

1 SHARES



## Day 2: Bootstrap 4 Grid System Tutorial and Examples

□ October 23, 2018 □ Cristina Conacei □ Tutorials

[DAY 1: BOOTSTRAP 4 CDN](#)

[DAY 3: BOOTSTRAP 4 FLEX](#)

Hello and welcome to the [second day of Bootstrap 4](#) 🎉 Today we will learn about the Bootstrap 4 Grid, what it is and how you can use it.

The Bootstrap 4 Grid System is used for responsive layouts. A responsive layout represents the way elements align in the page on different resolutions. It is important you understand how to use the grid before learning about any other Bootstrap 4 component. This is because whatever element you will choose to use, you will need to place it somewhere on the screen. Let's get started!

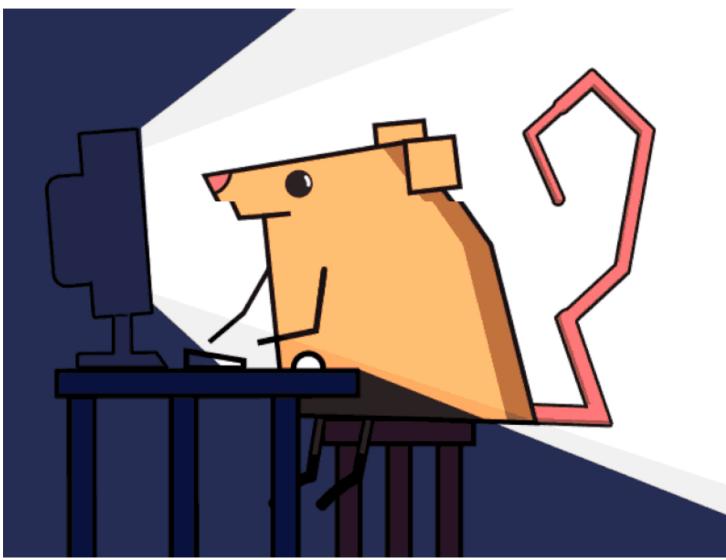


Photo credit to [Animade](#) for his shot.

This article is broken down into the following parts:

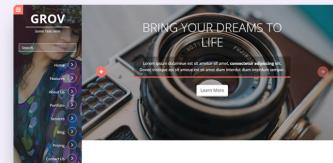
- Bootstrap 4 Containers
- Bootstrap 4 Rows
- Bootstrap 4 Columns
  - Basic Columns
  - Setting Sizes
  - Setting Breakpoints
  - Setting Sizes and Breakpoints
  - Offsetting Columns
  - Nesting Columns
- Further Reading and Watching

The Bootstrap 4 grid consists of containers, rows and columns. We will take them one by

### Learn Bootstrap 4 in 14 Days

1. Bootstrap 4 CDN and Starter Template
2. Bootstrap 4 Grid System
3. Bootstrap 4 Flex
4. Bootstrap 4 Typography
5. Bootstrap 4 Images
6. Bootstrap 4 Buttons
7. Bootstrap 4 Inputs
8. Bootstrap 4 Forms
9. Bootstrap 4 Tables
10. Bootstrap 4 Navbar
11. Bootstrap 4 Navigation
12. Bootstrap 4 Lists
13. Bootstrap 4 Cards
14. Bootstrap 4 Modals

### Popular Themes



GROV - A Responsive Theme



Aksh - Personal Portfolio Template



Precon - Multipurpose Business & Agency Template

[VIEW MORE POPULAR THEMES »](#)

The Bootstrap 4 grid consists of containers, rows and columns. We will take them one by one and explain them.

## Bootstrap 4 Containers

A Bootstrap 4 container is an element with the class `.container`. The container is the root of the Bootstrap 4 grid system and it is used to control the width of the layout. It contains all the elements in a page. This means your page should have the following structure: first the body of the HTML page, inside of it you add the container and all the other elements inside the container.

```
<body>
  <div class="container">
    ...
  </div>
</body>
```

The simple `.container` class sets the width of the layout depending on the width of the screen. It places the content in the middle of the page aligning it horizontally. There is equal space between the Bootstrap 4 container and the left and the right edge of the page. The `.container` scales down in width as the screen width narrows and becomes full-width on mobile. The width of the container is defined inside the Bootstrap 4 library for every screen size. You can see the exact [sizes here](#).

A full-width container takes 100% of the screen size regardless of the screen width. To use it you need to add the class `.container-fluid`. To see the differences between the 2 types of containers, you can open the following pen in your console and switch between screen sizes.

