December 8, 20

Styling Websites with CSS

https://app.pluralsight.com/library/courses/styling-websites-css/table-of-contents

**Course Overview**

-Types of CSS

-Embedded: a <style> element contains a stylesheet within the head of an HTML file

-<> Inline: Styles are added as an HTML attribute

- CSS external: Styles are stored in a separate CSS file

-The Box Model

-Writing CSS rules

-Styling text and adding fonts

-Sizing Elements

-Different types of CSS selectors

**Creating Style Rules**

-Course Overview and Project Introduction

-creating stylesheets

-writing style rules

-styling text

-sizing elements

-add images, backgrounds and borders

-CSSZenGarden.com, using the style sheet you can see the style changes

-Understanding Inline and Embedded CSS

-Demo: in the index.html open in VS code, and open html in live server. There are 3 different ways to add CSS: embedded, inline, external.

-an inline is adding them inline by applying them to a specific element using a style attribute. It only affects a SINGLE html element, mixes presentation and structure, which isn’t considered best practice.

Diagram

Description automatically generated

-Embedded styles: styles contained in a <style> element, specific to a single HTML file, time consuming, really easy to miss a file or end up with small differences over time

-External stylesheet: CSS is stored in its file, applied through a <link> element, changes can be made in a single location, easy to maintain, and make changes in one single location, one link can apply to multiple files.

-Creating and Linking a Stylesheet

-To start create new folder “Styles”, create new file “main.css”, add in lines 12-72 from index.html to main.css, now link the css file to index.html.

-now link index.html to main:

<link rel=”stylesheet” type=”text/css” href”./styles/main.css >

-<link> attributes:

-rel: explains the relationship

-type: text file; which is CSS in this case

-href: describe location: of CSS file

-Writing Rules:

-press enter twice to give organization

-first thing we write is the selector, what we want to select an html element. Use “p” for paragraph and add {} to enclose it.

**CSS Rule Syntax** i.e

p {

color: red; //add property and value to ‘p’

}

A picture containing text

Description automatically generated

How to change the default presentation of an HTML

**SUMMARY:**

-CSS can be added inline, embedded, or externally.

-External stylesheets are considered best practice.

-Styles are made up of rules containing property and value pair called declaration.

**Styling Text**

-Module Overview and Formatting CSS

-changing text size, weight and style

-aligning text

-adjusting leading and tracking

-using web-safe and custom fonts

-Go to brick icon>>search for Beautify>>select second option “Beautify” one that works with JavaScript, HTML & CSS

-Changing the Size, Weight, and Style of Text

-In main.css, scroll all the way to the bottom, choose selector of element we’ll be styling:

-

p {

font-size: 18px;

}

h3 {

font-size: 22px;

font-weight: normal; //since it’s already bold

}

li {

font-weight: bold;

}

figcaption {

font-style: italic;

}

-Altering Line-Height and Letter-spacing

-go back to list item, decrease white space in is line height property:

li {

font-weight: bold;

line-height: 1.5; //text readability for the web is 1.5

}

- tracking: space in between characters; press enter twice:

Create group selector >>

h1, h2, h3, h4, h5, h6 {

letter-spacing: 1px;

}

**Line-height:** changes the amount of whitespace between the lines of text.

**Letter-spacing:** changes the amount of space between character

-Aligning and Transforming Text

>> **Align text:**

h4 {

text-align: center; //can align it left, right, center or justify. Justify aligns it evenly between two things. They stretch to take the width of their container.

}

>> **text-transform :** alters the case of the text

In all of the heading group, we’ll use text-transform:

h1,h2,h3,h4,h5,h6 {

letter-spacing: 2px;

text-transform: capitalize ;//options are capitalize, full- width, inherit, lowercase, none, uppercase

}

-Using Fonts for the Web

-Go into 03 folder>Demos>03> promotions.html>> but also what we added in 02 main.css, we have to add in 03 main.css!!

Change font our heading our using.

Font-family property:

h1,h2,h3,h4,h5,h6 {

letter-spacing: 2px;

text-transform: capitalize;

font-family: Verdana, Arial, sans-serif;

}

Adding in a specific type of font you need “”, quotation marks, to let the browser know where the font family begins.

**Web-safe font:** a font that comes installed on most operating systems.

Text

Description automatically generated with low confidence

**Cssfontstack.com** to check a complete collection of web safe CSS font stacks. You can come in and see exactly what percentage of support the font has.

