HTML & CSS Workshop

Tyler Schwehr

What we'll cover

- Website composition
- HTML
- CSS
- Website project
- Tailwind

How websites work

- Hyper Text Transfer Protocol (HTTP) is the protocol that facilitates your ability to view websites.
- When you visit a website, your browser submits a HTTP request to the given web server, and the server sends back the relevant files
- At the most basic level, these are HTML and CSS files
- Browsers interpret these files and present you the result

The components of a website

- There are several distinct components to a website, some of these are interchangeable with different languages / methods
- HTML - Tells the browser what the content of the site is
- CSS - Tells the browser how to style and display that content
- Javascript - Programming code to perform tasks on the users' machine
- PHP - Server side code to complete all other logic for the website

What is HTML?

- HTML stands for Hyper Text Markup Language
- You modify how content is displayed by “tagging” it
- All tags use <> brackets
- Most tag types require closing such as <p>some text</p>
- You can always use W3Schools for reference

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

HTML Basics

- HTML documents have several distinct sections
- Firstly declare the document type and start HTML <!DOCTYPE html> <html>
- Next we have the header <head> this contains the metadata and externally linked components
- The body <body> contains the content
- Finally the footer <footer> which just denotes the footer content
- The are 6 headings tags <h1> <h2> <h3> <h4> <h5> <h6> each one is progressively smaller
- Paragraphs have in built formatting <p>
- Line breaks are self enclosing and don’t have a closing tag <br />
- The <title> tag sits in the header and displays the page name
- HTML elements can be nested within each other

HTML Attributes

- Some tags have additional information held within them
- These are known as attributes
- Attributes are assigned by an equals sign and must always receive a string
- The image tag is a great example of this <img src=“some_img.jpg”>
- An anchor tag or link tag uses a similar attribute

<a href=“some_link.com”>Click here</a>

- There are many other attributes out there, but beware some have been deprecated
- A great example is the width and height attributes for images are better done in CSS
- There is also an attribute for styling an element but we'll cover that further in the CSS section

Text Formatting

- Like a lot of HTML some aspects are deprecated, so we will only cover the current rules
- To bold some text we use the <strong> tags
- To add italics we use <em>
- For underlining we use <ins>
- You might be thinking that some of these tags are strange, you’re not wrong
- To put a line through the text we use the <del> tags
- We can highlight text with <mark>
- For subscript use <sub>
- Superscript uses <sup>

Tables & Lists

- There are a few components to tables
- Start by declaring the table <table>
- Next we declare the row <tr> this needs to be on every row
- In the top row we use <th> on every cell to note table headers
- In the following rows we use <td> tags to note the cells

- We can declare an unordered list or bullet point list using the <ul> tags
- To declare an ordered list we use the <ol> tags
- Both sets of lists use <li> to denote list items

Page structures

- Every element has a default display type, either block or inline.
- Block elements begin on a new line and consumes the full width of available space
- Inline does not start a new line and only takes up as much width as it needs
- The <p> tag is a block element, while the <strong> tag is an inline element
- Understanding these display types helps us create the structure of our websites
- The <div> tag is used for structuring a web page and to contain other elements
- The <div> tag is a block element
- <div> tags are typically assigned classes according to how they are used within a given context

Forms

- There are a range of different form elements, we’re only going to cover the most basic stuff
- The <form> tag denotes all included elements are within the same form
- The most important element within a form is the <input> tag
- The <input> tag uses a special attribute call type to further define it
- The <label> tag is used to create a label for a given <input> tag
- In the <input> tag we must give it a unique id with the id attribute, such as id=“first_name”
- In the <label> tag we can then reference this in the for attribute, such as for=“first_name”
- Finally every form must have a submit button, using the <input> tag with the type=“submit” attribute

Classes, IDs and responsiveness

- We’ve briefly mentioned these two attributes in the previous slides
- Classes and IDs play a key role in further development of your web page
- Classes identify a collection of elements, each element can be a member of many classes
- IDs identify a specific element, never use an ID twice on a website
- The use of IDs and classes will become more apparent in the CSS section coming up
- A website being responsive just means that it responds to the different screen sizes
- There are many ways to handle responsive designs
- The most basic action is to add this tag to the header <meta name=“viewport” content=“width=device-width, initial-scale=1.0”>

What is CSS?

