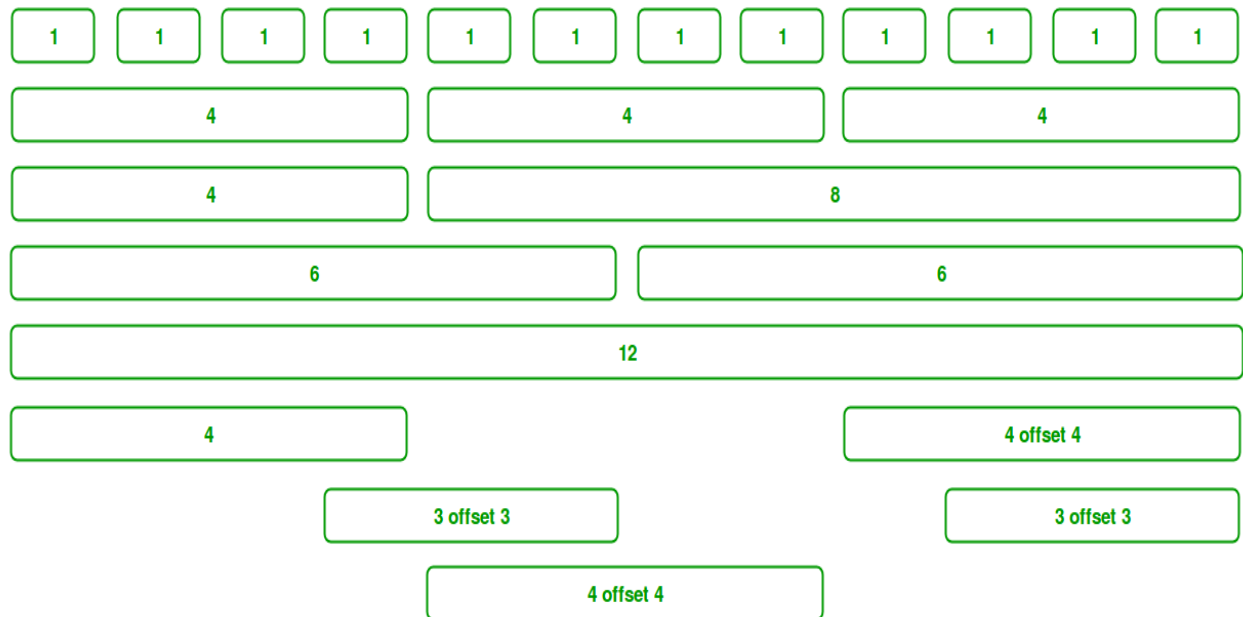


# The Grid System

Bootstrap Grid System allows up to 12 columns across the page. You can use each of them individually or merge them together for wider columns. You can use all combinations of values summing up to 12. You can use 12 columns each of width 1, or use 4 columns each of width 3 or any other combination.



## Grid Classes:

The Bootstrap grid system has four classes which can be combined to make more flexible layouts:

- xs – For Mobile Phones.
- sm – For Tablets/Phablets
- md – For Small-sized Desktops/Laptops
- lg – For Larger-sized Desktops/Laptops

## Components of Grid System:

We will be learning the Components of the Grid system one-by-one:

1. **Containers:** Bootstrap requires a containing element to wrap site contents and house our grid system. The word 'container' is a container of row elements and row elements are 'containers' of the column elements. You will understand it more in the later part of the article where we have dealt with columns.

Use 'container' for a responsive fixed width container.

