

Content

- **container-fluid**
- **img-responsive**
- **text responsive**
- **button responsive**
- **col-md-***
- **Font Awesome Icon (FAI) with link**
- **form-control**

Bootstrap: Use Responsive Design with Bootstrap Fluid Containers

Bootstrap will figure out how wide your screen is and respond by resizing your HTML elements - hence the name responsive design.

With responsive design, there is no need to design a mobile version of your website. It will look good on devices with screens of any width.

You can add Bootstrap to any app by adding the following code to the top of your HTML:

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-BVYiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous"/>
```

In this case, we've already added it for you to this page behind the scenes. Note that using either `>` or `/>` to close the link tag is acceptable.

To get started, we should nest all of our HTML (except the link tag and the style element) in a div element with the class **container-fluid**.

=====

Bootstrap: Make Images Mobile Responsive

First, add a new image below the existing one. Set its src attribute to `https://bit.ly/fcc-running-cats`.

It would be great if this image could be exactly the width of our phone's screen.

Fortunately, with Bootstrap, all we need to do is add the **img-responsive** class to your image. Do this, and the image should perfectly fit the width of your page.

=====

Bootstrap: Center Text with Bootstrap

Now that we're using Bootstrap, we can center our heading element to make it look better. All we need to do is add the class `text-center` to our h2 element.

`text-primary`, `text-danger` are also classes that can be used.

=====

Bootstrap: Create a Bootstrap Button

Bootstrap has its own styles for button elements, which look much better than the plain HTML ones.

Create a new button element below your large kitten photo. Give it the `btn` and `btn-default` classes, as well as the text of "Like".

Bootstrap: Create a Block Element Bootstrap Button

Normally, your button elements with the `btn` and `btn-default` classes are only as wide as the text that they contain. For example:

```
<button class="btn btn-default">Submit</button>
```

This button would only be as wide as the word "Submit".

By making them block elements with the additional class of `btn-block`, your button will stretch to fill your page's entire horizontal space and any elements following it will flow onto a "new line" below the block.

```
<button class="btn btn-default btn-block">Submit</button>
```

This button would take up 100% of the available width.

Note that these buttons still need the `btn` class.

Bootstrap: Taste the Bootstrap Button Color Rainbow

The `btn-primary` class is the main color you'll use in your app. It is useful for highlighting actions you want your user to take.

Replace Bootstrap's `btn-default` class with `btn-primary` in your button.

Note that this button will still need the `btn` and `btn-block` classes.

Bootstrap: Call out Optional Actions with btn-info

Bootstrap comes with several pre-defined colors for buttons. The btn-info class is used to call attention to optional actions that the user can take.

Create a new block-level Bootstrap button below your "Like" button with the text "Info", and add Bootstrap's **btn-info** and btn-block classes to it.

Note that these buttons still need the btn and btn-block classes.

Bootstrap: Warn Your Users of a Dangerous Action with btn-danger

Bootstrap comes with several pre-defined colors for buttons. The **btn-danger** class is the button color you'll use to notify users that the button performs a destructive action, such as deleting a cat photo.

Create a button with the text "Delete" and give it the class btn-danger.

Note that these buttons still need the btn and btn-block classes.

=====

Bootstrap: Use the Bootstrap Grid to Put Elements Side By Side

Bootstrap uses a responsive 12-column grid system, which makes it easy to put elements into rows and specify each element's relative width. Most of Bootstrap's classes can be applied to a div element.

Bootstrap has different column width attributes that it uses depending on how wide the user's screen is. For example, phones have narrow screens, and laptops have wider screens.

Take for example Bootstrap's **col-md-*** class. Here, md means medium, and * is a number specifying how many columns wide the element should be. In this case, the column width of an element on a medium-sized screen, such as a laptop, is being specified.

In the Cat Photo App that we're building, we'll use **col-xs-***, where xs means extra small (like an extra-small mobile phone screen), and * is the number of columns specifying how many columns wide the element should be.

Put the Like, Info and Delete buttons side-by-side by nesting all three of them within one **<div class="row">** element, then each of them within a **<div class="col-xs-4">** element.

The row class is applied to a div, and the buttons themselves can be nested within it.

Bootstrap: Create a Custom Heading

We will make a simple heading for our Cat Photo App by putting the title and relaxing cat image in the same row.

Remember, Bootstrap uses a responsive grid system, which makes it easy to put elements into rows and specify each element's relative width. Most of Bootstrap's classes can be applied to a div element.

Nest your first image and your h2 element within a single `<div class="row">` element. Nest your h2 element within a `<div class="col-xs-8">` and your image in a `<div class="col-xs-4">` so that they are on the same line.

=====

Bootstrap: Add Font Awesome Icons to our ButtonsPassed

Font Awesome is a convenient library of icons. These icons can be webfonts or vector graphics. These icons are treated just like fonts. You can specify their size using pixels, and they will assume the font size of their parent HTML elements.

You can include Font Awesome in any app by adding the following code to the top of your HTML:

```
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.1/css/all.css"
integrity="sha384-50oBUHEmvpQ+1lW4y57PTFmhCaXp0ML5d60M1M7uH2+nqUivzlebhndOJK28anvf"
crossorigin="anonymous">
```

In this case, we've already added it for you to this page behind the scenes.

The `i` element was originally used to make other elements italic, but is now commonly used for icons. You can add the Font Awesome classes to the `i` element to turn it into an icon, for example:

```
<i class="fas fa-info-circle"></i>
```

Note that the `span` element is also acceptable for use with icons.

=====

Bootstrap: Responsively Style Radio Buttons

You can use Bootstrap's `col-xs-*` classes on form elements, too! This way, our radio buttons will be evenly spread out across the page, regardless of how wide the screen resolution is.

Nest both your radio buttons within a `<div class="row">` element. Then nest each of them within a `<div class="col-xs-6">` element.

Note: As a reminder, radio buttons are input elements of type `radio`.

Bootstrap: Style Text Inputs as Form Controls

You can add the fa-paper-plane Font Awesome icon by adding `<i class="fa fa-paper-plane"></i>` within your submit button element.

Give your form's text input field a class of form-control. Give your form's submit button the classes btn btn-primary. Also give this button the Font Awesome icon of fa-paper-plane.

All textual `<input>`, `<textarea>`, and `<select>` elements with the class `.form-control` have a width of 100%.