Introduction to Basic Website Building using HTML and CSS

Johnson Le

1 Introduction

THIS document is intended for those interested in learning the basics of creating websites using HTML and CSS. Although there is some programming involved, no prior knowledge of programming is required. In this tutorial, the basics needed to produce a general website will be covered which includes the syntax of HTML and CSS and how they interact with each other.

2 Why Make a Website?

There are many reasons to build a website one of which includes building a personal, professional web space. This web space provides a place to document anything without restriction such as a one or two page resume. Personal websites helps recruiters find you much more easily when they go online to search for talent. Your website can be used as a personal resume website giving you a competitive edge and increasing the chance of landing a job. A website is dynamic and can be updated with any life altering moments building up one's online image for others to explore. Another reason involves business. Having a business or company website comes with many advantages and benefits such as being less expensive to advertise, increase accessibility for customers and providing access to information.

3 What is HTML and CSS?

3.1 HTML

HTML (Hypertext Markup Language) is the programming language of the web. It is not actually a programming language but a markup language. Although HTML can be used to implement style, it is mainly used to define the structure and content of a web page. HTML allows authors to:

- Publish documents online with headings, text, tables, lists, photos, etc.
- Retrieve online information via hypertext links.
- Include, video-clips, sound clips, spread-sheets, and other applications directly in their documents.

As mentioned before, HTML is actually a markup language. HTML utilizes ”tags” to define elements in the language labeling pieces of the content such as ”heading”, ”unordered list”, ”table”, and more. [1]

3.2 CSS

CSS (cascading style sheet) is a style sheet language independent from but used to compliment HTML by defining the layout and style of the structures. It can be used to define font type, font size, background color, background images, and so on. CSS also allows for presentation adaptation in regards to different devices ranging from small screens, large screens, or even printers. As noted before, HTML can be used to implement style, but the separation of the two makes it easier to maintain and allows for the same CSS file to be used across multiple web pages. [1]

4 Designing Your First Web Page

Before diving into coding part of website building, a design should be thought up and roughly sketched. This provides the author with a general approach and something to reference when coding.

4.1 Drawing A Web Page

When drawing out the web page, use shapes to form where content goes. Take a look at the example drawing in Fig.1. This step may seem simple, but I believe it is one of the most important steps to be successful and doesn't take too much time.

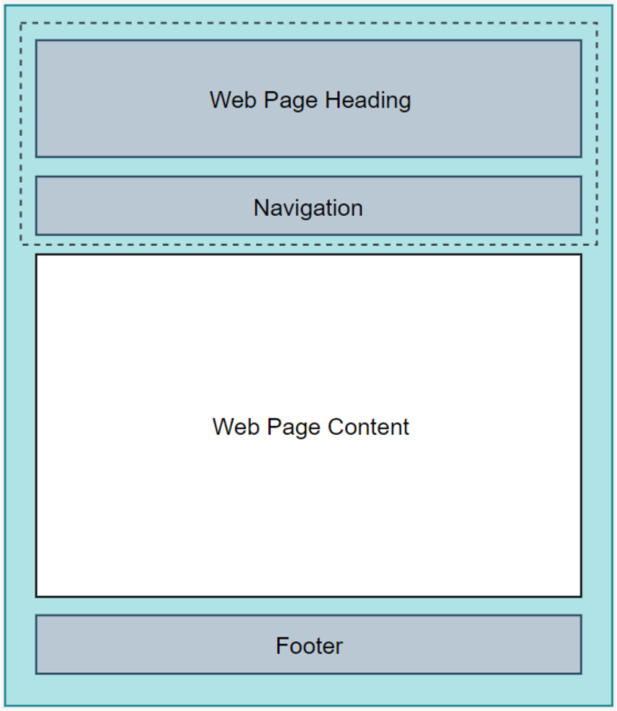


Fig. 1. Example Web Page Layout.

4.2 Sectioning

Sections are differentiated by all the rectangles. To allow for easier identification when diving into code, give simple section names for each rectangle. Sections can exist within each other such as the #top section. This is useful for organizational purposes. The # symbol represents ”id” so the dotted rectangle is a section with the id of top.

