HTML

As known as HTML5 and Hyper Text Makeup Language

Angle brackets - In HTML, the characters “<“and “>”are known as angle brackets.

HTML element (or simply, element) - HTML code that lives inside of angle brackets.

Opening tag - the first, or opening, HTML tag used to start an HTML element.

Closing tag - the second, or closing, HTML tag used to end an HTML element. Closing tags have a forward slash (/) inside of them

The <head> element will contain information about the page that is not displayed directly on the actual web page (you will see an example in the next exercise).

All of the code above demonstrates what is sometimes referred to as ["boilerplate code."](https://en.wikipedia.org/wiki/Boilerplate_code)

The term "boilerplate code" is used to describe the basic HTML code required to begin creating a web page. Without all of the elements in the boilerplate code, you will risk starting without the minimum requirements considered best practice.

HTML Dictionary

<!DOCTYPE <Type of file>> = Defines the file’s type

<html> = Type of text, everything associated with site written in HTML have to be inside it

<head> = Head is the tab on top of your browser as ”Youtube” or another site

<title> = It have to be inside of <head>, it defines title of your tab

<h(number)> = “Headings” how it can be called, it is a title text or subtitles with an adjustment on your size depending of the number.

<body> = The body of your site or file, everything that you will put on your body site have to be here.

<p> = It’s have to be inside the <body>, Create a paragraph

<ul> = To create a unordered list with bullet points

<ol> = To create a ordered list with numbers points

<li> = To add some text on your list

<a> = To add a link to other web, don’t forget the href

<img/> = To add an image to the web site, this element is special because it does not need to close the tag, be considered a self-closing element

<br /> = to break a line, adding a space between

<!-- --> = To add a comment in the code

<link> = You can use the <link> element to link the HTML and CSS files together. The <link> element must be placed within the head of the HTML file. It is a self-closing tag and requires the following three attributes:

1. href - like the anchor element, the value of this attribute must be the address, or path, to the CSS file.
2. type - this attribute describes the type of document that you are linking to (in this case, a CSS file). The value of this attribute should be set to text/css.
3. rel - this attribute describes the relationship between the HTML file and the CSS file. Because you are linking to a stylesheet, the value should be set to stylesheet.

If the CSS file is stored in the same [directory](https://en.wikipedia.org/wiki/Directory_%28computing%29) as your HTML file, then you can specify a [relative path](https://en.wikipedia.org/wiki/Path_%28computing%29#Absolute_and_relative_paths) instead of a URL.

<div> = It works as a container or a box. To use in CSS code, you must to use

Attributes

Href = To refer a link, used on <a>

Target = The target attribute specifies that a link should open in a new window

Src = Like the Href, it’s used to refer a source, like a directory or url, used on <img/>

Alt = The alt attribute is applied specifically to the <img> element. The value of alt should be a description of the image.

Class = to create a type of file or class, to organize the HTML and CSS code with (.name)

Id = to identify a element on the HTML code, CSS, you can use as (#name)

Values

\_blank = When the user click on the link, the \_blank opens this link in a new tab

CSS (**C**ascading **S**tyle **S**heets)

Although CSS is a different language than HTML, it is possible to write CSS code directly within an HTML file. This is possible because of the <style> element.

<style> = The <style> element allows you to write CSS code between its opening and closing tags. To use the <style> element, it must be placed inside of the head.

A Summary

1. HTML and CSS are kept in separate files in order to keep code maintainable and readable, as well as keep structure separate from styling.

2. The <style> element allows you to write CSS code within an HTML file.

3. A CSS stylesheet can be linked to an HTML file using the <link> element, which requires three attributes:

* href - set equal to the path of the CSS file.
* type - set equal to text/css.
* rel - set equal to stylesheet.

It's not enough to simply select an HTML element in a CSS file. To actually style the element, you need to specify two things inside the body of the selector:

1. Property - the property you'd like to style of that element (i.e., size, color, etc.).
2. Value - the value of the property (i.e., 18px for size, Blue for color, etc.).

Styling with CSS would be very inefficient if you were forced to manually style the same property across many elements.

For example, what if you wanted to change the color of 10 different elements to Aquamarine in CSS.

Fortunately, you can select multiple elements at once so that you can save time styling a shared property.

H1, h2, p{

color: Green;

}

There's a special selector that can instantly select every single element on the web page: the universal selector.

\*{

}

Just like HTML, CSS follows certain best practices for spacing and indentation.

h1 {

color: blue;

}

p {

color: red;

}

* CSS Rules

1. One space should be used between the selector and the opening curly brace ({).
2. No extra spaces should exist between opening and closing curly braces ({ and }) and CSS declarations (as in the example above).
3. Two spaces of indentation should be used for CSS declarations.
4. One line of spacing should exist between CSS rules. In the example above, there is one line of spacing between the CSS rule for the heading and the CSS rule for the paragraph.
5. The CSS rule above says: "Use the Garamond font for all <h1> elements on the web page. If that font is not available, use the Times font. If both of those fonts are not available, use any serif font pre-installed on the user's computer." The fonts specified after Garamond are the fallback fonts.

* CSS Dictionary

Font-size

Color

Background-color

Font-family

Rgb(Red, Green, Blue)

Hsl(Hue, Saturation(%), Lightness(%))

Line-height

Word-spacing

Letter-spacing

Font-weight

Text-transform

Font-style

Background-image

Background-position

Overflow

Max and min / height or width

* Colors
* Colors: Hexadecimals

There's an additional way to specify colors in CSS: hexadecimal color codes, often referred to as "hex color codes" for short.

Hex color codes also offer 16,777,216 color options, but they follow a different syntax.

When specifying an RGB color mixture, the values are in [base 10](https://en.wikipedia.org/wiki/Decimal). Hex color codes, however, use [base 16](https://en.wikipedia.org/wiki/Hexadecimal), or hexadecimal base (hence the name), to specify color mixtures.

**Note:** When a hex color code is composed of entirely of the same characters, the hex color can be abbreviated, like so:

h1 {

color: #FFFFFF;

color: #FFF; /\* This is the same color as above \*/

}

h2 {

color: #AA33BB;

color: #A3B; /\* This is the same color as above \*/

}

* Colors: Hue, Saturation and Lightness

The current revision of CSS, CSS3 (at the time of this writing), introduces a new way to specify colors using HSL colors.

HSL stands for **H**ue, **S**aturation, and **L**ightness. Specifically, this is what each means:

1. Hue - the technical term that describes what we understand as "color." In HSL, hue is represented on a color wheel. It can take on values between 0 and 360.
2. Saturation - the amount of gray in a given color. In HSL, saturation is specified using a percentage between 0% and 100%. The percentage 0% represents a shade of gray, whereas 100% represents full saturation.
3. Lightness - the amount of white in a given color. Similar to saturation, lightness is specified using a percentage between 0% and 100%. The percentage 0% represents black, whereas 100% represents white. 50% is normal.

* Colors: Opacity

You learned that RGB and hex color codes are two different methods of representing the same thing: color. However, there is one feature that RGB colors support that hex color codes do not: opacity.

Opacity is a measure of how transparent a color is. To modify opacity in RGB colors, CSS offers the rgba() value. Note the slight difference in rgb() and rgba().

The extra a character in the rgba() value is known as the alpha value. It represents the opacity of a color. The alpha value can be a number between 0 or 1, inclusive.

**Note:** The alpha value can also be used for HSL colors, using hsla().

* Colors: Compatibility

RGB colors, hex color codes, and HSL colors offer web developers an extraordinary amount of color customization options. As these properties become more advanced, however, it's important to keep in mind that not all users browse the Internet with the same browser, let alone the same version of a given browser.

How does this affect web development? Newer revisions of HTML and CSS affect older browsers. Older browsers, over time, will become dated (possibly obsolete) and not be able to support newer CSS features. For example, many older browsers do not support RGBa, HSL, or HSLa.

Because of this, we must include redundant color options in our CSS code that can cater to a wide audience of different browsers.

Specifically, we can add multiple CSS color declarations, just in case a user's browser can't support a certain declaration.

h1 {

color: rgb(22, 34, 88);

color: rgba(22, 34, 88, 0.4);

}

In CSS, the latter of multiple declarations takes priority. In the example above, if the user's browser supports rgba(), then that color will be applied to the heading. If it does not, then CSS will use the first rgb() color declaration, as a backup.

Using redundant declarations allow you to support as many users as possible across multiple versions of different Internet browsers.

* Fonts

If you've ever used a formatted word processor, chances are that you probably also used a feature that allowed you change the "type of font" you were typing in. The phrase "type of font" refers to the technical term [typeface](https://en.wikipedia.org/wiki/Typeface), or font family.

When setting typefaces on a web page, keep the following points in mind:

1. The font specified in a stylesheet must be installed on a user's computer in order for that font to display when a user visit the web page. We'll learn how to work around this issue in a later exercise.
2. You've probably noticed that we haven't been specifying a typeface in previous exercises of this course. How exactly does the browser know what typeface to use when displaying the web page? The default typeface for all HTML elements is Times New Roman. You may be familiar with this typeface if you have ever used a formatted word processor.
3. It's a good practice to limit the number of typefaces used on a web page to 2 or 3.
4. When the name of a typeface consists of more than one word, it must be enclosed in double quotes (otherwise it will not be recognized), like so:

Earlier, you learned that users *must* have the fonts specified in the stylesheet installed on their computer in order for their browser to display that font. What happens when a font is not installed on a user's computer?

Most computers have a small set of typefaces pre-installed. This small set includes serif fonts and sans-serif fonts, like Times New Roman and Arial, respectively.

When the stylesheet specifies a font not installed on a user's computer, the pre-installed fonts can be used as *fallback fonts* for users.

To use fallback fonts, the following syntax is required:

h1 {

font-family: Garamond, Times, serif;

}

* Fonts: Google Font

New fonts are constantly being developed. Because there are so many new fonts available, it would be unrealistic to expect users to have all of them installed on their computers.

Fortunately, you do not have to predict which fonts are installed on a user's computer. Many (but not all) of the new fonts that emerge on a daily basis are being centralized in directories made available for public use.

For example, Google offers [Google Fonts](https://fonts.google.com/), a directory of thousands of open-source fonts that are free to use by anyone.

To use these fonts, you can link to a specific Google Font in your HTML code and use it immediately in your stylesheet. Because the HTML file links directly to the Google Font, a user's browser can display the typeface you specify. This avoids having to determine whether or not that font is installed on a user's computer.

To use the google font, just use <link> with the Google Font’s directory.

* Fonts: Font Size

Changing the typeface isn't the only way to customize text. Often times, different sections of a web page and are highlighted by modifying the *font size*.

To change the size of text on your web page, you can use the font-size property.

p {

font-size: 18px;

}

There are three units of measurement for font size:

1. px - Represents the unit of *pixels*. The display of a computer monitor can be measured in pixels. A pixel is a small point on the display. How small? Most computer monitors have a resolution of 72 pixels per inch, so a pixel represents about 1/72nd of an inch. Pixels are sometimes also referred to as *points*. Pixels are used to set the exact size of an element, in this case, text.

p {

font-size: 18px;

}

2. ems - Pronounced just as it looks, "em." An em is equal to the width of the letter "m". Ems are a relative unit of measurement. They change the size of text relative to the parent element's size of text.

p {

font-size: 1.3em;

}

3. % - Percentages are also a relative unit of measurement. The default size of text in web browsers is 16 pixels, or 16px. When percentages are used, they set the size of text relative to this default size. For example, setting the font size to 200% would be equivalent to setting it to 32px.

p {

font-size: 150%;

}

* Fonts: Line height

Text on a web page must also be easy to read. When text is styled to appear larger, the vertical spacing between lines of text can decrease, creating text that is difficult to read, particularly in paragraphs.

To avoid this problem, you can modify the spacing between lines of text with the line-height property.

p {

line-height: 1.5em;

}

When the line height of an element is modified, you are manipulating the *leading* (pronounced "ledding") of the font. When the line height is increased, the spacing between lines of text increases, which can make text easier to read.

The line height can be modified using pixels or ems, but the unit of ems is preferred, since ems offer a spacing relative to the size of the text on the page.

* Fonts: Word Space

You can also increase the spacing between words in a body of text, technically known as *word spacing*.

To do so, you can use the word-spacing property:

h1 {

word-spacing: 0.3em;

}

The default amount of space between words is usually 0.25em. In the example above, the word spacing is set to 0.3em, which represents an increase of only .05em in word spacing.

It's not common to increase the spacing between words, but it may help enhance the readability of bolded or enlarged text. Note, again, that the preferred unit is ems.

* Fonts: Letter Space

You've learned how to increase the spacing between lines of text and words, but it's possible to get even more detailed: increasing the spacing between individual letters.

The technical term for adjusting the spacing between letters is called "kerning". Kerning can be adjusted with the letter-spacing property in CSS.

h1 {

letter-spacing: 0.3em;

}

Like word spacing, it's not common to increase the kerning in text, but sometimes it enhances the readability of uppercase text.

* Fonts: Bold

You've probably noticed **bolded** text across many different web sites. It's common to bold important headings or keywords.

In CSS, the font-weight property turns bold on or off.

p {

font-weight: bold;

}

In the example above, all paragraphs on the web page would appear bolded.

The font-weight property has a second value: normal. Why does it exist?

If we wanted all text on a web page to appear bolded, we could select all text elements and change their font weight to bold. If a certain section of text was required to appear normal, however, we could set the font weight of that particular element to normal, essentially "shutting off" bold for that element.

* Fonts: Style

You can also italicize words with the font-style property.

h3 {

font-style: italic;

}

The italic value causes text to appear in italics. The font-style property also has a normal value, for the same reasons discussed in the previous exercise.

* Fonts: Text transform

Text can also be styled to appear in either all uppercase or lowercase with the text-transform property.

h1 {

text-transform: uppercase;

}

The code in the example above formats all <h1> elements to appear in uppercase, regardless of the case used for the heading within the HTML code. Alternatively, the lowercase value could be used to format text in all lowercase.

Since text can be directly typed in all uppercase or lowercase within an HTML file, what is the point of a CSS rule that allows you to format [letter case](https://en.wikipedia.org/wiki/Letter_case)?

Depending on the type of content a web page displays, it may make sense to always style a specific element in all uppercase or lowercase letters. For example, a website that reports breaking news may decide to format all <h1> heading elements such that they always appear in all uppercase, as in the example above. It would also avoid uppercase text in the HTML file, which could make code difficult to read.

* Fonts: Text transform

No matter how much styling is applied to text (typeface, size, weight, etc.), text always appears on the left side of the browser.

The text-align property can be set to one of the following three values:

1. left - aligns text to the left hand side of the browser.
2. center - centers text.
3. right - aligns text to the right hand side of the browser

* HTML ID’s, Labels and etc.

ID’s serves to select a specific element on the HTML code. It helps when the programmer have to change very specific. To select a ID, use “#” before the ID’s name.

#botswana {

background-color: #56ABFF;

}

* HTML ID: Classes

IDs are great for labeling unique elements, but IDs have a limitation. Because unique IDs should not be used across multiple HTML elements, they are insufficient for quickly styling elements that should all share a specific style.

CSS offers a solution to this limitation. For multiple elements that should share the same styling, classes can be used to label them.

To label an element with a class, we can use the class attribute on an HTML element.

<h1 class="science">Scientist Discovers Important Cure</h1>

<h1 class="science">New Study Reveals The Importance of Sleep</h1>

In the example above, there are two headings with a class of science. Why?

HTML elements are often labeled with class names that reflect the content they represent on the web page. In the example above, perhaps a news company decided that all news headlines about science should be labeled with a class of science.

Now that you know how to label HTML elements with a class, we can style elements belonging to the same class at once. How exactly do you select them in CSS, though?

.science {

font-family: Georgia, Times, serif;

color: #A3B4C5;

text-transform: uppercase;

}

Using the dot before class’ name, you are able to modify the labels in the Class.

* Box models

All HTML elements live within a box. Elements on a web page are understood by the browser as "living" inside of a container, or a box. This is what is meant by the box model.

