FREE CODE CAMP TUTORIALS

With CSS, there are hundreds of CSS properties that you can use to change the way an element looks on your page.

When you entered <h2 style="color: red">CatPhotoApp</h2>, you were giving that individual h2 element an inline style.

That's one way to add style to an element, but a better way is by using CSS, which stands for Cascading Style Sheets.

At the top of your code, create a styleelement like this:

<style>  
</style>

Inside that style element, you can create a CSS selector for all h2 elements. For example, if you wanted all h2elements to be red, your style element would look like this:

<style>  
  h2 {color: red;}  
</style>

Note that it's important to have both opening and closing curly braces ({ and }) around each element's style. You also need to make sure your element's style is between the opening and closing style tags. Finally, be sure to add the semicolon to the end of each of your element's styles.

Import a Google Font

Now, let's import and apply a Google font (note that if Google is blocked in your country, you will need to skip this challenge).

First, you'll need to make a call to Google to grab the Lobster font and load it into your HTML.

Copy the following code snippet and paste it into the top of your code editor:

<link href="https://fonts.googleapis.com/css?family=Lobster" rel="stylesheet" type="text/css">

Now you can set Lobster as a font-family value on your h2 element.

Specify How Fonts Should Degrade

There are several default fonts that are available in all browsers. These include Monospace, Serif and Sans-Serif

When one font isn't available, you can tell the browser to "degrade" to another font.

For example, if you wanted an element to use the Helvetica font, but also degrade to the Sans-Serif font when Helvetica wasn't available, you could use this CSS style:

p {  
  font-family: Helvetica, Sans-Serif;  
}

Turn an Image into a Link

You can make elements into links by nesting them within an a element.

Nest your image within an a element. Here's an example:

<a href="#"><img src="https://bit.ly/fcc-running-cats" alt="Three kittens running towards the camera. "></a>

Remember to use # as your aelement's href property in order to turn it into a dead link.

Create a Form Element

You can build web forms that actually submit data to a server using nothing more than pure HTML. You can do this by specifying an action on your formelement.

For example:

<form action="/url-where-you-want-to-submit-form-data"></form>

Use HTML5 to Require a Field

You can require specific form fields so that your user will not be able to submit your form until he or she has filled them out.

For example, if you wanted to make a text input field required, you can just add the word required within your inputelement, you would use: <input type="text" required>

Use Clockwise Notation to Specify the Padding of an Element

Instead of specifying an element's padding-top, padding-right, padding-bottom, and padding-leftproperties, you can specify them all in one line, like this:

padding: 10px 20px 10px 20px;

These four values work like a clock: top, right, bottom, left, and will produce the exact same result as using the side-specific padding instructions.

**BOOTSTRAP**

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/3.3.1/css/bootstrap.min.css"/>

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

To get started, we should nest all of our HTML in a div element with the class container-fluid.

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.

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</h2>

Create a Bootstrap Button

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

Create a new button element below your large kitten photo. Give it the class btn and the text of "Like".

Normally, your button elements with a class of btn are only as wide as the text that they contain. For example:

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

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

Submit

Note that these buttons still need the btn class.

Add Bootstrap's btn-block class to your Bootstrap button.

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.

Add Bootstrap's btn-primary class to your button.

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

Call out Optional Actions with Button 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.

Warn your Users of a Dangerous Action

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.

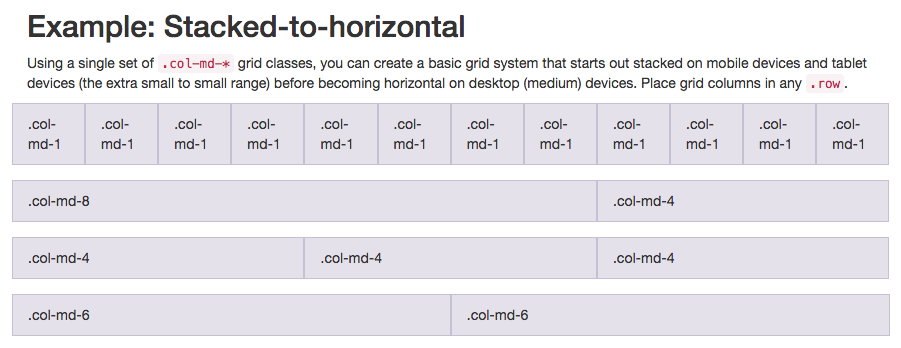
Create a button with the text "Delete" and give it the class btn-danger.

