Coding HTML with NVDA -Part 15(CSS Margins)

# 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 margins 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 most of you should know what margins are. If you have used Microsoft Word, we would have set the page margin. Now that would be the space in between the words on the page itself and the actual edge of the page. That is what is called the margin in Microsoft Word. Now in CSS, margins are used to create space around elements, outside of any defined borders. Now remember I did a video on the CSS box model and I said to imagine that every element has a box around it. You know it would have a top, right, bottom, and left side and the sides of that box would be its border. Outside of the border is where we have the margin. If you missed that video, you need to get back to it as well as the borders video while you are at it. Now margins are great, especially when you want to put some space in between your elements. Like say for example you have two buttons side to side and you don’t want them to be joined together, you want some amount of space in between them. Or if you have headings or paragraphs and you want a certain amount of space in between each paragraph. So those are different ways that margins can be used, as well as defining the space in between all the text and content on your webpage to the edge of the computer screen. So, in CSS, there are properties to set the margin for each side of an element (top, right, bottom, and left). To specify the margin for each side of an element you would use either:

* margin-top
* margin-right
* margin-bottom
* margin-left

Now each of these properties can have one of four values:

* auto

This allows the browser to calculate the margin

* length

specifies a margin in length units such as px, pt, cm, etc.

* %

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

* inherit

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

Here is an example of setting different margins for each side:

h1 {

  margin-top: 10px;

  margin-bottom: 20px;

  margin-right: 50px;

  margin-left: 80px;

}

# Shorthand

it is also possible to specify all the margin properties in one property. We can do this with the shorthand margin property. This property can have from 1 to 4 values and they must be written in this order:

* margin-top
* margin-right
* margin-bottom
* margin-left

So here is an example if the margin property has four values:

margin: 5px 10px 15px 20px;

so, you see that each value is separated by a space. In this example, the top margin is 5px, the right margin is 10px, the bottom margin is 15px, and the left margin is 20px.

Here is an example if the margin property has three values:

margin: 5px 10px 15px;

In this example, the top margin is 5px, the right and left margins are 10px, and the bottom margin is 15px.

What if the margin property has two values:

margin: 5px 10px;

In this example, the top and bottom margins are 5px and the right and left margins are 10px.

And if the margin property has one value:

margin: 5px;

all four margins are 5px.

Also note that you can use negative numbers for margins. Negative values draw the element closer to its neighbors than it would be by default.

# Auto value

You can set the margin property to auto to horizontally center the element within its container. The element will then take up the specified width, and the remaining space will be split equally between the left and right margins. Here is an example:

div {

  width: 125px;

  margin: auto;

}

# Inherit value

Now here is how the inherit value works. Say for instance we have a div element with a paragraph element within it. That would mean that the div element is the parent, and the paragraph element is the child. So, if we set the margins of the div element to be 5px and we want the same for the paragraph element that is in it, we can simply do this:

div {

  margin: 5px;

}

p {

  margin: inherit;

}

And there you go, so it would inherit the margin of its parent.

# Margin collapse

Before I end this video, there is one more thing you should know. Vertical margins (which are the top and bottom margins) of adjacent block level elements are collapsed into a single margin. The size of this space is equal to the largest of the two margins. This only happens on top and bottom margins, and not right and left margins. Block level elements are those that always start on a new line and take up the full width of their container. Some examples of block elements are the paragraph and div elements. You can find more information on block elements and a complete list of them via the link at the end of this document. So, say we have this example:

h1 {

  margin: 0 0 50px 0;

}

p {

  margin: 20px 0 0 0;

}

In this example, the h1 element has a bottom margin of 50 px and the paragraph element has a top margin of 20px. Say in the html code the h1 element was placed on top of the paragraph element. We would look at it and say that 50 + 20 is 70. So, the space between the 2 would be 70 px. But due to margin collapse, the space between the two would actually be the largest of the two, which is 50px.

# Links

Block level elements:

<https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements>