



SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IOT Including CS&BCT

**COURSE NAME : 19SB602 FULL STACK DEVELOPMENT FOR NEXT
GENERATION IOT**

III YEAR / VI SEMESTER

Unit II- FRONT-END MODULES

Topic :Basics, Containers, Colors Grids



Containers

Bootstrap requires a **containing element to wrap site contents.**

Containers are used to **pad the content inside of them**, and there are **two container classes** available:

1. The **.container** class provides a **responsive fixed width container**
2. The **.container-fluid** class provides a **full width container, spanning the entire width of the viewport.**



Fixed Container

.container class is to create a responsive, fixed-width container.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
  <script
src="https://cdn.jsdelivr.net/npm/jquery@3.7.1/dist/jquery.slim.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></scr
ipt>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"
></script>
</head>
<body>
```



```
<div class="container">  
  <h1>My First Bootstrap Page</h1>  
  <p>This part is inside a .container class.</p>  
  <p>The .container class provides a responsive fixed width container.</p>  
  <p>Resize the browser window to see that its width (max-width) will  
change at different breakpoints.</p>  
</div>  
  
</body>  
</html>
```



My First Bootstrap Page

This part is inside a `.container` class.

The `.container` class provides a responsive fixed width container.

Resize the browser window to see that its width (`max-width`) will change at different breakpoints.



Fluid Container

.container-fluid class is to create a full width container, that will always span the entire width of the screen (width is always 100%)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
  <script
src="https://cdn.jsdelivr.net/npm/jquery@3.7.1/dist/jquery.slim.min.js"></script>
  <script
```



```
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></scri  
pt>
```

```
<script  
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js">  
</script>  
</head>  
<body>
```

```
<div class="container-fluid">  
<h1>My First Bootstrap Page</h1>  
<p>This part is inside a .container-fluid class.</p>  
<p>The .container-fluid class provides a full width container, spanning the entire  
width of the viewport.</p>  
</div>
```

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



My First Bootstrap Page

This part is inside a `.container-fluid` class.

The `.container-fluid` class provides a full width container, spanning the entire width of the viewport.



Responsive Containers

You can also use the **.container-sm|md|lg|xl** classes to create responsive containers.

The **max-width** of the container will change on different screen sizes/viewports:



Class	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
.container-sm	100%	540px	720px	960px	1140px
.container-md	100%	100%	720px	960px	1140px
.container-lg	100%	100%	100%	960px	1140px
.container-xl	100%	100%	100%	100%	1140px



```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.cs
s">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></scri
pt>
  <script
```



```
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js">
</script>
```

```
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></scrip
t>
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<h1>Responsive Containers</h1>
```

```
<p>Resize the browser window to see the effect.</p>
```

```
</div>
```

```
<div class="container-sm border">.container-sm</div>
```

```
<div class="container-md mt-3 border">.container-md</div>
```

```
<div class="container-lg mt-3 border">.container-lg</div>
```

```
<div class="container-xl mt-3 border">.container-xl</div>
```

```
</body>
```

```
</html>
```



Responsive Containers

Resize the browser window to see the effect.

```
.container-sm
```

```
.container-md
```

```
container-lg
```

```
container-xl
```



Colors Grids

Bootstrap and other CSS frameworks often provide a system for managing **both layout grids and color schemes.**

In Bootstrap, the grid system helps you **organize the layout** of your webpage, and you **can customize the colors using predefined classes** or by defining your **own styles.**



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.item1 { grid-area: header; }
```

```
.item2 { grid-area: menu; }
```

```
.item3 { grid-area: main; }
```

```
.item4 { grid-area: right; }
```

```
.item5 { grid-area: footer; }
```

```
.grid-container {
```

```
  display: grid;
```

```
  grid-template-areas:
```

```
    'header header header header header header'
```

```
    'menu main main main right right'
```

```
    'menu footer footer footer footer footer';
```

```
  gap: 10px;
```

```
  background-color: #2196F3;
```

```
  padding: 10px;
```



```
.grid-container > div {  
  background-color: rgba(255, 255, 255, 0.8);  
  text-align: center;  
  padding: 20px 0;  
  font-size: 30px;  
}  
</style>  
</head>  
<body>  
  
