CHAPTER ONE

EXPLORING THE WEB

Chapter Objectives

At the end of the Chapter, the students should be able to;

- **identify the constituents of a web.**
- **visit some URLs.**
- **explore the web.**

Do you know that there are over 1 billion websites on the internet? These websites serve different purposes. One popular example is Google used by millions of users to source for information. Facebook is another examples that connects us with friends and family. Have you wondered how these websites work?

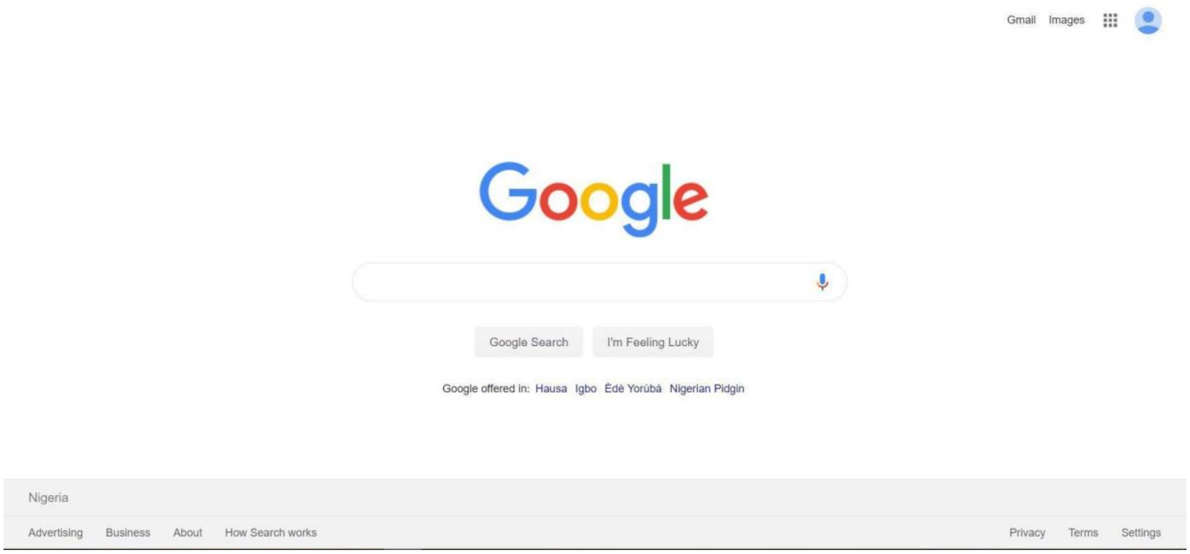


Figure 1.0: Google Search Page

How The Web Works

The web consists of four main parts communicating with one another:

- Web browser
- Web files
- Web server

- URL

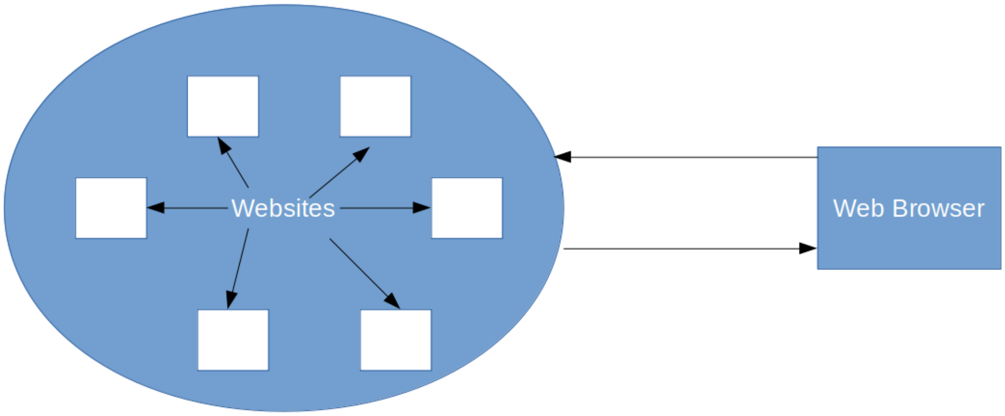


Figure 1.1: Constituents of the web.

The Web Browser

The web browser serves as the doorway to the internet, and its work is to:

- download files from the internet and render it in your browser, usually in the form of a webpage.
- take your inputs, such as “button clicks” and the texts you input into the text fields (like forms), and send them to the internet. Then, the internet can process the inputs and provide a response.

Examples of web browsers include:

- Google Chrome
- Mozilla Firefox
- Opera Mini
- UC Browser

The main function of a web browser is to display HTML (we will delve into HTML in the next Chapter). HTML is the code used to design or structure webpages. Each time a browser loads a webpage, it processes the HTML code, which may include texts, links, images and displays the result.

What is a Website?

A website is a collection of files, also called webpages, which make up the "world wide web". These webpages are stored on a **web server**. Web servers are computers that store webpages or files, and we access these web servers through the internet. To access these web files on web servers, we use a URL.

The URL

A URL, short for Uniform Resource Locator, is a unique address that is associated with websites and their pages. It enables you to access specific webpages that are stored on a web server. Without a URL, web addresses would look like this "**127.0.0.1".** How many of these numbers can you recall for all your favourite websites?

Examples of URLs are www.tqstem.org, www.ihstowers.com, www.google.com, www.facebook.com, etc. The URL of a website is usually seen in the address bar of a web browser. See figure 1.3.

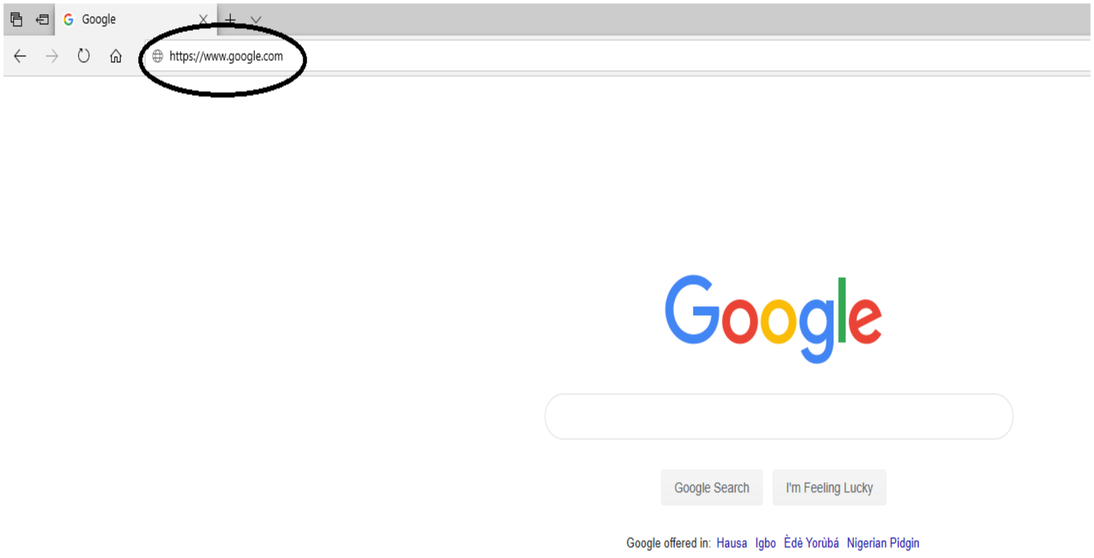


Figure 1.2: Locating a URL from a web browser

How browsers communicate with a web server?

Web browsers like Internet Explorer, Mozilla Firefox, Google Chrome are used for accessing information online. A web browser is called a client since it displays web content to the user.

A basic web browser and web server communication work like this:

- The user specifies a URL in a browser by typing it in the search bar of the browser. An example is https://www.google.com
- The browser makes a connection request to the web server through the internet.
- The web server accepts the connection request from the web browser and returns web content (webpages, audio, video or images) to the browser.
- After the server sends web content to the browser, the browser then displays all web content received.

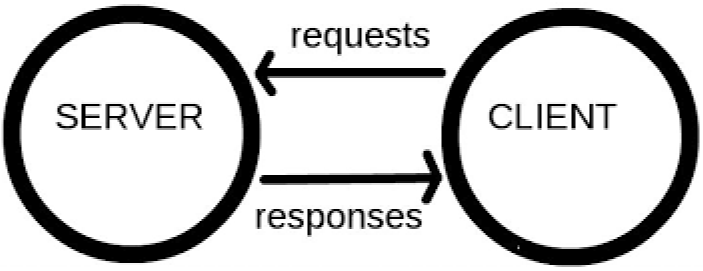


Figure 1.3

Summary of Chapter One

You have learned what a website is, and that it consists of a collection of webpages. You have also seen how a web browser and a web server communicate to display web contents to your browser. In the next Chapter, we will discuss what HTML is, and also look at the structure of an HTML code.

CHAPTER TWO

INTRODUCTION TO HTML

Chapter Objectives

At the end of this Chapter, the students should be able to:

- have an understanding of the basic HTML structure.

- identify head and body tags, and their uses.

- download and install a text editor on their computer with minimal supervision.

- create their first Hello World! page.

What is HTML?

After you have learned that a website is made up of webpages, the next natural question to ask is "*what a webpage is made up of."* HTML, short for Hypertext Markup Language, is the language used to create webpages. To understand what HTML looks like, let us find out by creating our first 'hello world' page.

Creating Your First Webpage

HTML has a layout structure. To understand this structure, let us start by creating a very basic webpage.

1. Click on Windows icon in your taskbar and search for **'notepad.'** Launch Notepad by clicking on it.

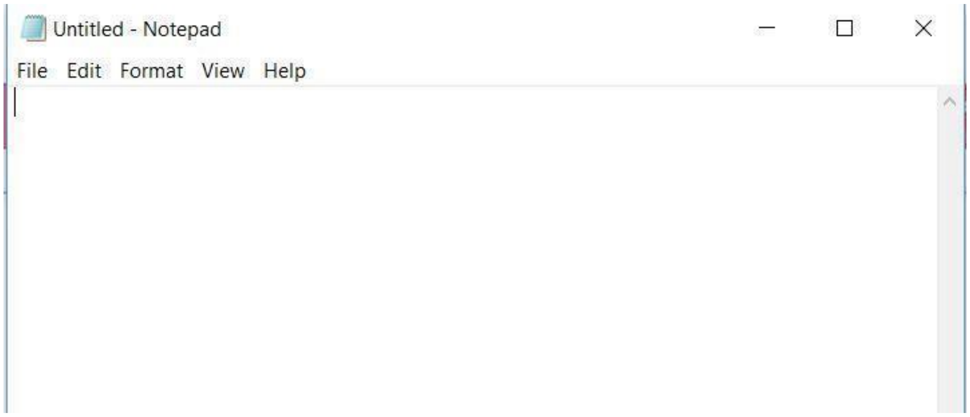
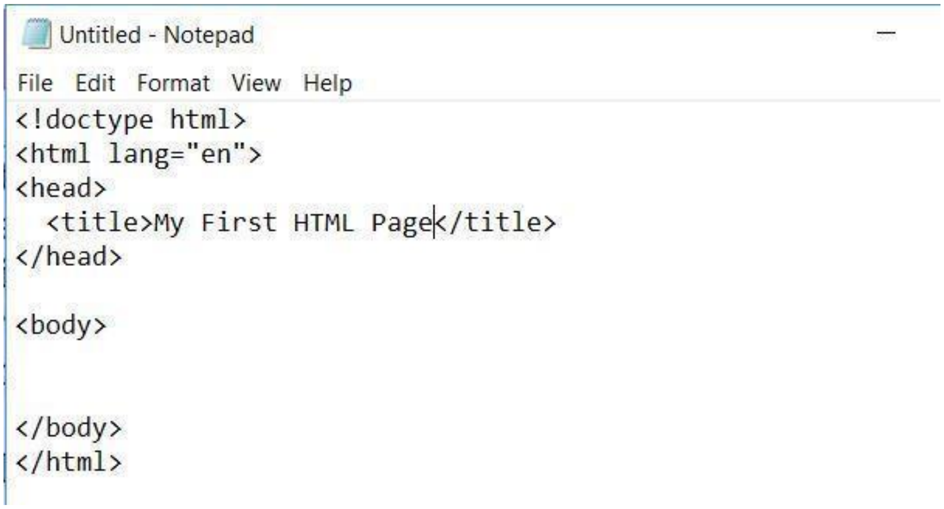


