10. Mastering Layout with bootstrap

By far the most useful part of Bootstrap is the grid, so this is the first thing you should master. The Bootstrap grid gives you a framework to create any layout that you can think of. Now, Bootstrap layouts have three main pieces. The first is containers, which can be either responsive fixed width that snap to certain breakpoints, or completely fluid containers which take up 100% of the width of the view port, which is the browser window or the width of your device. Inside containers, you can also place content in rows and columns.

Bootstrap uses a 12-column grid system that has breakpoints for extra small, small, medium, large, and extra large devices. If you're coming from a previous version of Bootstrap, note that there is an additional breakpoint for smaller devices. The grid is extremely flexible. You can create multiple layouts for different breakpoints, reset, offset, nest, and even customize the order of the columns at different breakpoints. Bootstrap's grid is very powerful.

If you can think of a layout, you can build it easily and responsively with the Bootstrap grid. The more time you spend learning how to use it, the easier it will be to get your work done.
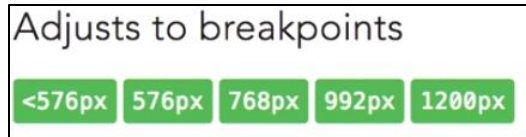
Containers and rows

One of the main reasons people use Bootstrap is to have access to an amazing grid that makes layout easy to do and responsive. This is one of the most important things you should master in Bootstrap, so let's take a look. First of all, the grid is a responsive 12-column system for creating just about any layout you can think of. It uses a technology called Flexbox that makes it easier to create complex layout with minimal code. In order to work with the grid, you need to master three simple components.

The first is containers, which can be used with or without the grid to `align-content` either to the viewport, or center it around a set of breakpoints. Next is rows and columns. They work together to allow you to create the layouts. The rows prepare the columns for layout, and the columns are complex and extremely flexible, so we'll cover them in a separate video in detail. So first, let's go ahead and take a look at containers. There are two different types of containers.

`container` class is used for regular containers, which center content and snap to certain grid points.

`container-fluid` are fluid containers, which are always the full width of the viewport, which means the width of the device or the browser window. One of the reasons you use a container is because you get a 15 pixel padding on each side to make sure it works well with backgrounds and other elements. Now here's the breakpoints that the regular container will adjust to. First is anything smaller than 576 pixels, and so the bootstrap grid adjusts to content that is smaller than 576 pixels, and then some of these other breakpoints.

Rows and columns

Bootstrap's grid system allows up to 12 columns across the page. If you do not want to use all 12 column individually, you can group the columns together to create wider columns:

Bootstrap's grid system is responsive, and the columns will re-arrange depending on the screen size: On a big screen it might look better with the content organized in three columns, but on a small screen it would be better if the content items were stacked on top of each other.

Grid Classes

The Bootstrap 4 grid system has five classes:

- `.col`– (extra small devices - screen width less than 576px)
- `.col-sm`– (small devices - screen width equal to or greater than 576px)
- `.col-md`– (medium devices - screen width equal to or greater than 768px)
- `.col-lg`– (large devices - screen width equal to or greater than 992px)
- `.col-xl`– (xlarge devices - screen width equal to or greater than 1200px)

The classes above can be combined to create more dynamic and flexible layouts.

Tip: Each class scales up, so if you wish to set the same widths for `sm` and `md`, you only need to specify `sm`.
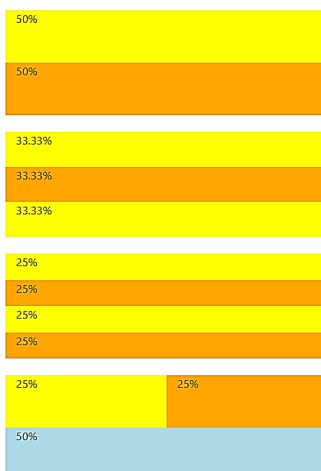
Screen Breakpoints

Some Bootstrap 4 grid system rules:

- Rows must be placed within a `.container` (fixed-width) or `.container-fluid` (full-width) for proper alignment and padding
- Use rows to create horizontal groups of columns
- Content should be placed within columns, and only columns may be immediate children of rows
- Predefined classes like `.row` and `.col-sm-4` are available for quickly making grid layouts
- Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on .rows
- Grid columns are created by specifying the number of 12 available columns you wish to span. For example, three equal columns would use three `.col-sm-4`
- Column widths are in percentage, so they are always fluid and sized relative to their parent element

Example) Create row with different width:

Computer view

50%	50%

33.33%	33.33%	33.33%

25%	25%	25%	25%

25%	25%	50%

Smartphone view

50%
50%

33.33%
33.33%
33.33%

25%
25%
25%
25%

25%	25%

50%

```
<div class="container-fluid">
    <!-- Control the column width, and how they should appear on different
devices -->
    <div class="row">
      <div class="col-sm-6" style="background-color:yellow;">50%</div>
      <div class="col-sm-6" style="background-color:orange;">50%</div>
    </div>
    <br>

    <div class="row">
      <div class="col-sm-4" style="background-color:yellow;">33.33%</div>
      <div class="col-sm-4" style="background-color:orange;">33.33%</div>
      <div class="col-sm-4" style="background-color:yellow;">33.33%</div>
    </div>
    <br>

    <!-- Or let Bootstrap automatically handle the layout -->
    <div class="row">
      <div class="col-sm" style="background-color:yellow;">25%</div>
      <div class="col-sm" style="background-color:orange;">25%</div>
      <div class="col-sm" style="background-color:yellow;">25%</div>
      <div class="col-sm" style="background-color:orange;">25%</div>
    </div>
    <br>

    <div class="row">
      <div class="col" style="background-color:yellow;">25%</div>
      <div class="col" style="background-color:orange;">25%</div>
      <div class="col-sm-6" style="background-color:blue;">50%</div>
    </div>
```