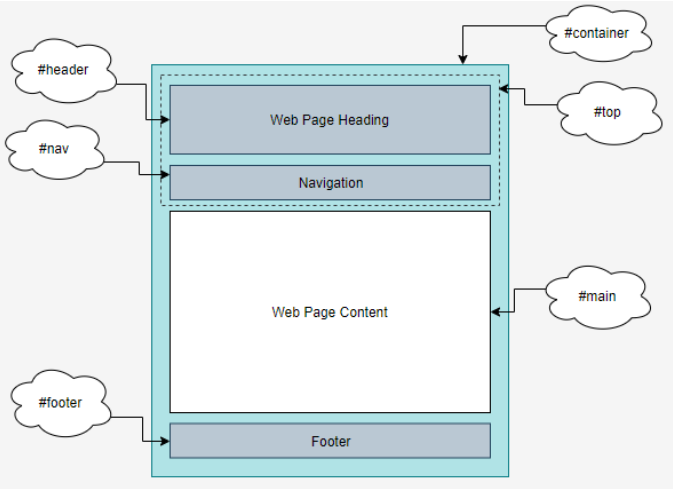


Fig. 2. Labeling Each Section

5 Creating a HTML file

Now that we have drawn out the desired web page, we can now convert it into code. Although you can do all your coding in any text editor such as Notepad for windows or TextEdit for mac, I would like to provide some of my personal recommended software to code on.

- Atom: My personal favorite. It is free and lightweight. It has many available themes and tools/packages such as ”Atom HTML Preview” which displays the website as you code.
- Notepad++: A favorite among many. It is free but much more robust than Notepad.
- Eclipse: A more complicated IDE (integraded development environment) with many plugins. Also free.

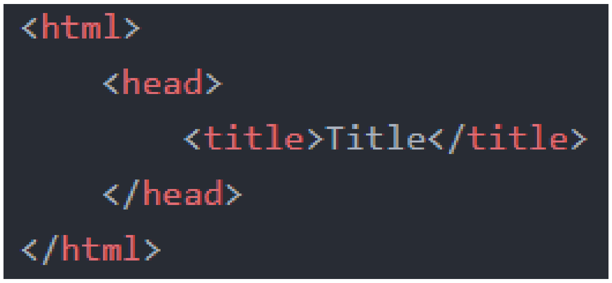
This tutorial will not cover how to obtain and setup these text editors. They are not necessary for this tutorial but can prove to be useful. [2]

5.1 The First File!

It is very important to keep organized so creating folders to house all the necessary files will prove useful when things begin to get complex. Create a folder in a convenient location such as your Desktop. Within the folder, create a new file called ”index.html”. Right click the file and click ”edit” or ”open with” to select the desired text editor. Generally, every HTML based website will have an index.html which indicates the home page that visitors first see. To open the website locally, just navigate to the file and open it.

6 The HTML and Head Tag

A typical HTML file should always begin with the ⟨html⟩ tag and is the container for all other HTML elements. The ⟨html⟩ tag tells the browser that this is an HTML document. The first two elements to create are the ⟨head⟩ and ⟨title⟩ tag. ⟨head⟩ tag houses the content that will be displayed at the tabbed area of the website. The ⟨title⟩ provides the tab with some text. When done correctly, it should look like the following:



```
<html>
    <head>
        <title>Title</title>
    </head>
</html>
```

Fig. 3. First Couple Tags

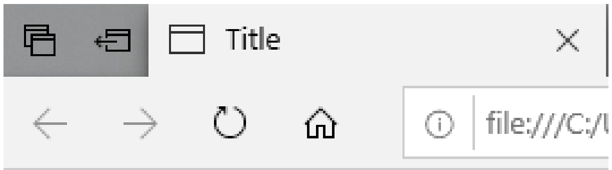


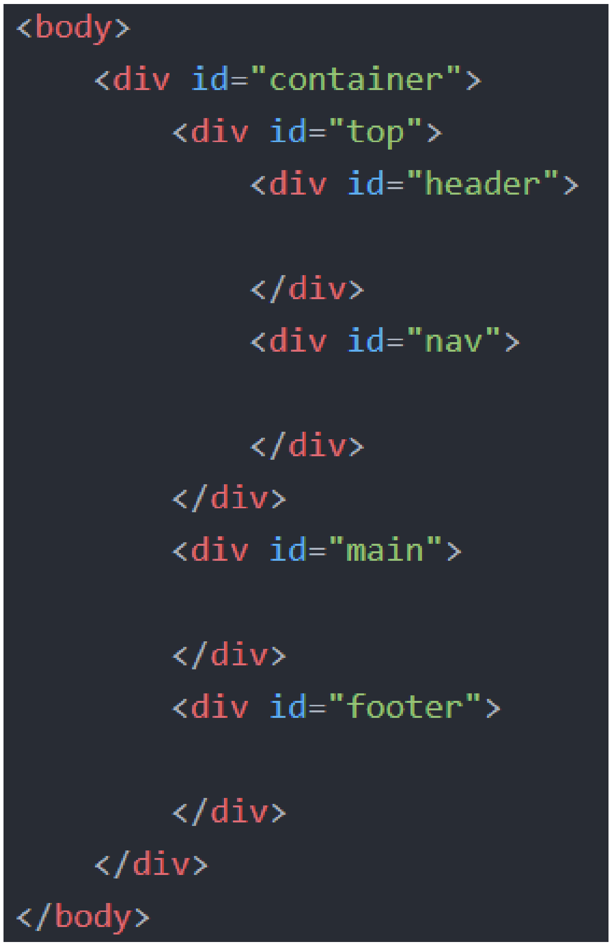
Fig. 4. HTML Head Output

Notice how every tag has a closing tag. This is very important. Each tag represents an element and if there are elements inside each other, they are implemented by encompassing one another between the opening and closing tags.

7 The Main Content

7.1 Body Tag and Sections

The ⟨body⟩ tag defines the main content seen on the HTML document. Within this tag, sections will be divided up organized into ⟨div⟩ tags. The ⟨div⟩ tag defines a section in the web page such as the ones described in Fig.2. Although it can be done without it, sections are vital to keeping neat and organized code. It will also allow for easier styling as each section can be targeted individually. Taking a look at Fig.2 again, the following is what the ⟨body⟩ tag will encompass:



```
<body>
    <div id="container">
        <div id="top">
            <div id="header">

            </div>
            <div id="nav">

            </div>
        </div>
        <div id="main">

        </div>
        <div id="footer">

        </div>
    </div>
</body>
```

Fig. 5. Body Tag Example Structure

Take time to study this structure and compare it with Fig.2. Notice that some sections are part of other section as seen in the ⟨div id=”container¨⟩.

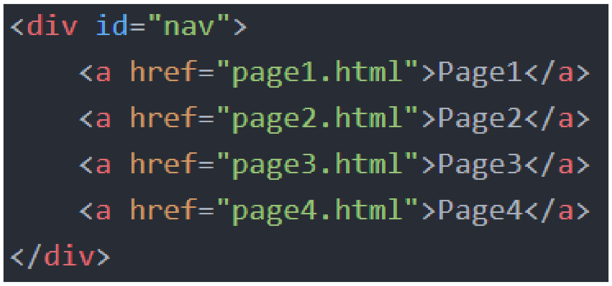
7.2 Header Tag Explained

Now that a structure is created, content can be implemented. At the top of the pre-drawn web page shown on Fig.2, the header section will house a web page heading. To accomplish this, ⟨header⟩ will be used. The ⟨header⟩ element represents a set of pre-defined elements. These range from ⟨h1⟩ to ⟨h6⟩ where ⟨h1⟩ represents the most important heading and ⟨h6⟩ represents the least important.

For the ⟨header⟩ tag we will want to use ⟨h1⟩ to represent the web-page title. The code will be
⟨h1⟩Your Website Title⟨/h1⟩
This should be sandwiched in the ⟨header⟩ section. For this tutorial, that is all that will be done in HTML for the header section.

7.3 Navigation

Navigation is important to link multiple web pages to each other. Some important web pages that could be included are a home, about, and contact page. There are many others but these are the main three seen in most websites. Before diving into the required tags to do this, lets look at how the code should look.



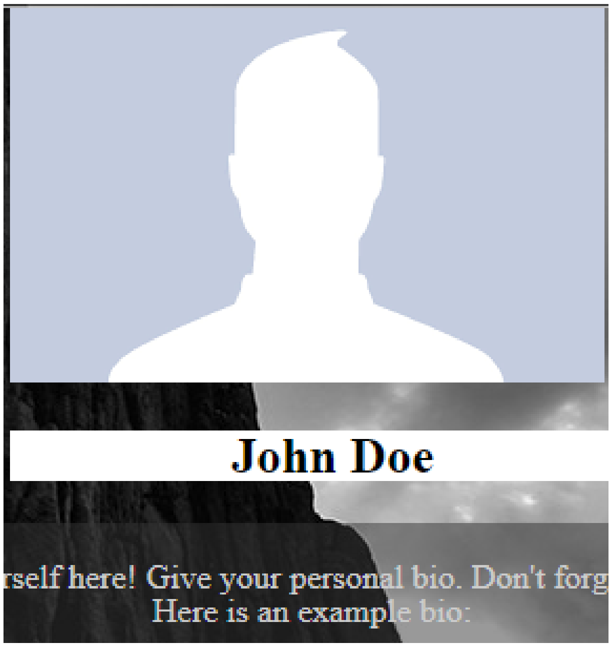
```
<div id="nav">
    <a href="page1.html">Page1</a>
    <a href="page2.html">Page2</a>
    <a href="page3.html">Page3</a>
    <a href="page4.html">Page4</a>
</div>
```