Figure 2.0: Blank Notepad

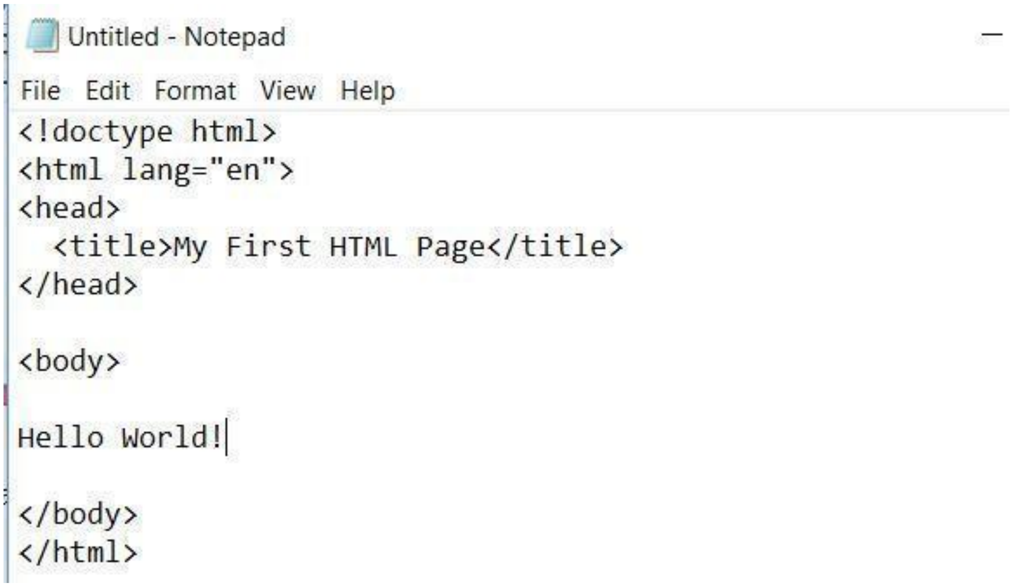
2. In the blank notepad, type the HTML template exactly as shown in figure 2.1.



```
<!doctype html>
<html lang="en">
<head>
  <title>My First HTML Page</title>
</head>

<body>


</body>
</html>
```

Figure 2.1: My First HTML Page

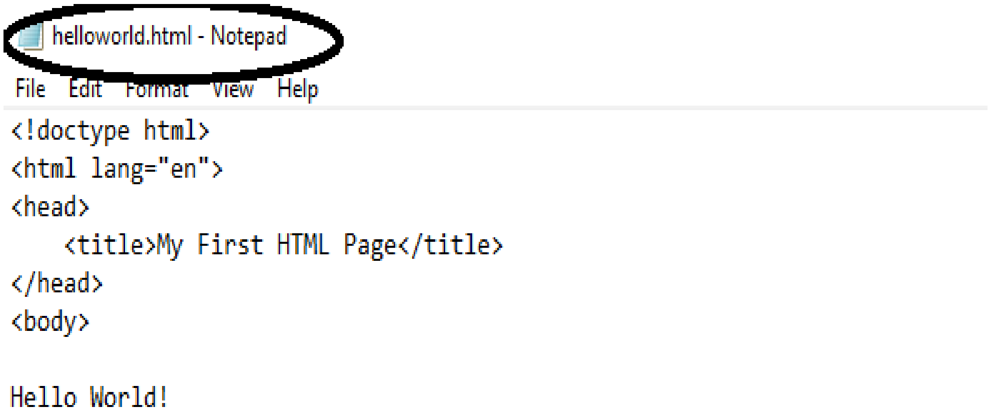
3. In the blank notepad, type your “hello world” HTML exactly as shown in figure 2.2.



```
<!doctype html>
<html lang="en">
<head>
  <title>My First HTML Page</title>
</head>

<body>

Hello World!

</body>
</html>
```

Figure 2.2 Hello World! HTML template

4. Save the notepad file as “helloworld.html” in some folder location. You should see the name of your file written at the top of your notepad as shown in figure 2.3.

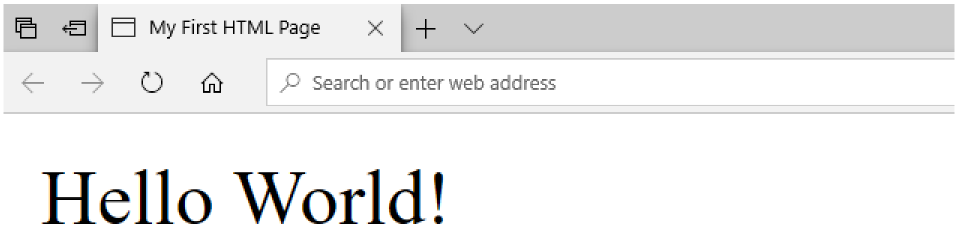


```
<!doctype html>
<html lang="en">
<head>
    <title>My First HTML Page</title>
</head>
<body>

Hello World!
```

Figure 2.3: HelloWorld HTML code

Note: All HTML files are saved with the "**.html**" file extension. A file extension is a set of characters usually attached to the end of a file name. File extensions show what kind of file you are working with for example **john.jpg** shows that you are working with an image since it ends with a **.jpg** file extension. We will discuss more on file extensions in chapter five.

5. How do we see our first HTML webpage on our web browser? First, navigate to the path where your ***helloworld.html*** is saved. Right click on the file and click open. Your browser is now showing Hello World! on the webpage as shown in figure 2.4 below.



Hello World!

Figure 2.4: hello world in the browser

Now that you have seen how to create a hello world page using notepad, let us take a more careful look at our template in figure 2.1.

Understanding the HTML Structure

Just as a human body is made up of cells, HTML is made up of tags. Each tag has a purpose in HTML on a webpage. Examples include <html>, <head>, <title>, <body>, </html>, </head>, </title>, </body>. Just as how you put words together in English to communicate a message, you place HTML tags together to get a specific output on your webpage.

Note: Most HTML tags comes in pairs; an opening tag *<tag>* and a corresponding closing tag *</tag>*. The opening tag is written inside this: **<** and this: **>** while the closing tag is written between this: **</** and this: **>**.

<!doctype html>

HTML DOCTYPE is an instruction to the web browser about the version of the markup language in which the page is written. The declaration must be the very first thing in your HTML document. **<!doctype html>** is not an HTML tag but an instruction to the web browser about what version of HTML the page is written in.

<HTML> Tag

<html> and *</html>* tags are like start and end points of the HTML file. All HTML files must start with *<html>* tag and end with *</html>* tag.

```
<html>
<head>
        <title>My First Title</title>
</head>
<body>
        Hello World!
</body>
</html>
```

<HEAD> Tag

The next important tag in HTML file is the *<head>* tag. In more advanced webpages, it contains scripts, styles and the *<title>* tag. You will learn more about the *<head>* tag as we proceed, when learning advanced styling and programming in HTML.

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

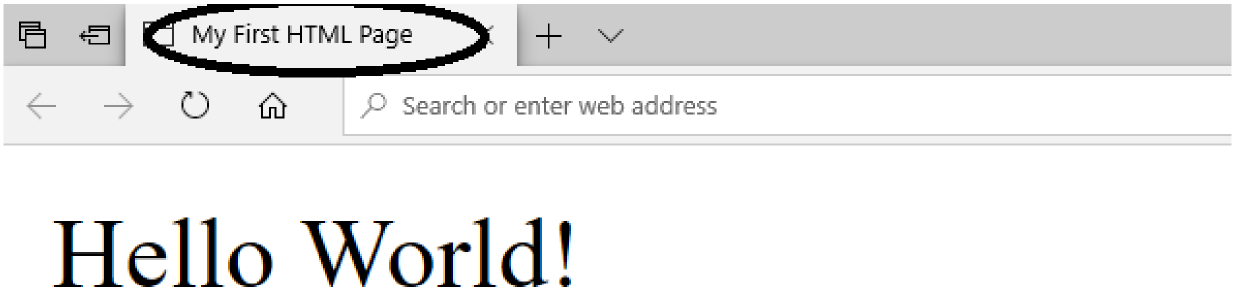
Note: The opening *<head>* tag comes immediately after *<html>* tag. The closing *</head>* tag comes before *<body>* tag. The *<head>* tag contains another tag called *<title>* tag.

<TITLE> Tag

The <title> tag helps to title your webpages. You can write some text between *<title>* and closing *</title>* tags; whatever you write between these pairs will appear in the title area your webpage. It must be written within the *<head>* tags however.

```
<html>
<head>
        <title>My First Title</title>
</head>
<body>
        Hello World!
</body>
</html>
```

Here is the result of the *<title>* tag in the browser window as shown in figure 2.10. See where "My First HTML Page" is displayed?



Hello World!

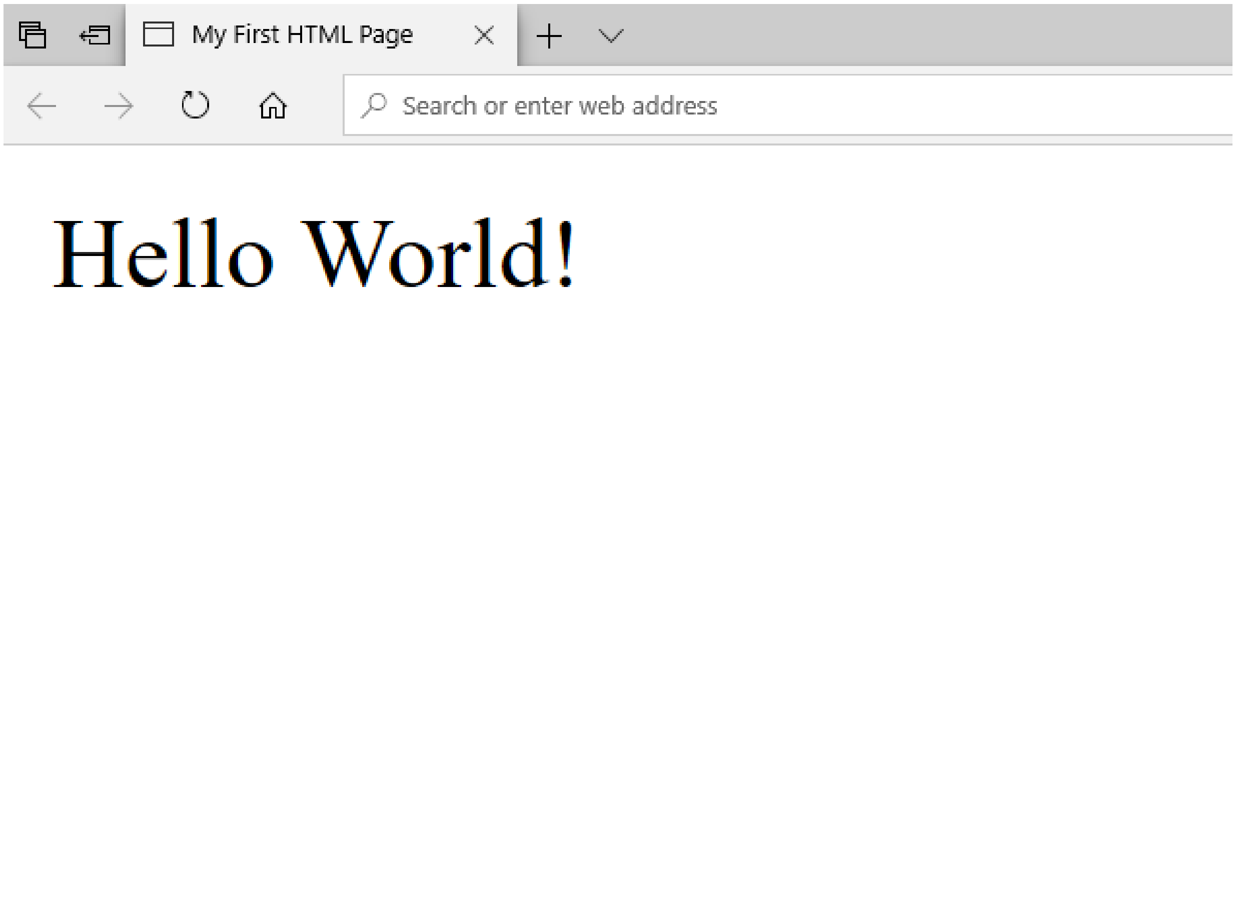
Figure 2.5: Page showing how the <title> tag displays on a browser.

Note: Note that<*title*> and <*head*> tags are optional in an HTML document. In upcoming Chapters, you will find that we have not added these two tags in many of the examples because they are not required in those examples. However, it is a good practice to include both <*title*> and <*head*> tags in your HTML.

<BODY> Tag

The most important and primary tag in HTML is the <*body*> tag. Whatever you write inside <*body*> tags; will be shown in the browser. Remember your first "Hello World!" webpage?

```
<html>

<head>
        <title>My First Title</title>

</head>

<body>

        Hello World!

</body>


</html>
```



Hello World!

Figure 2.6

Using A Text Editor

Working with Notepad can be challenging and difficult. A text editor is a software built to help programmers write better code, and make some processes easier. Most editors have intellisense, a code completion feature that speeds up the coding process by reducing typos and common mistakes.

In this book, we will use Sublime text editor because of its ease of use and simplicity. You can download the latest version by visiting https://sublimetext.com.

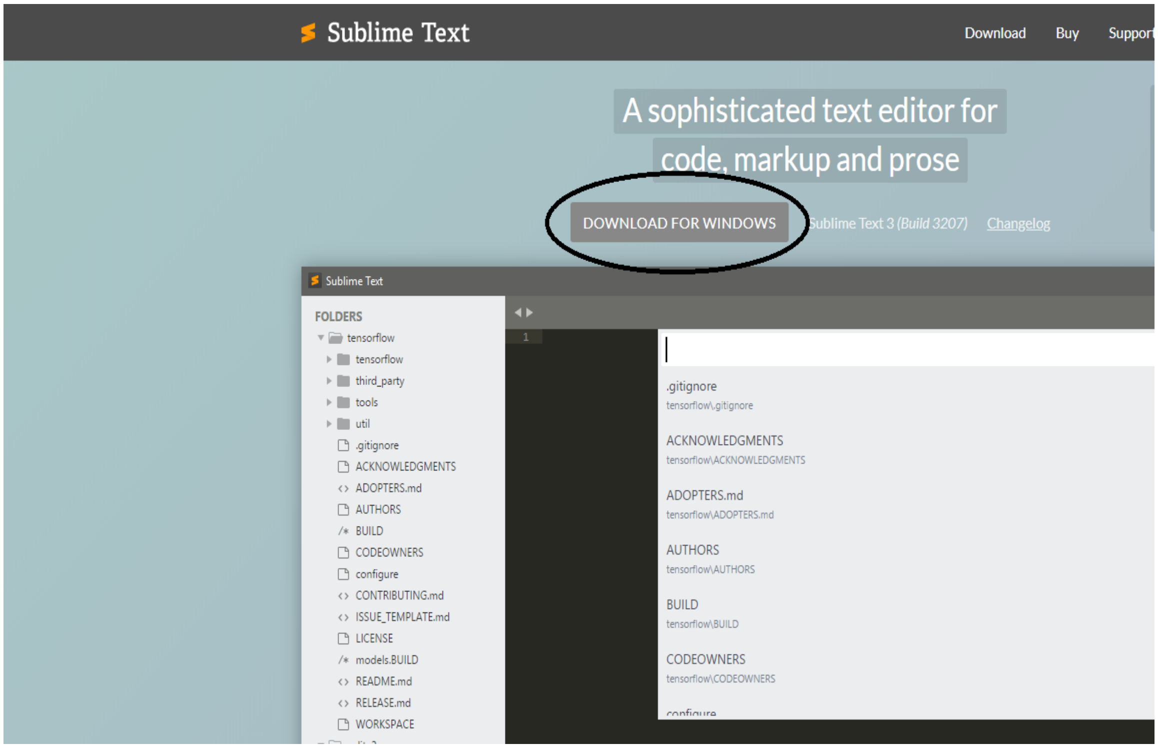


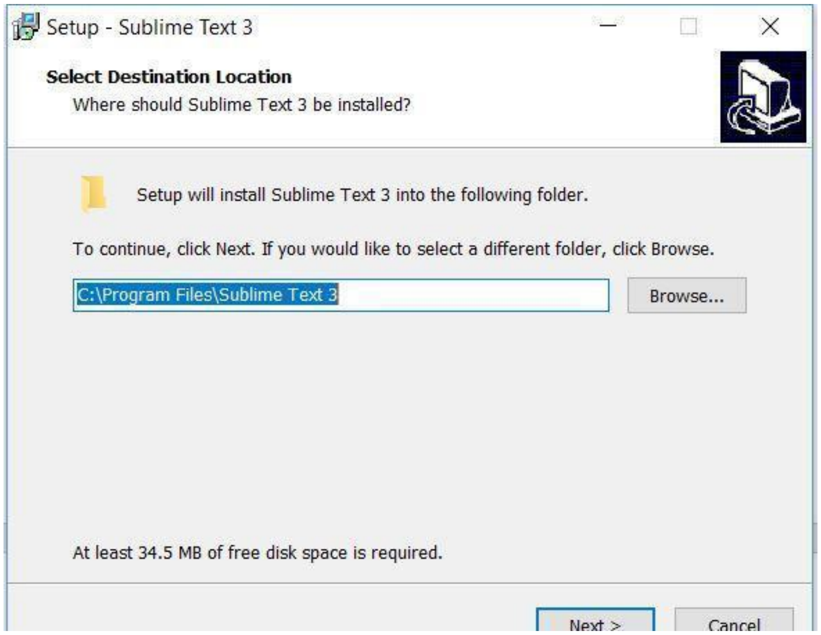
Figure 2.7

How to Install Sublime Text

The latest version of the sublime text editor as at the writing of this book is *sublime text 3 (Build 3207)*. You might notice a new release when you visit the website. Download the latest version instead.

STEPS

1. Open the application setup which you downloaded from the sublime text official site. Double-click on the downloaded setup to launch the setup wizard. Follow the installation process by clicking **‘Next’** until installation is complete.



Setup - Sublime Text 3

Select Destination Location

Where should Sublime Text 3 be installed?

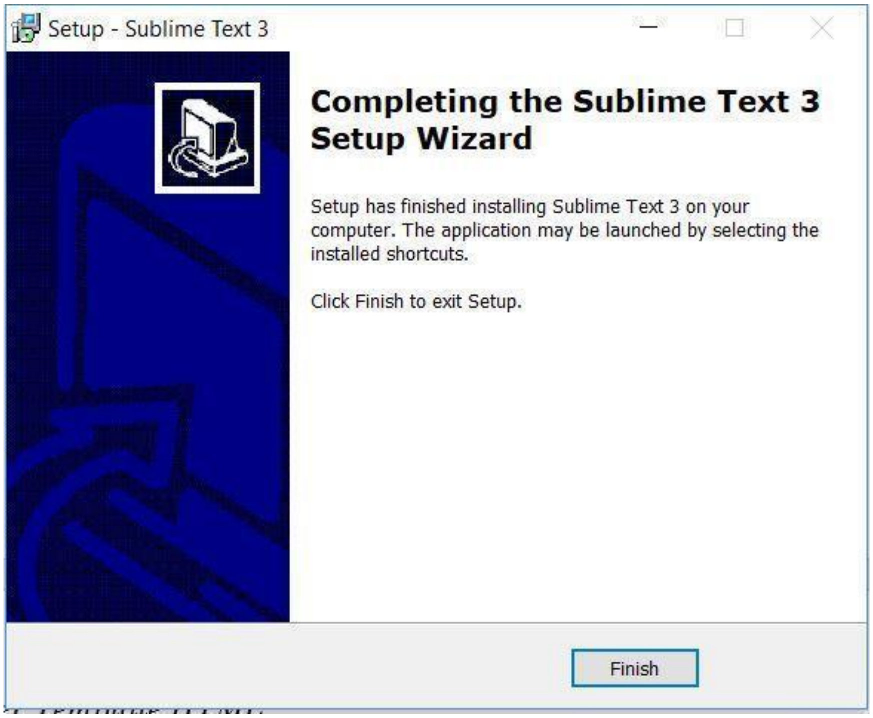
Setup will install Sublime Text 3 into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Sublime Text 3

At least 34.5 MB of free disk space is required.

Figure 2.8: Installing Sublime text



Setup - Sublime Text 3

Completing the Sublime Text 3 Setup Wizard

Setup has finished installing Sublime Text 3 on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

Figure 2.9: Sublime installation process complete

Congratulations!!! You have now successfully installed sublime text on your computer, now let us create a hello world page which will display “Hello World!” and “Hello John”.

Creating a Hello World Page using Sublime

Steps

1. First we open sublime by searching for sublime on your computer. You can press the key with the windows logo on your keyboard and then search for **sublime** and then press enter.

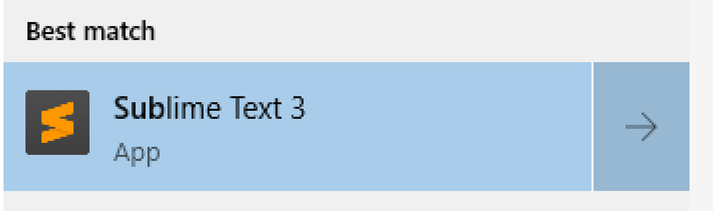


Figure 2.10 searching for sublime

2. Next we create a new file by clicking on **file** at the top left corner of your text editor and then select **new file** from the drop down menu.

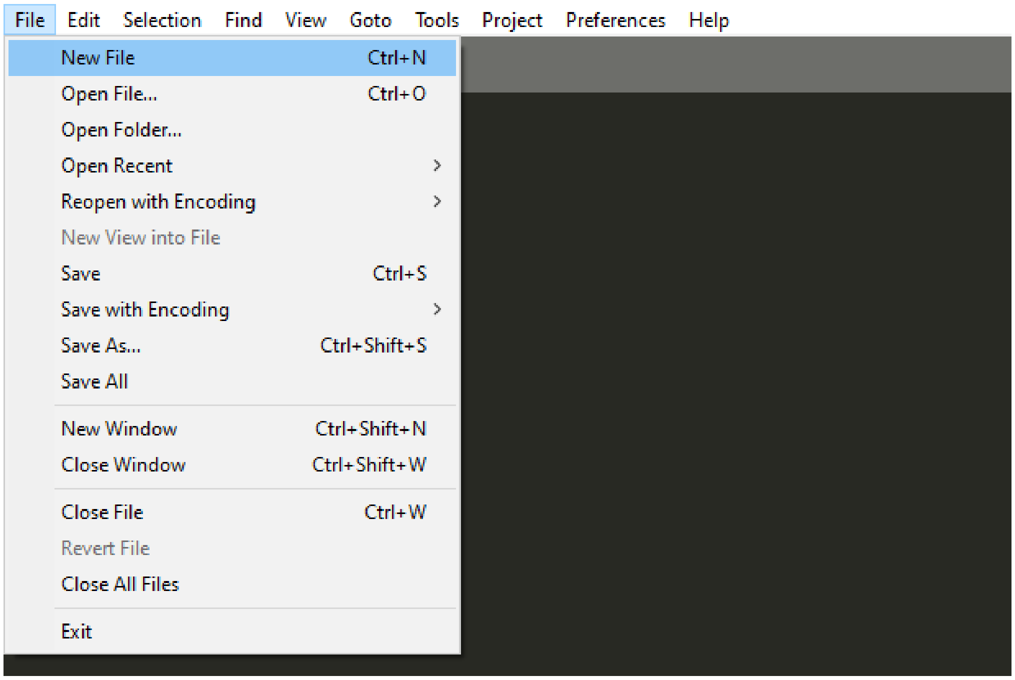
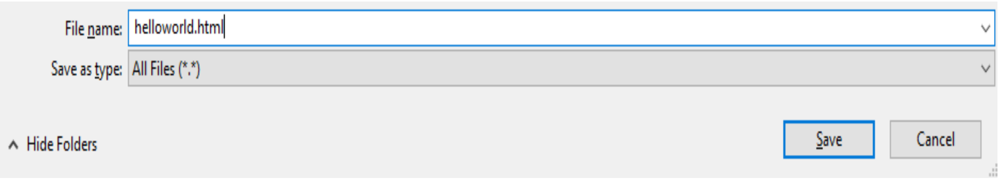


Figure 2.11 creating new file

3. Save your newly created file on your desktop and save it as **helloworld.html** as seen in figure 2.12 below.

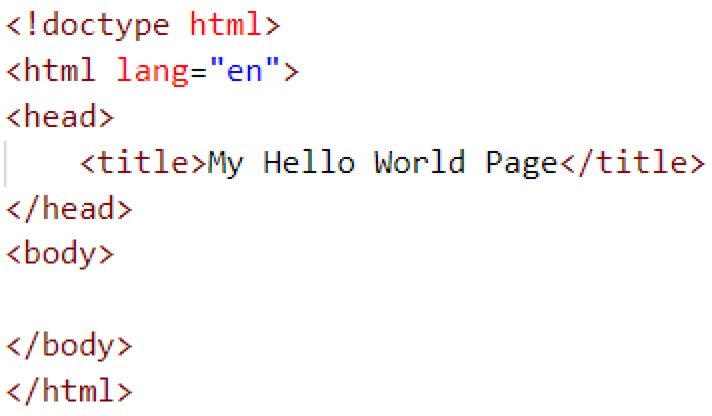


File name: helloworld.html

Save as type: All Files (*.*)

Figure 2.12 saving new file

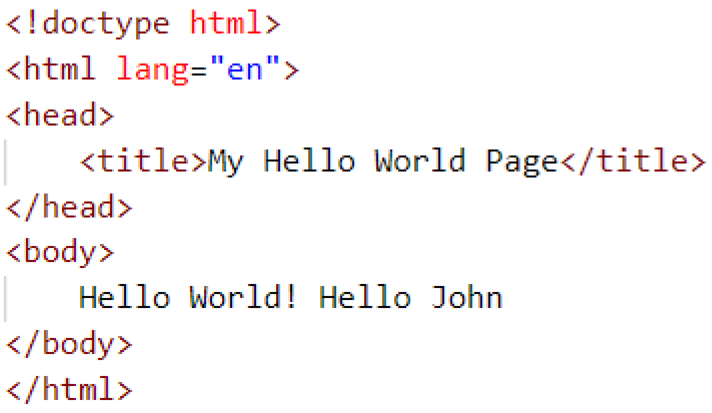
4. Next we create a basic HTML boilerplate like we did previously in figure 2.2



```
<!doctype html>
<html lang="en">
<head>
    <title>My Hello World Page</title>
</head>
<body>

</body>
</html>
```

Figure 2.13 basic boilerplate

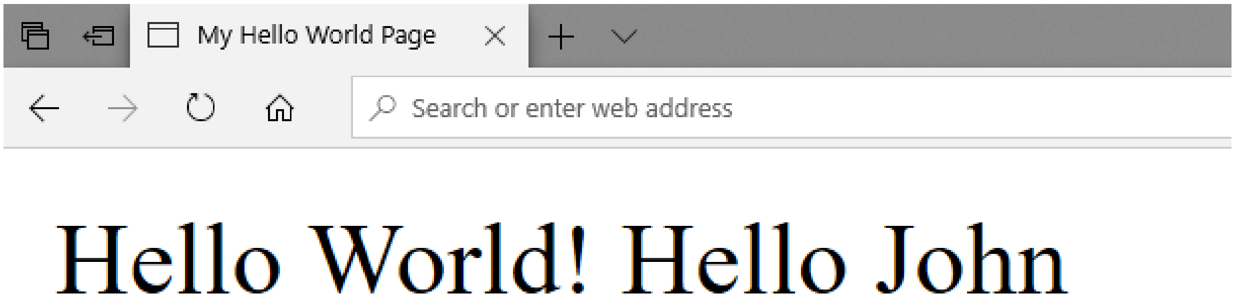
5. Let us create some content for our biolerplate by giving it a title and also adding “Hello World!” and “Hello John”.



```
<!doctype html>
<html lang="en">
<head>
    <title>My Hello World Page</title>
</head>
<body>
    Hello World! Hello John
</body>
</html>
```

Figure 2.14 basic boilerplate with content

6. Now that we are done creating our helloworld page, let us run our HTML code in our browser.



Hello World! Hello John

Figure 2.15 browser results

Summary of Chapter Two

In this chapter you learnt about the constituents of a basic HTML structure, you learnt about *<!doctype html>* as well as html and head tags and why they are useful. You also learnt how to create a hello world page using a text editor.

CHAPTER THREE

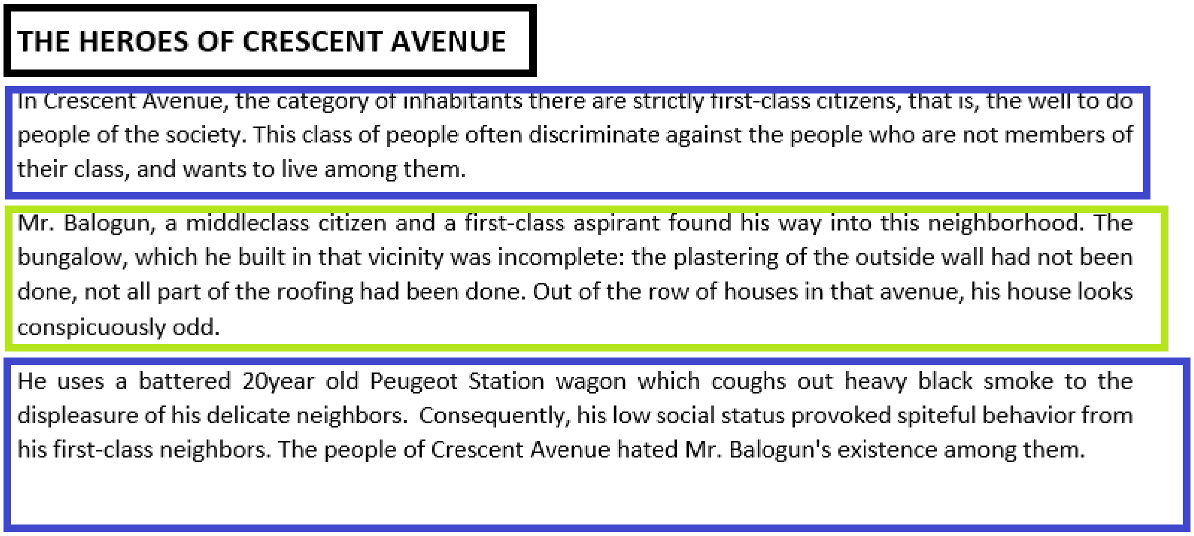
HEADINGS AND PARAGRAPHS

Chapter Objectives

At the end of this Chapter, the students should be able to:

- **correctly use HTML tags.**
- **create headers for their portfolio website.**
- **identify the different methods of formatting text using HTML tags.**
- **create the contact page for their portfolio website.**

In the previous chapter, you were introduced to the fundamentals of HTML, and now have a basic understanding of HTML tags and layouts of webpages. Now, you will learn about *Heading* and *Paragraph* tags. Just as in English, in HTML, headings are used for writing "*titles*" of a section of the webpage while paragraphs are the body.



THE HEROES OF CRESCENT AVENUE

In Crescent Avenue, the category of inhabitants there are strictly first-class citizens, that is, the well to do people of the society. This class of people often discriminate against the people who are not members of their class, and wants to live among them.

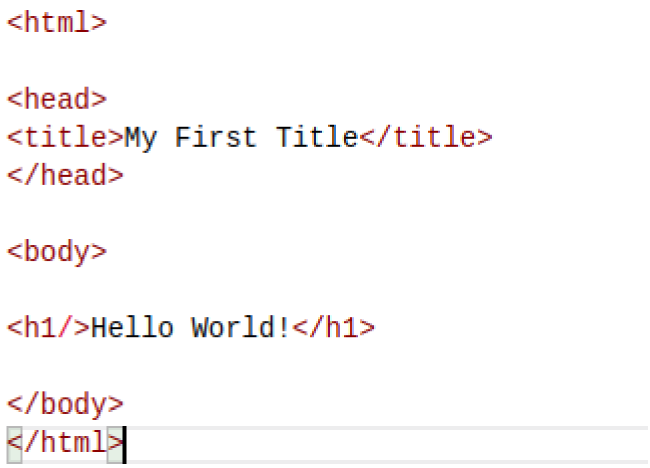
Mr. Balogun, a middleclass citizen and a first-class aspirant found his way into this neighborhood. The bungalow, which he built in that vicinity was incomplete: the plastering of the outside wall had not been done, not all part of the roofing had been done. Out of the row of houses in that avenue, his house looks conspicuously odd.

He uses a battered 20year old Peugeot Station wagon which coughs out heavy black smoke to the displeasure of his delicate neighbors. Consequently, his low social status provoked spiteful behavior from his first-class neighbors. The people of Crescent Avenue hated Mr. Balogun's existence among them.

Figure 3.0: Sample text showing a heading and a paragraph

Headings

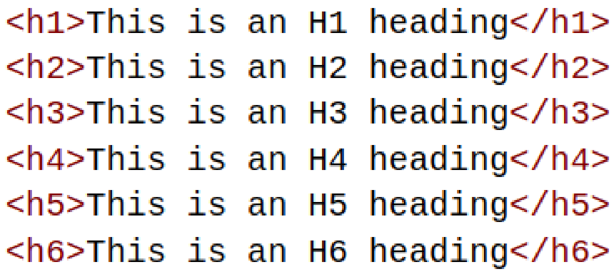
Headings vary by font size. You can write a heading using *<h1>, <h2>, <h3>,* to *<h6>* tags. <h1> is the biggest tag in this category with 32 pixels in size (we will discuss more about pixels in chapter 5) while <h6> is the smallest and has a font size of about 10.72 pixels.



```
<html>

<head>
<title>My First Title</title>
</head>

<body>

<h1/>Hello World!</h1>

</body>
</html>
```

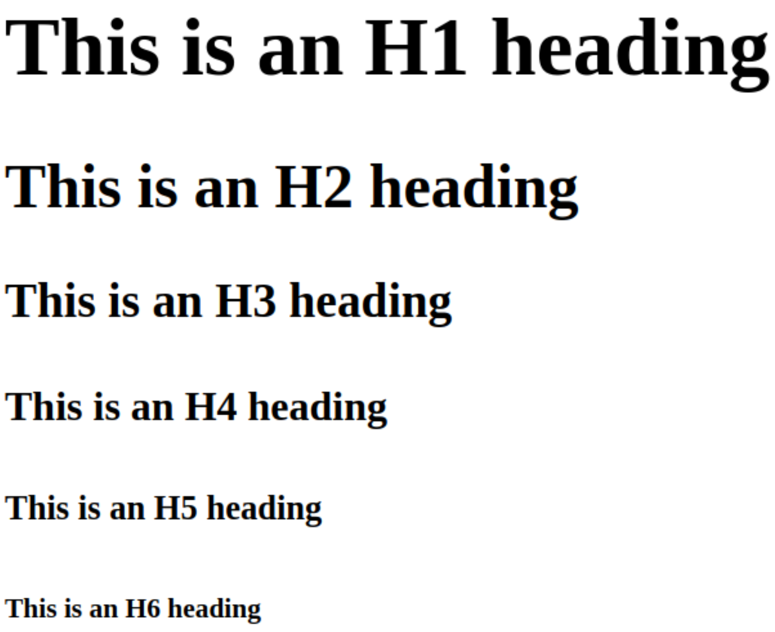
Figure 3.1: Sample code of showing "Hello World" heading with an h1 tag.

Similar to h1, you can use h2, h3, h4, h5 or h6 tags.



```
<h1>This is an H1 heading</h1>
<h2>This is an H2 heading</h2>
<h3>This is an H3 heading</h3>
<h4>This is an H4 heading</h4>
<h5>This is an H5 heading</h5>
<h6>This is an H6 heading</h6>
```

Figure 3.3. H tags in HTML



This is an H1 heading

This is an H2 heading

This is an H3 heading

This is an H4 heading

This is an H5 heading

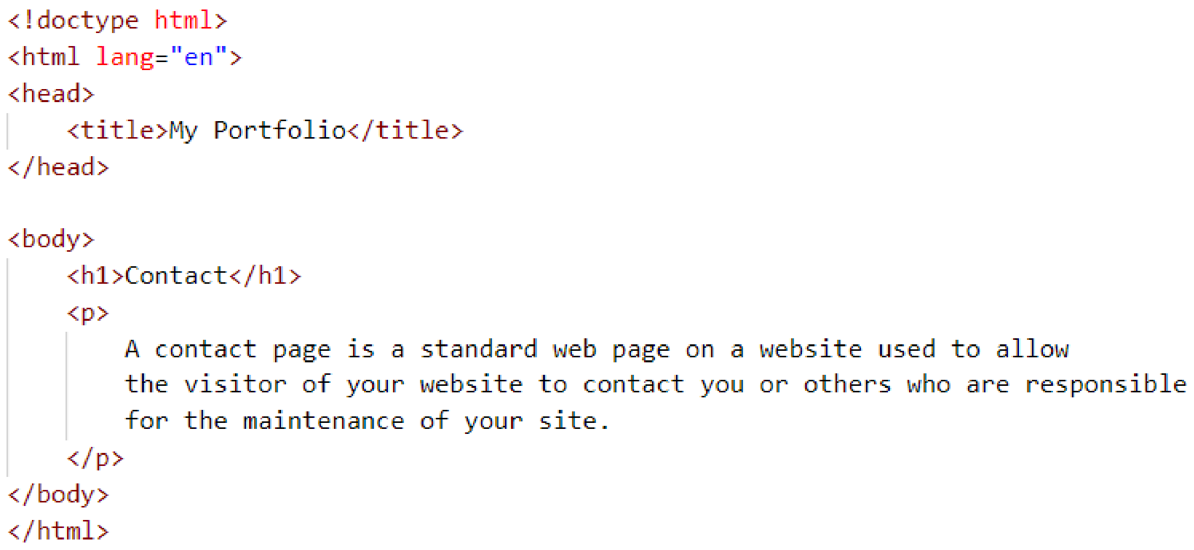
This is an H6 heading

Figure 3.4: All H tags displayed in a browser from h1 to h6

In your webpage, you can choose a specific heading number tag based on what size of the title you want to show.