- CSS stands for Cascading Style Sheets
- CSS is used to describe how to style the HTML elements on a web page
- They can be layered to create interesting and complex designs
- CSS allows you to design many pages at once with minimal work but can be complex if poorly managed

```
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

CSS Hierarchy

- CSS can be written at three different levels, externally, internally, and inline
- These layers have an intrinsic hierarchy

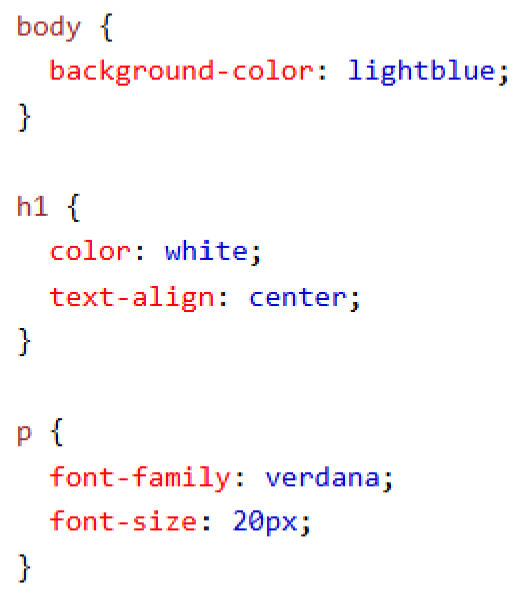
1. Inline
2. Internal
3. External

- Where a conflict exists between two levels, the browser will use the CSS written at the highest level

- Inline CSS is written directly into the html element with the attribute ‘style’
- Internal CSS is written within <style> tags and kept within the header section
- External CSS is written in a separate file and referenced within the header of the HTML
- <link rel=“stylesheet” href=“mystyle.css” />

CSS Selectors

- Within a style sheet you must declare which elements you would like to style for a given section, you can select more than one
- There are many ways in which you can use selectors but there are three basic ways
- Element selection by simply stating the element type to be changed
- Class selection that selects all elements that belong to a given class, you use the class name preceded by a full stop .class_name
- ID selection changes the element with that specific ID, you use the ID name preceded by the pound symbol #id_name



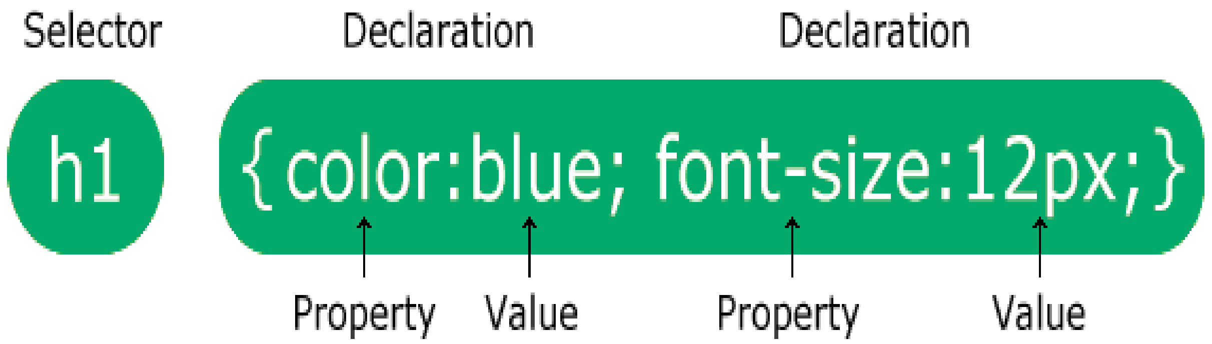
```
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

CSS Syntax

- The syntax for CSS is sets of property:value pairs
- This syntax is focusing on external/internal CSS, inline CSS is a little different
- As mentioned in the previous slide, we start a statement with our selectors
- Following the selector you encapsulate the code block with curly braces
- Each property uses a colon to separate the value and every line ends with a semi-colon



CSS ~~Colours~~ *Colors*

- Colors can be used in a variety of areas, such as background, text, borders and more
- There are pre-defined colors that can be used by string, such as ‘blue’, ‘red’, ‘yellow’, you can google to find the full range
- You can also pass colours by rgb values, hex values, or hsl values
- Coloring background uses the property “background-color”
- Coloring text uses the property “color”
- Rgb values require an rgb declaration and parentheses, like this “rgb(180, 99, 12)”
- Hex values are denoted by the # symbol, such as #6f42ae

