Coding HTML with NVDA -Part 16(CSS Padding)

# Introduction

Hey guys what’s up it’s your girl Thee Quinn here and I’m back with another video in the html series. In today’s video I will be showing you how to add padding to your html elements with CSS. But before we get into it, please be sure to like the video if you love the content, subscribe if you are new and turn on my notification bell to be alerted whenever I post the next video. Also note that the instrumentals for this video have been provided by Inner Sanctum Entertainment Ltd. But without further ado, let’s get right into it.

# Start of tutorial

So, in today’s video I am introducing you guys to padding in CSS. If you watched the box model video, I described to you the whole model of CSS elements. You should imagine that they have a box around them. So, imagine the box around the element on the screen. It would have a top, left, bottom, and right side. Now remember that the content of the element would be within that box and the padding is the space between the content and the actual border or the sides of that box. In CSS, Padding is used to create space around an element's content, inside of any defined borders. So, while a margin is the space on the outside of an elements border, the padding is the space between the elements content and the border. So, let’s take a button for example. If we have a blue button with white text, it would be like a blue rectangle if its not a rounded button, and it would have the white text on it. The padding would be the space in between where the edge of that blue rectangle is and the white text itself. So, there are properties for setting the padding for each side of an element (top, right, bottom, and left). These are:

* padding-top
* padding-right
* padding-bottom
* padding-left

All the padding properties can have the following values:

* Length

This specifies a padding in a length unit such as px, pt, cm, etc.

For more on length units, you can check out my CSS units video.

* %

This specifies a padding in a percentage of the width of the containing element

* Inherit

This specifies that the padding should be inherited from the parent element

Note: Unlike CSS margins, negative values are not allowed for padding. Here is an example for setting the padding for all sides of a paragraph element:

p {

 padding-top: 5px;

 padding-right: 10px;

 padding-bottom: 15px;

 padding-left: 20px;

}

# Shorthand

it is also possible to specify all the padding properties in one property. The padding property shortens the code and can have from 1 – 4 values. The padding property is a shorthand property for the following individual padding properties in this order :

* padding-top
* padding-right
* padding-bottom
* padding-left

now here is an example if the padding property has four values:

padding: 5px 10px 15px 20px;

In this example, the top padding is 5px, the right padding is 10px, the bottom padding is 15px and the left padding is 20px.

Now If the padding property has three values, for example:

padding: 5px 10px 15px;

in this example, the top padding is 5px, the right and left paddings are 10px and the bottom padding is 15px.

And what if it has two values? Here is an example:

padding: 5px 10px;

in this example, the top and bottom paddings are 5px and the right and left paddings are 10px.

And if the padding property has one value, for example:

padding: 5px;

In this example, all four paddings are 5px.

# padding and width

there is something to note when using padding with an element that has a specified width. Remember that each element has a border around its contents which is like a box. And the padding is the space between the element’s content and the border. Where the content is, is called the content area. Now, the CSS width property specifies the width of the content area of an element. So, if an element has a specified width, if padding is added to that element it will be added to the total width of the element. For example, if we set the width of a paragraph element to 100px, and add a padding of 10px on all sides of it:

p {

  width: 100px;

  padding: 10px;

}

In this example the actual width of the paragraph element would be 120px. That is the 100 for the width that we set, 10px for the left margin and 10px for the right margin. This will make the element wider than how we want it. In order to fix this and keep the width to what we set it as, we can use the box-sizing property. When we use this property, if we increase the padding, the space of the content area will decrease, leaving the total width of the element as we set it. So, let’s use the same example with the box-sizing property:

p {

 width: 100px;

 padding: 10px;

 box-sizing: border-box;

}

Now the width of the paragraph element will stay at 100.