

# Layout.

## Objectives:

By the end of this chapter, you should be able to:

- Explain the difference between `.container` and `.container-fluid`
- Explain how the grid system works in Bootstrap
- Compare and contrast `xl`, `lg`, `md`, `sm`, and `auto` columns
- Add offsets to columns
- Add margins and padding to elements using helper classes

## Layout

Probably the most commonly used feature of Bootstrap is its grid system. With the right combination of CSS classes, you can build sophisticated grids (based on 12 columns) to lay out your page. Bootstrap will handle the responsive design based on the classes that you choose.

### Containers

Any layout that you build with Bootstrap should begin with a container. According to the [docs](#), "Containers are the most basic layout element in Bootstrap and are **required when using our default grid system**" (emphasis original). We'll get to the grid system momentarily, but for now let's talk about containers.

A container is just a `div` with one of two classes: either `container` or `container-fluid`. The difference is that `container` elements have fixed widths that change at a finite number of break points, while `container-fluid` elements always take up 100% of the available width.

Here's a quick example that highlights the visual difference between `.container` and `.container-fluid`:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Bootstrap Container Example</title>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
beta/css/bootstrap.min.css"
integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vs1U6fwYXjCFTcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M"
crossorigin="anonymous">
</head>
<body>
  <div class="container alert alert-primary">
    <h1>Here is some content inside of a container!</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. A dolore ea sunt
porro, aliquam ipsa, ipsam, necessitatibus earum similique iure asperiores. Unde
quidem voluptatibus enim nam dolorum, harum provident, totam?</p>
```

```

</div>
<div class="container-fluid alert alert-secondary">
  <h1>Here is some content inside of a container-fluid!</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Beatae ipsam
quos minus deleniti maxime eum ullam praesentium corporis veniam, in vitae itaque
modi consequatur saepe tempore. Ratione labore, animi laudantium.</p>
</div>
</body>
</html>

```

Note that as you drag the window to decrease the width, the `container` element has a few different fixed widths until you get to very narrow screens. The `container-fluid` element, however, always takes up 100% of the width. The `alert` classes also come with bootstrap, and provide some default stylings for simple messages you may want to display to the user. [Here's](#) a link to the full list of available alert classes.

## The Grid System

Many CSS frameworks, Bootstrap included, come with some sort of grid system. A grid system is a way to organize the content on your page in terms of rows and columns. For example, maybe you're building a page that has a sidebar and a main content area. With a grid system, you can specify how many columns you want the sidebar to take up, and how many columns you want the main content area to take up.

Bootstrap's grid system uses a twelve-column grid layout. This means that when you're using the system, you can specify to bootstrap how many columns you want your content to take up, out of a possible maximum of twelve columns.

Let's take a look at an example of the grid system in action. Once again, let's use alerts to clearly highlight where columns are showing up on the screen.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Bootstrap Grid System Example</title>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
beta/css/bootstrap.min.css"
integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vs1U6fwYXjCfTcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M"
crossorigin="anonymous">
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col-6 alert alert-primary">I always take up half a row.</div>
      <div class="col-6 alert alert-secondary">I always take up half a row.</div>
    </div>
    <div class="row">
      <div class="col-12 col-md-8 alert alert-success">I take up the whole row on
extra small screens, 2/3rds on medium (or larger) screens.</div>

```

```

    <div class="col-6 col-md-4 alert alert-danger">I take up half a row on
extra small screens, 1/3rd on medium (or larger) screens.</div>
</div>
<div class="row">
    <div class="col-12 col-sm-6 col-md-4 alert alert-warning">I take up a row
on extra small screens, half a row on small screens, and 1/3rd on medium (or
larger) screens.</div>
    <div class="col-12 col-sm-6 col-md-4 alert alert-info">I take up a row on
extra small screens, half a row on small screens, and 1/3rd on medium (or larger)
screens.</div>
    <div class="col-6 col-md-4 col-lg-12 alert alert-dark">I take up half a row
on extra small screens, 1/3rd on medium (or larger) screens, and a full row on
large screens.</div>
</div>
</div>
</body>
</html>

```

There's a lot going on here in terms of the classes we're using, but before we dig into the code, take a look at what happens to the columns as you adjust the width of the window. You should see that the columns adjust and take up different fractions of the window's width as you adjust that width.

Let's try to understand how Bootstrap's grid system works. In order to make use of the grid system, you first need to have a container, created with either the `container` or `container-fluid` class. After that, you can create a row for the grid using the `row` class. Inside of the `rows`, you should only place column classes, and finally, your content will live inside of the columns.

As you can see in the example above, you have many different options for column classes. In our first row, we've created two columns that each take up 6 of the available 12 columns, or 50% each.

Our next row is a bit more complicated, because each div has multiple column classes assigned to it. The first div has a `col-12` class, meaning it will take up all 12 columns. But it also has a `col-md-8` class; this means that it will take up only 8 columns on a screen that's medium-sized or larger. Similarly, the second div will take up 4 columns on a medium (or larger) screen, and 6 columns otherwise. Since  $12 + 6$  is greater than 12, on smaller screens the two columns stack, while on medium screens they fit nicely in the row.

The third row is similar, just with more options for the layout based on the screen size. Bootstrap has breakpoints and associated column classes for five different screen sizes: extra small, small, medium, large, and extra large. Here's a quick summary of the sizes and associated classes:

Size	Extra Small	Small	Medium	Large	Extra Large
Break Point	N/A	576px	768px	992px	1200px
Class Prefix	<code>col-</code>	<code>col-sm-</code>	<code>col-md</code>	<code>col-lg</code>	<code>col-xl</code>

You can also have your columns auto-adjust to the available width by simply giving them a class of `col`. When you do this, the columns will take up all of the available width. Here's an example:

```

<!DOCTYPE html>
<html lang="en">

```

```

<head>
  <meta charset="UTF-8">
  <title>Bootstrap Grid System Example, Part 2</title>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
beta/css/bootstrap.min.css"
integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vs1U6fwYXjCftcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M"
crossorigin="anonymous">
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col alert alert-secondary">I'm one of three columns!</div>
      <div class="col alert alert-primary">I'm one of three columns!</div>
      <div class="col alert alert-secondary">I'm one of three columns!</div>
    </div>
    <div class="row">
      <div class="col alert alert-warning">I'm one of four columns!</div>
      <div class="col alert alert-success">I'm one of four columns!</div>
      <div class="col alert alert-danger">I'm one of four columns!</div>
      <div class="col alert alert-info">I'm one of four columns!</div>
    </div>
    <div class="row">
      <div class="col-7 alert alert-light">I take up 7/12 columns!</div>
      <div class="col alert alert-dark">I take up half the remainder!</div>
      <div class="col alert alert-dark">I also take up half the remainder!</div>
    </div>
  </div>
</body>
</html>

```

As long as there's enough space to fit the contents of the columns, the columns in the first row each take up one third of the width, while the columns in the second row take up one fourth of width, even though all of these divs use the same class: `col`. Similarly, the last row specifies that the first div should take up seven columns, while the other two divs should split the remaining width.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Bootstrap Grid System Example, Part 3</title>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
beta/css/bootstrap.min.css"
integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vs1U6fwYXjCftcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M"
crossorigin="anonymous">
</head>
<body>
  <div class="container">

```