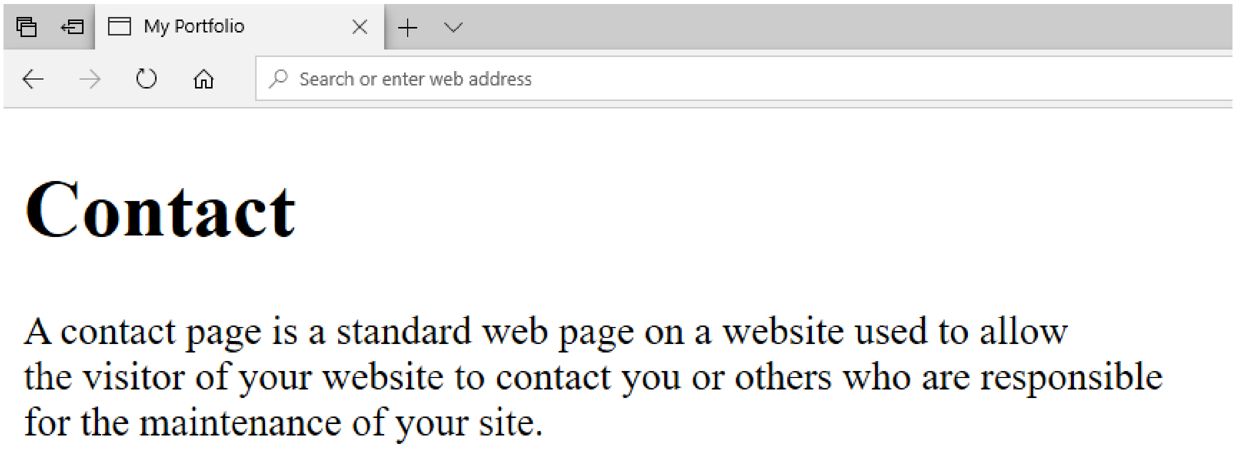
Paragraphs

Paragraphs are used to provide long blocks of text. We use the *<p>* tag to show paragraph on a webpage. See figure 3.5.



```
<!doctype html>
<html lang="en">
<head>
    <title>My Portfolio</title>
</head>

<body>
    <h1>Contact</h1>
    <p>
        A contact page is a standard web page on a website used to allow
        the visitor of your website to contact you or others who are responsible
        for the maintenance of your site.
    </p>
</body>
</html>
```

Figure 3.5 shows a paragraph tag.

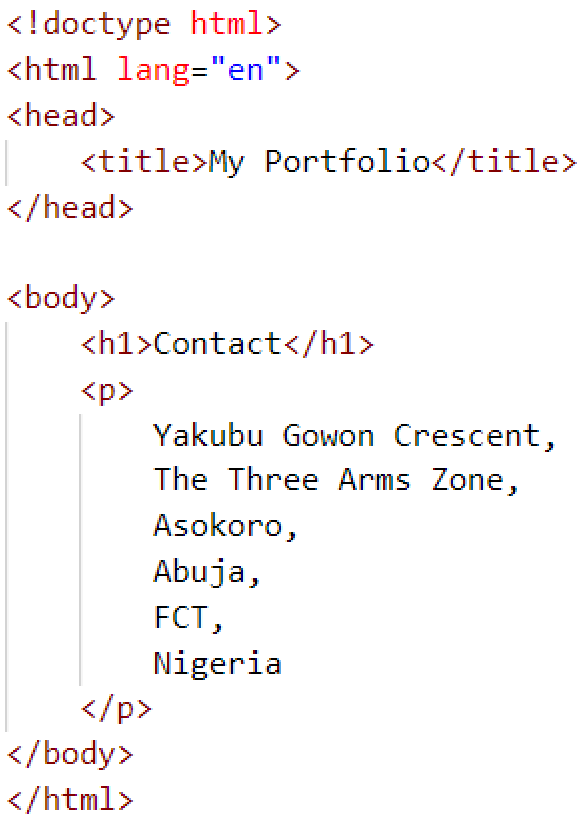


Contact

A contact page is a standard web page on a website used to allow
the visitor of your website to contact you or others who are responsible
for the maintenance of your site.

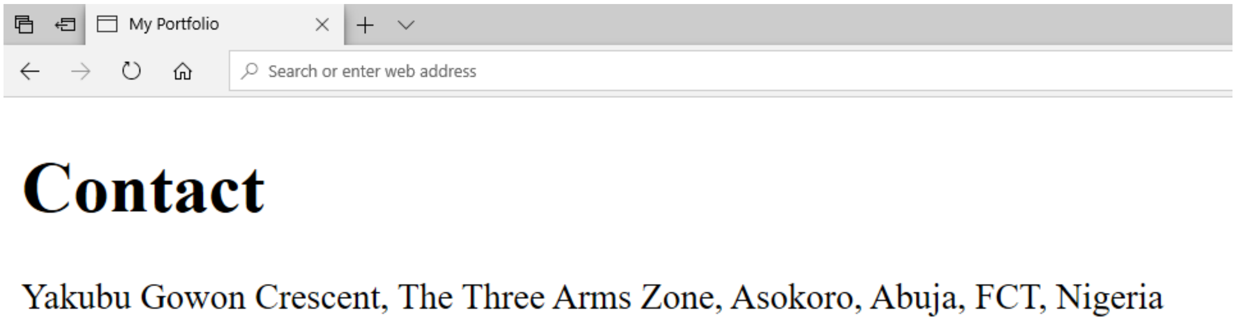
Figure 3.6 Code output in the browser.

We placed a random text between *<p>* tags in figure 3.7, now let us put John's office address between the *<p>* tags.



```
<!doctype html>
<html lang="en">
<head>
    <title>My Portfolio</title>
</head>

<body>
    <h1>Contact</h1>
    <p>
        Yakubu Gowon Crescent,
        The Three Arms Zone,
        Asokoro,
        Abuja,
        FCT,
        Nigeria
    </p>
</body>
</html>
```

Figure 3.7 John's address sample code



Contact

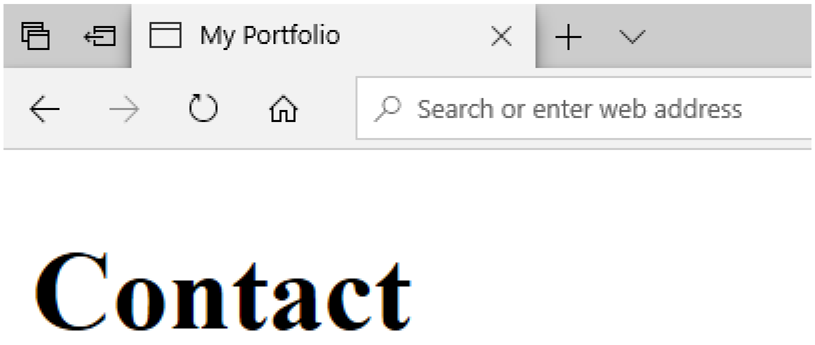
Yakubu Gowon Crescent, The Three Arms Zone, Asokoro, Abuja, FCT, Nigeria

Figure 3.8 John's address displayed in the browser.

Building Your Portfolio Website

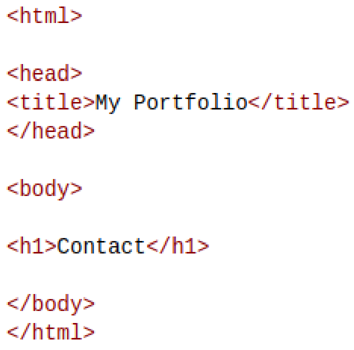
Throughout the remaining Chapters of this book, you will learn to a portfolio website. A portfolio website shows information about you, as well as the kind of work you do. For instance, if you are a web designer, your portfolio website will include personal information about yourself, your contact details, and samples of your previous projects (websites). In this book, you develop your own portfolio, following the same process we will use for our imaginary friend, John, a web developer like yourself. John's portfolio website will include a home page, a contact page, and an about page. Now, let us build a contact page for John's portfolio website.

First, we will create a "contact" title for John's contact page. Then, we will create an HTML file and name it *contact.html*. You can see the sample HTML code to build the contact page in figure 3.6 below.



Contact

Figure 3.5: Contact Title



```
<html>

<head>
<title>My Portfolio</title>
</head>

<body>

<h1>Contact</h1>

</body>
</html>
```

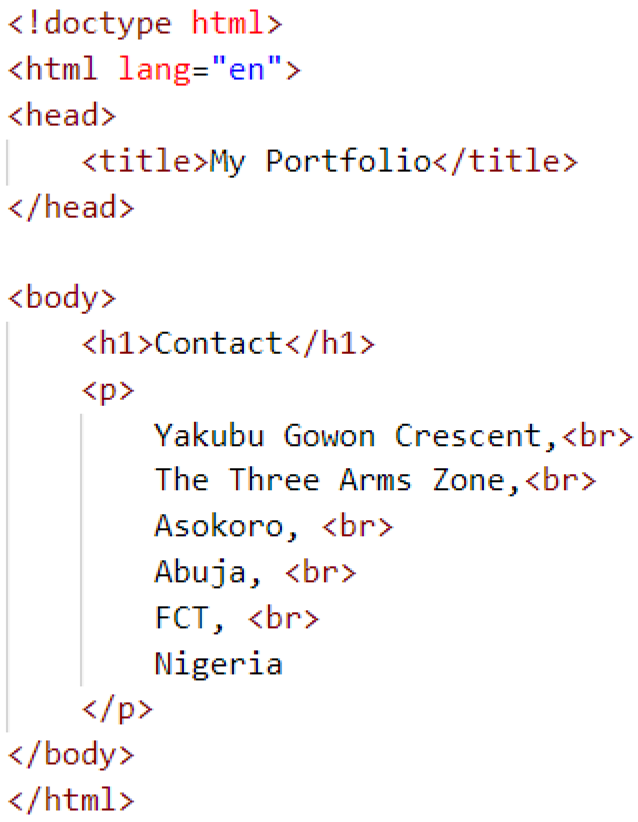
Figure 3.9: Contact Title. Sample code

You can see that the **Contact Title** is wrapped between h1 tags.

Now that you are getting how things work, let us now move on to how paragraphs are used.