Use 'container-fluid' for a full width container, spanning the entire width of your viewport.

```
</header>

<div class="container">
  <div class="row"><div class="well">First Row</div></div>
  <div class="row"><div class="well">Second Row</div></div>
</div>

<div class="container">
  <div class="row">

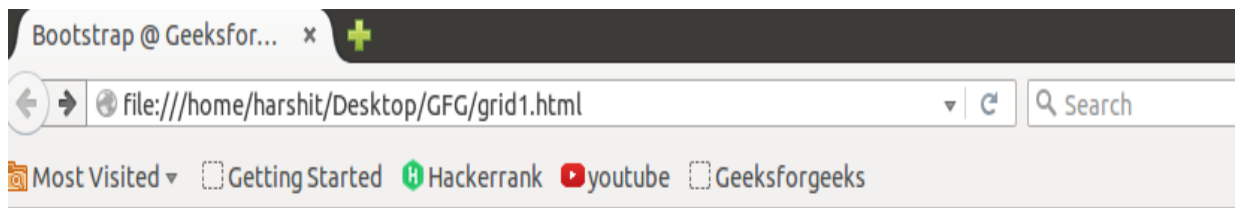
    <div class="col-xs-6 col-sm-4 col-md-6 col-lg-4"><div class="well"> First Column </div></div>

    <div class="col-xs-3 col-sm-4 col-md-6 col-lg-4">
      <div class="well"> Second Column
        <div class="row">
          <div class="col-xs-6 col-sm-4 col-md-6 col-lg-4"> <div class="well">Nested I</div></div>
          <div class="col-xs-3 col-sm-4 col-md-6 col-lg-4"> <div class="well">Nested II</div></div>
          <div class="col-xs-3 col-sm-4 col-md-6 col-lg-4"> <div class="well">Nested III</div></div>
        </div>
      </div>
    </div>

    <div class="col-xs-3 col-sm-4 col-md-6 col-lg-4"><div class="well">Third Column</div></div>

  </div>
</div>

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



# GeeksforGeeks

A computer science portal for geeks.

[Visit our website](#)

[Like us](#) On facebook

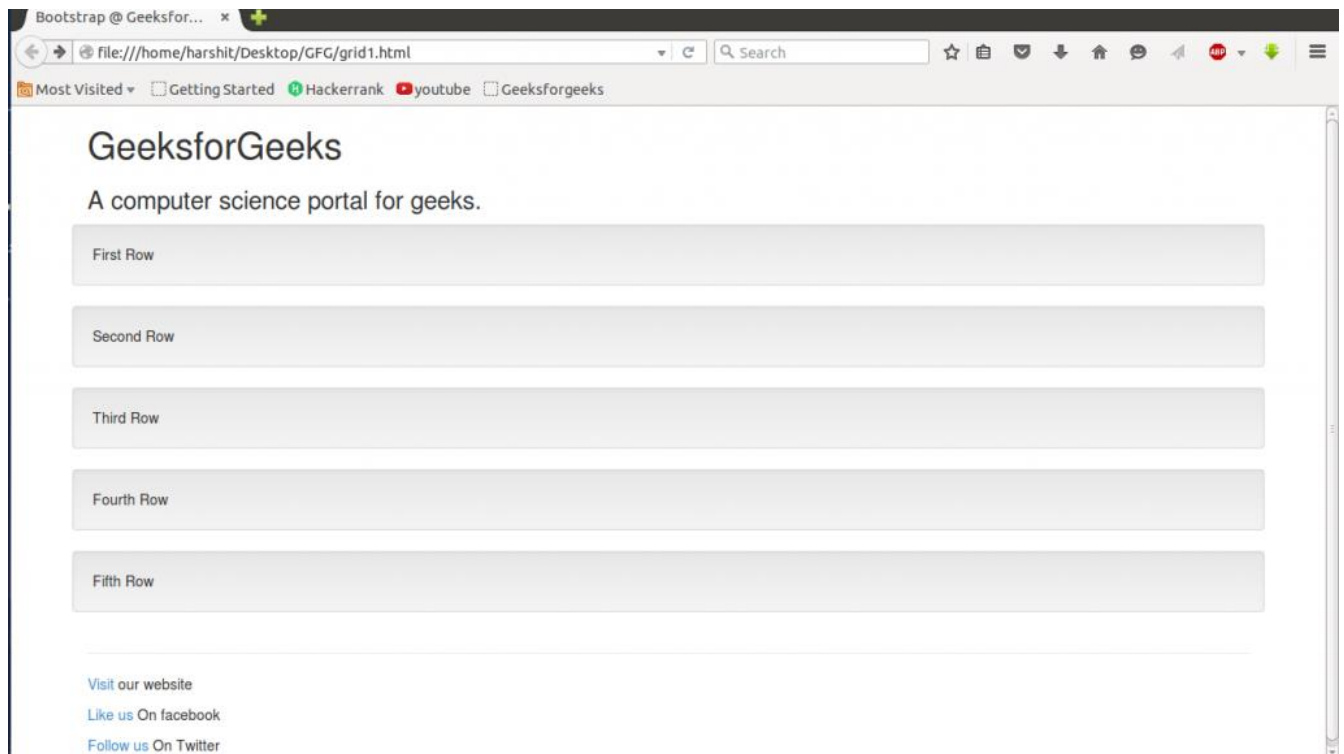
[Follow us](#) On Twitter

*Note: The tag defines a division or a section in an HTML document. The tag is used to group block-elements to format them with CSS.*

2. **Rows:** Rows must be placed within a 'container' or 'container-fluid' for proper alignment and padding. We use rows to create horizontal groups

of columns.

