Coding HTML with NVDA -Part 21(Text Shadow and Fonts)

# 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 properties for adding shadow and customizing the font for your text. 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

I know that you are excited to get into fonts, but let’s first discuss text shadow. The text-shadow property is used to add shadow to text. Its values are arranged in this order:

h-shadow

v-shadow

blur-radius

color

This is what each value represents:

* h-shadow

This is required and is the position of the horizontal shadow.

Negative values are allowed for this value

* v-shadow

This is required and is the position of the vertical shadow.

Negative values are allowed for this value

* blur-radius

This is optional and specifies the radius of the blur

The default value is 0

* color

This is optional and is the color of the shadow.

This can be set by using the various methods that CSS provides for setting colour such as an HSL value, HEX code, RGB code, or predefined colour name.

You can find out more about those by watching my CSS colour video.

* none

This is the default value and specifies no shadow

* initial

This sets this property to its default value.

* inherit

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

So only the h-shadow and v-shadow values are mandatory for the text-shadow property

Here is a simple example, where the horizontal and vertical shadows are set to 2px:

h1 {

  text-shadow: 2px 2px;

}

Here is a complete example with all the values:

h1 {

  text-shadow: 2px 2px 5px red;

}

Note that it is also possible to add more than one shadow to the text. To do this, add a comma separated list of shadows. For example, here is a text-shadow with red and blue neon glow:

h1 {

  text-shadow: 0 0 3px #ff0000, 0 0 5px #0000ff;

}

Now let’s move on to fonts.

# Fonts

Choosing the right font for your website is critical, as it can make or break your website. If you choose the wrong font, maybe one that is too distracting or hard to read, your visitors might just close your website. If you choose the right font that is attractive and easy to read, visitors will actually be interested in what your website has to offer. The right colour and size for your font is also key. Let’s start off with the font family.

# Font family

In CSS there are five generic font families, these are:

1. Serif fonts

These have a small stroke at the edges of each letter. They create a sense of formality and elegance. Some examples include:

Times New Roman

Georgia

Garamond

1. Sans-serif fonts

These have clean lines which means no small strokes attached. They create a modern and minimalistic look. Some examples are:

Arial

Verdana

Helvetica

Note that on computer screens, sans-serif fonts are considered easier to read than serif fonts.

1. Monospace fonts

With this, all the letters have the same fixed width and they create a mechanical look. Some examples are:

Courier New

Lucida Console

Monaco

1. Cursive fonts

These imitate human handwriting. Some examples are:

Brush Script MT

Lucida Handwriting

1. Fantasy fonts

These are decorative or playful fonts. Some examples are:

Copperplate

Papyrus

All of the different font names belong to one of these generic font families. In CSS, we use the font-family property to specify the font of a text. Note that if the font name is more than one word, it must be in quotation marks, like this:

"Times New Roman".

Web safe fonts are fonts that are universally installed across all browsers and devices. However, there is always a chance that a font is not found or is not installed properly. Therefore, it is very important to always use fallback fonts. This means that you should add a list of similar backup fonts in the font-family property. If the first font does not work, the browser will try the next one, and the next one, and so on. This will ensure maximum compatibility between browsers or operating systems. Start with the font you want, and end with a generic family. This will let the browser pick a similar font in the generic family if no other fonts are available. Also note that the font names should be separated with a comma. For example:

\* {

  font-family: "Times New Roman", Times, serif;

}

The following list are the best web safe fonts for HTML and CSS:

From the sans-serif family we have:

* Arial
* Verdana
* Tahoma
* Trebuchet MS

From the Serif family we have:

* Times New Roman
* Georgia
* Garamond

From the monospace family:

* Courier New

From the Cursive family we have:

* Brush Script MT

Please note that before you publish your website, always check how your fonts appear on different browsers and devices. Below are some commonly used font fallbacks, organized by the 5 generic font families:

* Serif

"Times New Roman", Times, serif

Georgia, serif

Garamond, serif

* Sans-serif

Arial, Helvetica, sans-serif

Tahoma, Verdana, sans-serif

"Trebuchet MS", Helvetica, sans-serif

Geneva, Verdana, sans-serif

* Monospace

"Courier New", Courier, monospace

* Cursive

"Brush Script MT", cursive

* Fantasy

Copperplate, Papyrus, fantasy

Please note that you can also use google fonts. At the end of this document, , you can find a link to the google fonts and a guide on how to use them. Now let’s move on to font style.

# style

The font-style property is mostly used to specify italic text. This property has three values:

* normal

With this the text is shown normally

* italic

With this the text is shown in italics

* oblique

With this the text seems as if it is leaning.

oblique is very similar to italic, but it is less supported

here is an example:

p {

  font-style: italic;

}

Now let’s move on to the font weight

# Weight

The font-weight property sets how thick or thin characters in text should be displayed. The possible values are:

* normal

This is default and defines normal characters.

* bold

This defines thick characters

* bolder

This defines thicker characters

* lighter

This defines lighter characters

* 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000

These numbers define from thin to thick characters.

400 is the same as normal, and 700 is the same as bold

* initial

This sets this property to its default value.

* inherit

with this, it inherits this property from its parent element.

Here is an example:

p {

  font-weight: 100;

}

Now let’s move on to the font size

# Size

The font-size property is used to set the size of the text. When setting font size, always use the proper HTML tags, like <h1> to <h6> for headings and <p> for paragraphs. This is so that there will be a difference in size for each element. For example, you wouldn’t want your heading level one and your paragraph elements having the same size. The value for the font-size property can be set with both absolute and relative CSS units. You can remember that I spoke about these in my HTML part 13 video in which I covered CSS units. For an in-depth tutorial on using CSS units to set the font size, please refer to that video. You can find it easily on my website by using the search bar, and the link will aalso be provided at the end of this document. Note that if you do not specify a font size, the default size for normal text, like paragraphs, is 16px. Here is a simple example of setting the font-size with pixels:

h1 {

  font-size: 20px;

}

Now let’s move on to the shorthand property.

# shorthand

To shorten the code, it is also possible to specify all the individual font properties in one property. The font property is a shorthand property for these values in this specific order:

* font-style
* font-variant
* font-weight
* font-size/line-height
* font-family

Here are some things to note:

The font-size and font-family values are required. All the other values are optional, and will be set to their default if they are not specified.

font-variant is a font property used if you want to put the text in small caps. To do this you would set its value to small-caps. Otherwise the default value is normal.

If using line-height, it must immediately follow font-size, after a "/", like this:

16px/3

Note that the line-height property sets the space between lines. If you want to learn more about the line-height property, you can check out my last video in which I taught text transformation and line spacing (link at the end of the document)

Here is an example

p {

  font: italic small-caps bold 12px/30px Georgia, serif;

}

# links

CSS colour video:

https://youtu.be/Y5vrpJd7vk8

Google fonts

https://www.w3schools.com/css/css\_font\_google.asp

CSS units:

https://youtu.be/rTXIYL3agcY

CSS transformation and line spacing:

https://youtu.be/\_aaDRGsNF1Y