**Use Responsive Design with Bootstrap Fluid Containers**

In the HTML5 and CSS section of freeCodeCamp we built a Cat Photo App. Now let's go back to it. This time, we'll style it using the popular Bootstrap responsive CSS framework.

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-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+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.

First, add a new image below the existing one. Set its src attribute to https://cdn.freecodecamp.org/curriculum/cat-photo-app/running-cats.jpg.

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.

# 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.

Remember that you can add several classes to the same element by separating each of them with a space, like this:

<h2 class="red-text text-center">your text</

# 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.

# 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.

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.

# 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.

# 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 class to it.

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

# 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.

# 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.

# 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.

# Ditch Custom CSS for Bootstrap

We can clean up our code and make our Cat Photo App look more conventional by using Bootstrap's built-in styles instead of the custom styles we created earlier.

Don't worry - there will be plenty of time to customize our CSS later.

Delete the .red-text, p, and .smaller-image CSS declarations from your style element so that the only declarations left in your style element are h2 and thick-green-border.

Then delete the p element that contains a dead link. Then remove the red-text class from your h2 element and replace it with the text-primary Bootstrap class.

Finally, remove the smaller-image class from your first img element and replace it with the img-responsive class.

# Use a span to Target Inline Elements

You can use spans to create inline elements. Remember when we used the btn-block class to make the button fill the entire row?

normal button

btn-block button

That illustrates the difference between an "inline" element and a "block" element.

By using the inline span element, you can put several elements on the same line, and even style different parts of the same line differently.

Using a span element, nest the word love inside the p element that currently has the text Things cats love. Then give the span the class text-danger to make the text red.

Here's how you would do this for the p element that has the text Top 3 things cats hate:

<p>Top 3 things cats <span class

# 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.

Notice how the image is now just the right size to fit along the text?

# 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.

**Add Font Awesome Icons to our Buttons**

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+nqUivzIebhndOJK28anvf" 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.

Use Font Awesome to add a thumbs-up icon to your like button by giving it an i element with the classes fas and fa-thumbs-up. Make sure to keep the text Like next to the icon.

**Add Font Awesome Icons to all of our Buttons**

Font Awesome is a convenient library of icons. These icons can be web fonts 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.

Use Font Awesome to add an info-circle icon to your info button and a trash icon to your delete button.

**Note:** You can use either i or span elements to complete this challenge.

# 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.

**Responsively Style Checkboxes**

Since Bootstrap's col-xs-\* classes are applicable to all form elements, you can use them on your checkboxes too! This way, the checkboxes will be evenly spread out across the page, regardless of how wide the screen resolution is.

Nest all three of your checkboxes in a <div class="row"> element. Then nest each of them in a <div class="col-xs-4"> element.

# 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%.

# Line up Form Elements Responsively with Bootstrap

Now let's get your form input and your submission button on the same line. We'll do this the same way we have previously: by using a div element with the class row, and other div elements within it using the col-xs-\* class.

Nest both your form's text input and submit button within a div with the class row. Nest your form's text input within a div with the class of col-xs-7. Nest your form's submit button in a div with the class col-xs-5.

This is the last challenge we'll do for our Cat Photo App for now. We hope you've enjoyed learning Font Awesome, Bootstrap, and responsive design!

# Create a Bootstrap Headline

Now let's build something from scratch to practice our HTML, CSS and Bootstrap skills.

We'll build a jQuery playground, which we'll soon put to use in our jQuery challenges.

To start with, create an h3 element, with the text jQuery Playground.

Color your h3 element with the text-primary Bootstrap class, and center it with the text-center Bootstrap class.

# House our page within a Bootstrap container-fluid div

Now let's make sure all the content on your page is mobile-responsive.

Let's nest your h3 element within a div element with the class container-fluid.

# Create a Bootstrap Row

Now we'll create a Bootstrap row for our inline elements.

Create a div element below the h3 tag, with a class of row.

# Split Your Bootstrap Row

Now that we have a Bootstrap Row, let's split it into two columns to house our elements.

Create two div elements within your row, both with the class col-xs-6.

# Create Bootstrap Wells

Bootstrap has a class called well that can create a visual sense of depth for your columns.

Nest one div element with the class well within each of your col-xs-6 div elements.

# Add Elements within Your Bootstrap Wells

Now we're several div elements deep on each column of our row. This is as deep as we'll need to go. Now we can add our button elements.

Nest three button elements within each of your div elements having the class name well.

# Apply the Default Bootstrap Button Style

Bootstrap has another button class called btn-default.

Apply both the btn and btn-default classes to each of your button elements.

# Create a Class to Target with jQuery Selectors

Not every class needs to have corresponding CSS. Sometimes we create classes just for the purpose of selecting these elements more easily using jQuery.

Give each of your button elements the class target.

# Add id Attributes to Bootstrap Elements

Recall that in addition to class attributes, you can give each of your elements an id attribute.

Each id must be unique to a specific element and used only once per page.

Let's give a unique id to each of our div elements of class well.

Remember that you can give an element an id like this:

<div class="well" id="center-well">

Give the well on the left the id of left-well. Give the well on the right the id of right-well.

# Label Bootstrap Wells

For the sake of clarity, let's label both of our wells with their ids.

Above your left-well, inside its col-xs-6 div element, add an h4 element with the text #left-well.

Above your right-well, inside its col-xs-6 div element, add an h4 element with the text #right-well.

# Label Bootstrap Buttons

Just like we labeled our wells, we want to label our buttons.

Give each of your button elements text that corresponds to its id selector.

# Use Comments to Clarify Code

When we start using jQuery, we will modify HTML elements without needing to actually change them in HTML.

Let's make sure that everyone knows they shouldn't actually modify any of this code directly.

Remember that you can start a comment with <!-- and end a comment with -->

Add a comment at the top of your HTML that says Code below this line should not be changed