Fig. 6. Navigation Code Example

⟨a⟩ tag defines a hyperlink that links the current page to another. href is an attribute of the ⟨a⟩ tag and stands for hypertext reference. The href attribute specifies the relation between the element and another file. Looking at Fig.6, the first navigation item is linked to ”page1.html” and is labeled as ”Page1.” When activated, the website will load the Page1 web page which is done by opening ”page1.html” file. Add as many web pages as wanted.

8 Web Page Content

This section can include anything. For the purpose of this tutorial, I will be utilizing this section for a personal bio example to introduce visitors to the website. Every personal bio should have a professional picture and some block of text to introduce oneself. The goal here is to create something like the following:



John Doe

rself here! Give your personal bio. Don't forg
Here is an example bio:

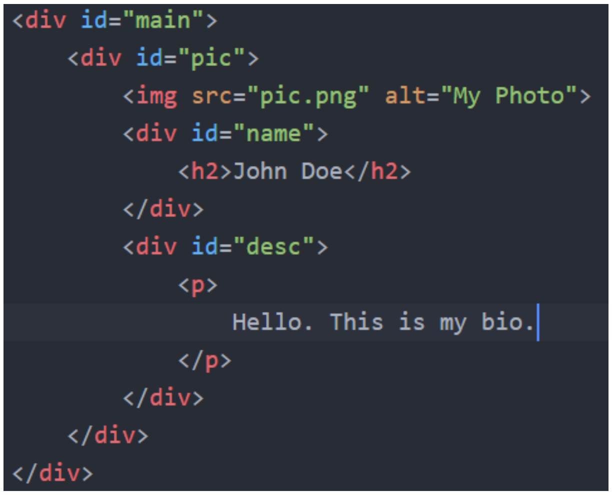
Fig. 7. Personal Bio Example

As you may have noticed, Fig.7 example has three sections; profile picture, name, and some text. Just like we have done before to create sections, the same should be done here under inside the main section.

8.1 Personal Bio

Every personal bio should have an appropriate photo. To implement a photo into html, I would suggest placing the image within the same directory or within a sub-folder called images. The ⟨img⟩ tag is one of the few tags that does not need a closing. Another tag that will be useful in this section is the paragraph tag ⟨p⟩ where text and be inputted to be displayed on the website. The following snippet of code is what the main content section should be like.

The ⟨header⟩ tag was covered earlier. Breaking down the ⟨img⟩ tag, it is actually empty



```
<div id="main">
    <div id="pic">
        <img src="pic.png" alt="My Photo">
        <div id="name">
            <h2>John Doe</h2>
        </div>
        <div id="desc">
            <p>
                Hello. This is my bio.
            </p>
        </div>
    </div>
</div>
```

Fig. 8. Code for Main Section

and only contains attributes. The two main attributes are src and alt. The src attribute specifies the location of the image. This can either be local or on the web. The alt attribute provides an alternate text for an image that will display if the image cannot be processed. [3]

