#### Use CSS Selectors to Style Elements

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 style element like this:

<style>  
</style>

Inside that style element, you can create a CSS selector for all h2elements. For example, if you wanted all h2 elements 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.

Delete your h2 element's style attribute and instead create a CSS style element. Add the necessary CSS to turn all h2 elements blue.

Use a CSS Class to Style an Element

Classes are reusable styles that can be added to HTML elements.

Here's an example CSS class declaration:

<style>  
  .blue-text {  
    color: blue;  
  }  
</style>

You can see that we've created a CSS class called blue-text within the <style> tag.

You can apply a class to an HTML element like this:

<h2 class="blue-text">CatPhotoApp</h2>

**Note that in your CSS style element, classes should start with a period. In your HTML elements' class declarations, classes shouldn't start with a period.**

Inside your style element, change the h2 selector to .red-text and update the color's value from blueto red.

Give your h2 element the classattribute with a value of 'red-text'.

#### Style Multiple Elements with a CSS Class

Remember that you can attach classes to HTML elements by using class="your-class-here" within the relevant element's opening tag.

Remember that CSS class selectors require a period at the beginning like this:

.blue-text {  
  color: blue;  
}

But also remember that class declarations don't use a period, like this:

<h2 class="blue-text">CatPhotoApp</h2>

Apply the red-text class to your h2and p elements.

#### Set the Font Family of an Element

You can set an element's font by using the font-family property.

For example, if you wanted to set your h2 element's font to Sans-serif, you would use the following CSS:

h2 {  
  font-family: Sans-serif;  
}

Make all of your p elements use the Monospace font.

#### 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.

Apply the font-family of Lobsterto your h2 element.

#### pecify 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-Seriffont when Helvetica wasn't available, you could use this CSS style:

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

Now comment out your call to Google Fonts, so that the Lobster font isn't available. Notice how it degrades to the Monospace font.

#### Add Images to your Website

You can add images to your website by using the img element, and point to a specific image's URL using the src attribute.

An example of this would be:

<img src="https://www.your-image-source.com/your-image.jpg">

All img elements **must** have an altattribute. The text inside an altattribute is used for screen readers to improve accessibility and is displayed if the image fails to load.

Lets add an alt attribute to our imgexample above:

<img src="https://www.your-image-source.com/your-image.jpg" alt="Author standing on a beach with two thumbs up. ">

Note that in most cases, imgelements are self-closing.

Try it with this image:

https://bit.ly/fcc-relaxing-cat

***ALT SPECIFIES WHAT SHOULD BE DISPLAYED INCASE PHOTO DOESN’T LOAD.***

#### Size your Images

CSS has a property called width that controls an element's width. Just like with fonts, we'll use px (pixels) to specify the image's width.

For example, if we wanted to create a CSS class called larger-image that gave HTML elements a width of 500 pixels, we'd use:

<style>  
  .larger-image {  
    width: 500px;  
  }  
</style>

Create a class called smaller-imageand use it to resize the image so that it's only 100 pixels wide.

**Note**  
Due to browser implementation differences, you may need to be at 100% zoom to pass the tests on this challenge.

#### Add Borders Around your Elements

CSS borders have properties like style, color and width

For example, if we wanted to create a red, 5 pixel border around an HTML element, we could use this class:

<style>  
  .thin-red-border {  
    border-color: red;  
    border-width: 5px;  
    border-style: solid;  
  }  
</style>

Create a class called thick-green-border that puts a 10-pixel-wide green border with a style of solidaround an HTML element, and apply that class to your cat photo.

Remember that you can apply multiple classes to an element by separating each class with a space within its class attribute. For example:

<img class="class1 class2">

#### Add Rounded Corners with a Border Radius

Your cat photo currently has sharp corners. We can round out those corners with a CSS property called border-radius.

You can specify a border-radiuswith pixels. Give your cat photo a border-radius of 10px.

Note: this waypoint allows for multiple possible solutions. For example, you may add border-radius to either the .thick-green-border class or .smaller-imageclass.