```
14
15 </head>
16
17 <body>
18
19   <header>
20     <div class="container">
21       <h1>GeeksforGeeks</h1>
22       <h3>A computer science portal for geeks.</h3>
23     </div>
24   </header>
25
26   <div class="container">
27     <div class="row"><div class="well">First Row</div></div>
28     <div class="row"><div class="well">Second Row</div></div>
29     <div class="row"><div class="well">Third Row</div></div>
30     <div class="row"><div class="well">Fourth Row</div></div>
31     <div class="row"><div class="well">Fifth Row</div></div>
32   </div>
33
34
35
36   <footer>
37     <div class="container">
38       <hr>
39       <p> <a href="http://www.geeksforgeeks.org">Visit </a> our website </p>
40       <p> <a href="https://www.facebook.com/GeeksforGeeks-316764689022/timeline/">Like us</a> On facebook</p>
41       <p> <a href="https://twitter.com/geeksforgeeks">Follow us </a> On Twitter</p>
42     </div>
43   </footer>
44
45
```



adds a rounded border around an element with a gray background color and some padding.

3. **Columns:** Grid columns are created by specifying the number of twelve available columns you wish to span. For example, three equal columns would use three `col-lg-4`.

Remember we can use all the grid classes (`xs`, `sm`, `md` and `lg`) for different screen sizes.