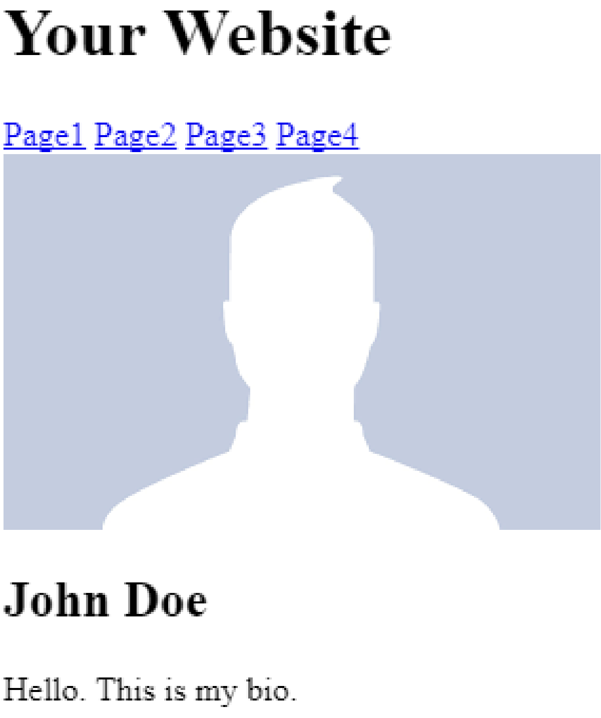
9 Halfway There

The HTML structure for the web page is done and final result of just the HTML file looks like what is shown in Fig.9.

So far, it looks pretty plain. It has good structure but not much flare. This is where CSS comes into play allowing for complete customization.

10 The CSS File

Create a new file and name it anything with a file extension of .css. I recommend placing this file in a sub directory as well such as stylesheets or anything along those lines. For this tutorial, the CSS file will be main.css. Before diving into CSS, the HTML file created earlier needs to be linked with main.css or it will not see it. To do this, use the following line:
⟨link rel=”stylesheet” type=”text/css” href =”file location”⟩
This should be placed somewhere near the top of the HTML file usually right under the ⟨html⟩ tag or even in the ⟨head⟩ tag. The rel attribute indicates what kind of relationship the link will be. The type attribute in this case is a CSS file. Again, href attribute indicates where a file is.



Your Website

Page1 Page2 Page3 Page4

John Doe

Hello. This is my bio.

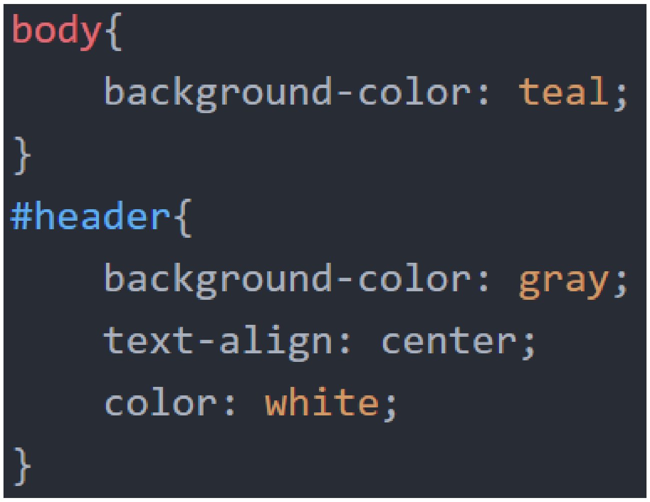
Fig. 9. HTML Result

10.1 Targeting Tags and Syntax

After linking the CSS file to the HTML, we can begin customizing the look of the web page. To target a section to be modified, simply do:
#sectionName{}
All attributes for customizing will go inbetween the curly braces. Looking back at Fig.1 for example, the web page has a teal background. To add a background of a simple color, use
background-color: ”color”;
Don’t forget the semi-colon! In CSS, semi-colons indicate that an argument has ended. The # symbol indicates ”id” and sectionName will be the area to be targeted. Non-sections such as ⟨head⟩ and ⟨body⟩ tags can be targeted as well. This is done similiarly to sections but without the # symbol.

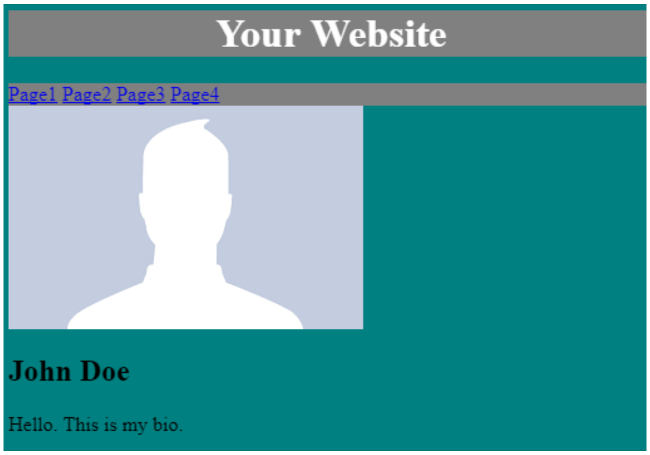
10.2 Header Target Example

In the case of Fig.1, the header wants to be gray. The text should also be aligned in the middle. Lets also change the text color and make it white. Lets take a look at how things should look so far.



```
body{
    background-color: teal;
}
#header{
    background-color: gray;
    text-align: center;
    color: white;
}
```

Fig. 10. Header and Body CSS



Hello. This is my bio.