#### Make Circular Images with a Border Radius

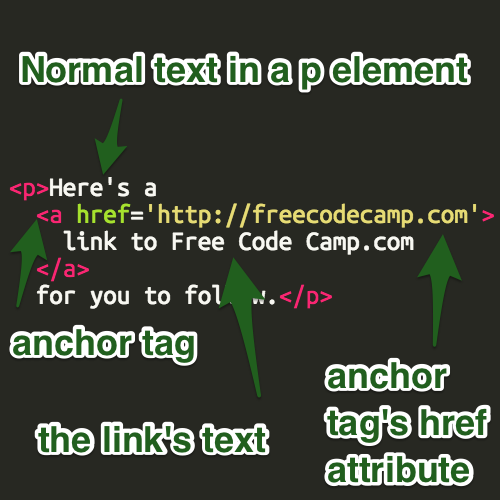
In addition to pixels, you can also specify a border-radius using a percentage.

Give your cat photo a border-radiusof 50%.

#### Link to External Pages with Anchor Elements

a elements, also known as anchorelements, are used to link to content outside of the current page.

Here's a diagram of an a element. In this case, the a element is used in the middle of a paragraph element, which means the link will appear in the middle of a sentence.

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

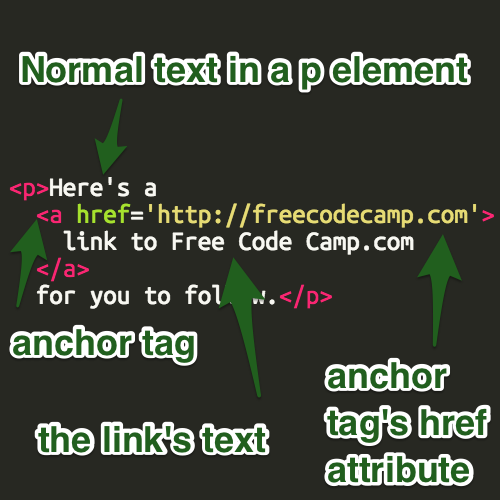
Here's an example:

<p>Here's a <a href="http://freecodecamp.org"> link to Free Code Camp</a> for you to follow.</p>

Create an a element that links to http://freecatphotoapp.com and has "cat photos" as its anchor text.

#### Nest an Anchor Element within a Paragraph

Again, here's a diagram of an aelement for your reference:

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

Here's an example:

<p>Here's a <a href="https://freecodecamp.org"> link to Free Code Camp</a> for you to follow.</p>

Nesting just means putting one element inside of another element.

Now nest your existing a element within a new p element (just after the existing h2 element) so that the surrounding paragraph says "View more cat photos", but where only "cat photos" is a link, and the rest of the text is plain text.

#### Make Dead Links using the Hash Symbol

Sometimes you want to add aelements to your website before you know where they will link.

This is also handy when you're changing the behavior of a link using jQuery, which we'll learn about later.

Replace the value of your aelement's href attribute with a #, also known as a hash symbol, to turn it into a dead link.

<a href=”#”>

#### Turn an Image into a Link

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

Nest your image within an aelement. 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.

Place the existing image element within an anchor element.

Once you've done this, hover over your image with your cursor. Your cursor's normal pointer should become the link clicking pointer. The photo is now a link.

#### Create a Bulleted Unordered List

HTML has a special element for creating unordered lists, or bullet point-style lists.

Unordered lists start with a <ul>element. Then they contain some number of <li> elements.

For example:

<ul>  
  <li>milk</li>  
  <li>cheese</li>  
</ul>

would create a bullet point-style list of "milk" and "cheese".

Remove the last two p elements and create an unordered list of three things that cats love at the bottom of the page.

#### Create an Ordered List

HTML has a special element for creating ordered lists, or numbered-style lists.

Ordered lists start with a <ol>element. Then they contain some number of <li> elements.

For example:

<ol>  
  <li>Garfield</li>  
  <li>Sylvester</li>  
</ol>

would create a numbered list of "Garfield" and "Sylvester".

Create an ordered list of the top 3 things cats hate the most.

#### Create a Text Field

Now let's create a web form.

Text inputs are a convenient way to get input from your user.

You can create one like this:

<input type="text">

Note that input elements are self-closing.

Create an input element of type text below your lists.

#### Add Placeholder Text to a Text Field

Your placeholder text is what appears in your text input before your user has input anything.

You can create placeholder text like so:

<input type="text" placeholder="this is placeholder text">

Set the placeholder value of your text input to "cat photo URL".

