

Grid in CSS

Assignment

Q1. Create an image gallery using a CSS grid.

Ans:- <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Grid</title>

<link rel="stylesheet" href="./style.css">

</head>

<body>

<div class="gallery">

<div class="box col2" style="background-image: url(https://images.unsplash.com/photo-1673894019491-4c569a62fdca?ixlib=rb-4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=435&q=80);"></div>

<div class="box" style="background-image: url(https://images.unsplash.com/photo-1674072311078-35d03d1104be?ixlib=rb-4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=1332&q=80);"></div>

<div class="box row2" style="background-image: url(https://images.unsplash.com/photo-1674068331911-555750338b11?ixlib=rb-

4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=1074&q=80);"></div>

<div class="box" style="background-image: url(https://images.unsplash.com/photo-1673754472479-84c988b6386c?ixlib=rb-4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=387&q=80);"></div>

<div class="box row2" style="background-image: url(https://images.unsplash.com/photo-1675068711500-537d82cb4463?ixlib=rb-4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=387&q=80);"></div>

<div class="box" style="background-image: url(https://images.unsplash.com/photo-1675124516483-b7d21f60e6d9?ixlib=rb-4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=464&q=80);"></div>

</div>

</body>

</html>

Style.css

```
*{
    margin: 0;
    box-sizing: border-box;
    padding: 0;
}

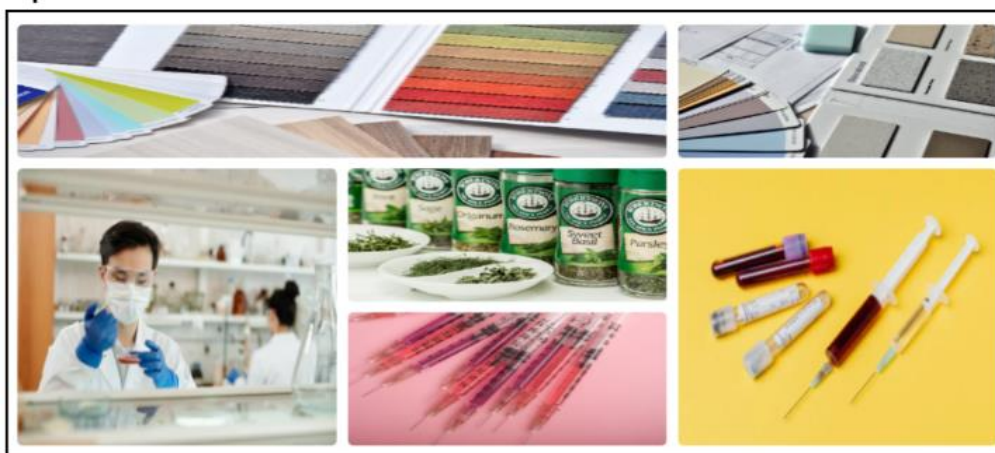
.gallery{
    margin: 75px auto;
    display: grid; width: 100%; max-width: 1000px;
```

```

grid-template-columns: repeat(3, 1fr);
grid-auto-rows: 200px;
gap: 10px;
}
.box{
border-radius: 10px;
background-size: cover;
background-position: bottom;
}
.row2{
grid-row: span 2;
}
.col2{
grid-column: span 2;
}

```

Output:-



Q2. Write code to arrange containers with texts A, B, C, and D as shown in the below image.

Ans:- <!DOCTYPE html>

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Grid</title>
```

```
  <link rel="stylesheet" href="style.css">
```

```
</head>
```

```
<body>
```

```
  <div class="grid-class">
```

```
    <div class="grid-item">A</div>
```

```
    <div class="grid-item">B</div>
```

```
    <div class="grid-item">C</div>
```

```
    <div class="grid-item">D</div>
```

```
  </div>
```

```
</body>
```

```
</html>
```

Style.css:

```
* {
```

```
  margin: 0;
```

```
  box-sizing: border-box;
```

```
  padding: 0;
```

```
}
```

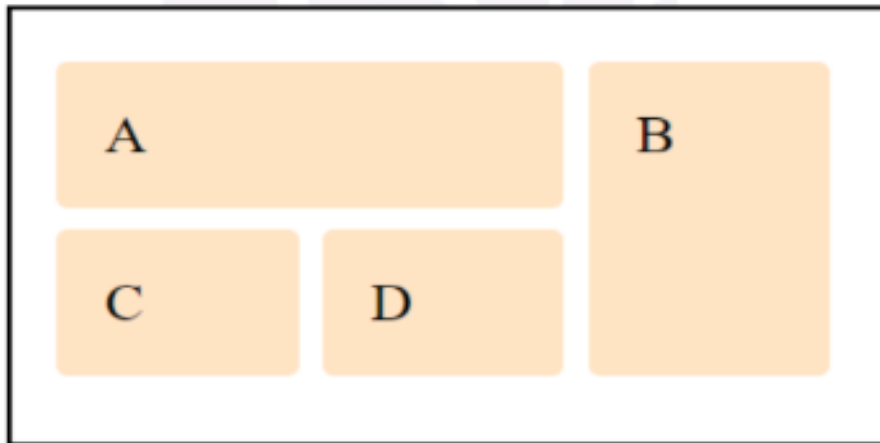
```
.grid-class{  
    width: 100%;  
    max-width: 800px;  
    margin: 100px auto;  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    grid-auto-rows: 180px;  
    gap: 10px;  
}
```

```
.grid-item{  
    padding: 20px;  
    background-color: rgb(220, 178, 218);  
    border-radius: 10px;  
}
```

```
.grid-class :first-child{  
    grid-column: span 2;  
}
```

```
.grid-class :nth-child(2){  
    grid-row: span 2;  
}
```

Output:



Q3.Explain the use of grid-auto-row and grid-auto-column using code examples.

Ans:- <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>grid-auto</title>

<link rel="stylesheet" href="./style.css">

</head>

<body>

<div class="grid-container">

<div class="grid-item">Item 1</div>

<div class="grid-item">Item 2</div>

<div class="grid-item">Item 3</div>

<div class="grid-item">Item 4</div>

</div>

```
</body>
```

```
</html>
```

Style.css

```
*{
```

```
  margin: 0;
```

```
  box-sizing: border-box;
```

```
  padding: 0;
```

```
}
```

```
.grid-container {
```

```
  display: grid;
```

```
  gap: 10px;
```

```
  grid-template-columns: 1fr 1fr;
```

```
  grid-template-rows: 1fr 1fr;
```

```
  grid-auto-rows: 100px; /* All implicitly created rows will be 100px  
tall */
```

```
  grid-auto-columns: 150px; /* All implicitly created columns will be  
150px wide */
```

```
}
```

```
.grid-item {
```

```
  background-color: lightblue;
```

```
  border: 1px solid gray;
```

```
  border-radius: 4px;
```

```
}
```

Q4. Write CSS to show numbers as shown in the figure, without altering the below html code.

Ans:- <!DOCTYPE html>

<html lang="en">

<head>

 <meta charset="UTF-8">

 <meta name="viewport" content="width=device-width, initial-scale=1.0">

 <title>nth-evenly</title>

 <link rel="stylesheet" href="style.css">

</head>

<body>

 <div class="container">

 <div class="box box1">1</div>

 <div class="box box2">2</div>

 <div class="box box3">3</div>

 <div class="box box4">4</div>

 <div class="box box5">5</div>

 <div class="box box6">6</div>

 <div class="box box7">7</div>

 <div class="box box8">8</div>

 </div>

</body>

</html>

Style.css:-

```
*{  
    margin: 0;  
    box-sizing: border-box;  
    padding: 0;  
}  
  
.container{  
    margin: 100px auto;  
    border: 2px solid black;  
    width: 700px;  
    display: flex;  
    flex-wrap: wrap;  
    gap: 10px;  
    padding: 20px 0 20px 20px;  
}  
  
.box{  
    width: 100px;  
    padding: 20px;  
    background-color: rgb(202, 201, 201);  
    border-radius: 10px;  
}
```

```
.container :nth-child(odd){  
    background-color: rgb(81, 81, 81);  
    color: #ffffff;  
}
```

```
.box1 {  
    order: 1;  
}
```

```
.box2 {  
    order: 2;  
}
```

```
.box3 {  
    order: -1;  
}
```

Q5. Explain the difference between justify-items and justify-self using code examples.

Ans:- <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>justify-content vs justify-self</title>

```
<link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="grid-container">
    <div class="grid-item">Item 1</div>
    <div class="grid-item item2">Item 2 (aligned to the end)</div>
  </div>
</body>
</html>
```

Style.css

```
.grid-container {
  display: grid;
  grid-template-columns: 1fr 1fr;
  grid-gap: 10px;
  justify-items: center; /* Default horizontal alignment for all grid
items */
}

.grid-item {
  background-color: lightblue;
  padding: 10px;
  border: 1px solid blue;
}

.item2 {
```

```
    justify-self: end; /* Override the default alignment for this specific
item */
}
```

COMPLETE