1. Problem Statement

Create an image gallery using a CSS grid.

Solution:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Image Gallary</title>

<style>

.container{

display: grid;

grid-template-columns: repeat(3, 1fr);

grid-template-rows: repeat(3, 250px);

gap: 20px;

padding: 20px;

}

.image1{

grid-column-start: 1;

grid-column-end: 3;

grid-row-start: 1;

grid-row-end: 2;

}

.image2{

grid-column-start: 2;

grid-column-end: 3;

grid-row-start: 2;

grid-row-end: 3;

}

.image3{

grid-column-start: 3;

grid-column-end: 4;

grid-row-start: 1;

grid-row-end: 2;

}

.image4{

grid-column-start: 1;

grid-column-end: 2;

grid-row-start: 2;

grid-row-end: 4;

}

.image5{

grid-column-start: 2;

grid-column-end: 3;

grid-row-start: 3;

grid-row-end: 4;

}

.image6{

grid-column-start: 3;

grid-column-end: 4;

grid-row-start: 2;

grid-row-end: 4;

}

img{

height: 100%;

width: 100%;

border-radius: 10px;

}

</style>

</head>

<body>

<div class="container">

<div class="image1"><img src="/Grid Assignment/image/download1.jpeg" alt=""></div>

<div class="image2"><img src="/Grid Assignment/image/download2.jpeg" alt=""></div>

<div class="image3"><img src="/Grid Assignment/image/download3.png" alt=""></div>

<div class="image4"><img src="/Grid Assignment/image/images.jpeg" alt=""></div>

<div class="image5"><img src="/Grid Assignment/image/images1.jpeg" alt=""></div>

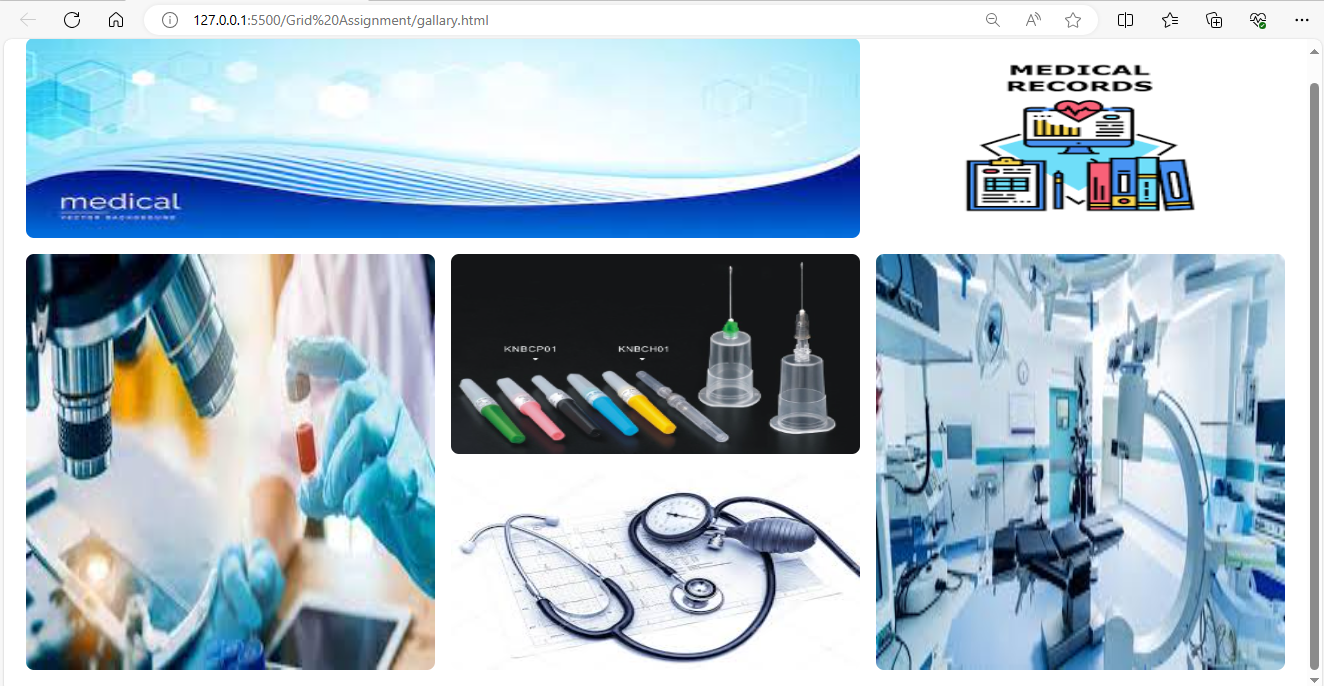
<div class="image6"><img src="/Grid Assignment/image/images2.jpeg" alt=""></div>

</div>

</body>

</html>

Output:



1. Problem Statement

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

Solution:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>box image</title>

<style>

.container{

display: grid;

grid-gap: 10px;

grid-template-columns: 100px 100px 100px;