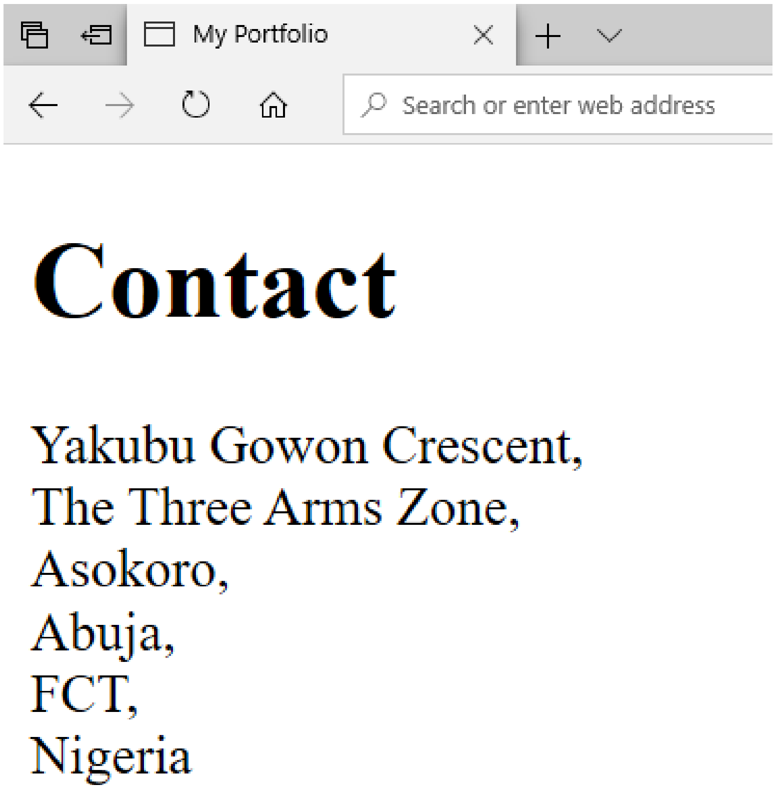
Introducing the <br> tag

The break tag, called the <br> tag, is used to move a paragraph, or a section of a paragraph to a new line. There is a better way to re-format John's address with this tag. Look at the code snippet in figure 3.11 below.



```html
<!doctype html>
<html lang="en">
<head>
    <title>My Portfolio</title>
</head>

<body>
    <h1>Contact</h1>
    <p>
        Yakubu Gowon Crescent,<br>
        The Three Arms Zone,<br>
        Asokoro, <br>
        Abuja, <br>
        FCT, <br>
        Nigeria
    </p>
</body>
</html>
```

Figure 3.10 John's address displayed in the browser. Sample code



Contact

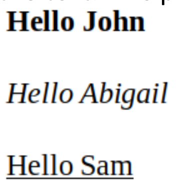
Yakubu Gowon Crescent,
The Three Arms Zone,
Asokoro,
Abuja,
FCT,
Nigeria

Figure 3.11 John’s address displayed in the browser with *<br>* tags

John’s contact page looks a lot better after adding break tags “*<br>*”

Formatting Your Webpage

When you are reading a book or browsing a webpage, you must have observed that certain parts of the text are Bold (**Bold**) or underlined (underline) or in Italic (*Italic*). This is to call attention to that part of the text. This process is called “Formatting”.



Hello John

Hello Abigail

Hello Sam

Figure 3.12 Samples of Formatted text

Formatting Tags in HTML

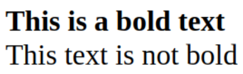
The following HTML tags help you to put specified formats in specified segments of your webpage.

- **<b>** - Bold text
- **<strong>** - Important text
- **<i>** - Italic text
- **<em>** - Emphasized text
- **<u>** - Underlined text
- **<small>** - Small text
- **<sub>** - Subscript text
- **<sup>** - Superscript text

The <b> tag

```
<b>This is a bold text</b><br>
This text is not bold
```

Figure 3.14 HTML code to make a text bold



This is a bold text
This text is not bold

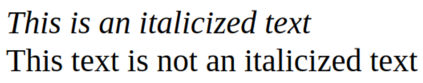
Figure 3.13 browser results

The <i> tag

This tag helps to italicize your text.

```
<i>This is an italicized text</i><br>
This text is not an italicized text
```

Figure 3.16 HTML code to italicize text



This is an italicized text
This text is not an italicized text

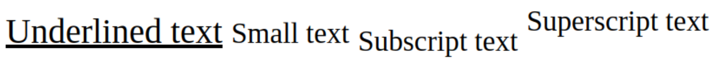
Figure 3.17 Browser results

Take a look at the list below to understand how other text formatting works:

- **<u>** - <u>Underlined text</u>
- **<small>** - <small>Small text</small>
- **<sub>** - <sub>Subscript text</sub>
- **<sup>** - <sup>Superscript text</sup>

```
<u>Underlined text</u>
<small>Small text</small>
<sub>Subscript text</sub>
<sup>Superscript text</sup>
```

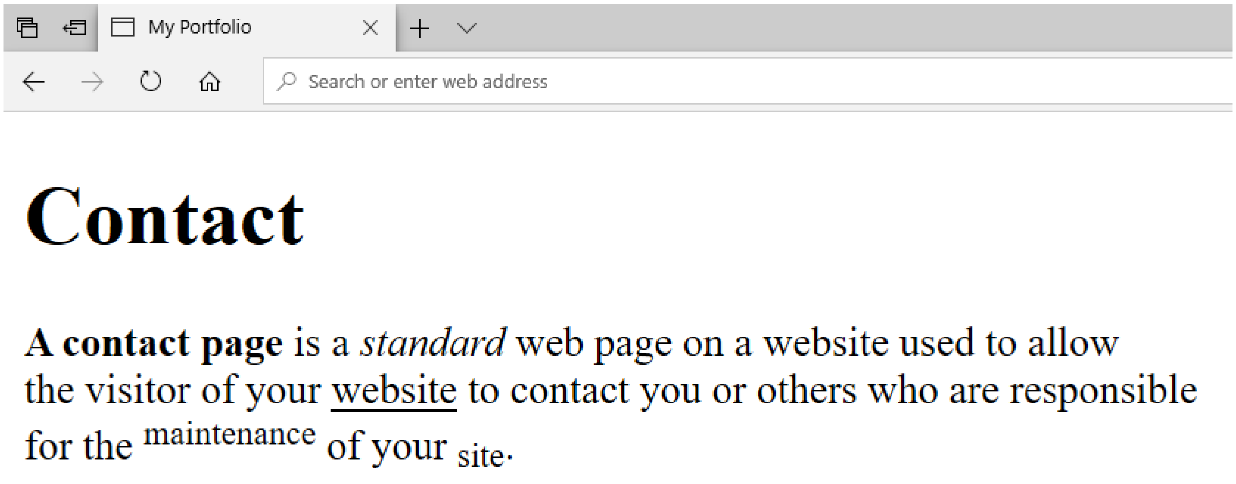
Figure 3.18 HTML code for various text formatting styles



Underlined text Small text Subscript text Superscript text

Figure 3.14 Browser results

Mixed formatting

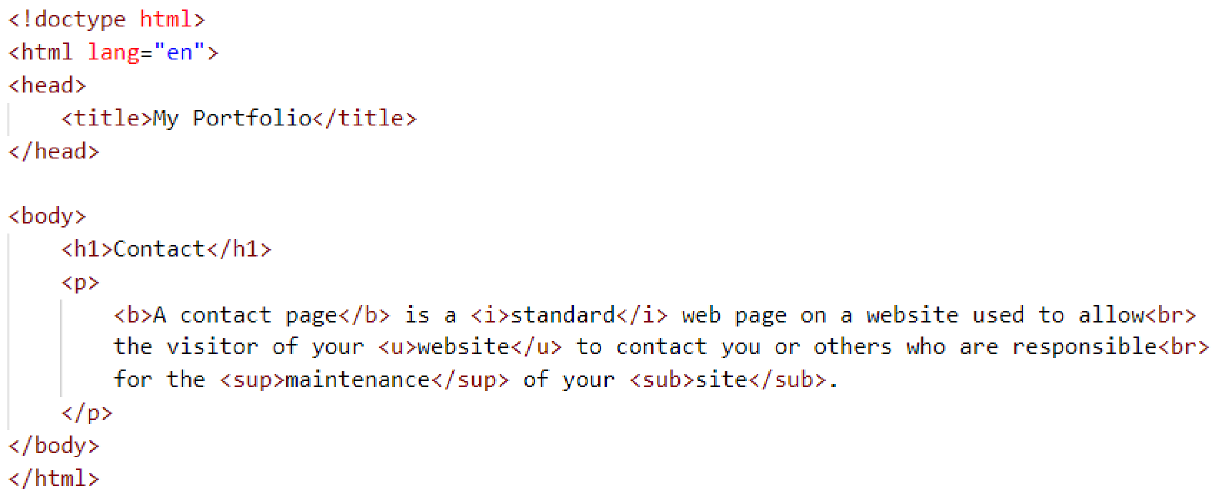


Contact

A contact page is a *standard* web page on a website used to allow the visitor of your website to contact you or others who are responsible for the maintenance of your site.

Figure 3.15 Random text mixed formatting

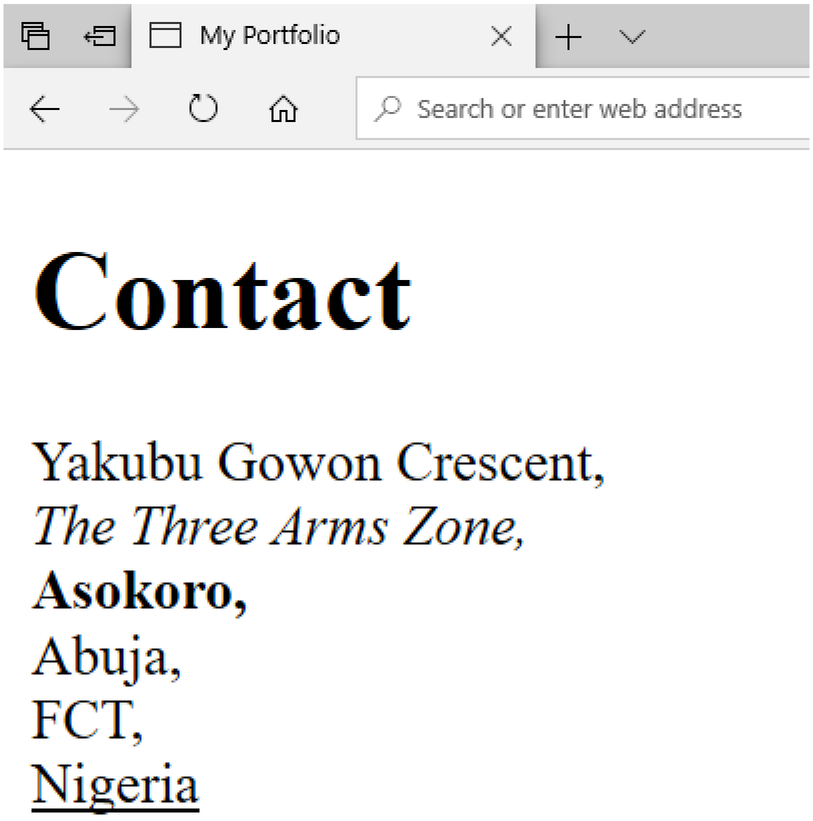
In figure 3.20 above, you will see that different ways of formatting text have been mixed with normal texts. Interesting, right? How does it work? Look at the HTML tags below (see figure 3.21) which was used to create this kind of formatting.



```
<!doctype html>
<html lang="en">
<head>
    <title>My Portfolio</title>
</head>

<body>
    <h1>Contact</h1>
    <p>
        <b>A contact page</b> is a <i>standard</i> web page on a website used to allow<br>
        the visitor of your <u>website</u> to contact you or others who are responsible<br>
        for the <sup>maintenance</sup> of your <sub>site</sub>.
    </p>
</body>
</html>
```

Figure 3.16 Random text mixed formatting. Sample code

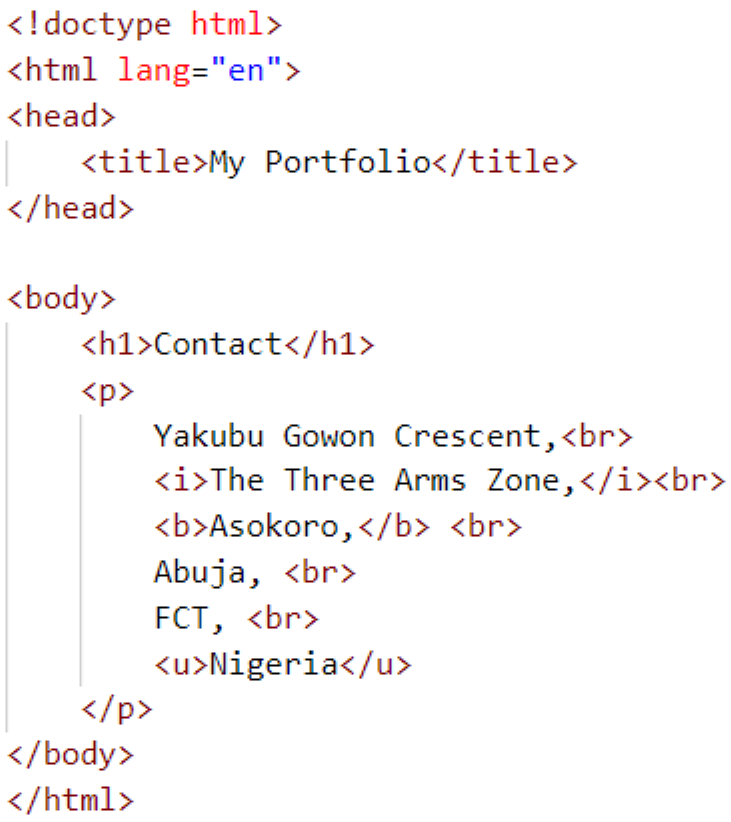
Let us now apply mixed formatting to our contact page



Contact

Yakubu Gowon Crescent,
The Three Arms Zone,
Asokoro,
Abuja,
FCT,
Nigeria

Figure 3.17 Contact page mixed formatting.



```
<!doctype html>
<html lang="en">
<head>
    <title>My Portfolio</title>
</head>

<body>
    <h1>Contact</h1>
    <p>
        Yakubu Gowon Crescent,<br>
        <i>The Three Arms Zone,</i><br>
        <b>Asokoro,</b> <br>
        Abuja, <br>
        FCT, <br>
        <u>Nigeria</u>
    </p>
</body>
</html>
```

Figure 3.18 Contact page mixed formatting. Sample code

Summary of Chapter Three

You have just learnt how to use heading and paragraph tags. You also learnt how to create a website portfolio, and as a sample, you created a contact page for John to show his contact information. Also, you can now effectively make use of text formatting on a webpage which is majorly used to call attention to that part of the text.

