrapping, we can use the auto-fit or auto-fill keywords.

### auto-fill value

- auto-fill FILLS the row with as many columns as it can fit. So it creates implicit columns whenever a new column can fit, because it's trying to FILL the row with as many columns as it can. The newly added columns can and may be empty, but they will still occupy a designated space in the row.
- This value creates an extra blank columns if all columns/elements of grid would be fit inside of the grid container and leaves an empty space.
- Check the difference using grid-lines(dev tools).

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    *
    {
      box-sizing: border-box;
    }
    html,body
    {
      margin: 0px;
      padding: 0px;
    }
  </style>
</head>
<body>
```

```
.container    {
  width: 700px;
  background-color: silver;
  margin: 45px auto;
  font-size: 30px;
  color: white;
  display: grid;
  grid-template-columns: repeat(auto-fill,100px);
  grid-template-rows: repeat(2,150px);
}
.item1      {
  background-color: red;
}
.item2      {
  background-color: blue;
}
.item3      {
  background-color: green;
}
.item4      {
  background-color: yellow;
}
</style>
</head>
<body>
```

```
<div class="container">
  <div class="item item1">Item1 </div>
  <div class="item item2">Item2 </div>
  <div class="item item3">Item3 </div>
  <div class="item item4">Item4 </div>
</div>
```

```
</body>
</html>
```

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    *
    {
      box-sizing: border-box;
    }
    html,body
    {
      margin: 0px;
      padding: 0px;
    }
    .container
    {
      width: 700px;
      background-color: silver;
      margin: 45px auto;
      font-size: 30px;
      color: white;
      display: grid;
      grid-template-columns: repeat(auto-fill,100px);
      grid-template-rows: repeat(2,150px);
    }
    .item1
    {
      background-color: red;
    }
    .item2
    {
      background-color: blue;
    }
    .item3
    {
      background-color: green;
    }
```



```
.item4    {  
    background-color: yellow;  
    grid-column-end: -1;  
}  
</style>  
</head>  
<body>  
  
<div class="container">  
    <div class="item item1">Item1 </div>  
    <div class="item item2">Item2 </div>  
    <div class="item item3">Item3 </div>  
    <div class="item item4">Item4 </div>  
</div>  
  
</body>  
</html>
```



## auto-fit value

- "auto-fit" FITS the CURRENTLY AVAILABLE columns into the space by expanding them so that they take up any available space. The browser does that after FILLING that extra space with extra columns (as with auto-fill ) and then collapsing the empty ones.
- Doesn't creates an extra track.
- Check the difference using grid-lines(dev tools).

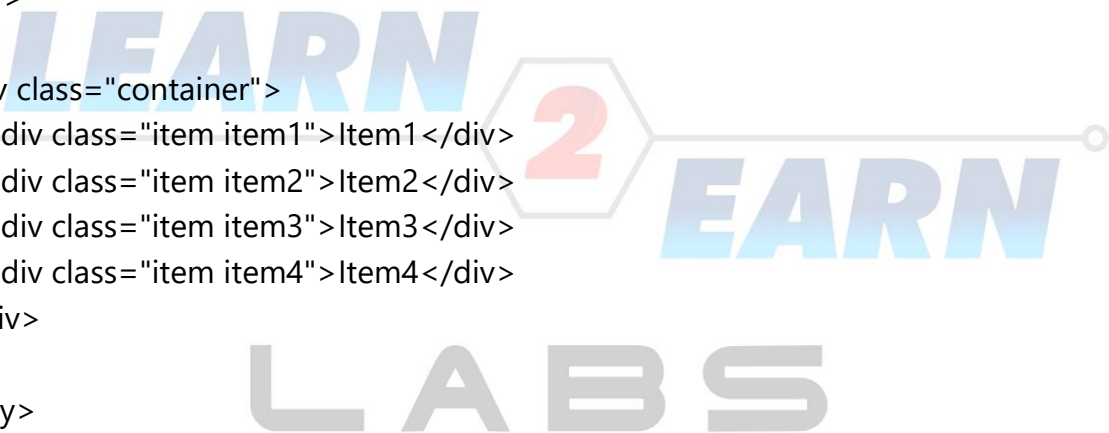
## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container
    {
      width: 700px;
      background-color: silver;
      margin: 45px auto;
      font-size: 30px;
      color: white;
      display: grid;
      grid-template-columns: repeat(auto-fit,100px);
      grid-template-rows: repeat(2,150px);
    }
```

```
.item1    {
  background-color: red;
}
.item2    {
  background-color: blue;
}
.item3    {
  background-color: green;
}
.item4    {
  background-color: yellow;
}
</style>
</head>
<body>

<div class="container">
  <div class="item item1">Item1</div>
  <div class="item item2">Item2</div>
  <div class="item item3">Item3</div>
  <div class="item item4">Item4</div>
</div>