}

.box{

background-color:blanchedalmond;

color: black;

border-radius: 5px;

padding: 35px;

font-size: 20px;

}

.box1{

grid-column: 1/3;

grid-row: 1;

}

.box2{

grid-column: 3;

grid-row:1/3 ;

}

.box3{

grid-column: 1;

grid-row: 2;

}

.box4{

grid-column: 2;

grid-row: 2;

}

</style>

</head>

<body>

<div class="container">

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

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

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

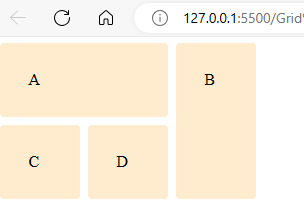
<div class="box box4">D</div>

</div>

</body>

</html>

Output:



1. Problem Statement

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

Solution:

The grid-auto-rows property sets a size for the rows in a grid container. This property affects only rows with the size not set and the grid-auto-columns property in CSS grid is used to specify the size of columns that are automatically created by the grid.

Example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.container{

display: grid;

grid-template-areas: "X X";

grid-template-rows: 50px ;

grid-auto-rows: 250px;

}

.box{

border: 2px solid black;

background-color: aqua;

padding: 10px;

}

</style>

</head>

<body>

<div class="container">

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

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

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

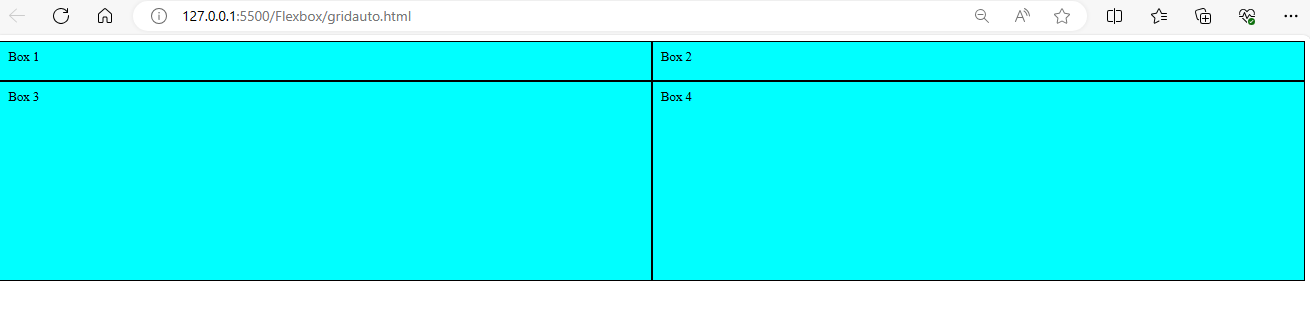
<div class="box">Box 4</div>

</div>

</body>

</html>

Output:



1. Problem Statement

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

Solution:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Box Altering</title>

<style>

body{

margin: 45px;

}

.box{

background-color: #464343;

color: #fff;

border-radius: 5px;

padding: 20px;

font-size: 150%;

order: 1;

}

.box:nth-child(even) {

background-color: #ccc;

color: #000;

}

.container {

width: 600px;

display: grid;

grid-template-columns: repeat(6, 100px);

grid-gap: 10px;

}

.box1 {

order: 3;

}

.box2 {

order: 6;

}

.box8 {

order: 2;

}

</style>

</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>

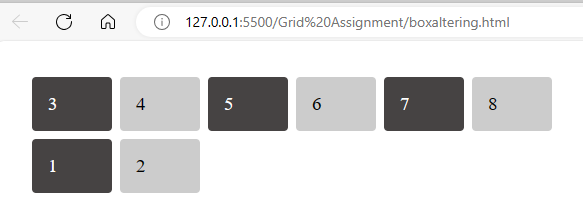
<div class="box box8">8</div>

</div>

</body>

</html>

Output:



1. Problem Statement

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

Solution:

The main difference between them is that justify-items apply to grid containers and justify-self applies to grid-items.

Example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.container {

Justify-item: center;

display: grid;

grid-template-columns: auto auto auto;

border: 1px solid black;

}

.container > div {

padding: 20px;

background-color: bisque;

border: 1px solid black;

}

</style>

</head>

<body>

<div class="container">

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

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

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

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

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

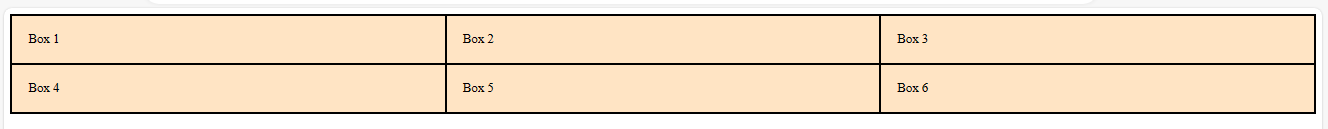
<div class="box6">Box 6</div>

</div>

</body>

</html>

Output:



After using justify-item: center;

Output:

