Coding HTML with NVDA -Part 11(Introduction to CSS)

# Introduction

# Start of tutorial

So, for the past 10 videos in the series, I have been teaching you guys how to use html to create and display different elements to the user on your webpage. But what I didn’t teach you, is how to style those elements and make them look good. That is where CSS comes in. CSS stands for Cascading Style Sheets and is the language used to style and layout your html elements. So how does it work you may ask? to define how to display html elements, CSS uses rules. A CSS rule consists of a selector and a declaration block. To write a CSS rule, you first have to put the selector, then the declaration block. The selector points to the HTML element you want to style. The declaration block is then surrounded by curly braces. You can get these braces by pressing shift and either your left or right square brackets, which are to the right of your P. The declaration block can contain one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, separated by a colon. Multiple CSS declarations are separated with semicolons. The basic syntax for a CSS rule is:

selector {

 property1: value;

 property2: value;

}

Now let’s focus on the selectors.

# Selectors

The selectors in CSS are used to select the html element or elements that you want to style. Now there are many ways to do this, as there are various types of selectors in CSS. This is where you need to pay attention and listen closely.

## The element selector

This type of selector selects the html elements based on their element name. This means that whatever rule you write, will apply to all the elements of that type in your html document. For example, if you want to write a rule that applies to all the heading level ones in your html document, you would do this:

h1 {

 property: value;

}

## The ID Selector

The id selector uses the id attribute of an HTML element to select a specific element. What is this ID attribute you may ask? The HTML id attribute is used to specify a unique id for an HTML element. You cannot have more than one element with the same id in an HTML document. The value of the id attribute must be unique within the HTML document. The id attribute is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific id. So its like if you were in a school setting, where each student has their unique id. No student has the same id. In the html code, to give an element a unique id, in its starting tag, you would use id as the attribute and assign the unique name as its value. For example:

id="myName"

Note that the id name must contain at least one character, cannot have any spaces, cannot start with a number sign, and must begin with a letter. Now back to the id selector in CSS.

In order to select an element with a specific id, type the number sign or hashtag, then the id of the element. For example, the CSS rule below will only be applied to the element with the id “myName”:

#myName {

 property: value;

}

Now let’s move on to the next type of selector.

## The Class Selector

The class selector selects HTML elements with a specific class attribute. What is the class attribute you may ask? The HTML class attribute is used to specify a class for an HTML element. Multiple HTML elements can share the same class. If we continue with the analogy that I mentioned above, Each student has their own unique id, but more than one student can belong to a class. The class attribute is often used to point to a class name in a style sheet. It can also be used by JavaScript to access and manipulate elements with the specific class name. To assign an element to a class, You would use the class attribute in the starting tag of that element, with the name of the class being the value. For example:

class="myClass"

HTML elements can also belong to more than one class. To define multiple classes, separate the class names with a space. For example:

class="myClass1 myClass2"

Different HTML elements can also point to the same class name. This means that a h1 element and a paragraph element can be in the same class. Please note that your class names must begin with a letter. Now back to the class selector

To select elements with a specific class, write a period or full stop, then the class name. For example, all the elements in the boy class will be styled by this rule:

.boy {

 property: value;

}

You can also specify that only specific HTML elements should be affected by a class. In this example, only paragraph elements in the boy class will be styled by the rule:

p.boy {

 property: value;

}

Now let’s move on to the next selector.

## The Universal selector

The universal selector selects all HTML elements on the page. This is represented by the asterisk. For example, the rule below will affect all the elements in the document:

\* {

 property: value;

}

Now to the last selector.

## The Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions. This is used to group the html elements that have the same styling as one. For example, if you want to bold both the heading level one and the paragraph elements, you don’t have to write two separate rules like this:

h1 {

 property: value;

}

p {

 property: value;

}

You can easily group them into one rule. To group selectors, separate each selector with a comma. So the code will be simplified to:

h1, p {

 property: value;

}

And those are the most commonly used selectors in CSS. You can view the others via the link at the end of this document.

# Inserting CSS

Now that you know how the CSS rule works, There are 3 ways in which you can insert CSS into your html documents. You can do this by using external CSS, internal CSS or in line CSS.

# Internal CSS

As the name suggests, this is where the CSS code will be located within the html document. This method can be used if you are creating a website with only one page, or if a single page has a unique style. In order to do this, your CSS code must be placed in between an opening and closing style tag, within the head section of the document. For example:

<head>

 <style>

 h1 {

 property: value;

 }

 </style>

</head>

# External CSS

Now as this name suggests, the CSS code will be located outside of the html document. This is ideal to use when you have a website with more than one page, and they all share common styles. This is also great for separating your CSS from the html, which makes the code cleaner and more organized. An external style sheet can be coded in any html editor, but the file must be saved with the .css extension. In order to use an external style sheet, you would use the link element.

All of your html pages will have to include a reference to the css file inside the head section, using this link element. For example:

<head>

 <link rel="stylesheet" href="style.css">

</head>

So, it is obvious that the link element is an empty element. Now in the example above, the rel attribute names a relationship of the linked document to the current document. So, in this case it is a style sheet. The href attribute is used to point to the location of the file. The url can be both relative and absolute. In this case the file is called style.css. In the actual CSS file, you would just write out the rules as normal, omitting the opening and closing style tags.

## In line CSS

This may be used to apply a unique style for a single element. To use inline styles, we would add the style attribute to the relevant element. The style attribute can contain any CSS property. Here is a simple example:

<h1 style="property1:value;property2:value;">This is a heading</h1>

As you can see, this method can complicate the code and is not usually recommended.

# Multiple style sheets

If some properties have been defined for the same selector in different style sheets, the value from the last defined style sheet will be used. For example, if you set a rule in the external style sheet for the paragraphs to have blue text, but in the internal CSS, you have a rule that say that the text in all paragraph elements should be yellow. If the external style sheet is defined before the internal css, the paragraph text will be yellow. However, if the internal style is defined before the link to the external style sheet, the text for the paragraph elements will be blue. Please note however, that in line css will overwrite both internal and external CSS.

There is also another method to overwrite other CSS rules. This can be done by using the important property. This property will override ALL previous styling rules for that specific property on that element. The important property is represented by an exclamation mark, followed by the word important. This is placed between the value of the property and the semi colon. For example:

p {

 property: value !important;

}

Also an important note before we end this video. You may also want to use comments in your css code in order to let anyone reading it know what each block of code is doing. CSS comments starts with /\* and ends with \*/. Any text placed between these will be ignored by the browser. Of course, in internal CSS, these comments have to be in between the style tags. For example:

/\* This is a single-line comment \*/

/\* This is

a multi-line

comment \*/

link to other selectors:

https://www.w3schools.com/css/css\_selectors.asp