How To Create a Simple Web Page

Introduction

This guide to creating a simple web page is targeted at beginners who are interested in web development or creating their own website and want a brief introduction into some of the essential knowledge needed. In this guide, we'll introduce two of the components which make up a web page: HTML and CSS.

The goal, by the end of this guide, will be to provide the reader with a good starting point to begin their web design journey.

What this guide *won't* do is go particularly in depth into the tools introduced in this guide, as that would turn this guide into a book, nor will it teach the user how to host their web page following the end of the guide.

Requirements

Following this guide requires that the reader is technologically literate and has access to a desktop or laptop with a browser and text editor on it.

Prologue

Understanding the Internet and theWorld WideWeb will help provide context behind some of the actions we perform later in this guide.

The Internet began as ARPANET, a military funded project which aimed at connecting some universities across the U.S.
It has rapidly expanded since its inception in 1969 forming the global network connected by a backbone of physical cables that span across the whole world.

While the internet and the World Wide Web are often used interchangeably, they are two different concepts. The World WideWeb is the common language/protocol used by the interconnected computers that are part of the Internet to communicate and display web pages. It was invented by Tim Burners-Lee, who became the first web developer, along with the first browser and website.

He is also the inventor of the first component of a web page that we'll be covering: Hypertext Markup Language (HTML).

HTML

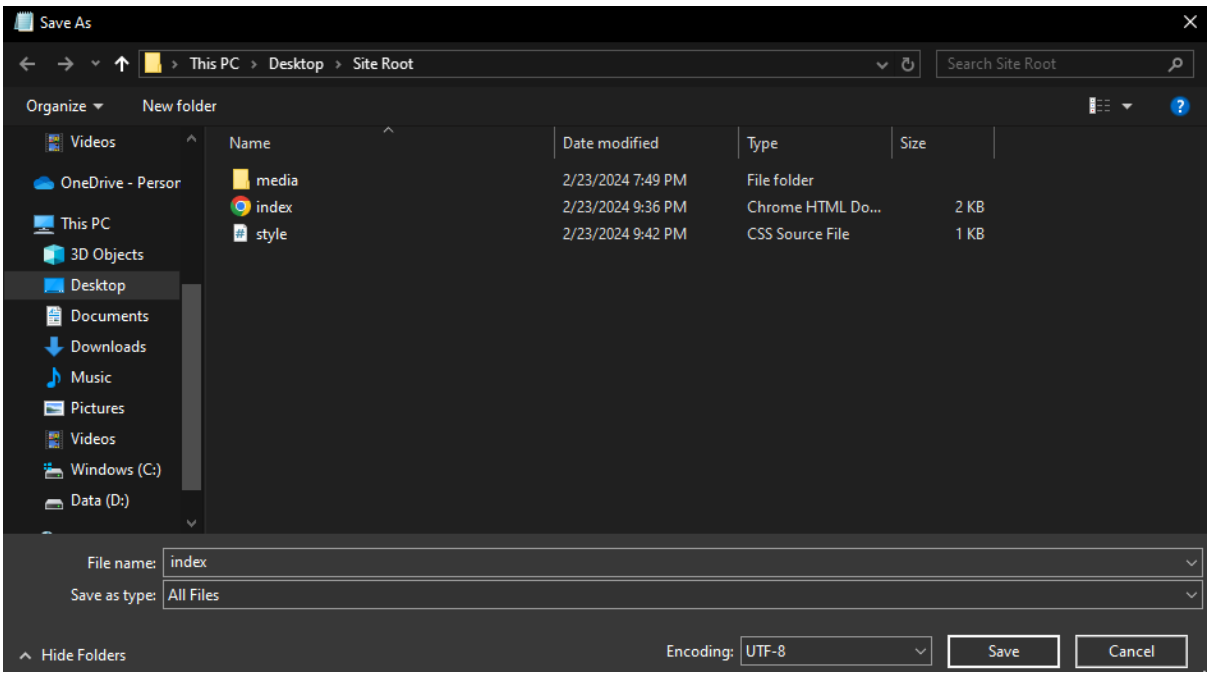
Hypertext (a.k.a. links) refers to text on a device that has a reference to a resource at another location, for example a link on Google to a Wikipedia page.A markup language is a computer language used to define a document's structure and format.

Put together, **HTML is a language we use to define a document's structure and format and to connect multiple resources on the web through hyperlinks**. To begin writing in HTML we'll need to first open our browser and text editor.

For this guide I'll be using Windows' Notepad because it's widely accessible and well-known, however people serious about web development may want to use an integrated development environment (IDE).