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 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.

Here's a diagram of how Bootstrap's 12-column grid layout works:

**[](https://i.imgur.com/FaYuui8.png)**

Note that in this illustration, the col-md-\* class is being used. Here, mdmeans 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 Deletebuttons 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.

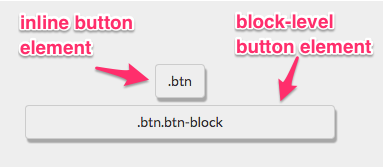
Then delete the p element that contains a dead link. Then remove the red-text class from your h2 element and replace it with the text-primaryBootstrap class.

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

Use Spans for 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?

This image illustrates the difference between inline elements and block-level elements:

**[](https://i.imgur.com/O32cDWE.png)**

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

Nest the word "love" in your "Things cats love" element below within a spanelement. Then give that span the class text-danger to make the text red.

Here's how you would do this with the "Top 3 things cats hate" element:

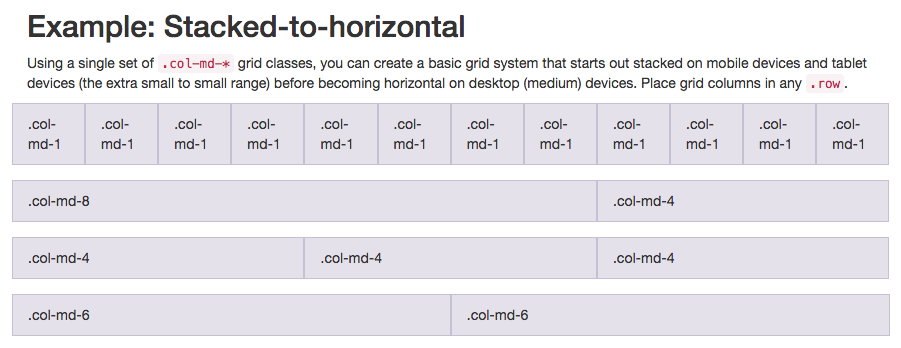
<p>Top 3 things cats <span class = "text-danger">hate:</span></p>

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.

Here's a diagram of how Bootstrap's 12-column grid layout works:

**[](https://i.imgur.com/FaYuui8.png)**

Note that in this illustration, the col-md-\* class is being used. Here, mdmeans 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.

Nest your first image and your h2element within a single <div class="row"> element. Nest your h2element 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?

Add Font Awesome Icons to our Buttons

Font Awesome is a convenient library of icons. These icons are vector graphics, stored in the .svg file format. 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 add Font Awesome to any app just by including it by adding the following code to the top of your HTML:

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css"/>

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 add the Font Awesome classes to the i element to turn it into an icon, for example:

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

Use Font Awesome to add a thumbs-upicon to your like button by giving it an ielement with the classes fa and fa-thumbs-up. For delete class=”fa fa-trash” & for info class=”fa fa-info-circle”.

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

Responsively Style Checkboxes

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

Nest all 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.

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 inputwithin 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 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-6div 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 well div elements.

Apply the Default Bootstrap Button Style

Bootstrap has another button class called btn-default.

Apply both the btn and btn-defaultclasses to each of your buttonelements.

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 divelements 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 a h4 element with the text #left-well.

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

Give Each Element a Unique ID

We will also want to be able to use jQuery to target each button by its unique id.

Give each of your buttons a unique id, starting with target1 and ending with target6.

Make sure that target1 to target3are in #left-well, and target4 to target6 are in #right-well.

**JQUERY**

Adding jQuery to Your Web Pages

There are several ways to start using jQuery on your web site. You can:

* Download the jQuery library from jQuery.com
* Include jQuery from a CDN, like Google

Downloading jQuery

There are two versions of jQuery available for downloading:

* Production version - this is for your live website because it has been minified and compressed
* Development version - this is for testing and development (uncompressed and readable code)

Both versions can be downloaded from [jQuery.com](http://jquery.com/download/).