**Font stack** is a fall safe option just in case the first couple of fonts doesn’t work on the browser.

-Using Externally Hosted Fonts

Diagram

Description automatically generated

Go to fonts.google.com, search font “Cookie”, choose select this style, then search for Lato and select regular 400 weight, and select the regular italic weight, top right corner you can view your selected font families, copy the link, paste the link on every html document that you’d want to use those fonts, PASTE ABOVE THE STYLESHEET!! That way we import the font prior.

Now add those fonts to main.css,

Font-family: ‘Cookie’,

One thing about importing font is that the browser makes headings bold and we didn’t import Cookie bold. It’s going to create it’s own version this is called browser synthesis. We can use normal value for font weight property.

Graphical user interface, text, application, chat or text message

Description automatically generated

-Adjust the letter-spacing from 2, to 1px.

-If we want to be able to apply a certain font to not only paragraphs but to list items and such, we can apply it to a parent element because the **font-family property is inherited by descendant elements from their parent element.** If we use the biggest parent element like the body, it would be a good way to apply all of our text. So scroll and change the font on the body.

Graphical user interface, text, application, chat or text message

Description automatically generated

**Sizing Elements**

-Module Overview

-Changing an element’s size

-Understanding different types of units

-Adding colors and borders

-Working with images

-The CSS Box Model

-Changing the Color and the Size of Elements

-Styling promotional items by adding a background color to these elements: 04 folder > Demo > 04 > styles/main.css & promotion.html

Target one element, use **class selector**: selects a set of elements with matching class attributes. Add markup on html first, on first <div class=”promotion”>, then add it to the other divs on the promotions. Then switch back over to main.css, for class selection, use a (.) period:

* .promotion {
* background-color: #23CEA6;
* color: white; //change the color from the parent element
* width: 450px; // We have a CSS property called the width property, that will let us change the width of element.
* text-align: center;

}

-Next add an image above each promotional heading in promotion.html:

- <img src="images/products/applepie.jpg" alt="Apple Pie">

Because the image takes over the div box, we’ll have to adjust the image by using relative units.

-Relative and Absolute Units

-absolute unit is using px or not even adjusting the images.

-relative units are flexible and they’re calculated relative to the size of their parent element. Target the image in main.css:

.promotion img {

width: 50%; //doing a percentage we don’t have to recalculate every time.

}

A picture containing diagram

Description automatically generated

-Adding Borders and Outlines

-When trying to add border property value in CSS: the options are width, border style, and color value. The border color also depends on the text color if you don’t specify a color value for the border.

Table

Description automatically generated with low confidence

Text

Description automatically generated

To apply a border to just one side of the element: border-top, border-right, border-left, border-bottom.

-Add outlines using the outline property. Just like the border property, outline has 3 values as well: width, style, color. Border adds to content box element but the outline just outlines the box element.

-Adding Margins and Padding

-Add an image to other promotion div. We can add additional class by just using a space so the cheesecake promotion div is:

<div class=”promotion promotion-purple”>

<img src="images/products/cheesecake.jpg" alt="Cheese cake">

-In main.css,

.promotion-purple {

background-color: #a593c2;

border-color: #23CEA6;

}

Padding property: allow to add whitespace inside an element: by text and above the image as well. Padding with no specification of “left, right, bottom, top” will add equal padding to that element.

Text

Description automatically generated

Padding: top right bottom left; << it’ll affect, respectivately, the top, the right, the bottom and the left.

Padding: top/bottom left/right; << If we added two values to the padding shorthand, the first value would affect the top and bottom, and the second would be applied to the left and right.

Padding: top/bottom/left/right; << we can apply one value like we just did and it will get applied to all four sides equally at the same time.

Margin: adds whitespace outside of an element

Margin: top right bottom left; <<respectively apply to the top, right, bottom, left.

Margin: top/bottom left/right; << two values apply to the top and bottom and left and right.

Margin: top/bottom/left/right; <<applies one value equally at the same time

Graphical user interface, text

Description automatically generated

\*\*Watch out for **vertical margin collapse**. If you have margins touching, the largest gets applied instead of both. If they’re equal, only one is getting applied. \*\*

-Understanding the Box Model

Graphical user interface

Description automatically generated

**-Box Model:** Model that helps us identify exactly how the width and height of an element is calculated.

-An element could either have it’s width and height be calculated by the browser or set specifically in CSS using the width and height property.