```

    <div class="row">
      <div class="col-6 ml-auto mr-auto alert alert-primary">I'm in the middle!
    </div>
  </div>
  <div class="row">
    <div class="col-4 mr-auto alert alert-success">I'm on the left.</div>
    <div class="col-4 alert alert-warning">I'm on the right.</div>
  </div>
</div>
</body>
</html>

```

You can also offset columns if you want to add some space between them or around them. Under the hood, the grid system in version 4 of Bootstrap uses flexbox, which means there are several ways you can offset columns. One approach is to set the `justify-content` property on the row. Bootstrap provides you with classes to modify this property on the rows.

Another approach is to modify the margins. Bootstrap gives you two classes which can be helpful for offsetting columns using margin: `ml-auto`, which sets the margin-left to auto, and `mr-auto`, which sets the margin-right to auto. Here's an example of the two approaches:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Bootstrap Grid System Example, Part 4</title>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
beta/css/bootstrap.min.css"
integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vs1U6fwYXjCftcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M"
crossorigin="anonymous">
</head>
<body>
  <div class="container">
    <div class="row justify-content-center">
      <div class="col-6 alert alert-primary">I'm in the middle using justify-
content!</div>
    </div>
    <div class="row justify-content-between">
      <div class="col-4 alert alert-success">I'm on the left using justify-
content.</div>
      <div class="col-4 alert alert-warning">I'm on the right using justify-
content.</div>
    </div>
    <div class="row">
      <div class="col-6 ml-auto mr-auto alert alert-primary">I'm in the middle
using margin!</div>
    </div>
    <div class="row">

```

```

        <div class="col-4 mr-auto alert alert-success">I'm on the left using
margin.</div>
        <div class="col-4 alert alert-warning">I'm on the right using margin.</div>
    </div>
</div>
</body>
</html>

```

As you can see, the first two rows offset the columns by modifying the `justify-content` property. The second two rows have the same effect, but go about it using `margin`.

We can also nest rows inside of columns. Every new row gains access to its own 12-column grid system. Here's an example that uses this kind of nesting:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Bootstrap Grid System Example, Part 5</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
beta/css/bootstrap.min.css"
integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vs1U6fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M"
crossorigin="anonymous">
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col-8 alert alert-primary">
                I'm a column!
                <div class="row">
                    <div class="col-7 alert alert-success">I'm a nested column!</div>
                    <div class="col-5 alert alert-warning">Me too!</div>
                </div>
            </div>
            <div class="col-4 alert alert-secondary">Don't forget about me - I'm a
column too!</div>
        </div>
    </div>
</body>
</html>

```

The grid system has some other features as well, but this covers the basics. If you're using Bootstrap to build a responsive site, the grid system is going to be one of your most commonly-used tools. If you'd like to read more about it, check out the [docs](#).

## Margin and Padding

In the previous example, we saw how Bootstrap classes like `mr-auto` and `ml-auto` could be used to add margin to elements on the page.

Bootstrap comes with a number of classes to help add space between elements on the page. These elements all come in the form `{property}{sides}-{size}` or `{property}{sides}-{breakpoint}-{size}`, where `property`, `sides`, `breakpoint`, and `size` are one of the following:

- `property`: `m` (margin) or `p` (padding)
- `sides`: `t` (top), `b` (bottom), `l` (left), `r` (right), `x` (left and right), or `y` (top and bottom). You can also leave this value blank to target all directions.
- `breakpoint`: `sm`, `md`, `lg`, or `xl`. You can also leave this value blank to target `xs`.
- `size`: a number from 0 to 5. Larger numbers indicate more space.

Here's a quick example highlighting some of these classes. As you can see, these classes allow to quickly add margin or padding to elements on the page without needing to modify a stylesheet.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Bootstrap Form Example</title>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
beta/css/bootstrap.min.css"
integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vslU6fwYXjCftcEpHbNj0lyAFsXTsjBbfaDjzALeQsN6M"
crossorigin="anonymous">
</head>
<body>
  <h1>Spacing classes in Bootstrap</h1>
  <hr>
  <div class="alert alert-primary px-5 py-md-3">I have lots of padding (but
slightly less on medium screens)!</div>
  <div class="alert alert-primary px-0 py-0 m-5">I have lots of margin, but no
padding!</div>
  <div class="alert alert-danger p-0 m-0">I have neither margin nor padding :
(</div>
  <div class="alert alert-success pt-5 pb-0 pl-2 pr-4 mt-0 ml-5 mr-1 mb-5">My
spacing is all over the place!</div>
  <hr>
</body>
</html>
```

For more on these Bootstrap classes, check out the [docs](#).