The jQuery library is a single JavaScript file, and you reference it with the HTML <script> tag (notice that the <script> tag should be inside the <head> section):

<head>  
<script src="jquery-3.2.1.min.js"></script>  
</head>

**Tip:** Place the downloaded file in the same directory as the pages where you wish to use it.

**Do you wonder why we do not have type="text/javascript" inside the <script> tag?**  
  
This is not required in HTML5. JavaScript is the default scripting language in HTML5 and in all modern browsers!

jQuery CDN

If you don't want to download and host jQuery yourself, you can include it from a CDN (Content Delivery Network).

Both Google and Microsoft host jQuery.

To use jQuery from Google or Microsoft, use one of the following:

Google CDN:

<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  
</head>

[Try it Yourself »](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_lib_google)

Microsoft CDN:

<head>  
<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.2.1.min.js"></script>  
</head>

[Try it Yourself »](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_lib_microsoft)

**One big advantage of using the hosted jQuery from Google or Microsoft:**  
  
Many users already have downloaded jQuery from Google or Microsoft when visiting another site. As a result, it will be loaded from cache when they visit your site, which leads to faster loading time. Also, most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.

Learn how Script Tags and Document Ready Work

Now we're ready to learn jQuery, the most popular JavaScript tool of all time. Don't worry about JavaScript itself - we will cover it soon.

Before we can start using jQuery, we need to add some things to our HTML.

First, add a script element at the top of your page. Be sure to close it on the following line.

Your browser will run any JavaScript inside a script element, including jQuery.

Inside your script element, add this code: $(document).ready(function() { to your script. Then close it on the following line (still inside your scriptelement) with: });

We'll learn more about functions later. The important thing to know is that code you put inside this function will run as soon as your browser has loaded your page.

This is important because without your document ready function, your code may run before your HTML is rendered, which would cause bugs.

Target HTML Elements with Selectors Using jQuery

Now we have a document ready function.

Now let's write our first jQuery statement. All jQuery functions start with a $, usually referred to as a dollar sign operator, or as bling.

jQuery often selects an HTML element with a selector, then does something to that element.

For example, let's make all of your button elements bounce. Just add this code inside your document ready function:

$("button").addClass("animated bounce");

Note that we've already included both the jQuery library and the Animate.css library in the background so that you can use them in the editor. So you are using jQuery to apply the Animate.css bounceclass to your button elements.

Additionally make sure to use $("button").addClass("animated bounce"); instead of $('button').addClass("animated bounce"); since single-quote selectors will not pass our tests.

Target Elements by Class Using jQuery

You see how we made all of your buttonelements bounce? We selected them with $("button"), then we added some CSS classes to them with .addClass("animated bounce");.

You just used jQuery's .addClass()function, which allows you to add classes to elements.

First, let's target your div elements with the class well by using the $(".well") selector.

Note that, just like with CSS declarations, you type a . before the class's name.

Then use jQuery's .addClass()function to add the classes animatedand shake.

For example, you could make all the elements with the class text-primaryshake by adding the following to your document ready function:

$(".text-primary").addClass("animated shake");

Target the same element with multiple jQuery Selectors

Now you know three ways of targeting elements: by type: $("button"), by class: $(".btn"), and by id $("#target1").

Although it is possible to add multiple classes in a single .addClass() call, let's add them to the same element in *three separate ways*.

Using .addClass(), add only one class at a time to the same element, three different ways:

Add the animated class to all elements with type button.

Add the shake class to all the buttons with class .btn.

Add the btn-primary class to the button with id #target1.

#### **Note** You should only be targeting one element and adding only one class at a time. Altogether, your three individual selectors will end up adding the three classes shake, animated, and btn-primary to #target1.

#### Remove Classes from an element with jQuery

In the same way you can add classes to an element with jQuery's addClass()function, you can remove them with jQuery's removeClass() function.

Here's how you would do this for a specific button:

$("#target2").removeClass("btn-default");

Let's remove the btn-default class from all of our button elements.

#### **Change the CSS of an Element Using jQuery**

We can also change the CSS of an HTML element directly with jQuery.

jQuery has a function called .css()that allows you to change the CSS of an element.

Here's how we would change its color to blue:

$("#target1").css("color", "blue");

This is slightly different from a normal CSS declaration, because the CSS property and its value are in quotes, and separated with a comma instead of a colon.

Delete your jQuery selectors, leaving an empty document ready function.