CHAPTER FOUR

LINKS

Chapter Objectives

At the end of this chapter, the students should be able to:

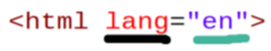
- **identify the HTML attributes and how they are used.**
- **identify one of the most important elements in HTML called Hyperlink.**
- **move from one web page to another using hyperlink.**
- **create a contact page and an about us page for their portfolio website.**
- **switch between each page of their portfolio website.**

HTML Attributes

In previous chapters, we learned that a webpage is a text file that contains both content and HTML tags, and it is saved as an HTML document using the ***.html*** file extension. HTML includes dozens of tags that describe the structure of webpages. For instance, the HTML tags *<p>* and *</p>* mark the start and the end of a paragraph area, while *<html>* and *</html>* indicate the start and end of a webpage.

An HTML element consists of everything from the start tag to the end tag, including content, and represents a distinct part of a webpage such as a paragraph or heading. For example, "***<title> Webpage Example </title>***" is an HTML element that sets the title of a webpage. In common usage, when web designers say "Use a '***p***' element to define a paragraph," or something similar, they meant, using a starting *<p>* tag to mark the beginning of a paragraph and an ending *</p>* tag to mark the end of the paragraph.

You can enhance HTML elements by using **attributes**, which define additional information about the contents of an element. They appear on the opening tag of the element and are made up of two parts which are: a name and a value, separated by an equals sign (=) as seen in figure 4.0 below.

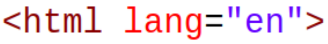


```
<html lang="en">
```

Figure 4.0: HTML attribute showing name and value.

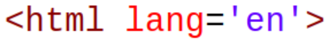
In the figure above, the part underlined in black on the left-hand side of the equals sign is the name of the attribute, meaning that the attribute's name is ***lang*** which is used to set the language of an HTML document. The attribute's name indicates what kind of extra information you are supplying about the element's content. It should be written in lowercase (small letters).

The part underlined in green on the right-hand side of the equals sign is the value of the attribute. The attribute has a value of ***en***; ***en*** means that we set the language of the HTML document to English. The value is the information or setting for the attribute. To write the value of an attribute, you wrap the value around double quotes or single quotes as seen in figure 4.1 and 4.2 respectively below.



```
<html lang="en">
```

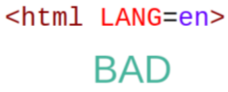
Figure 4.1 HTML attribute value surrounded by double quotes.



```
<html lang='en'>
```

Figure 4.2 HTML attribute value surrounded by single quotes.

HTML5 allows you to use uppercase attribute names and omit the quote marks, but this is not recommended. See figure 4.3



```
<html LANG=en>
```

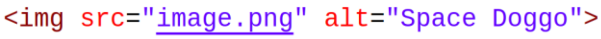
Figure 4.3 This method of the naming attribute is not recommended in HTML

Some Common HTML Attributes

Let us take a closer look at the list of some of the most common HTML attributes, you will learn how to use them later on.

alt

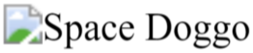
The attribute ***alt*** sets up an alternative text for an image element if the image element cannot be loaded on the browser. For example, if you upload a picture to your website, and for some reasons it cannot be loaded for a user, the alternative text of your choice will be shown instead. Using this attribute is important as a means of making your website easily accessible to disabled users. Blind or visually impaired people use screen readers to browse the internet. By providing an alternative text for visually impaired individuals to read, you allow them to understand the content fully.



```
<img src="image.png" alt="Space Doggo">
```

Figure 4.4 "alt" attribute

When you put this within the body tag *<body>* of your HTML document, you will see the result in your browser. The page is loaded and the value we specified in our **alt** attribute is displayed in the browser. See figure 4.5 below.

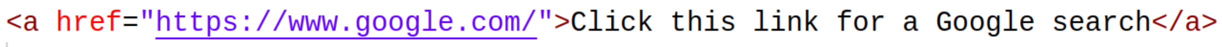


Space Doggo

Figure 4.5 Alternative text

href

HTML href attribute adds a URL destination for a link and therefore creates a hyperlink (we will look at hyperlinks later on in this chapter). Hyperlink works as the button which can take you to any other webpage.

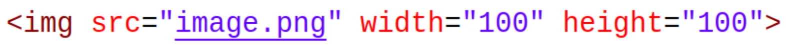


```
<a href="https://www.google.com/">Click this link for a Google search</a>
```

Figure 4.6 Href attribute

width & height

With **width**, you can change the width of the element; width extends horizontally (i.e. from left to right). While with **height**, you can adjust the height of the element, height extends vertically (i.e. from top to bottom). **Note:** Over-extending these attributes may make your elements look ridiculous.



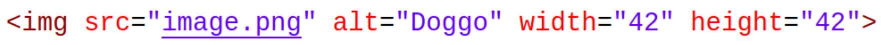
```
<img src="image.png" width="100" height="100">
```

Figure 4.7 Width and Height attributes

In the CSS chapters, you will learn how to set these properties with more flexibility using ***CSS***.

src

If you want to include an image on your website, you will definitely have to use the **src** attribute. **src** is an abbreviation for *source*. You use it to define a source from which the picture is taken.

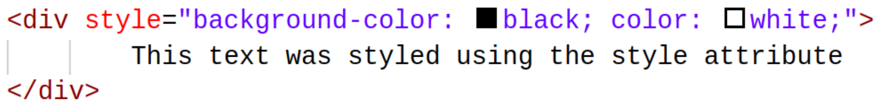


```
<img src="image.png" alt="Doggo" width="42" height="42">
```

Figure 4.8 src attribute

style

The **style** attribute allows you to add *CSS* code to an element. It can change the design of an element-its color, font, alignment or even add decorations. Using the **style** attribute is also called *inline style,* we will understand the style attribute later on in the CSS chapters. It is a nice property for small elements, but overusing it may create a very messy code.

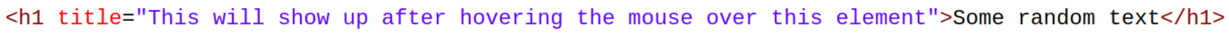


```
<div style="background-color: ■black; color: □white;">
        This text was styled using the style attribute
</div>
```

Figure 4.9 style attribute

title

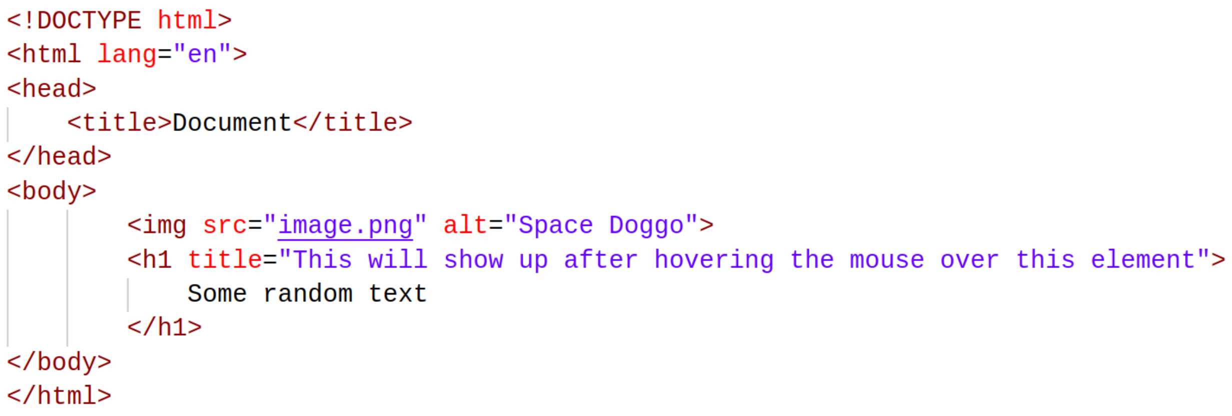
HTML **title** attribute adds additional information for another tag. If you add it, whenever you place your mouse on that particular element, it will show a tooltip with the title you created. You should not overuse this attribute however; your website might look messy.



```
<h1 title="This will show up after hovering the mouse over this element">Some random text</h1>
```

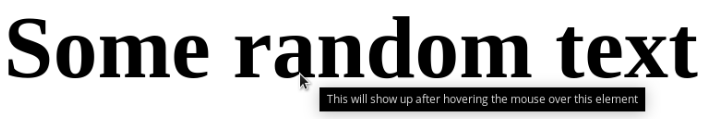
Figure 4.10 title attribute

When you put an h1 tag with a title attribute in an HTML document, and run it in your browser, you see additional information when you place your mouse over the text as shown in figure 4.12 below.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Document</title>
</head>
<body>
        <img src="image.png" alt="Space Doggo">
        <h1 title="This will show up after hovering the mouse over this element">
            Some random text
        </h1>
</body>
</html>
```

Figure 4.11 title attribute in HTML document



Some random text

This will show up after hovering the mouse over this element

Figure 4.12 title attribute showing additional information in the browser

Hyperlinks

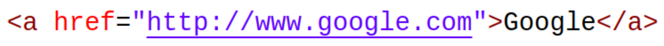
In the previous chapter, you learned about formatting text. A very important HTML element is the **Hyperlink** element. The hyperlink helps to move from one webpage to the other. Clicking on the huperlink on one webpage takes you to the link webpage or web address. In the HTML world, we say that we **navigate** from one page to another using Hyperlink.

You will commonly come across the following types of links:

- Links from one website to another;
- Links from one page to another on the same website;
- Links from one part of a web page to another part of the same page;
- Links that open in a new browser window;
- Links that automatically start up your email program and address a new email to someone.

Writing Links

Links are created using the *<a>* element, this is called the anchor. Users can click on anything between the opening *<a>* tag and the closing *</a>* tag. You can specify which page you want to link to, using the href attribute.

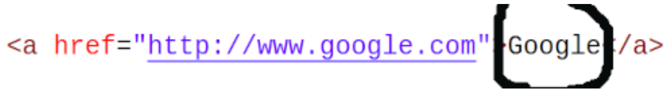


```
<a href="http://www.google.com">Google</a>
```

Figure 4.13 anchor tag <a> example

The text that is in between the opening *<a>* tag and closing *</a>* tag is known as the link text. Wherever possible, your link text should explain where visitors will be taken if they click on it (rather than just saying "*click here*"). You can see the link to the Google search page that was created in figure 4.13 above.

Many users navigate websites by scanning the text for links. The link text can help users find what they want. This will give them a positive impression of your site and may encourage them to stay glued to it for long.