#### 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 form element.

For example:

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

Nest your text field in a formelement. Add the action="/submit-cat-photo" attribute to this form element.

#### Add a Submit Button to a Form

Let's add a submit button to your form. Clicking this button will send the data from your form to the URL you specified with your form's action attribute.

Here's an example submit button:

<button type="submit">this button submits the form</button>

Add a submit button to your formelement with type submit and "Submit" as its text.

#### 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 input element, you would use: <input type="text" required>

Make your text input a requiredfield, so that your user can't submit the form without completing this field.

Then try to submit the form without inputing any text. See how your HTML5 form notifies you that the field is required?

Note: This field does not work in Safari.

#### Create a Set of Radio Buttons

You can use radio buttons for questions where you want the user to only give you one answer.

Radio buttons are a type of input

Each of your radio buttons should be nested within its own labelelement.

All related radio buttons should have the same name attribute.

Here's an example of a radio button:

<label><input type="radio" name="indoor-outdoor"> Indoor</label>

Add a pair of radio buttons to your form. One should have the option of indoor and the other should have the option of outdoor.

#### Create a Set of Checkboxes

Forms commonly use checkboxes for questions that may have more than one answer.

Checkboxes are a type of input

Each of your checkboxes should be nested within its own label element.

All related checkbox inputs should have the same name attribute.

Here's an example of a checkbox:

<label><input type="checkbox" name="personality"> Loving</label>

Add to your form a set of three checkboxes. Each checkbox should be nested within its own labelelement. All three should share the name attribute of personality.

#### Check Radio Buttons and Checkboxes by Default

You can set a checkbox or radio button to be checked by default using the checked attribute.

To do this, just add the word "checked" to the inside of an input element. For example:

<input type="radio" name="test-name" checked>

Set the first of your radio buttonsand the first of your checkboxes to both be checked by default.

#### Nest Many Elements within a Single Div Element

The div element, also known as a division element, is a general purpose container for other elements.

The div element is probably the most commonly used HTML element of all. It's useful for passing the CSS of its own class declarations down to all the elements that it contains.

Just like any other non-self-closing element, you can open a div element with <div> and close it on another line with </div>.

Try putting your opening div tag above your "Things cats love" pelement and your closing div tag after your closing ol tag so that both of your lists are within one div.

Nest your "Things cats love" and "Things cats hate" lists all within a single div element.

#### Give a Background Color to a Div Element

You can set an element's background color with the background-colorproperty.

For example, if you wanted an element's background color to be green, you'd put this within your style element:

.green-background {  
  background-color: green;  
}

Create a class called silver-background with the background-color of silver. Assign this class to your div element.

#### Set the ID of an Element

In addition to classes, each HTML element can also have an idattribute.

There are several benefits to using id attributes, and you'll learn more about them once you start using jQuery.

id attributes should be unique. Browsers won't enforce this, but it is a widely agreed upon best practice. So please don't give more than one element the same id attribute.

Here's an example of how you give your h2 element the id of cat-photo-app:

<h2 id="cat-photo-app">

Give your form element the id cat-photo-form.

#### Use an ID Attribute to Style an Element

One cool thing about id attributes is that, like classes, you can style them using CSS.

Here's an example of how you can take your element with the idattribute of cat-photo-element and give it the background color of green. In your style element:

#cat-photo-element {  
  background-color: green;  
}

Note that inside your style element, you always reference classes by putting a . in front of their names. You always reference ids by putting a # in front of their names.

Try giving your form, which now has the id attribute of cat-photo-form, a green background.

#### Adjusting the Padding of an Element

Now let's put our Cat Photo App away for a little while and learn more about styling HTML.

You may have already noticed this, but all HTML elements are essentially little rectangles.

Three important properties control the space that surrounds each HTML element: padding, margin, and border.

An element's padding controls the amount of space between the element and its border.

Here, we can see that the green box and the red box are nested within the yellow box. Note that the red box has more padding than the green box.

When you increase the green box's padding, it will increase the distance between the text padding and the border around it.

Change the padding of your green box to match that of your red box.

#### Adjust the Margin of an Element

An element's margin controls the amount of space between an element's border and surrounding elements.

Here, we can see that the green box and the red box are nested within the yellow box. Note that the red box has more margin than the green box, making it appear smaller.