Step 1: Setting Up Your HTML File

1. **Start by opening your text editor and saving your file using the .html file extension.** Conventionally, the file name for the main page of a site is usually index.html, but you can give it any name.



Note: where it says, "save as type" on the bar at the bottom of the window I changed it to "all files", this is to ensure my file is saved as the right file type. Without it, it may be saved as a text document called index.html with the .txt extension.

2. **With your HTML file created, try to open it in your browser.** In my case, the default app to open HTML files is Google Chrome. If your HTML file is not opening in your browser, you may need to change the default application the file is set to open with.

The resulting web page should appear, similar to the screenshot above, empty! This is because we haven't added any content to our HTML file.

Step 2: Writing an HTML Document

There are two types of text in any markup language: text meant for the user and text meant for the computer.

To begin creating an HTML document we'll need to learn the rules for creating HTML tags, the keywords our browser uses to format and display content. An example of an HTML tag would be the image (`<img>`) tag.

Rules For HTML Tags:

- **All HTML tags must be enclosed within a set of angle brackets (<>)**.
- **For most tags, when opening a tag <> you must close it </>.** This means that tags often come in pairs. For example, the tag for a paragraph is the tag pair.
    - A closing tag always starts with a forward slash / followed by the tag name.
    - A tag without a closing tag is called a self-closing tag.

- **Anything that is placed between a pair of tags becomes its contents.** This is partially the reason behind self-closing tags, as they themselves often represent the content.

- **Tags may have attributes that modify the tag's behavior**. In the case of some tags, attributes are required. Attributes are added within the angle brackets, separated from the tag name and other attributes by a space.
    - Some attributes modify tags simply by being added, but many often require further input which is assigned to it using an equal sign (=) and quotes. For example, the <img> tag, a self-closing tag used to display an image on a web page, requires the source (src) attribute with a path to our image (<img src="image/path/here.png">).

Now we can set up our HTML document using the essential tags, which are part of every HTML document. Switch to your HTML file and follow along:

1. The first line of any HTML document will always be the doctype declaration (<!DOCTYPE html>). This line tells our browser that our file is an HTML document.

1. Then, we create a pair of <html></html> tags. This tag defines the start and end of our HTML document.

2. Between our <html> tag pair, we add in a <head> and <body> tag pair. These should be added consecutively, rather than between each other.
    - Whitespace doesn't affect how our computer reads our document, meaning we could have our entire HTML document written on one line. However, web developers will make use of new lines and indentations to organize their HTML tags and content.
    - The <head> tag pair is the container for our website's metadata, such as the page's title or links to the CSS files our page uses.
- The <body> tag pair is the container for the page's content. There should only be one <html>, <head>, and <body> tag pair in any HTML document.

From here, you've completed everything that your HTML file needs. Everything else is up to what you want to add to your HTML file; however, here are a couple of tags you may use frequently:

- The title <title></title> tag pair is used in the <head> to give your page a title, simply place the name you want between the tags like so: <title>My Page Title</title>.

- The paragraph tag pair used to create a container for a paragraph of text.

- The header <h1></h1>…<h6></h6> tag pairs are used to define a header of varying size where <h1> is the largest and <h6> is the smallest.

- The image <img> tag is used to display an image. It requires a src attribute with a link or path to the image.

- The unordered and ordered list tag pair are used to create lists, the former using bullet points and the latter using numbers. To create list items you need to nest list item tag pairs within.

- The anchor <a> tag pair is what we use to create links to other resources in our body, and a very important tag. It requires a ref attribute with a reference to the resource you want to link to.

- The link <link> tag is what we use to link to other resources in our head, and is also very important. It requires a rel attribute which describes the

relationship between the document and the resource.

- The division tag is a general tag used to divide content into sections, such as headers and footers.

- In recent years semantic HTML tags were introduced which work similarly, like <header></header>, <footer> </footer>, or <main></main>.

You can find a reference many more HTML tags here:
https://www.w3schools.com/tags

CSS

To this day the very first website ever, created by Tim Burners Lee, is hosted online (https://info.cern.ch). Here's what one of the pages looks like:

For modern standards this is simple and bland. Cascading Style Sheets (CSS), developed soon after HTML, is the second essential tool for a web developer and it aims to make websites more readable and visually appealing.