The screenshot shows a CodePen interface with tabs for 'HTML', 'SCSS', and 'Result'. The 'Result' tab is active, displaying two sections. The top section, with a light gray background, contains the text 'Hello! I am in a simple container.' The bottom section, with a dark teal background, contains the text 'Hello! I am in a full-width container.'. At the bottom of the result area, there are buttons for 'Resources', '1x', '0.5x', '0.25x', and 'Rerun'.

```
<div class="container">
  Hello! I am in a simple container.
</div>

<div class="container-fluid">
  Hello! I am in a full-width container.
</div>
```

## Bootstrap 4 Rows

Bootstrap 4 rows are horizontal slices of the screen. They are used only as wrappers for columns. To use them, you need the `.row` class.

```
<div class="row">
  ...
</div>
```

They are only used for containing columns. If you place other elements inside the row along with columns you will not get the expected result.

They have to be placed in containers. Containers can hold any element, but a row must be placed inside a container. If you don't do this, you will get a horizontal scroll on your page. This happens because rows have negative left and right margins of 15. The container has 15px paddings so it counteracts the margins.

The columns have to be children of the row. Otherwise they will not align. The rows and columns are created to work together in this strict hierarchy.

## Bootstrap 4 Columns

### Basic Columns

The screenshot shows a CodePen interface with tabs for 'HTML' and 'SCSS' on the left, and 'Result' on the right. At the top right is a 'CODEPEN' button with the text 'EDIT ON'. Below the tabs, there's a preview area containing several colored boxes. A large blue box at the top contains the text 'This is a single column.'. Below it, two purple boxes side-by-side contain 'This is the first column.' and 'This is the second column.'. At the bottom, four pink boxes arranged in a 2x2 grid contain 'This is the first column.', 'This is the second', 'This is the third', and 'This is the fourth'. At the bottom of the preview area, there are buttons for 'Resources', '1x', '0.5x', '0.25x', and 'Run'.

We can now get to the nice part of this tutorial, the Bootstrap 4 columns. Columns are great! They help you divide the screen horizontally. If you place a single column in your row, it will take up all the width. If you add two columns, they will each take 1/2 from the width. And so it goes for any number of columns.

```
<div class="container">
  <div class="row">
    <div class="col">
      ...
    </div>
  </div>
  <div class="row">
    <div class="col">
      ...
    </div>
    <div class="col">
      ...
    </div>
  </div>
  <div class="row">
    <div class="col">
      ...
    </div>
    <div class="col">
      ...
    </div>
    <div class="col">
      ...
    </div>
  </div>
</div>
```

**Side note:** Columns are not coloured. I just added colours for a more visually compelling description/so they look pretty.

### Setting Sizes

Using the `.col` class sets the width for the column dynamically. That means that depending on the number of columns in a row, the width of a column will be the width of the container divided by the number of columns.

But there is another way to define columns. You can use classes for columns and define their size. By default, the Bootstrap 4 grid consists of 12 columns. You can select any size from 1 to 12 for your column. If you want 3 equal columns, you can use `.col-4` for each one (because  $3 \times 4$  cols each = 12). Or you can set different sizes for them. Here are some examples:

The screenshot shows a CodePen interface similar to the previous one, with tabs for 'HTML' and 'SCSS' on the left, and 'Result' on the right. At the top right is a 'CODEPEN' button with the text 'EDIT ON'. The preview area contains four yellow boxes arranged in two rows of two. The top row has two boxes, both containing 'This is a col-6.'. The bottom row also has two boxes, containing 'This is a col-5.' and 'This is a col-7.' respectively. The boxes are styled with rounded corners and shadows.

Resources      1x 0.5x 0.25x      Rerun

```
<div class="row">
  <div class="col-6">
    ...
  </div>
  <div class="col-6">
    ...
  </div>
</div>

<div class="row">
  <div class="col-5">
    ...
  </div>
  <div class="col-7">
    ...
  </div>
</div>

<div class="row">
  <div class="col-3">
    ...
  </div>
  <div class="col-4">
    ...
  </div>
</div>

<div class="row">
  <div class="col-6">
    ...
  </div>
  <div class="col-7">
    ...
  </div>
</div>
```

If the sum of the cols in your row doesn't get to 12, then they don't fill the whole row. And if it goes beyond 12 then it will move to the next line (it will only display the sum of the first elements <=12 on the first line).

## Setting Breakpoints

If you take the example above and want to display it on mobile, you may run into some problems. Displaying 5 columns on mobile will make the content unreadable. This is where one of the most powerful Bootstrap 4 components comes into play. In order to have different layouts on different screens you won't need to write media queries, instead you can use the column breakpoints.