When you changed the background color of an element, you changed the background color of its entire box. When you aligned text, the text was aligned relative to the element's entire box. To truly create custom websites, you'll have to understand the box model.

* Box models: Height and Width

An element's box has two dimensions: a height and a width. In HTML, all boxes have default dimensions. These default dimensions are automatically set to hold the raw contents of the box.

These two properties can be specified with the following units of measurement:

1. Pixels - You learned about pixels when you learned about fonts. This unit lets you set the exact size of an element's box.
2. Ems - This unit sets the dimensions of the box relative to the size of the text within the box.
3. Percentages - This unit sets the dimensions of the box relative to the size of the box that it is encased in. For example, consider an element (a box) of class blue set to a height of 200 pixels and a width of 200 pixels. Inside of blue, consider another box of class red, set to a height of 37% and a width of 45%. The resulting dimensions of the red box would be a height of 74 pixels and a width of 90 pixels.

Developers often use ems or percentages to set box dimensions. Because many web pages are now accessed on mobile devices and other displays of differing sizes, ems and percentages allow boxes to scale depending on the size of a user's display.

* Box models: Min and Max

Because a web page can be viewed through displays of differing screen size, the content on the web page can suffer from those changes in size. To avoid this problem, CSS offers two properties that can limit how narrow or how wide an element's box can be sized to.

1. min-width - this property ensures a minimum width for an element's box.
2. max-width - this property ensures a maximum width for an element's box.

You can also limit the minimum and maximum *height* of an element.

1. min-height - this property ensures a minimum height for an element's box.
2. max-height - this property ensures a maximum height for an element's box.

* Box models: Overflow

When the value of the max-height property is set too low, the contents will spill outside of the box. How can we ensure that this doesn't happen?

The overflow property controls what happens to content when it spills, or *overflows*, outside of its box. It can be set to one of the following values:

1. hidden - when set to this value, any content that overflows be hidden from view.
2. scroll - when set to this value, a scrollbar will be added to the element's box so that the rest of the content can be viewed by scrolling.

* Borders: style

To set your style of border, use on of this key words to complete the code

1. solid - border is a solid line.
2. dashed - border is a series of lines or dashes.
3. dotted - border is a series of square dots.
4. double - border is two solid black lines.
5. groove - border is a groove (or carving).
6. inset - border appears to cut into the screen.
7. outset - border appears to pop out of the screen.
8. ridge - border appears as a picture frame.
9. hidden or none - no border.

* Borders: Width

To set the width of your border, you can change its thickness using “border-width”

It is also possible to also set the border-width property to one of the following named thicknesses:

1. thin
2. medium
3. thick

If necessary, you can change all four of the borders’ thickness:

border-width: 3px 1px 2px 1px;

(Top, Right, Left, Bottom)

If you'd like to be even more specific about the width of different sides of the border, you can use the following properties:

1. border-top-width
2. border-right-width
3. border-bottom-width
4. border-left-width

* Borders: Colors

To set your border’s color, use border-color with someone of styles to set this (RGB, HUE, #FF0000, Red):

border-color: rgb(22, 77, 100);

* Borders: Shorthand

The shorthand way of setting border style, width, and color can be achieved with the border property. Let us look at how we can decrease the amount of code bloat with this property.

border: 3px(thick) solid(style) rgb(22, 77, 100)(color);

* Borders: Radius

The corners of an element's border box can be modified with the border-radius property.

border-radius: 5px;

You can create a border that is a perfect circle by setting the radius equal to the height of the box, or to 100%.

border-radius: 100%;

* Content: Padding

The space between the contents of a box and the borders of a box is known as *padding*. In CSS, you can modify this space with the padding property.

padding: 10px;

The padding property is particularly useful at making text easier to read when the text has a border around it.

You can do as the borders too, set padding of top, right, bottom, left.

padding: 5px 10px 5px 10px;

If you want to be even more specific about the amount of padding on each side of a box's content, you can use the following properties:

1. padding-top
2. padding-right
3. padding-bottom
4. padding-left

* Content: Margin

The margin refers to the space directly outside of the box. You can adjust this spacing with the margin property.

margin: 20px;