-When padding is added to an element, then that padding gets added to the render box, which means that default width and height is either is determined by the browser or as you the author. So when we add padding, it gets added to that width and height.

-When we add borders, it gets added outside of that padding, it also affects the width and height of the element.

-Margin gets added as whitespace outside of the content box or the render box of an element, but they don’t affect an element’s width and height.

Graphical user interface

Description automatically generated

Let’s say we have an element with a width of 400 and height of 200. Add 40px of padding on all 4 sides, the new width of the element would be 480 by 280. If we were to add a 20px border on all 4 sides, the new width and height of the element would be 520x320. If we were to add 100px to the element on all 4 sides, the width and height of the element itself would be 520x320, since the margins doesn’t get added to the element.

\*\*\*One reason a website layout breaks is because there’s not enough width or height or room for an element in a specific space. Often the culprit is forgetting to add the padding and the amount of borders to the total width and height of the element to figure out how much space is needed. One great thing you can do is the browser will show you the actual box model of an element. So if we use the inspect, developer tool, here you can see the padding, margin information as well. \*\*\*

**SUMMARY:**

**-Changing the color and size of elements**

**-Relative and absolute units**

**-Borders and outlines**

**-Margins and padding**

**-The Box Model**

**Using Selectors**

-Module Overview

-Class & descendant selectors

-Combinators

-Pseudo-classes

-Pseudo-elements

-IDs and Classes

-ID selector: targets a single HTML element using a unique identifier. Use a (#) hash tag in CSS when associating an id tag with an html element. The downside of IDs is they can only be used one time for HTML page or HTML document. If an element only appears one time for a document, then that’s totally fine and ID can be an appropriate selector to use. In general, if you can use a class, it’s easier to maintain your CSS because class styles are reusable, since you can apply them to multiple elements.

-In 05 >> DEMO >> 05 >> promotion.html & main.css

-Add in promotion.html

<h4 id="sidebar-heading">Browse our pies</h4>

-Add

#sidebar-heading {

font-size: 24px;

}

-Combinators

-Combinators: a category of selectors based off the hierarchical structure of elements. They allow you to create styles based on HTML patterns and the relationships between elements.

-Our .promotion image selector is an example of this called a descendant combinator, in which it targets any descendants of the specified parent. We’re targeting any image inside of the promotion class.

-If our website is growing and we have new content patters starting to emerge, for example our sidebar now had been expanded to include subcategories within each category of pie. So we’ve expanded our sidebar navigation.

-Using a special combinator to target the outside list only, use a special character to signify a specific relationship. One of these is the child combinator, which targets only the direct child of an element and it uses a caret( > ) sign.

-Change the list in main.css using this method:

Graphical user interface, text

Description automatically generated

The ul and li becomes red but the embedded ul/li remains blue.

-If we target the embedded list, it is the direct child of the ordered list so we can target it like so.

- Graphical user interface, text, application

Description automatically generated



Text

Description automatically generated

-Pseudo Classes

-**Pseudo class:** selectors based on an element state or position. We can use pseudo classes to style some links.

Target left menu links:

#leftmenu a {

color: #199175;

}

When we hover over a link, it’s actually a class that we can target.

#leftmenu a:hoover {

color: #116552;

}

As you hover over the link, you can see the color change.

Other ex: focus: if someone uses keyboard navigation, you can see the browser has some default focus styles that adds a border to the element so you can see where you’re navigating.

-Other types of pseudo-classes can use include: location-based pseudo classes, first type of an element or last type.

-Go to pieoverview.html, rather than added classes to each individual element, we could use pseudo-class instead. Every time we have a table on our site, we aren’t required to add that class to each and every element, we already have that style in place to automatically style our table in that way.

-Back in main.css, we can target any **table row, tr element**, on our site. Use pseudo class to target a specific position in that table. Ex, **nth-of-type**, in the parentheses we can specify the number or position of what we want to target.

tr:nth-of-type(1) {

background-color: red;

}

This changes our first row of the table background color red in pieoverview.html. In addition to specifying a specific number, we could also specify keywords as well, such as odd and even. This would be great if we want to alternate different color rows.

tr:nth-of-type(odd) {

background-color: #b4ddd3;

}

tr:nth-of-type(even) {

background-color: #E0D7EF;

}

Table

Description automatically generated with low confidence

-Course Conclusions

Table

Description automatically generated with low confidence

Graphical user interface, application

Description automatically generated