To take full advantage of the 12 column grid, you have to master the use of columns.
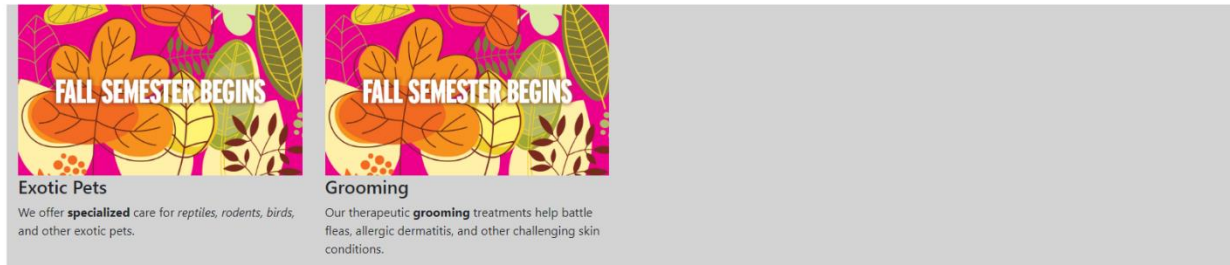

Multiple columns

The grid in Bootstrap is powerful because of the many ways that we can use it. One thing we can do is specify multiple-column breakpoints for each of our grid elements.

	Extra small < 576px	Small ≥ 576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-

Offsetting columns

Another way to control the grid is to offset columns in order to move their position on the grid. For example, a grid without offset looks as the following:

```
<div class="row" style="background: lightgray;">
   <section class="col-md-3" >
       <img src="images/image.jpg" alt="sample image" style="width:100%;" >
       <h4>Exotic Pets</h4>
       <p>We offer <strong>specialized</strong> care for <em>reptiles,
       rodents, birds,</em> and other exotic pets.</p>
   </section>
   <section class="col-md-3">
       <img src="images/image.jpg" alt="sample image" style="width:100%;" >
       <h4>Grooming</h4>
       <p>Our therapeutic <span class="font-weight-bold">grooming</span>
       treatments help battle fleas, allergic dermatitis, and other challenging
       skin conditions.</p>
   </section>
</div><!-- end of row -->
```

offset class in columns is used to shift but the number of columns over. From the previous example, if we add offset to **<section class="col-md-3 offset-sm-1">** the webpage will look as the following:

l

Nested column

You can easily put a whole new grade of columns inside an existing column. Now, that's called nesting. So, let's take a look at how that works. To nest columns, you simply create a new row inside an existing column. Now, this will create a new regular 12-column grid inside that existing column. And inside that column, you can use the same set of classes that you've been using so far. For example, we can create one main image with six sub-images as the following:

Exotic Pets
We offer **specialized** care for *reptiles, rodents, birds,* and other exotic pets.

HTML

```
<div class="row">
    <section class="col-sm-8">
      <img src="images/image.jpg" alt="sample image">
      <h4>Exotic Pets</h4>
      <p>We offer <strong>specialized</strong> care for <em>reptiles,
      rodents, birds,</em> and other exotic pets.</p>
    </section>
    <section class="col-sm-4">
      <div class="row">
        <div class="col-2 col-sm-4">
          <img src="images/image.jpg" class="img-thumbnail"></div>
        <div class="col-2 col-sm-4">
          <img src="images/image.jpg" class="img-thumbnail"></div>
        <div class="col-2 col-sm-4">
          <img src="images/image.jpg" class="img-thumbnail"></div>
        <div class="col-2 col-sm-4">
          <img src="images/image.jpg" class="img-thumbnail"></div>
        <div class="col-2 col-sm-4">
          <img src="images/image.jpg" class="img-thumbnail"></div>
        <div class="col-2 col-sm-4">
          <img src="images/image.jpg" class="img-thumbnail"></div>
      </div> <!-- end of subnets row -->
    </section>
</div><!—end of row -->
```

We can also set one main image with six sub-images below the main image:

Exotic Pets

We offer **specialized** care for *reptiles, rodents, birds,* and other exotic pets.

Custom order

One of the more advanced ways that you can control layouts is to customize the order of elements. It helps you to reorder the way that columns appear and you can do this at different screen widths. Class *col order-2* change the order of the element in a row

For example, if we have a session with col

```
<section class="col">
```


If we add order-3 to the first session, that session will be moved to the third position

```
<section class="col order-3">
```

Order

2. Grooming

Our therapeutic **grooming** treatments help battle fleas, allergic dermatitis, and other challenging skin conditions.

3. General Health

Wellness and senior exams, ultrasound, x-ray, and dental cleanings are just a few of our general health services.

1. Exotic Pets

We offer **specialized** care for *reptiles, rodents, birds,* and other exotic pets.