</body>
</html>
```



## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    *
    {
      box-sizing: border-box;
    }
    html,body
    {
      margin: 0px;
      padding: 0px;
    }
    .container
    {
      width: 700px;
      background-color: silver;
      margin: 45px auto;
      font-size: 30px;
      color: white;
      display: grid;
      grid-template-columns: repeat(auto-fit,100px);
      grid-template-rows: repeat(2,150px);
    }
    .item1
    {
      background-color: red;
    }
    .item2
    {
      background-color: blue;
    }
    .item3
    {
      background-color: green;
    }
```

```
.item4    {
    background-color: yellow;
    grid-column-end: -1;
}
</style>
</head>
<body>

<div class="container">
    <div class="item item1">Item1 </div>
    <div class="item item2">Item2 </div>
    <div class="item item3">Item3 </div>
    <div class="item item4">Item4 </div>
</div>

</body>
</html>
```



## Dynamic Grids

In order to make our grid arrangement dynamic we should use "minmax()" function along with "auto-fill" & "auto-fit" property.

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 30px;
      color: white;
      display: grid;
      grid-template-columns: repeat(auto-fit,minmax(200px,1fr));
      grid-template-rows: repeat(2,150px);
      grid-auto-rows: 150px;
    }
    .item1
    {
      background-color: red;
    }
  </style>

```

```
.item2    {
    background-color: blue;
}
.item3    {
    background-color: green;
}
.item4    {
    background-color: yellow;
}
</style>
</head>
<body>

<div class="container">
    <div class="item item1">Item1</div>
    <div class="item item2">Item2</div>
    <div class="item item3">Item3</div>
    <div class="item item4">Item4</div>
</div>

</body>
</html>
```

LABS

## 10 - The "fit-content" property

- The CSS fit-content property is an inbuilt property in CSS. This property is used to adjust the size according to this formula  $\min(\text{maximum size}, \max(\text{minimum size}, \text{argument}))$ . The fit-content() property use to defined function to put a limit on the maximum size of the division.
- Generally used in masonry grids / gallery.

### Example -- Simple Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container
    {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 30px;
      color: white;
      display: grid;
      grid-template-columns: fit-content(100px) fit-content(200px) 1fr;
      grid-template-rows: repeat(2,150px);
      grid-auto-rows: 150px;
    }
```



```
.item1    {
  background-color: red;
}
.item2    {
  background-color: blue;
}
.item3    {
  background-color: green;
}
.item4    {
  background-color: yellow;
}
.item6    {
  background-color: orange;
}
.item5    {
  background-color: blueviolet;
}
</style>
</head>
<body>

<div class="container">
  <div class="item item1">Lorem ipsum dolor sit amet consectetur, adipisicing elit.
  Nobis voluptate quia non modi eaque tempora ipsam harum nostrum sint
  quisquam? </div>
  <div class="item item2">Lorem, ipsum dolor sit amet consectetur adipisicing elit.
  Ad ipsum et, officiis optio tempore quos vel sapiente? Culpa, voluptas vitae.</div>
  <div class="item item3">Item3</div>
  <div class="item item4">Item4</div>
  <div class="item item5">Item5</div>
  <div class="item item6">Item6</div>
</div>
</body>
</html>
```

### Example -- With minmax()

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 30px;
      color: white;
      display: grid;
      grid-template-columns: fit-content(100px) fit-content(200px) 1fr;
      grid-template-rows: repeat(2,minmax(100px,min-content));
      grid-auto-rows: 150px;
    }
    .item1
    {
      background-color: red;
    }
    .item2 {
      background-color: blue;
    }
    .item3 {
      background-color: green;
    }
```

```
.item4    {
    background-color: yellow;
}
.item5    {
    background-color: orange;
}
.item6    {
    background-color: blueviolet;
}
</style>
</head>
<body>

    <div class="container">
        <div class="item item1">Lorem ipsum dolor sit amet consectetur, adipisicing elit.
        Nobis voluptate quia non modi eaque tempora ipsam harum nostrum sint
        quisquam?</div>
        <div class="item item2">Lorem, ipsum dolor sit amet consectetur adipisicing elit.
        Ad ipsum et, officiis optio tempore quos vel sapiente? Culpa, voluptas vitae.</div>
        <div class="item item3">Item3</div>
        <div class="item item4">Item4</div>
        <div class="item item5">Item5</div>
        <div class="item item6">Item6</div>
    </div>

</body>
</html>
```

## 11) Ordering Grid Items

- The order CSS property sets the order to layout an item in a flex or grid container. Items in a container are sorted by ascending order value and then by their source code order.
- The order property can be used to specify the order in which different items should be placed inside a grid.
- By default, every grid element have an order property 0.

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html,body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 30px;
      color: white;
      display: grid;
      grid-template-columns: repeat(3,1fr);
      grid-template-rows: repeat(2,200px);
      grid-auto-rows: 150px;
    }
```

```
.item1    {
  background-color: red;
  order: 1;
}
.item2    {
  background-color: blue;
}
.item3    {
  background-color: green;
}
.item4    {
  background-color: yellow;
}

.item6    {
  background-color: orange;
}
.item5    {
  background-color: blueviolet;
}
</style>
</head>
<body>
  <div class="container">
    <div class="item item1">Item1</div>
    <div class="item item2">Item2</div>
    <div class="item item3">Item3</div>
    <div class="item item4">Item4</div>
    <div class="item item5">Item5</div>
    <div class="item item6">Item6</div>
  </div>
</body>
</html>
```

## 12) Nested Grids

- In order to make nested grids make sure we have use "display:grid" with the parent div / grid container.
- These nested grids elements have their own grid lines.

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    *
    {
      box-sizing: border-box;
    }
    html,body
    {
      margin: 0px;
      padding: 0px;
    }
    .container
    {
      width: 80%;
      background-color: silver;
      margin: 45px auto;
      font-size: 30px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,1fr);
      grid-template-rows: repeat(2,200px);
      grid-auto-rows: 150px;
      grid-gap: 20px;
    }
```

```
.item1    {
  background-color: red;
  order: 1;    }
.item2    {
  background-color: blue;    }
.item3    {
  background-color: green;
  display: grid;
  font-size: 20px;
  grid-template-columns: repeat(3,1fr);
  grid-template-rows: repeat(1,1fr);
  grid-gap: 20px;
  padding: 20px;    }
.item4    {
  background-color: yellow;    }
.child1   {
  background-color: cadetblue;    }
.child2   {
  background-color: crimson;    }
.child3   {
  background-color:black ;    }
</style>
</head>
<body>
  <div class="container">
    <div class="item item1">Item1 </div>
    <div class="item item2">Item2 </div>
    <div class="item item3">
      <div class="child child1">Child1 </div>
      <div class="child child2">Child2 </div>
      <div class="child child3">Child3 </div>
    </div>
    <div class="item item4">Item4 </div>
  </div>
</body>
</html>
```

### 13) Overlapping Grid Items

- When grid elements overlap one another then this is the case of Overlapping grid items.
- We should use "z-index" property to change the stack order of elements.

#### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      box-sizing: border-box;
    }
    html, body {
      margin: 0px;
      padding: 0px;
    }
    .container {
      width: 300px;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2, 150px);
      grid-template-rows: repeat(2, 150px);
      grid-auto-rows: 150px;
    }
    .item {
      width: 150px;
```



```
}  
.item1    {  
  background-color: red;  
  grid-column: 1/span 2;  
  grid-row: 1/span 2;  
  align-self: start;    }  
.item2    {  
  background-color: blue;  
  grid-column: 1/span 2;  
  grid-row: 1/span 2;  
  align-self: center;  
  justify-self:center;    }  
.item3    {  
  background-color: green;  
  grid-column: 1/span 2;  
  grid-row: 1/span 2;  
  align-self: end;  
  justify-self:end;    }  
  
</style>  
</head>  
<body>  
  
  <div class="container">  
    <div class="item item1">Item1 Lorem ipsum dolor sit amet consectetur adipisicing  
elit. Quae, laborum? </div>  
    <div class="item item2">Item2 Lorem ipsum dolor sit amet consectetur adipisicing  
elit. Quae, laborum? </div>  
    <div class="item item3">Item3 Lorem ipsum dolor sit amet consectetur adipisicing  
elit. Quae, laborum? </div>  
  </div>  
  
</body>  
</html>
```

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    *
    {
      box-sizing: border-box;
    }
    html,body
    {
      margin: 0px;
      padding: 0px;
    }
    .container
    {
      width: 300px;
      background-color: silver;
      margin: 45px auto;
      font-size: 20px;
      color: white;
      display: grid;
      grid-template-columns: repeat(2,150px);
      grid-template-rows: repeat(2,150px);
      grid-auto-rows: 150px;
    }
    .item
    {
      width: 150px;
    }
    .item1
    {
      background-color: red;
      grid-column: 1/span 2;
      grid-row: 1/span 2;
      align-self: start;
      z-index: 1;
    }
```

```
.item2    {
    background-color: blue;
    grid-column: 1/span 2;
    grid-row: 1/span 2;
    align-self: center;
    justify-self:center;
}
.item3    {
    background-color: green;
    grid-column: 1/span 2;
    grid-row: 1/span 2;
    align-self: end;
    justify-self:end;
}
</style>
</head>
<body>

<div class="container">
    <div class="item item1">Item1 Lorem ipsum dolor sit amet consectetur adipisicing
elit. Quae, laborum? </div>
    <div class="item item2">Item2 Lorem ipsum dolor sit amet consectetur adipisicing
elit. Quae, laborum? </div>
    <div class="item item3">Item3 Lorem ipsum dolor sit amet consectetur adipisicing
elit. Quae, laborum? </div>
</div>

</body>
</html>
```