```
16
17 <body>
18
19   <header>
20     <div class="container">
21       <h1>GeeksforGeeks</h1>
22       <h3>A computer science portal for geeks.</h3>
23     </div>
24   </header>
25
26   <div class="container">
27     <div class="row"><div class="well">First Row</div></div>
28     <div class="row"><div class="well">Second Row</div></div>
29   </div>
30
31   <div class="container">
32     <div class="row">
33       <div class="col-lg-4">
34         <div class="well">First Column</div>
35       </div>
36       <div class="col-lg-4">
37         <div class="well">Second Column</div>
38       </div>
39       <div class="col-lg-4">
40         <div class="well">Third Column</div>
41       </div>
42     </div>
43   </div>
44
45
46
```

Bootstrap @ Geeksfor... x

File:///home/harshit/Desktop/GFG/grid1.html

Most Visited Getting Started Hackerrank youtube Geeksforgeeks

# GeeksforGeeks

A computer science portal for geeks.

First Row

Second Row

First Column

Second Column

Third Column

[Visit our website](#)  
[Like us On facebook](#)  
[Follow us On Twitter](#)

But, we would face a major problem in this case. Notice that for devices like mobiles(xs) and tablets(sm), the columns would stack upon one another. If we want our columns to stay in the same row, no matter what the size of device is, then we must use xs. But then, sometimes it looks ugly.

So, we can build even more dynamic and powerful layouts with using more than one choice for columns. Let's say we define the attributes for 4 columns as-

What this means is:

For **larger devices(lg)**, there would be a single row for all the four columns( $3*4=12$ ), each of grid size 3.

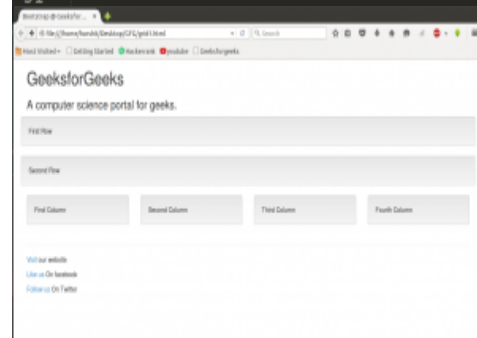
For **medium sized devices(md)**, there would be two rows with two columns each ( $6*2 + 6*2=12+12$ ). Each column would be of grid size 6.

For **small sized devices(sm)**, there would be two rows. First row would have 3 columns and second row would have a single column .( $4*3 + 4*1=12+4$ ). Each column is of grid size 4.

For **extra small devices(xs)**, there would again be a single row and all the four columns in it( $3*4=12$ ), each of grid size 3.

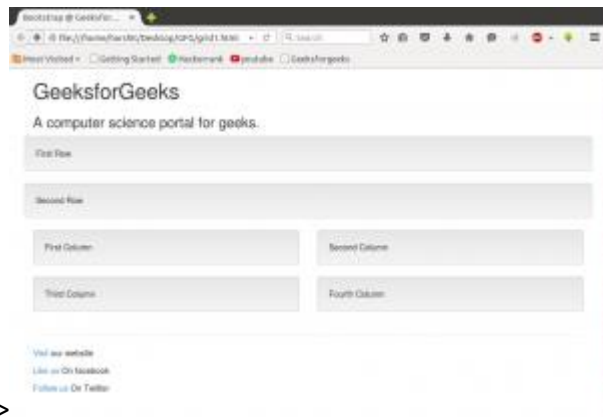
You can try this by squeezing a browser from an end and notice the orientations of the columns as the size decreases.