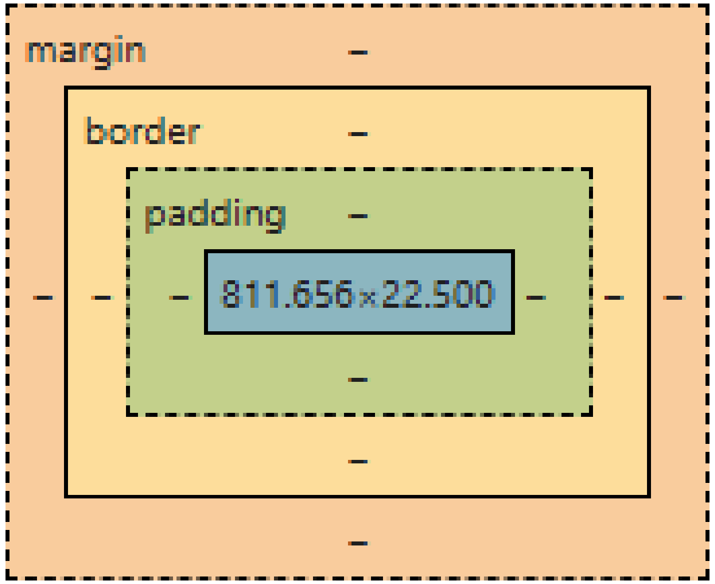
CSS Borders

- Borders are split between specific properties and a composite property
- A composite property allows you to declare multiple values at the same time
- Here is an example “border: 2px, solid, red”

- There are many styles for borders, which you can find on w3schools, for now we’ll look at solid
- A border style uses the property ‘border-style’
- Border width uses ‘border-width’
- Border color is ‘border-color’
- You can round the corners with ‘border-radius’

CSS Margin & Padding

- Padding is the space between the element and the border, where margin in the space between the border and another element
- Each can be declared targeting a specific side, such as ‘padding-top’
- Or they can be used as a composite ‘padding: 10px 25px 10px 25px’
- The values describe the top, right, bottom, and left in that order



CSS Width & Height

- The basic width property and height property are quite straight-forward, behaving as you would expect
- You can also enforce a minimum width/height with the ‘min-width’ property
- The same is true for maximum width/height using the ‘mx-height’ property
- So far we’ve been using pixels to define all of our measurements, which us the ‘px’ suffix
- You can also use real world units such as ‘cm’ and ‘mm’
- We also have access to relative measurements, ‘%’ is percentage relative to the parent element
- ‘vh’ and ‘vw’ are relative to the viewport height and width

CSS Text Styling

- We already covered color, but there are many different style options for text, we'll cover a few key points
- Firstly we can align the text with 'text-align'
- We can alter the capitalization using 'text-transform'
- We can also adjust the line height with 'line-height', which we typically use an 'em' value, which is relative to the font size
- Not every computer has access to all fonts, so we declare a set of fonts known as a family, this creates a failsafe if a font isn't found
- An example would be 'font-family: Arial, Helvetica, Roboto;'
- Font size is declared as 'font-size'
- Font weight describes the thickness of the font and uses the property 'font-weight', not all fonts can use this property

CSS Positioning

- There are several ways of positioning elements on a page, the default is static
- Positions are declared using the ‘position’ property
- Relative position is one that is relative to its static position, using directional properties such as ‘left’, ‘right’, ‘top’, and ‘bottom’ will move its relative position
- The fixed position is fixed to your viewport, given a position using directional properties
- The sticky position switches between relative and fixed, it’s great for navbars
- Sticky positioning requires a directional property to denote where is should stick to

CSS Pseudo Classes

- Pseudo classes describe the different states of a given element
- We declare the pseudo class by following a property with a colon and the pseudo class name
- Here is an example of the syntax “div:hover”
- The <a> tag is a key area where pseudo classes get used
- The “a:link” is used for unclicked links
- The “a:visited” is used for links previously visited
- The “a:hover” is for when you hover over the element
- And the “a:active” is used for a currently selected link
- These pseudo classes can be used on several different elements

CSS Pseudo Elements

- Pseudo elements are used to select specific parts of an element, such as the first line, the space before an element, and the markers of a list
- Pseudo elements use a similar syntax to pseudo classes, but instead use a double colon between the property and the pseudo element name
- Here is an example “p::first-line”
- The “p::first-line” allows us to modify the properties of the first line of every <p> element
- In a similar fashion the “p::first-letter” lets us access the first letter
- We can use “p::before” and “p::after” to insert content either before or after an element
- The ::selection can be used to style parts of the page selected by the user

CSS Variables

- Just like with programming, variables are used to store data
- I typically use variables for my color palette and font choices, but you can use them in other ways
- Variables make it very easy to test/change colors for your whole website in one stroke
- Variables can be declared globally or locally for a specific selection type
- Variables are declared with double dash, variable name, colon, and variable value
- Here is an example “--white: #ffffff“
- To declare it globally we use the root pseudo class “:root”
- To call a variable we use the “var(--white)” function, with the variable name in the parentheses

Design your own website - LazOne

Now that we've covered the basics of HTML and CSS, your job is to build a simple website.

Remember to use external resources like W3Schools as a reference.

Ask for help as you need it.

Set up a new folder for your project, and keep all the files together, this will make it easier to reference each file.

~~Once you've had some time to work on this, we'll cover tailwind next~~

Tailwind is hard to learn, but the first step is understanding vanilla CSS.

Thank you

Tyler Schwehr