# ***STYLING TEXT***

[SOUND] There are ton of CSS properties

that effect the way text is displayed.

I'm not going to attempt to cover every single one of them.

Instead, we'll take a look at a few that illustrate the concepts behind their

stylings.

Let's jump straight into the code editor and take a look.

Play video starting at ::21 and follow transcript0:21

Okay, so I'm in Sublime Text and

I'm looking at the file called styling-text-before.htlm.

Let me quickly rearrange this screen so you can see the browser, and

the code editor at the same time.

Okay so the first thing we want to do here, is we have a class called style, and

we're applying this class to the second paragraph, in our html file,

we have two paragraphs here.

The second one is what we're applying this to.

As you can see in the browser, there's a couple of paragraphs that are being shown,

and both of them at the moment have the default styling from the browser,

so our task here is to style the second paragraph just to see a comparison.

So the first thing we want to do usually is specify font-family.

So font-family is the property name.

And the value for font-family could vary.

And in fact, I provided you here a link right straight in the HTML page, and

we'll go ahead and click that link.

It basically gives you commonly used font combinations.

So what you usually put in the font-family is the value is combinations of fonts.

And the reason you put not just one font, but a few fonts is because

when you specify font-family in this way, you're relying on the client's computer,

you're relying on the user's computer to have that font installed.

And it's possible that a particular font will not be installed on the user's

computer, so you want to provide some options, and at the very least, you want

to provide whether you want a serif type of font, or sans-serif type of font.

Because every computer will have some sort of a default serif or sans-serif font.

And just in case you don't know the difference between Serif and Sans-Serif,

Serifs are the fonts that not only have the lines, but

also have a bit of an embellishment at the end of each line.

So, as you can see, there's these little squigglies at the end of each line.

As opposed to Sans-Serif, they're just very straight lines.

Personally, I prefer this particular font combination,

so I'll copy it and paste it right here.

Give it a semicolon and let's go back, let's save the file, and refresh.

And now you see that now we have,

Arial most probably as the font for our paragraph.

Next let's change the color.

Now you've seen me use color multiple times already but,

the truth matter is while it's probably okay to use red,

green, blue, predefined color names in testing for real websites,

you would want to use a hexadecimal value for a particular color.

And whatever color you want to have is something that obviously is up to you.

So I'll pick this color and this is a hexadecimal value.

And the way this color is specified is, the first two numbers is red, second two

numbers is green, and the last two numbers is blue, ff being the highest blue value.

So, basically it's the same RGB specification that a lot of colors

are specified as, except it's expressed in hexadecimal, so let's save it, refresh.

As you can see now, it turned blue.

Now there are various shortcuts for specifying colors, and

I'll let you look it up yourself, but needless to say,

if you specify a six digit hexadecimal color, it's certainly good enough.

The property font-style is the one that specifies whether or

not you want something to be italic or normal.

And we'll give it an italic to just change it a little bit.

We'll save it, refresh, and now you see the font turned italic.

Next thing is font-weight.

Now font-weight can be specified from basically normal to bold.

And you can also specify it using numbers.

So if I specify it all the way to 800, 900, and bold.

Well 900 will be pretty bold so if I save it and refresh it, it will be pretty bold,

but in practice people just use the word bold as a predefined keyword so

that's also good enough and about equals 900 anyway.

Next we'll specify font size.

And we'll give it 24 pixels.

And this will obviously give it a little bit of a bigger font size than it is right

now, as you can see right there.

Just about every browser out there has a default font size of 16 pixels.

And that should give you an idea as to what an on-style text would look like and

by on-styled I mean the one that has the browser default styling applied to it.

Now don't confuse this with points, these are pixels.

Points are used in print when you do Microsoft Word or something like that, but

not on screen.

On screen you use pixels.

Pixels are what's considered absolute unit of measurement for size.

However, they do have a bit of a relative component to them.

Pixels are relative to the viewing device.

For low DPI or low dots per inch devices,

one pixel is one device pixel dot of the display.

For printers and high-resolution screens, one pixels implies multiple device pixels.

Point is, higher DPI devices will give you sharper text, because for every

pixel that it's drawing, it's actually drawing multiple pixels on the device.

With all that said,

pixels is still considered an absolute unit of measurement.

We'll talk about relative units of measurements,

as far as font size is concerned, in just a little bit.

But for this short segment, let me just show you a couple of more

text properties that you might want to know about.

There's another one called text transform and

it allows you to basically control how your text looks.