When you increase the green box's margin, it will increase the distance between its border and surrounding elements.

Change the margin of the green box to match that of the red box.

#### Add a Negative Margin to an Element

An element's margin controls the amount of space between an element's border and surrounding elements.

If you set an element's margin to a negative value, the element will grow larger.

Try to set the margin to a negative value like the one for the red box.

Change the margin of the green box to -15px, so it fills the entire horizontal width of the yellow box around it.

#### Add Different Padding to Each Side of an Element

Sometimes you will want to customize an element so that it has different padding on each of its sides.

CSS allows you to control the padding of an element on all four sides with padding-top, padding-right, padding-bottom, and padding-left properties.

Give the green box a padding of 40px on its top and left side, but only 20px on its bottom and right side.

#### Add Different Margins to Each Side of an Element

Sometimes you will want to customize an element so that it has a different margin on each of its sides.

CSS allows you to control the marginof an element on all four sides with margin-top, margin-right, margin-bottom, and margin-leftproperties.

Give the green box a margin of 40px on its top and left side, but only 20px on its bottom and right side.

#### 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.

Use Clockwise Notation to give the ".green-box" class a padding of 40pxon its top and left side, but only 20pxon its bottom and right side.

#### Use Clockwise Notation to Specify the Margin of an Element

Let's try this again, but with marginthis time.

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

margin: 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 margin instructions.

Use Clockwise Notation to give the element with the green-box class a margin of 40px on its top and left side, but only 20px on its bottom and right side.

#### Style the HTML Body Element

Now let's start fresh and talk about CSS inheritance.

Every HTML page has a bodyelement.

We can prove that the body element exists here by giving it a background-color of black.

We can do this by adding the following to our style element:

body {  
  background-color: black;

#### Inherit Styles from the Body Element

Now we've proven that every HTML page has a body element, and that its body element can also be styled with CSS.

Remember, you can style your bodyelement just like any other HTML element, and all your other elements will inherit your body element's styles.

First, create a h1 element with the text Hello World

Then, let's give all elements on your page the color of green by adding color: green; to your bodyelement's style declaration.

Finally, give your body element the font-family of Monospace by adding font-family: Monospace; to your body element's style declaration

#### Prioritize One Style Over Another

Sometimes your HTML elements will receive multiple styles that conflict with one another.

For example, your h1 element can't be both green and pink at the same time.

Let's see what happens when we create a class that makes text pink, then apply it to an element. Will our class override the body element's color: green; CSS property?

Create a CSS class called pink-textthat gives an element the color pink.

Give your h1 element the class of pink-text.

#### verride Styles in Subsequent CSS

Our "pink-text" class overrode our body element's CSS declaration!

We just proved that our classes will override the body element's CSS. So the next logical question is, what can we do to override our pink-textclass?

Create an additional CSS class called blue-text that gives an element the color blue. Make sure it's below your pink-text class declaration.

Apply the blue-text class to your h1 element in addition to your pink-text class, and let's see which one wins.

Applying multiple class attributes to a HTML element is done with a space between them like this:

class="class1 class2"

Note: It doesn't matter which order the classes are listed in the HTML element.

However, the order of the classdeclarations in the <style> section are what is important. The second declaration will always take precedence over the first. Because .blue-text is declared second, it overrides the attributes of .pink-text

#### Override Class Declarations by Styling ID Attributes

We just proved that browsers read CSS from top to bottom. That means that, in the event of a conflict, the browser will use whichever CSS declaration came last.

But we're not done yet. There are other ways that you can override CSS. Do you remember id attributes?

Let's override your pink-text and blue-text classes, and make your h1 element orange, by giving the h1element an id and then styling that id.

Give your h1 element the idattribute of orange-text. Remember, id styles look like this:

<h1 id="orange-text">

Leave the blue-text and pink-text classes on your h1 element.

Create a CSS declaration for your orange-text id in your styleelement. Here's an example of what this looks like:

#brown-text {  
  color: brown;  
}

Note: It doesn't matter whether you declare this css above or below pink-text class, since id attribute will always take precedence.

#### Override Class Declarations with Inline Styles

So we've proven that id declarations override class declarations, regardless of where they are declared in your style element CSS.

There are other ways that you can override CSS. Do you remember inline styles?

Use an in-line style to try to make our h1 element white. Remember, in line styles look like this:

<h1 style="color: green">

Leave the blue-text and pink-text classes on your h1 element.

#### Override All Other Styles by using Important

Yay! We just proved that in-line styles will override all the CSS declarations in your style element.

But wait. There's one last way to override CSS. This is the most powerful method of all. But before we do it, let's talk about why you would ever want to override CSS.

In many situations, you will use CSS libraries. These may accidentally override your own CSS. So when you absolutely need to be sure that an element has specific CSS, you can use !important

Let's go all the way back to our pink-text class declaration. Remember that our pink-text class was overridden by subsequent class declarations, id declarations, and in-line styles.

Let's add the keyword !important to your pink-text element's color declaration to make 100% sure that your h1 element will be pink.

An example of how to do this is:

color: red !important;

#### Use Hex Code for Specific Colors

Did you know there are other ways to represent colors in CSS? One of these ways is called hexadecimal code, or hex code for short.

We usually use decimals, or base 10 numbers, which use the symbols 0 to 9 for each digit. Hexadecimals (or hex) are base 16 numbers. This means it uses sixteen distinct symbols. Like decimals, the symbols 0-9 represents values zero to nine. Then A,B,C,D,E,F represent values ten to fifteen. Altogether, 0 to F can represent a digit in hexadecimal, giving us 16 total possible values. You can find more information about [**hexadecimal numbers here**](https://en.wikipedia.org/wiki/Hexadecimal).

In CSS, we can use 6 hexadecimal digits to represent colors, two each for the red (R), green (G), and blue (B) components. For example, #000000is black and is also the lowest possible value. You can find more information about the [**RGB color system here**](https://en.wikipedia.org/wiki/RGB_color_model).

Replace the word black in our bodyelement's background-color with its hex code representation, #000000.

#### Use Hex Code to Mix Colors

To review, hex codes use 6 hexadecimal digits to represent colors, two each for red (R), green (G), and blue (B) components.

From these three pure colors (red, green, and blue), we can vary the amounts of each to create over 16 million other colors!

For example, orange is pure red, mixed with some green, and no blue. In hex code, this translates to being #FFA500.

The digit 0 is the lowest number in hex code, and represents a complete absence of color.

The digit F is the highest number in hex code, and represents the maximum possible brightness.

Replace the color words in our styleelement with their correct hex codes.

|  |  |
| --- | --- |
| **Color** | **Hex Code** |
| Dodger Blue | #2998E4 |
| Green | #00FF00 |
| Orange | #FFA500 |
| Red | #FF0000 |

#### Use Abbreviated Hex Code

Many people feel overwhelmed by the possibilities of more than 16 million colors. And it's difficult to remember hex code. Fortunately, you can shorten it.

For example, red's hex code #FF0000can be shortened to #F00. This shortened form gives one digit for red, one digit for green, and one digit for blue.

This reduces the total number of possible colors to around 4,000. But browsers will interpret #FF0000 and #F00 as exactly the same color.

Go ahead, try using the abbreviated hex codes to color the correct elements.

|  |  |
| --- | --- |
| **Color** | **Short Hex Code** |
| Cyan | #0FF |
| Green | #0F0 |
| Red | #F00 |
| Fuchsia | #F0F |

#### Use RGB values to Color Elements

Another way you can represent colors in CSS is by using RGB values.

The RGB value for black looks like this:

rgb(0, 0, 0)

The RGB value for white looks like this:

rgb(255, 255, 255)

Instead of using six hexadecimal digits like you do with hex code, with RGByou specify the brightness of each color with a number between 0 and 255.

If you do the math, the two digits for one color equal 16 times 16, which gives us 256 total values. So RGB, which starts counting from zero, has the exact same number of possible values as hex code.

Let's replace the hex code in our body element's background color with the RGB value for black: rgb(0, 0, 0)

#### Use RGB to Mix Colors

Just like with hex code, you can mix colors in RGB by using combinations of different values.

Replace the color words in our styleelement with their correct RGB values.

|  |  |
| --- | --- |
| **Color** | **RGB** |
| Blue | rgb(0, 0, 255) |
| Red | rgb(255, 0, 0) |
| Orchid | rgb(218, 112, 214) |
| Sienna | rgb(160, 82, 45) |