

Sql server, .net and c# video tutorial

Free C#, .Net and Sql server video tutorial for beginners and intermediate programmers.

Bootstrap 3 grid classes

Suggested Video Tutorials

Part 1 - What is bootstrap

Part 2 - Setting up bootstrap

Part 3 - Bootstrap grid system

In this video we will discuss **Bootstrap 3 grid classes** and their use.

Here are the different grid classes that are available in Bootstrap 3

Device	Bootstrap Class
Extra small devices - Mobile Phones (< 768px)	.col-xs-*
Small devices - Tablets (≥768px)	.col-sm-*
Medium devices - Desktops (≥992px)	.col-md-*
Large devices - Desktops (≥1200px)	.col-lg-*

Grid classes for a given screen size apply to that screen size and larger unless another declaration overrides it. Let us understand what we mean by this statement with an example.

For example, we want four equal columns on both medium and large devices.



For medium devices the class that we use is col-md-*. Since we want 4 equal columns, we would use four col-md-4 columns as shown below. So with the following HTML we get four equal columns on both medium and large devices. This is because col-md-* class is applied to both medium and large devices as we have not used col-lg-*.

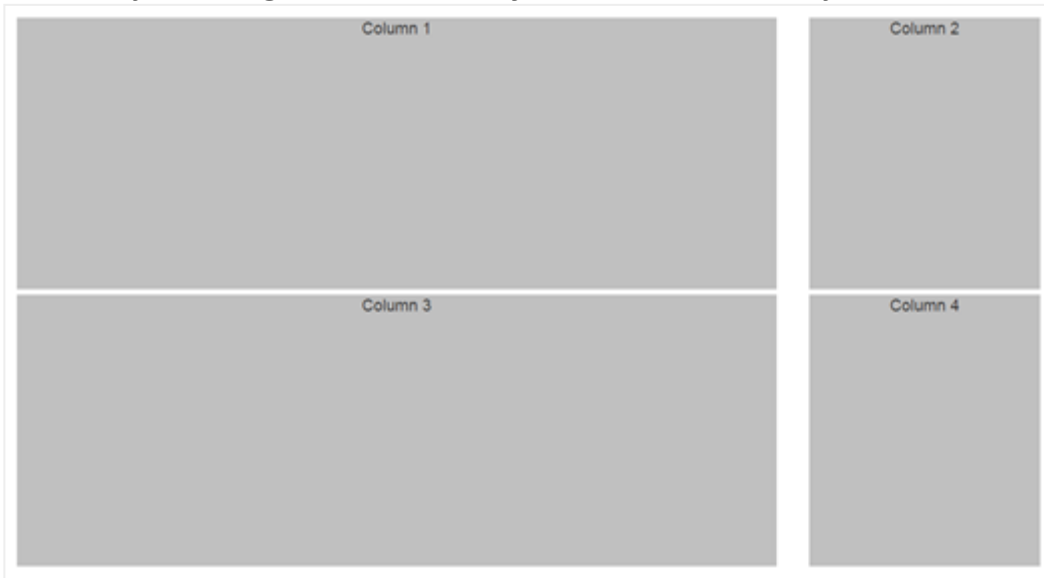
```
<div class="container">  
  <div class="row">
```

```

<div class="col-md-3">
  <div class="customDiv">Column 1</div>
</div>
<div class="col-md-3">
  <div class="customDiv">Column 2</div>
</div>
<div class="col-md-3">
  <div class="customDiv">Column 3</div>
</div>
<div class="col-md-3">
  <div class="customDiv">Column 4</div>
</div>
</div>
</div>

```

If you want, you can change this on a large device by using col-lg-* class along with col-md-* class. Let us say on a large device we want just 2 columns in every row with 3:1 ratio.



To achieve this we would modify our HTML as shown below. So with these 2 classes in places we get 4 equal columns on a medium device and 2 columns with 3:1 ratio on a large device.

```

<div class="container">
  <div class="row">
    <div class="col-md-3 col-lg-9">
      <div class="customDiv">Column 1</div>
    </div>
    <div class="col-md-3 col-lg-3">
      <div class="customDiv">Column 2</div>
    </div>
    <div class="col-md-3 col-lg-9">
      <div class="customDiv">Column 3</div>
    </div>
    <div class="col-md-3 col-lg-3">
      <div class="customDiv">Column 4</div>
    </div>
  </div>
</div>

```

However, on a small device notice the columns are stacked on top of each other with each column spanning across the 12 column grid system. If you want you can change this default behaviour using `col-sm-*` class. Let us say on a small device we want 2 equal columns in every row.



To achieve this we would modify our HTML as shown below. So with these 3 classes in places we get 4 equal columns on a medium device, 2 columns with 3:1 ratio on a large device and 2 equal columns on a small device.

```
<div class="container">
  <div class="row">
    <div class="col-md-3 col-lg-9 col-sm-6">
      <div class="customDiv">Column 1</div>
    </div>
    <div class="col-md-3 col-lg-3 col-sm-6">
      <div class="customDiv">Column 2</div>
    </div>
    <div class="col-md-3 col-lg-9 col-sm-6">
      <div class="customDiv">Column 3</div>
    </div>
    <div class="col-md-3 col-lg-3 col-sm-6">
      <div class="customDiv">Column 4</div>
    </div>
  </div>
</div>
```

On an extra small device, the columns are stacked on top of each other with each column spanning across the 12 column grid system. If you want you can change this default behaviour very easily using `col-xs-*` class.

Now, if you remove all the grid classes except `col-sm-6`. In other words if we modify the code as shown below, we get 2 equal columns on small, medium and large devices. This is because grid classes for a given screen size apply to that screen size and larger unless another declaration

overrides it.

```
<div class="container">
  <div class="row">
    <div class="col-sm-6">
      <div class="customDiv">Column 1</div>
    </div>
    <div class="col-sm-6">
      <div class="customDiv">Column 2</div>
    </div>
    <div class="col-sm-6">
      <div class="customDiv">Column 3</div>
    </div>
    <div class="col-sm-6">
      <div class="customDiv">Column 4</div>
    </div>
  </div>
</div>
```

By now I believe you have understood the use of these grid classes. With these classes, you have complete control over the layout of your website on different screen sizes.

WWW.PRAGIMTECH.COM

**CLICK HERE FOR THE FULL
BOOTSTRAP TUTORIAL PLAYLIST**

**[HTTPS://TWITTER.COM/KUDVENKAT](https://twitter.com/kudvenkat)
[HTTPS://FACEBOOK.COM/PRAGIMTECH](https://facebook.com/pragimtech)**

Share



4 comments:



Kalapala Babu Rao May 17, 2016 at 11:31 PM

This GridDemo program is working in IE, but not in Chrome and Firefox.
What is the problem.

[Reply](#)

Replies



Siva GS May 31, 2016 at 8:39 AM

Reset your screen zoom size to default in Chrome and Firefox, the demo works well.

[Reply](#)



San Ali May 19, 2016 at 6:35 AM

Hi Venkat Sir Please make few videos on iOS Programming in Swift as it is like a C# ...

[Reply](#)

Anonymous July 22, 2016 at 7:04 PM

At your service man

[Reply](#)

Enter your comment...

Comment as: WhackyCoder ▾

[Sign out](#)

[Publish](#)

[Preview](#)

☐ Notify me

If you like this website, please share with your friends on facebook and Google+ and recommend us on google using the g+1 button on the top right hand corner.

Links to this post

[Create a Link](#)



[Home](#)



[View web version](#)

Powered by [Blogger](#).