Select target1 and change its color to red.

#### **Disable an Element Using jQuery**

You can also change the non-CSS properties of HTML elements with jQuery. For example, you can disable buttons.

When you disable a button, it will become grayed-out and can no longer be clicked.

jQuery has a function called .prop()that allows you to adjust the properties of elements.

Here's how you would disable all buttons:

$("button").prop("disabled", true);

#### Disable only the target1 button.

#### **Change Text Inside an Element Using jQuery**

Using jQuery, you can change the text between the start and end tags of an element. You can even change HTML markup.

jQuery has a function called .html()that lets you add HTML tags and text within an element. Any content previously within the element will be completely replaced with the content you provide using this function.

Here's how you would rewrite and emphasize the text of our heading:

$("h3").html("<em>jQuery Playground</em>");

jQuery also has a similar function called .text() that only alters text without adding tags. In other words, this function will not evaluate any HTML tags passed to it, but will instead treat it as the text you want to replace the existing content with.

Change the button with id target4 by emphasizing its text.

#### **Remove an Element Using jQuery**

Now let's remove an HTML element from your page using jQuery.

jQuery has a function called .remove()that will remove an HTML element entirely

Remove element target4 from the page by using the .remove() function.

#### **Use appendTo to Move Elements with jQuery**

Now let's try moving elements from one div to another.

jQuery has a function called appendTo() that allows you to select HTML elements and append them to another element.

For example, if we wanted to move target4 from our right well to our left well, we would use:

$("#target4").appendTo("#left-well");

Move your target2 element from your left-well to your right-well

#### **Clone an Element Using jQuery**

In addition to moving elements, you can also copy them from one place to another.

jQuery has a function called clone()that makes a copy of an element.

For example, if we wanted to copy target2 from our left-well to our right-well, we would use:

$("#target2").clone().appendTo("#right-well");

Did you notice this involves sticking two jQuery functions together? This is called function chaining and it's a convenient way to get things done with jQuery.

Clone your target5 element and append it to your left-well.

#### **Target the Parent of an Element Using jQuery**

Every HTML element has a parentelement from which it inheritsproperties.

For example, your jQuery Playgroundh3 element has the parent element of <div class="container-fluid">, which itself has the parent body.

jQuery has a function called parent()that allows you to access the parent of whichever element you've selected.

Here's an example of how you would use the parent() function if you wanted to give the parent element of the left-well element a background color of blue:

$("#left-well").parent().css("background-color", "blue")

Give the parent of the #target1element a background-color of red.

#### **Target the Children of an Element Using jQuery**

Many HTML elements have childrenwhich inherit their properties from their parent HTML elements.

For example, every HTML element is a child of your body element, and your "jQuery Playground" h3 element is a child of your <div class="container-fluid"> element.

jQuery has a function called children() that allows you to access the children of whichever element you've selected.

Here's an example of how you would use the children() function to give the children of your left-well element the color of blue:

$("#left-well").children().css("color", "blue")

Give all the children of your #right-well element a color of orange.

#### **Target a Specific Child of an Element Using jQuery**

You've seen why id attributes are so convenient for targeting with jQuery selectors. But you won't always have such neat ids to work with.

Fortunately, jQuery has some other tricks for targeting the right elements.

jQuery uses CSS Selectors to target elements. target:nth-child(n) css selector allows you to select all the nth elements with the target class or element type.

Here's how you would give the third element in each well the bounce class:

$(".target:nth-child(3)").addClass("animated bounce");

Make the second child in each of your well elements bounce. You must target the children of element with the targetclass.

#### **Target Even Numbered Elements Using jQuery**

You can also target all the even-numbered elements.

Here's how you would target all the odd-numbered elements with class targetand give them classes:

$(".target:odd").addClass("animated shake");

Note that jQuery is zero-indexed, meaning that, counter-intuitively, :oddselects the second element, fourth element, and so on.

Try selecting all the even-numbered elements and giving them the classes of animated and shake.

#### **Use jQuery to Modify the Entire Page**

We're done playing with our jQuery playground. Let's tear it down!

jQuery can target the body element as well.

Here's how we would make the entire body fade out: $("body").addClass("animated fadeOut");

But let's do something more dramatic. Add the classes animated and hinge to your body element.