<h1>Grid Layout</h1>
```



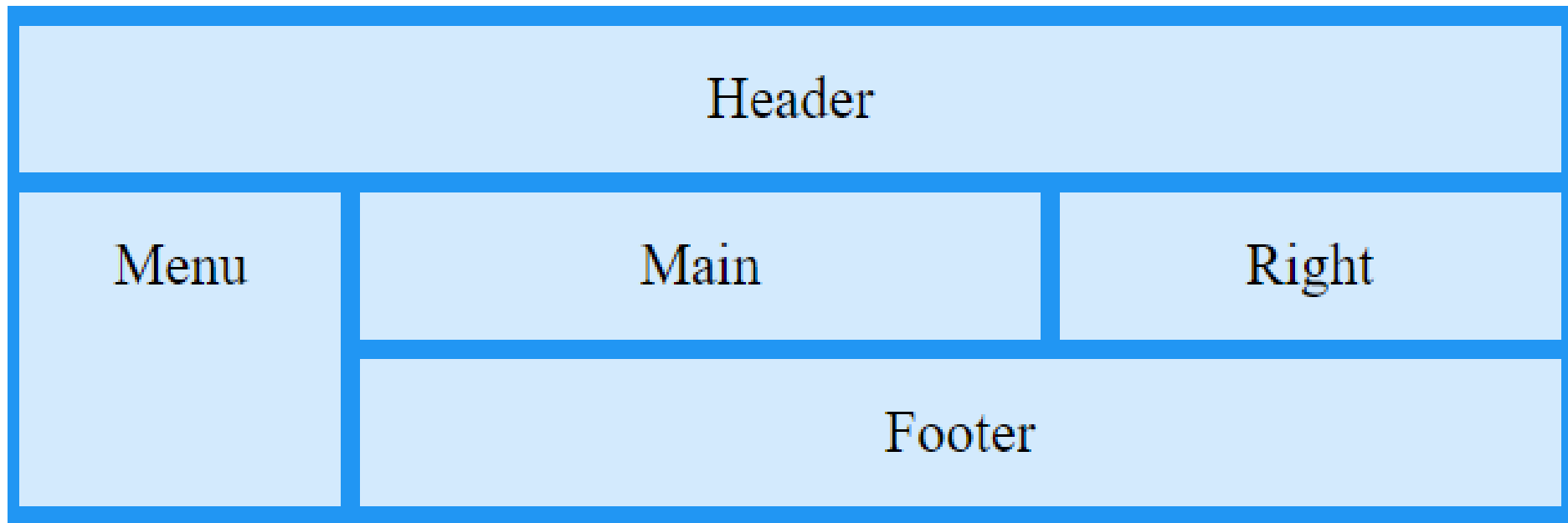

The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning:

```
<div class="grid-container">  
  <div class="item1">Header</div>  
  <div class="item2">Menu</div>  
  <div class="item3">Main</div>  
  <div class="item4">Right</div>  
  <div class="item5">Footer</div>  
</div>  
  
</body>  
</html>
```



Grid Layout

The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning:





Grid Layout

The CSS Grid Layout Module offers a **grid-based layout system**, with **rows and columns**, making it easier to design web pages without having to use floats and positioning.

Browser Support

The grid properties are supported **in all modern browsers**.

Grid Elements

A grid layout consists of a **parent element**, with **one or more child elements**.



```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  background-color: #2196F3;
  padding: 10px;
}
.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
```



```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Grid Elements</h1>
```

```
<p>A Grid Layout must have a parent element with the  
<em>display</em> property set to <em>grid</em> or <em>inline-  
grid</em>.</p>
```

```
<p>Direct child element(s) of the grid container automatically  
becomes grid items.</p>
```



```
<div class="grid-container">  
  <div class="grid-item">1</div>  
  <div class="grid-item">2</div>  
  <div class="grid-item">3</div>  
  <div class="grid-item">4</div>  
  <div class="grid-item">5</div>  
  <div class="grid-item">6</div>  
  <div class="grid-item">7</div>  
  <div class="grid-item">8</div>  
  <div class="grid-item">9</div>  
</div>  
</body>  
</html>
```

Grid Elements

A Grid Layout must have a parent element with the *display* property set to *grid* or *inline-grid*.

Direct child element(s) of the grid container automatically becomes grid items.

1	2	3
4	5	6
7	8	9



Any Query????

Thank you.....