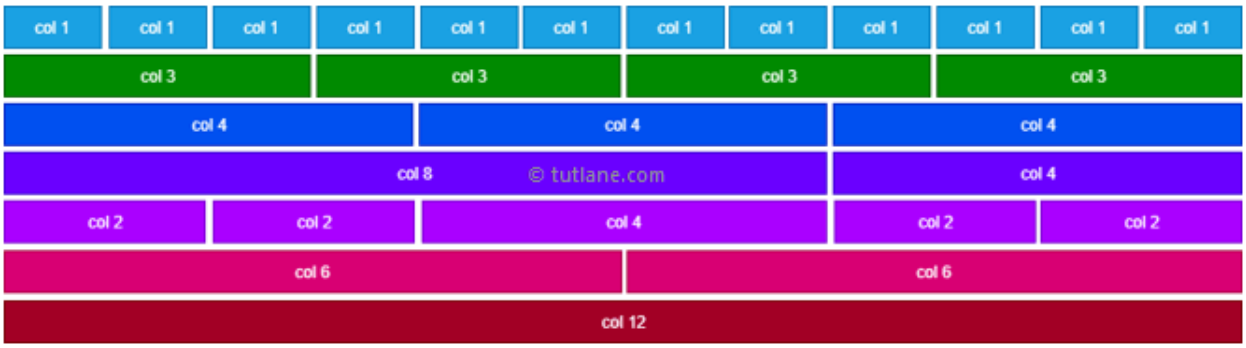
What is the Bootstrap Grid System?

It's a **layout system** that helps you organize content on a webpage in **rows and columns**, making it responsive (works on mobile, tablet, and desktop).

Think of it like a big **table with 12 invisible columns** across the screen.

- You can decide how many of those 12 columns each element should take.
- Bootstrap automatically adjusts for different screen sizes.



How it works

1. **Container** → Wraps your grid. (`.container` or `.container-fluid`)
2. **Row** → Creates a horizontal group. (`.row`)
3. **Columns** → Divide the row into up to **12 parts** using `.col`.

Bootstrap Grid Classes

Bootstrap 4 has included 5 predefined grid classes to scale the content depending on the device or viewport size.

- .col-*
- .col-sm-*
- .col-md-*

- .col-lg-*
- .col-xl-*

Here, the asterisk (*****) is the span width of the column from **1** to **12**. The following table lists how the grid system classes will work across multiple devices.

Class	Device Type	Width
.col-*	Extra Small	<576px
.col-sm-*	Small	≥576px
.col-md-*	Medium	≥768px
.col-lg-*	Large	≥992px
.col-xl-*	Extra Large	≥1200px

Structure of Bootstrap Grid

To create responsive web page layouts using a bootstrap grid system, we need to use rows and columns within the container or container-fluid like as shown below.

```
<div class="container">
  <div class="row">
   <div class="col-*-*"></div>
   <div class="col-*-*"></div>
   <div class="col-*-*"></div>
  </div>
</div>
```

If you observe the above code, we created a grid layout by specifying the rows and columns within .container. Here, the first star (*) will indicate the responsiveness, and that should be either sm, md, lg, or xl. The second star (*) will indicate a number, and that should be from **1** to **12** for each row.

Every time while we create a grid layout, we need to place the content within the columns (.col or .col-*-*) and those columns must be the children of rows (.row) and those rows must be placed inside of a container (.container or .container-fluid).

Example

```
<div class="container">
  <div class="row">
    <div class="col-4">Column 1 (4/12)</div>
    <div class="col-8">Column 2 (8/12)</div>
  </div>
</div>
```

This creates **2 columns**:

- First takes **4 parts** (out of 12)
- Second takes **8 parts**

Responsive Grid

Bootstrap has **breakpoints** for different screen sizes:

- `col-` → Extra small (phones)
- `col-sm-` → Small devices (≥576px)
- `col-md-` → Medium devices (≥768px)
- `col-lg-` → Large devices (≥992px)
- `col-xl-` → Extra large (≥1200px)
- `col-xxl-` → Very large screens (≥1400px)

Example

```
<div class="row">
  <div class="col-sm-6 col-lg-4">Box 1</div>
  <div class="col-sm-6 col-lg-4">Box 2</div>
  <div class="col-sm-12 col-lg-4">Box 3</div>
</div>
```

On **small screens**:

- Box 1 = half width
- Box 2 = half width
- Box 3 = full width

On **large screens**:

- Each box = 1/3 of the row

Auto Layout

If you don't specify numbers, columns will share space equally:

```
<div class="row">
  <div class="col">One</div>
  <div class="col">Two</div>
  <div class="col">Three</div>
</div>
```

Each takes **4/12** (equal widths).

In short:

- Bootstrap grid = 12 columns per row.
- Use `col-[size]-[number]` to control how wide a column is on different screens.
- Makes layouts flexible & responsive without writing custom CSS.