Fig. 11. Header and Body Style

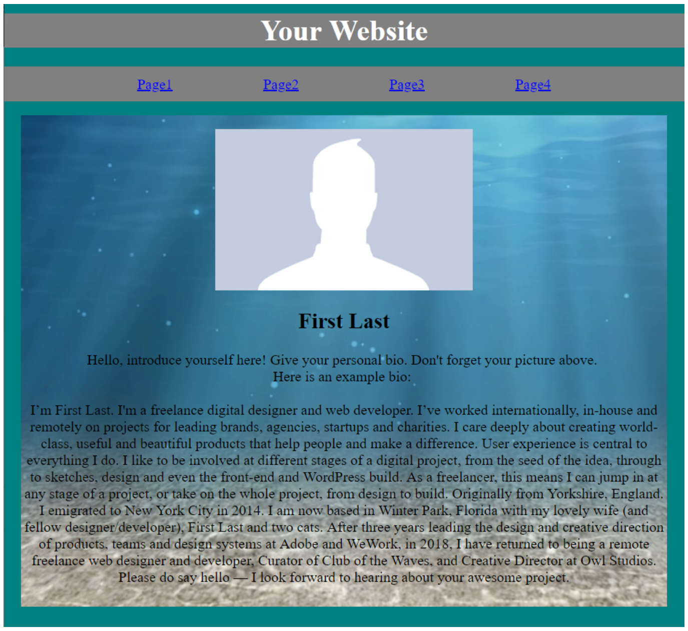
To align text, use the ”text-align: center;” Center can also be changed to left and right as well. The color attribute changes the font color.

There are a multitude of other attributes that can be used but this tutorial will only cover the basic ones. Take a look at the next section for a list of commonly used attributes.

11 Common CSS Attributes

color	sets the color of the text
text-align	aligns the text in an element
text-indent	indents the first line
font-family	defines the font in element
font-size	size of font
background-color	indicates the background color
background-image	sets the background as an image
padding	defines spaces between content

Using these common attributes, you have access to personalizing any web page. To conclude this tutorial, take a look at the example website below using all the material covered in this document.



Hello, introduce yourself here! Give your personal bio. Don't forget your picture above.
Here is an example bio:

I’m First Last. I'm a freelance digital designer and web developer. I’ve worked internationally, in-house and remotely on projects for leading brands, agencies, startups and charities. I care deeply about creating world-class, useful and beautiful products that help people and make a difference. User experience is central to everything I do. I like to be involved at different stages of a digital project, from the seed of the idea, through to sketches, design and even the front-end and WordPress build. As a freelancer, this means I can jump in at any stage of a project, or take on the whole project, from design to build. Originally from Yorkshire, England. I emigrated to New York City in 2014. I am now based in Winter Park, Florida with my lovely wife (and fellow designer/developer), First Last and two cats. After three years leading the design and creative direction of products, teams and design systems at Adobe and WeWork, in 2018, I have returned to being a remote freelance web designer and developer, Curator of Club of the Waves, and Creative Director at Owl Studios. Please do say hello — I look forward to hearing about your awesome project.

Fig. 12. Final Web Page

This is a very simple, sleek web page design with a nice background image encompassing and highlighting the main content.

12 Conclusion

The material covered in this tutorial was meant to be a comprehensive introduction to website building using HTML and CSS. Summarizing what they are, HTML is used to structure a web page and CSS is used to give style to the structure. With everything that was covered in this document, you the reader should be able to study and produce your very own website without any prior knowledge of programming.

It may take some time starting out but with practice, you'll be able to create and customize your websites efficiently and reuse old CSS files as templates for multiple web pages.

References

[1] "Html & css - w3c." https://www.w3.org/standards/webdesign/htmlcss. (Accessed on 11/24/2018).

[2] "14 best free html editors." https://blog.capterra.com/best-free-html-editors/. (Accessed on 11/24/2018).

[3] "Html images." https://www.w3schools.com/html/html$_images.asp.(Accessedon11/28/2018).$