Let's say you want to display, for example, 2 columns one after another vertically on small screens and on the same line on bigger screens. You will need to specify the breakpoint where the columns go on the same line.

In our example, we want the columns to arrange horizontally starting with the laptop. The breakpoint for the laptop resolution is `.col-lg`. If you open the Codepen in another window and see the page with a higher resolution, you will see the columns being aligned horizontally.

HTML      SCSS      Result      EDIT ON CODEPEN

Resources      1x 0.5x 0.25x      Rerun

The breakpoint for laptops is `.col-lg`. The code for this example looks like this:

```
<div class="row">
  <div class="col-lg">
    ...
  </div>
```

```
<div class="col-lg">
  ...
</div>
</div>
```

If you wanted for the 2 columns to go on the same line starting with larger mobile phones ( $\geq 576px$ ) you would use `.col-sm`, for tablets ( $\geq 768px$ ) `.col-md` and for extra large screens ( $\geq 1200px$ ) `.col-xl`.

It is useful to keep in mind that up to the given breakpoint, your columns will align vertically. This is one example of the fact that Bootstrap 4 is mobile-first. The default styling is for mobile devices and you specifically need to write what's the behaviour for bigger screens. And for the resolutions bigger than your breakpoint they will align horizontally.

## Setting Sizes and Breakpoints

You can combine the sizes and breakpoints and use a single class with the format `.col-[breakpoint]-[size]`. For example, if you want 3 columns of different sizes to align on a row starting with the laptop resolution you need to do this:

```
<div class="row">
  <div class="col-md-4">
    ...
  </div>
  <div class="col-md-3">
    ...
  </div>
  <div class="col-md-5">
    ...
  </div>
</div>
```

And the result will look like this:

The screenshot shows a CodePen interface with tabs for 'HTML', 'SCSS', and 'Result'. The 'Result' tab is active, displaying a laptop screen. Inside the laptop window, there are two horizontal boxes. The top box is pink and contains the text 'This is a column.'. The bottom box is purple and also contains the text 'This is a column.'. Below the laptop window, there is a toolbar with 'Resources', a zoom selector ('1x | 0.5x | 0.25x'), and a 'Rerun' button.

But what if you want to display 1 column per row on mobile devices, 2 columns per row on tablets and 4 starting with laptops? Then you add multiple classes for a single column to describe the behaviour for every resolution. Using multiple classes, you specify that the content will take 6 slots on tablets and 3 on laptops. The result will show like this:

The screenshot shows a CodePen interface with tabs for 'HTML', 'SCSS', and 'Result'. The 'Result' tab is active, displaying a laptop screen. Inside the laptop window, there are four horizontal boxes arranged in a 2x2 grid. The top-left box is blue and contains 'This is a column.'. The top-right box is purple and contains 'This is a column.'. The bottom-left box is blue and contains 'This is a column.'. The bottom-right box is purple and contains 'This is a column.'. Below the laptop window, there is a toolbar with 'Resources', a zoom selector ('1x | 0.5x | 0.25x'), and a 'Rerun' button.

Please open the CodePen on different screen sizes to see the differences. And here is the code behind it:

```
<div class="row">
  <div class="col-sm-6 col-lg-3">
    ...
  </div>
  <div class="col-sm-6 col-lg-3">
    ...
  </div>
  <div class="col-sm-6 col-lg-3">
    ...
  </div>
  <div class="col-sm-6 col-lg-3">
    ...
  </div>
</div>
```

```
</div>
```

As an exercise, you can try and create rows with different number of columns depending on the screensize on your own. You can fork the CodePen above and see its behaviour in your browser console.

## Offsetting Columns

If you don't want your columns to be next to each other, you can use the class `.offset-[breakpoint]-[size]` together with the `.col-[breakpoint]-[size]`. Using this class is the same as adding an empty column before your column. Here is a simple example:

The screenshot shows a CodePen interface with tabs for 'HTML', 'SCSS', and 'Result'. The 'Result' tab is active, displaying a layout with two purple rounded rectangular boxes. The left box contains the text 'This is an offset column.' and the right box contains 'This is a column.'. Below the preview area, there are buttons for 'Resources', '1x', '0.5x', '0.25x', and a 'Rerun' button.

```
<div class="row">
  <div class="col-sm-4 offset-sm-4">
    ...
  </div>
  <div class="col-sm-4">
    ...
  </div>
</div>
```

You can use the class on any column in the row. Here are some more examples:

The screenshot shows a CodePen interface with tabs for 'HTML', 'SCSS', and 'Result'. The 'Result' tab is active, displaying a layout with four colored rounded rectangular boxes arranged in a 2x2 grid. The top-left box is pink and contains 'This is a column.'. The top-right box is pink and contains 'This is an offset column.'. The bottom-left box is purple and contains 'This is an offset column.'. The bottom-right box is blue and contains 'This is an offset column.'.

```
<div class="row">
  <div class="col-sm-4">
    ...
  </div>
  <div class="col-sm-4 offset-sm-4">
    ...
  </div>
</div>

<div class="row">
  <div class="col-sm-4 offset-sm-2">
    ...
  </div>
  <div class="col-sm-4 offset-sm-2">
    ...
  </div>
</div>

<div class="row">
  <div class="col-sm-6 offset-sm-3">
    ...
  </div>
</div>
```

## Nesting Columns

This may come as a surprise, but you can add a row inside a column. That row (which

can have its own columns) will be nested inside the first column of the outer row.

will have the width of its parent column will then be divided into 12 (smaller) columns that you can reference through the `.col-*` classes. Let's take a look at what happens when we add a new row in a column:

The screenshot shows a CodePen interface with tabs for 'HTML', 'SCSS', and 'Result'. The 'Result' tab is active, displaying a grid layout. The HTML code in the editor is as follows:

```
<div class="row">
  <div class="col-sm-8">
    ...
    <div class="row">
      <div class="col-sm-5">
        ...
        <div class="col-sm-7">
          ...
        </div>
      </div>
    </div>
  </div>
<div class="col-sm-4">
  ...
</div>
</div>
```

The grid consists of two main columns. The first column contains a blue box labeled 'This is the first division.' and a purple box labeled 'This is the second division.' The second column contains a blue box labeled 'This is the first division.'.

This wraps up the basic knowledge regarding the Bootstrap 4 responsive grid system. This is a very important piece when learning to design responsively. If you have questions, please let me know in the comments, I will be glad to answer. I hope you will celebrate your achievement 😊



Photo credit to [Jonas Mosesson](#) for his shot.

## Further Reading & Watching

- [Official Bootstrap Documentation for the Grid](#)
- [This Video Tutorial](#) about the Grid from Scrimba

DAY 1: BOOTSTRAP 4 CDN

DAY 3: BOOTSTRAP 4 FLEX

Sharing is caring!

[Share](#)

[Tweet](#)

[LinkedIn](#)

[1 SHARES](#)

bootstrap 4



### Cristina Conacel

Cristina is a web developer and the owner of BootstrapBay. She likes foxes, clean design, writing blog posts and creating themes that are useful to other developers.

#### What do you think?

16 Responses



0 Comments    BootstrapBay

Login

Recommend

Tweet

Share

Sort by Best



Start the discussion...

LOG IN WITH



OR SIGN UP WITH DISQUS

Name

Be the first to comment.

#### ALSO ON BOOTSTRAPBAY

##### [Day 5: Bootstrap 4 Images Tutorial and Examples](#)

2 comments • a year ago



conacelelena — Hello Fatimah,The images I have used are uploaded to Amazon s3 and they take a bit longer to display, but they eventually do. I will

##### [The 7 Best Personal Websites We've Ever Seen](#)

5 comments • 2 years ago



Darryl Snow — Will throw mine into the ring:  
<https://yourweb.expert>

##### [Bootstrap Table Made Easy](#)

2 comments • 3 years ago



pavel — Thank you for the feedback, Sebastien.

##### [The Advantage of Creating Proposals Which Are Webpages](#)

1 comment • 3 months ago



Web Developer — wow

[Subscribe](#)

[Add Disqus to your site](#)

[Disqus' Privacy Policy](#)

**DISQUS**

□ DAY 1: BOOTSTRAP 4 CDN AND STARTER  
TEMPLATE

DAY 3: BOOTSTRAP 4 FLEX TUTORIAL AND  
EXAMPLES □

#### Company

Blog

About Us

RSS Feed

#### Help & Support

FAQ

Submission Guidelines

Payment Rates

Contact & Support

#### Resources

Resource List

Free Stock Photos

Graphic Design Tool

React Native Templates

#### Legal

Licenses

Terms & Conditions

Privacy Policy

#### Join our community

Your Email...

JOIN

Follow us



SELL TEMPLATES

Earn 55% and  
more on all sales

AFFILIATE

Earn 30% on all  
sales you refer

41,680  
Members

12,489  
Downloads

77  
Authors

215  
Themes

© 2019 BootstrapBay. All rights reserved.

Secure Payment: FastSpring