Coding HTML with NVDA -Part 17(CSS Height and Width)

# 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 introducing you to the CSS height and width properties. 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, by now you should know that when we are measuring height, we are talking about how high something is. So, measuring it from top to bottom. And when we are talking about width, we are talking about how wide something is. So, measuring it from left to right. Now in CSS, the height property is used to set the height of an element, and the width property is used to set the width. The height and width properties do not include padding, borders, or margins. If you don’t know what those are, you can go back to the three previous videos in this series where I discuss those. So, the height or width properties only sets the height or width of the content area of the element. The height and width properties can have the following values:

* auto

This is the default value. With this value, the browser calculates the height and width.

* Length units

This defines the height or width in length units such as px, cm, etc.

* %

This Defines the height/width in a percentage of the containing block.

* Initial

This sets the height or width to its default value.

* Inherit

The height or width will be inherited from its parent.

Here is an example of setting the height and width for the paragraph element:

p {

height: 200px;

width: 50%;

}

This element has a height of 200 px and its width is half of its parent’s width. So, if a div element is its parent and its width is 100, then the paragraph elements width would be 50.

# Max-width

The max-width property is used to set the maximum width of an element. If the content is larger than the maximum width, it will automatically change the height of the element. If the content is smaller than the maximum width, the max-width property has no effect. The value of the max-width property overrides the width property. This prevents the value of the width property from becoming larger than max-width. The max-width can be specified by one of the following:

* length values
* %
* initial
* inherit

none (This is the default and Means that there is no maximum width).

The max-width property works great in cases when the browser window is smaller than the width of the element. In this case, the browser will add a horizontal scroll bar to the page which may seem a bit tacky. In this situation, if we use max-width instead, it will improve the browser's handling of small windows. For example, This element has a height of 100 pixels and a max-width of 300 pixels.

p {

  max-width: 300px;

  height: 100px;

}

# Min-width

The min-width property defines the minimum width of an element. If the content is smaller than the minimum width, the minimum width will be applied. If the content is larger than the minimum width, the min-width property has no effect. This prevents the value of the width property from becoming smaller than min-width. The possible values for this property are length units, %, initial, or inherit.

# Max-height

The max-height property defines the maximum height of an element. If the content is larger than the maximum height, it will overflow. The overflow property defines how the container will handle the overflowing content. If the content is smaller than the maximum height, the max-height property has no effect. The value of the max-height property overrides the height property. This prevents the value of the height property from becoming larger than max-height. The possible values for this property are length units, %, initial, inherit, or none. Here is an example:

p {

  width: 300px;

  max-height: 100px;

}

# Overflow

Now as I said before, the overflow property specifies what should happen if an element's box has overflowing content. When the content of an element is too big to fit in the space that is specified, the overflow property specifies whether to clip the content or to add a scroll bar. This property only works for block elements with a specified height. Here are the possible values of the overflow property:

* visible

This is the default value. With this value the overflow is not clipped and renders outside the element’s box.

* hidden

With this value the overflow is clipped and the rest of the content will be invisible. The content can be scrolled programmatically.

* clip

With this value the overflow is clipped and the rest of the content will be invisible. Scrolling is forbidden

* scroll

With this value the overflow is clipped, but a scroll bar is added to see the rest of the content.

* auto

With this value if overflow is clipped, a scroll-bar should be added to see the rest of the content.

* initial

This sets this property to its default value.

* inherit

With this value it inherits this property from its parent element.

Here is an example of using the overflow property:

p {

width: 300px;

height: 60px;

overflow: visible;

}

# Min-height

The min-height property defines the minimum height of an element. If the content is smaller than the minimum height, the minimum height will be applied. If the content is larger than the minimum height, the min-height property has no effect. This prevents the value of the height property from becoming smaller than min-height. The possible values for this property are length units, %, initial, or inherit. Here is an example:

p {

  min-height: 200px;

}