Let’s review what you learned so far:

* The basic anatomy of CSS syntax written for both inline styles and stylesheets.
* Some commonly used CSS terms, such as *ruleset*, *selector*, and *declaration*.
* CSS inline styles can be written inside the opening HTML tag using the style attribute.
* Inline styles can be used to style HTML, but it is not the best practice.
* An internal stylesheet is written using the <style> element inside the <head> element of an HTML file.
* Internal stylesheets can be used to style HTML but are also not best practice.
* An external stylesheet separates CSS code from HTML, by using the “.css”.file extension.
* External stylesheets are the best approach when it comes to using HTML and CSS.
* External stylesheets are linked to HTML using the <link> element.

Take this knowledge to the next lesson, where you start learning how to select HTML elements to style!

Throughout this lesson, you learned how to select HTML elements with CSS and apply styles to them. Let’s review what you learned:

* CSS can select HTML elements by type, class, ID, and attribute.
* All elements can be selected using the universal selector.
* An element can have different states using the pseudo-class selector.
* Multiple CSS classes can be applied to one HTML element.
* Classes can be reusable, while IDs can only be used once.
* IDs are more specific than classes, and classes are more specific than type. That means IDs will override any styles from a class, and classes will override any styles from a type selector.
* Multiple selectors can be chained together to select an element. This raises the specificity but can be necessary.
* Nested elements can be selected by separating selectors with a space.
* Multiple unrelated selectors can receive the same styles by separating the selector names with commas.

Great work this lesson. With this knowledge, you’ll be able to use CSS to change the look and feel of websites to make them look great!

Incredible work! You used CSS to alter text and images on a website. Throughout this lesson, you learned concepts including:

* The font-family property defines the typeface of an element.
* font-size controls the size of text displayed.
* font-weight defines how thin or thick text is displayed.
* The text-align property places text in the left, right, or center of its parent container.
* Text can have two different color attributes: color and background-color. color defines the color of the text, while background-color defines the color behind the text.
* CSS can make an element transparent with the opacity property.
* CSS can also set the background of an element to an image with the background-image property.
* The !important flag will override any style, however it should almost never be used, as it is extremely difficult to override.

We’ve completed our extensive tour of the colors in CSS! Let’s review the key information we’ve learned.

There are four ways to represent color in CSS:

* Named colors—there are more than 140 named colors, which you can review [here](https://developer.mozilla.org/en-US/docs/Web/CSS/color_value).
* Hexadecimal or hex colors
  + Hexadecimal is a number system with has sixteen digits, 0 to 9 followed by “A” to “F”.
  + Hex values always begin with # and specify values of red, blue, and green using hexadecimal numbers such as #23F41A.
  + Six-digit hex values with duplicate values for each RGB value can be shorted to three digits.
* RGB
  + RGB colors use the rgb() syntax with one value for red, one value for blue and one value for green.
  + RGB values range from 0 to 255 and look like this: rgb(7, 210, 50).
* HSL
  + HSL stands for hue (the color itself), saturation (the intensity of the color), and lightness (how light or dark a color is).
  + Hue ranges from 0 to 360 and saturation and lightness are both represented as percentages like this: hsl(200, 20%, 50%).
* You can add opacity to color in RGB and HSL by adding a fourth value, a, which is represented as a percentage.

In this lesson, we covered the four properties of the box model: height and width, padding, borders, and margins. Understanding the box model is an important step towards learning more advanced HTML and CSS topics. Let’s take a minute to review what you learned:

* The box model comprises a set of properties used to create space around and between HTML elements.
* The height and width of a content area can be set in pixels or percentages.
* Borders surround the content area and padding of an element. The color, style, and thickness of a border can be set with CSS properties.
* Padding is the space between the content area and the border. It can be set in pixels or percent.
* Margin is the amount of spacing outside of an element’s border.
* Horizontal margins add, so the total space between the borders of adjacent elements is equal to the sum of the right margin of one element and the left margin of the adjacent element.
* Vertical margins collapse, so the space between vertically adjacent elements is equal to the larger margin.
* margin: 0 auto horizontally centers an element inside of its parent content area, if it has a width.
* The overflow property can be set to display, hide, or scroll, and dictates how HTML will render content that overflows its parent’s content area.
* The visibility property can hide or show elements.

Let’s review what you’ve learned so far:

* The position property allows you to specify the position of an element.
* When set to relative, an element’s position is relative to its default position on the page.
* When set to absolute, an element’s position is relative to its closest positioned parent element. It can be pinned to any part of the web page, but the element will still move with the rest of the document when the page is scrolled.
* When set to fixed, an element’s position can be pinned to any part of the web page. The element will remain in view no matter what.
* When set to sticky, an element can stick to a defined offset position when the user scrolls its parent container.
* The z-index of an element specifies how far back or how far forward an element appears on the page when it overlaps other elements.
* The display property allows you to control how an element flows vertically and horizontally in a document.
* inline elements take up as little space as possible, and they cannot have manually adjusted width or height.
* block elements take up the width of their container and can have manually adjusted heights.
* inline-block elements can have set width and height, but they can also appear next to each other and do not take up their entire container width.
* The float property can move elements as far left or as far right as possible on a web page.
* You can clear an element’s left or right side (or both) using the clear property.