Capitalize it, lowercase.

If I capitalize, for

example, save it and refresh, you could see that every word now gets capitalized.

But there's more.

There's lowercasing.

I could have the entire text get lowercased.

Or I could get the entire text get uppercased.

And let's leave it that way.

Another useful property is text-align.

And what it allows you to do, it allows you to either center, right,

justify, left the text within its block level element.

For this one, let's go ahead and right justify it.

Let's refresh, and here you go.

We'll continue this lecture with part two, and

we'll speak about relative font sizing.

[MUSIC]

Play video starting at ::5 and follow transcript0:05

Okay let's talk a little bit about relative font sizing.

And for that we're going to take a look at font size before that HTML.

And it's located in the same folder as the previous file.

Now there are two units of measurements that are relative that we're going

to talk about and that is percent and ems.

Let's start with percent first of all let's specify that our body tag

should have fun size of a 120 %.

And what that means is Is that we want to take whatever the default size is,

and by default I mean whatever the browser provides by default, and

increase it by 120%.

You can actually do a quick calculation and

figure out what pixel size that would be.

Most browsers have 16 pixels as default text size.

So 120% of 16 would a little bit over 19 pixels.

So if we refresh the page,

we'll see that we increased the font size in the entire HTML document.

Let's now target the very first text here that says 2em text.

So lets go ahead and actually use an inline style since we're just testing and

specify the font size to be 2em.

And is a unit of measurement that is equivalent to the width of the letter

m in this particular font that we're using.

It sounds a little bit confusing at first, but

the bottom line is it's a relative size.

It's relative to something.

And for this div where it says 2em text, what this is saying is it's

relative to whatever the font size is at the moment we're applying it.

Since we set the font size to 120% in the entire body,

this div also received at 120% and when you combine this font size with this one,

this basically says I want to increase the font two times of whatever a 120 was.

So if we take a look at this text this 2em should really increase twice as large as

this word regular text.

So if we save the file and refresh, and you can see now everything

since that div includes the other divs they got increased.

That looks just about right that it's twice the size of the regular text.

Now you might find this confusing.

Didn't we just override the font size and say it's 2em?

Why is 120% playing together with 2em instead of having 2em override the 120%,

and that's kind of where the game of the relative sizing.

When you specify them on subsequent elements,

they don't have an overriding effect, they have a cumulative effect instead.

Because what you're saying here is whatever the size of this div is,

I want you to increase it.

Or I want you to make 2ems out of it, meaning, make it twice as large.

So I'm sure you guessed now, how do I make this 4em text actually 4em?

If I specify style and say font-size, what should I put here?

Should I put 4em?

Well that would be a mistake.

If I put 4em it will really be the 2em that it's already applying to this

times 4em, which would make it 8em, right.

But what we actually need to do is put 4em.

In order to do that, I just specify 2em, again.

So, it means that this 4em text will be twice as large as the 2em text right here.

So let's save the file and

refresh and you can see now it's twice as large in this 2em.

This text also got caught into this because it is a child element

of our div that we just styled.

so how so I make it 2em again?

Well, the answer is, you don't overwrite it, you just decrease it.

So let's go ahead and say style=\_"font-size: and

in order to get back to 2em I need to half the size of my text.

So in order to do that I'll just say .5em and now when I refresh my text now

is back to the same size as it was before, because I halved the relative size of 4em.

A couple of more quick points before we finish up.

First it's a pretty common practice, that if you want to increase the font-size in

the entire document, that you specify percentage and

you apply it usually to the most parent tag, which is usually the body tag.

And technically speaking, you could have continued to use percentages even here.

So 2em really translates to 200% so if I put 200% right here and

save it and refresh you could see nothing really changed.

However it's more common to use ems when you're talking about relative sizes

within the documents itself.

But one thing you should know is that even though you could use percentages here I

highly recommend you don't mix and match the percentage, ems and

certainly not the absolute value units all over the place.

It could get very confusing and it'll become a mess.

So it's better to keep things consistent.

Point number two is, even though we increase the font size here to 120,

using CSS rule,

a user of your website can do the same thing just by increasing the zoom.

For example here, I will increase the zoom a couple of times.

As you could see, I'm all the way to 200% zoom, and hopefully now,

you see the value of having relative units of measurement for font size.

You could see that the visual size relationship between these different divs

remains the same, even though they're all now very big.

This text is still half of this text, and this text is still half of this text.