Step 1: Creating Our CSS File

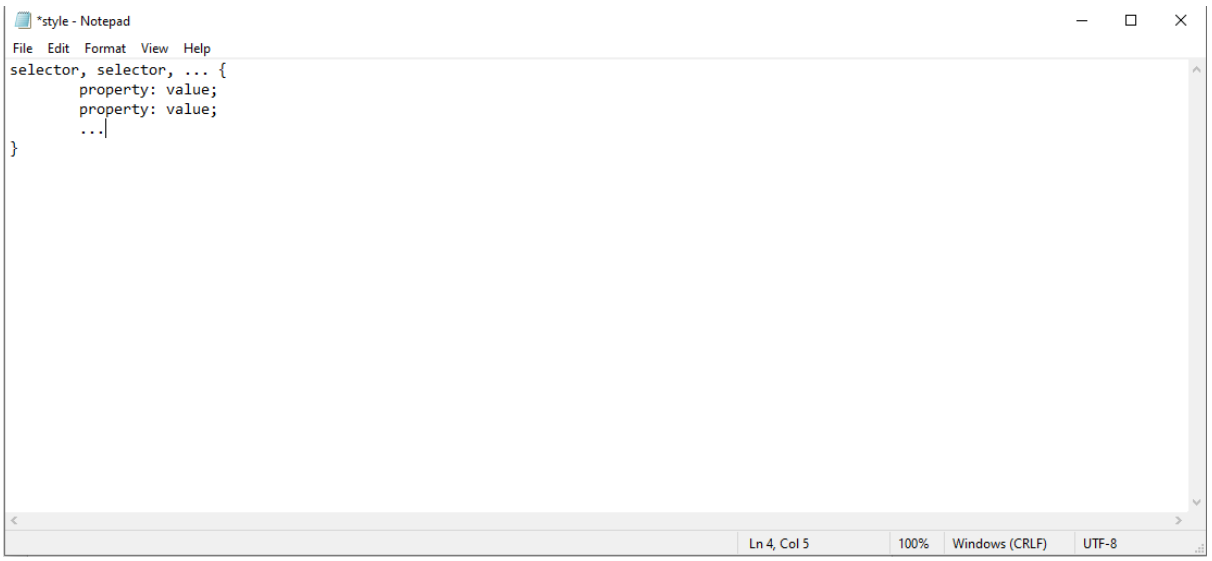
The steps are very similar to creating an HTML file:

1. In your text editor, create a new file and save it using the .css extension. Make sure to save your CSS file in the same folder as your HTML file. A common name to use is style.css but you can name it anything.

2. To be able to see what our CSS does, we'll need to link our CSS file to our HTML file. Going back to our HTML file, in the <head> section, add in the <link> tag in a format like this: <link rel="stylesheet" href="style.css">. The rel attribute defines the relationship and the href attribute specifies the location of the resource.

Step 2: Writing Our CSS Rules

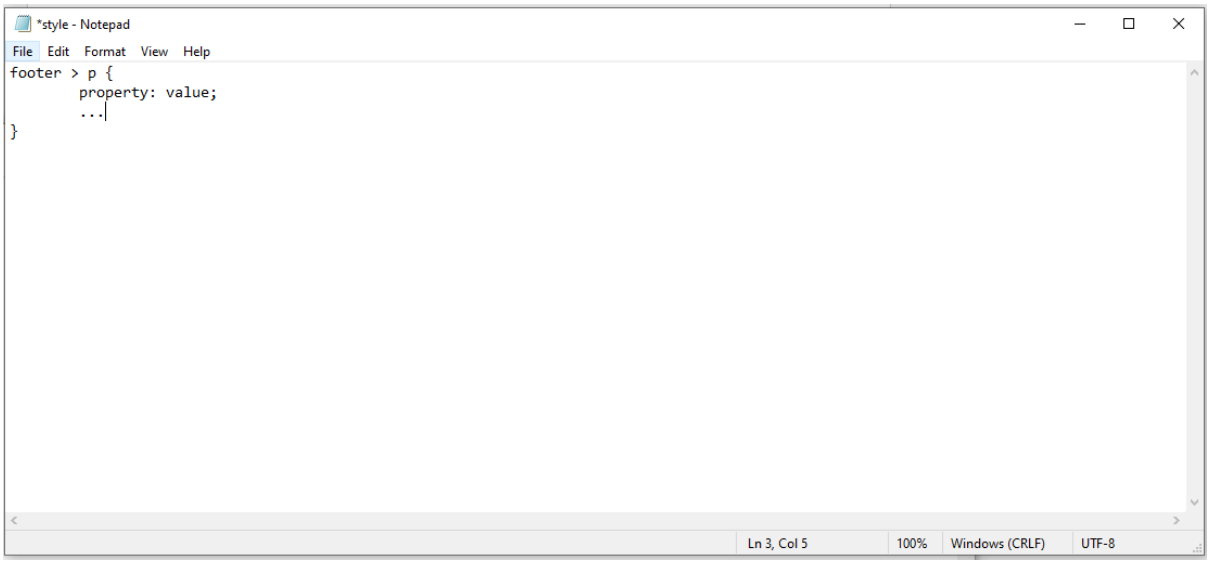
The sections of CSS which change how tags in our HTML appear are called CSS rules. CSS rules are simple, consisting of three components: the selectors, the properties, and the values.