```
18
19 <header>
20   <div class="container">
21     <h1>GeeksforGeeks</h1>
22     <h3>A computer science portal for geeks.</h3>
23   </div>
24 </header>
25
26 <div class="container">
27   <div class="row"><div class="well">First Row</div></div>
28   <div class="row"><div class="well">Second Row</div></div>
29 </div>
30
31 <div class="container">
32   <div class="row">
33     <div class="col-xs-3 col-sm-4 col-md-6 col-lg-3">
34       <div class="well">First Column</div>
35     </div>
36     <div class="col-xs-3 col-sm-4 col-md-6 col-lg-3">
37       <div class="well">Second Column</div>
38     </div>
39
40     <div class="col-xs-3 col-sm-4 col-md-6 col-lg-3">
41       <div class="well">Third Column</div>
42     </div>
43
44     <div class="col-xs-3 col-sm-4 col-md-6 col-lg-3">
45       <div class="well">Fourth Column</div>
46     </div>
47   </div>
48 </div>
49
50
51
```

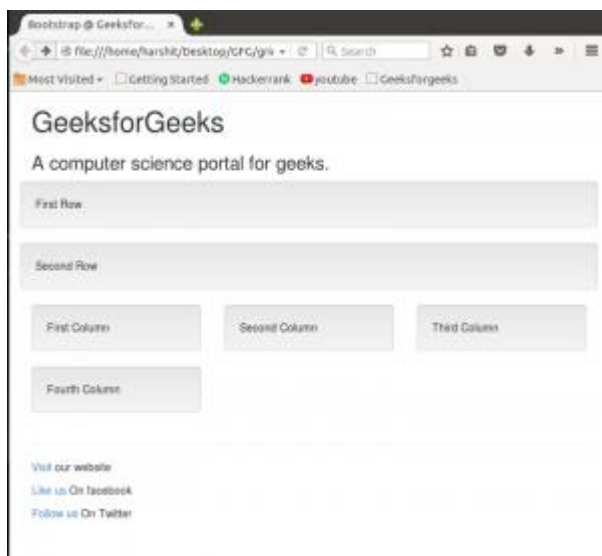


<-- For Larger-sized Desktops/Laptops (lg)

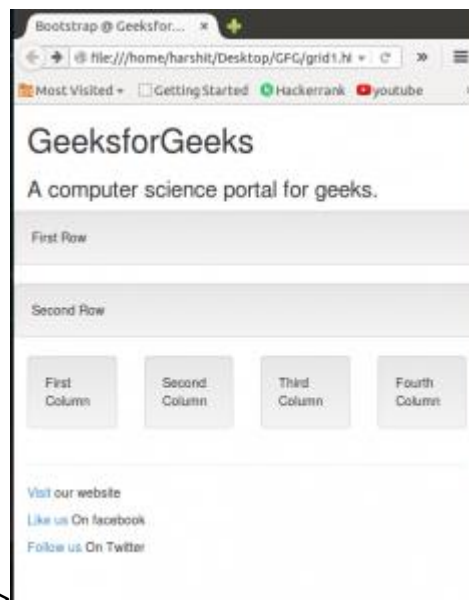




For Small-sized Desktops/Laptops (md) →



<-- For Tablets/Phablets (sm)



For Mobile Phones (xs)→

**Column Resets:** With the four tiers of grids available, we are bound to run into issues where at certain breakpoints, the columns don't quite clear right as one is taller( has more text) than the other column. A command called **clearfix** is

there which fixes any issues regarding that viewport. We just need to write a div command with class clearfix after the block where the column isn't clearing right.

Let's say we have an issue with the md and sm viewport. But if we are using

But then it is causing problems for other viewport (maybe lg and xs) . So, what we can do is make clearfix visible only for the md and sm block or hide all other blocks (lg and xs).

or

**Columns Offset:** We can move the columns to the right by x columns using col-md-offset-x in the class.

This change results in making an offset of 2 grid columns before the fourth column.

**Nesting Columns:** For nesting columns within a column, we need to add a new row and set of columns. Nested rows should include a set of columns that add

up to 12 or less than that.

```
</header>

<div class="container">
  <div class="row"><div class="well">First Row</div></div>
  <div class="row"><div class="well">Second Row</div></div>
</div>

<div class="container">
  <div class="row">

    <div class="col-xs-6 col-sm-4 col-md-6 col-lg-4"><div class="well"> First Column </div></div>

    <div class="col-xs-3 col-sm-4 col-md-6 col-lg-4">
      <div class="well"> Second Column
        <div class="row">
          <div class="col-xs-6 col-sm-4 col-md-6 col-lg-4"> <div class="well">Nested I</div></div>
          <div class="col-xs-3 col-sm-4 col-md-6 col-lg-4"> <div class="well">Nested II</div></div>
          <div class="col-xs-3 col-sm-4 col-md-6 col-lg-4"> <div class="well">Nested III</div></div>
        </div>
      </div>
    </div>

    <div class="col-xs-3 col-sm-4 col-md-6 col-lg-4"><div class="well">Third Column</div></div>

  </div>
</div>

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

# GeeksforGeeks

A computer science portal for geeks.

First Row

Second Row

First Column

Second Column

Nested  
I

Nested  
II

Nested  
III

Third Column

[Visit](#) our website

[Like us](#) On facebook

[Follow us](#) On Twitter