[MUSIC] Your choice for placing styles in one

versus another place doesn't only affect how reusable the styles are, but it also affects which style declarations

override other style declarations. Let's jump into the code and

take a look at an example. Okay, on the Sublime Text, and I'm looking

at the file called styleplacement.html. Let me go ahead and quickly rearrange

the screen, so you can see the browser and the code editor at the same time. So here's our HTML document, and so far,

we've been specifying all of our styles inside the style tag that sits inside

of the head tag of our HTML document. And in this case, we're specifying that

every h2 in our HTML document should have the text color maroon. As you could see that subheadings,

the h2's, are, in fact, maroon. However, there are other places

that you could specify a CSS style. For example, you could specify a CSS style

directly on the element by providing the style attribute and the CSS styles

that should apply to that style. The only thing that's missing from these

styles is the target, and it's for a good reason, since we're inside the element, so

we don't really need a target any more. But the style decorations

are exactly the same as before, and they're also terminated by semicolons,

and you could put here more than one, just keep terminating

them with semicolons. In this case, this I am centered content,

is text aligned center. And as you can see in the page, it is

in fact getting centered in the page. This is called inline styling,

and as you might have guessed, this is the least reusable way

of styling elements there is. So for example, if you wanted to style

something similarly within the same page, you would literally have to copy and paste the styles somewhere

else in order to achieve that. For that reason, this is the least

preferred method of styling, and you should really try to stay as

far away from this as possible. I personally suggest that

you only use this for some very quick testing just to see

what the style would look like. The style and location that we usually use in this class

is a little better than an inline style. It's better because I

could specify classes and target multiple elements

with my selectors here. However, what happens if I have

multiple pages in my website and I want them to look consistent? Which means specifying the styles inside

the head tag over one particular HTML page is really not going to

help me with the others. That means I really need some sort

of other way to specify styles that are external to the HTML page and kind

of link them back in into my HTML page. Well, this is what external

style sheets are all about. And you can see that we're specifying

one here with a tag called link, and we're telling the browser

that it's the style sheet and its location using

the familiar href attribute. In this case,

we're specifying it to be a relative URL. So style.css better live in the same

directory as our styleplacement.html file. And in fact, it does, and

this is what it looks like. External style sheets are just

styles with lists of CSS rules. There's no particular tags that

have to surround any of this. So as you can see in this style.css,

we're specifying that the entire body, meaning the entire viewport,

its background color should be gray, and the font size should be 130% of

whatever the default font size is. And if you look to our page, you could see

that the fonts are a little bit bigger than usual, and

certainly the background is gray. That styling came from style.css. While we will continue in the class to

use the style tag inside the head tag so it's a little more convenient to showcase

the styles, in a real-world website, you almost always want all your styles

to be external to your HTML page. Which means writing your styles

inside some style.css external file is actually the way to go. To summarize, placing styles inline

is great for quick testing but try to avoid that if you're actually

coding a real-world website. Real-world websites almost always

use external style sheets, which means you take all of your styles

and you place them in an external file, and then you link it back in

into usually several HTML pages. That technique doesn't only

reuse your CSS styles but also promotes consistent look

throughout your entire website. And on top of that, since you externalize

your CSS styling into one file, each subsequent page in your

website will not have to redownload the CSS styles over and over again. While we use the head styles for just

demonstration purposes, in the real world, they're usually used to override

some external ones, however, even that practice is

a little bit questionable. If you're trying to overwrite styles

that are external to your HTML page and you have no control over, it's better to

create your own external CSS file, and then specify it below the external CSS

file that you have no control over. However, that does get a little bit into

conflict resolution between styles, and that is exactly what we're

going to talk about next.