The code block above models what a CSS rule looks like. Note that our properties are placed between a set of curly braces after our selector and that we use a semicolon at the end of the values. The curly braces signify the container for one rule, and the semicolon the computer know we are done modifying one property.

Selectors

Our selectors are generally the tags we want to modify, and we can put any number of selectors if they are separated by a comma. We can also define relationships between selectors to increase specificity using special symbols.
For example, to only select paragraphs <p> nested within our footer <footer> we would write something like this:



The more specific the target of a CSS rule, the more likely it is to take precedence over other rules.

Specificity is important in CSS. CSS files are read from top to bottom, so specificity helps us decide which rules take precedence when conflicts occur.

To learn more about CSS specificity, like the order of priority, check out this resource: https://www.w3schools.com/css/css_specificity.asp

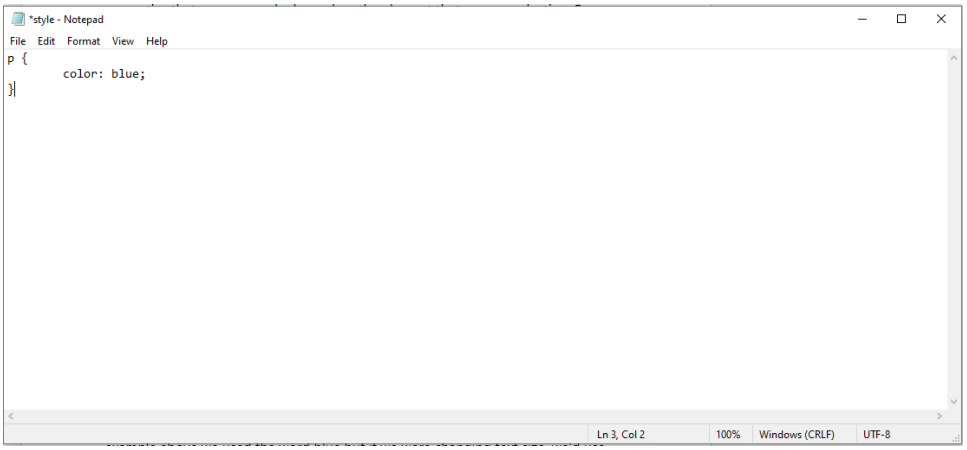
To learn all the relational symbols, you can use for selectors, check out this resource: https://www.w3schools.com/css/css_combinators.asp

---

Properties and Values

Our properties are what we want to change about the tag that we are selecting. The properties that you can apply depend on the element that you are selecting. For example, we font of an image but we can change the font of a paragraph.

The value is what we are using to modify the property, such as the font we are changing to. An example of a simple CSS rule that changes a paragraph color would appear like so:



The types of values accepted by a property depend on the property itself. In our example above we used the word blue but if we were changing text size, we'd use numbers followed by the units.
There are too many properties to cover in this guide, but here's a list of some common ones:

- The color property which allows you to change the color of text. The value can be the name of common colors like white, can be in hexadecimal like #0F0F0F, or can be RGB values like (255, 255, 255).
- The background-color property lets you change the background color of a container.
- The font-size property allows you to adjust the text's size.
- The font-family property allows you to change the font family.
- The height and width property lets you change a container's dimensions.

To find a list of CSS properties, check out this resource:
https://www.w3schools.com/cssref/index.php

Conclusion and Next Steps

With this covered you now have a basic understanding of two of the essential tools for web development. If you'd like to see what's possible using just these two tools I've uploaded a sample project repository on my GitHub (https://github.com/Bland-Life/instructions-enc) for reference. Download the files and from there you make any changes or see what I've done to get an idea of how to create a simple web page.

However, the web design journey doesn't end here. If you're serious about web development I suggest that you check out the third essential tool needed to develop a website: JavaScript. JavaScript is a programming language which makes web pages dynamic, however being a programming language it is difficult to learn and more so to master. To those intent on following this path, I wish you good luck!