```
<a href="http://www.google.com">Google</a>
```

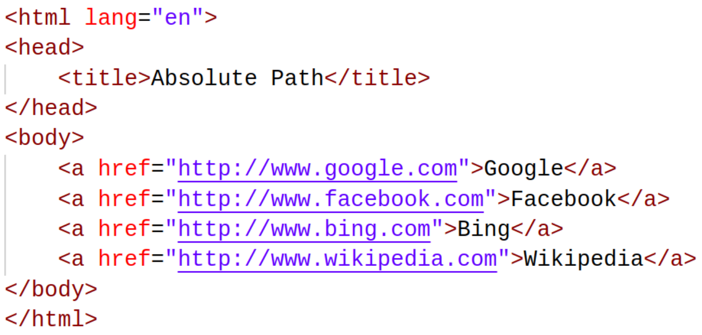
Figure 4.14 Google link text

To write a good link text, you can think of words people might use when searching for the page you are linking to. For example, rather than write "**Social Media**" you can use something more specific, such as "**Facebook**"

Linking to Other Sites

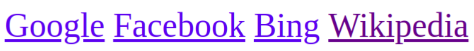
Links can be created using the *<a>* element which has an attribute called *href*. The value you specify in the *href* attribute is the page you want users to go to when they click on a link on your website. For example, if you want your users to go to the Google search page, you specify the Google URL (***http://www.google.com***) within the href attribute value. Users can click on anything that appears between the opening the *<a>* tag and the closing *</a>* tag, and they will be taken to that page specified in the *href* attribute.

When linking to a different website, the value of the href attribute will be the full web address for the site, this is known as an absolute URL. See examples of absolute paths in figure 4.15 below. An absolute URL starts with the domain name for that site for example, http://www.google.com, and it can be followed by the path to a specific page, for example http://www.facebook.com/home. If no page is specified, the site will only display the homepage.



```
<html lang="en">
<head>
    <title>Absolute Path</title>
</head>
<body>
    <a href="http://www.google.com">Google</a>
    <a href="http://www.facebook.com">Facebook</a>
    <a href="http://www.bing.com">Bing</a>
    <a href="http://www.wikipedia.com">Wikipedia</a>
</body>
</html>
```

Figure 4.15 Absolute URLs



Google Facebook Bing Wikipedia

Figure 4.16 Absolute URLs displayed in the browser

By default, browsers show links in blue, with an underline. Also links that have already been clicked are highlighted with a darker shade of purple.

Linking to Other Article Pages on the Same Site

When you are linking to other web pages within the same site, you do not need to specify the domain name ("*http://*") in the URL. You can make use of a shorthand known as a relative URL.

If all the web pages of the site are in the same folder, then the value of the href attribute will just be the name of the file. If you have different pages of a site in different folders, then you can use a slightly different method to indicate where the page is, in relation to the current page. We will learn more about this method in the next chapter.

In figure 4.17 below, you can see a simple folder structure. A folder was created and named **portfolio,** and within the **portfolio** folder, there are three HTML files named **about**, **contact** and **home**. There is also another folder within that portfolio folder called **image**. The image folder is called a subfolder because it lies within another folder. The image folder is where we will store the images which we will use for our portfolio website.

The *home.html* file will be the home page of your portfolio website. It is usually the first page that users see when they visit your site. The *contact.html* file will display your contact details and that includes your name, phone number, address and email. And the ***about*** page is going to display a few sentences about you, (it could be your age, favourite sport, favourite food, as well as your hobbies).

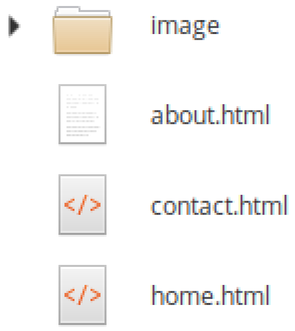
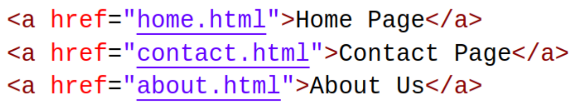


Figure 4.17 Folder structure

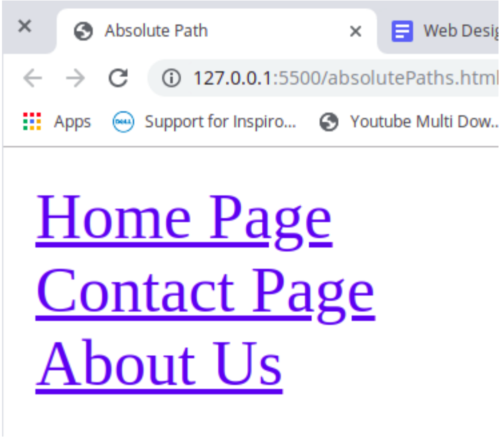


```
<a href="home.html">Home Page</a>
<a href="contact.html">Contact Page</a>
<a href="about.html">About Us</a>
```

Figure 4.18 Relative URLs displayed in the browser

Result:

If you look at the code above in figure 4.18, you will see that the *<a></a>* tags contain links that use relative URLs. When you click on the link text of each *<a>* element, you will be able to navigate to each web page located within your folder. Clicking on the Home Page will run the *home.html* file that is within your directory in your browser. Directories are simply folders that lie within your folder structure. Clicking on the Contact Page will run the *contact.html* file in your browser. The *contact.html* file contains your contact details for your portfolio website.



Home Page
Contact Page
About Us

Figure 4.19 Relative URL browser results

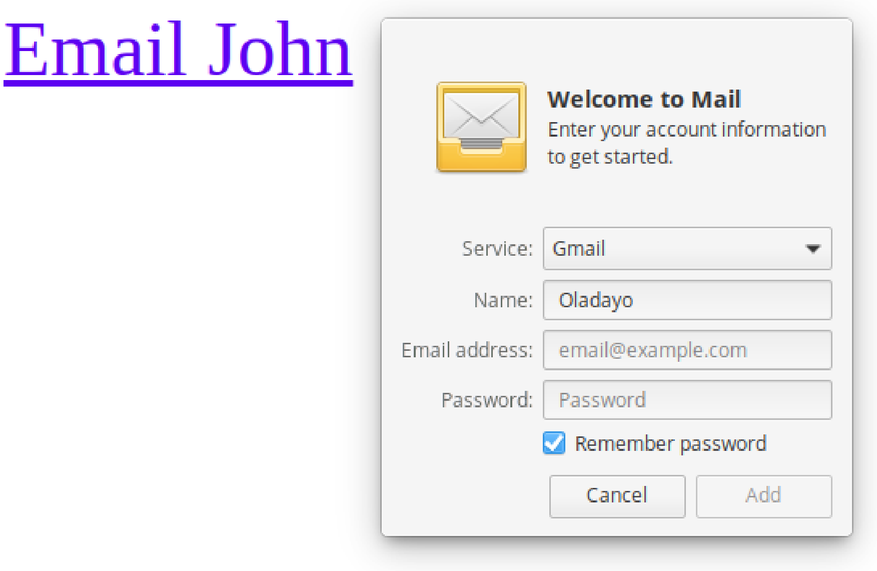
Relative URLs help when building a site on your computer because you can create links between pages.

Email Links (Using *mailto*)

To create a link that starts up a user's email program and sends an email to a specified email address (for example, sending an email from john@gmail.com to joseph@gmail.com), you use the *<a>* element. However, in this case, the value of the href attribute starts with *mailto:* and it is followed by the email address you want the email to be sent to.

```
<a href="mailto:john@gmail.com">Email John</a>
```

Figure 4.20 Email links HTML example



Email John

Welcome to Mail

Enter your account information to get started.

Service:	Gmail
Name:	Oladayo
Email address:	email@example.com
Password:	Password

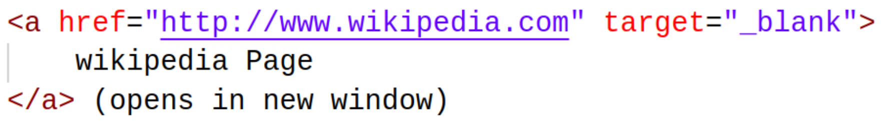
☑ Remember password

Figure 4.21 Email links browser results

In figure 4.21 above, you can see that the email link looks just like any other link but, when it is clicked on, the user's email program will open a new email message box and address it to the person specified in the link. The user’s email program might look different on your computer depending on your email client.

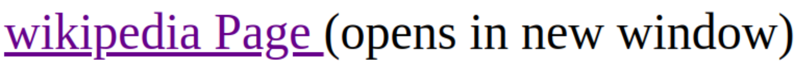
Opening Article Links in a New Window (Target)

If you want a link to open in a new browser window, you can use the target attribute on the opening *<a>* tag. The value of this attribute should be *_blank*. When the user of your website clicks on a link, the site that has been specified in the *href* attribute will be opened in a new browser window. One of the most common reasons you might want a link to be opened in a new window is if it points to another website.



```
<a href="http://www.wikipedia.com" target="_blank">
    wikipedia Page
</a> (opens in new window)
```

Figure 4.22 Using “target” HTML example



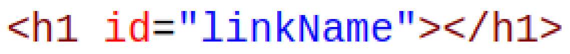
wikipedia Page (opens in new window)

Figure 4.20 “target” browser results

Linking to a Specific Part of the Same Page

When you are at the top of a long web page, you might want to access contents that are at the bottom part of the web page. For instance, you have a word document that is 1000 pages long, if you are on the first page and you want to access contents on page 800, you will have to scroll from page 1 to page 800 and, scrolling in this situation can be tiresome.

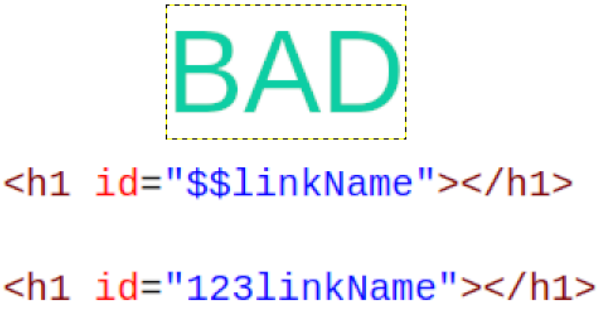
HTML provides a way for you to add shortcut links to any part of a web page. From our previous example, you can create a link from page 1 to page 800 in order to access your contents faster without having to scroll through hundreds of pages. You can also add a link from page 800 back to the top (i.e. page one) Before you can link to a specific part of a page, you need to identify the points in the page that the link will go to. You do this using the *id* attribute (which can be used on every HTML element).



```
<h1 id="linkName"></h1>
```

Figure 4.23 id attribute containing link name

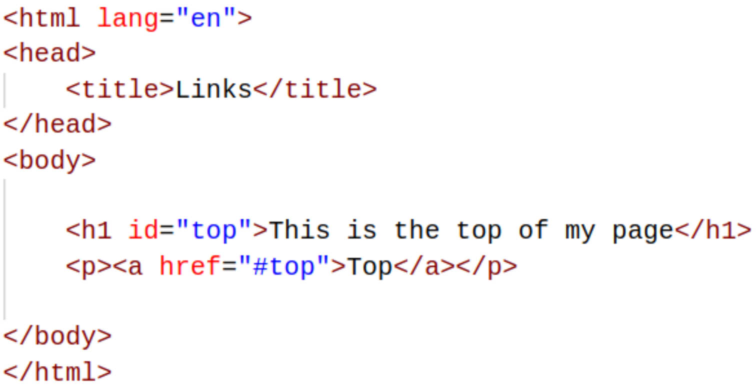
You can see that the *<h1>* element in figure 4.23 has been given an *id* attribute that identifies that particular section of the page. The value of the *id* attribute should start with a letter or an underscore (not a number or any other character) and, on a single page, no two *id* attributes should have the same value.



```
<h1 id="$$linkName"></h1>

<h1 id="123linkName"></h1>
```

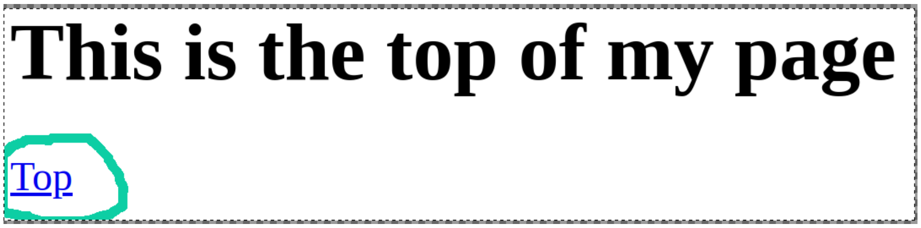
Figure 4.24 Non recommended way of writing id attributes for linking

To link to an element that uses an *id* attribute, you use the *<a>* element again. But the value of the *href* attribute starts with the # symbol, followed by the value of the *id* attribute of the element you want to link to. In figure 4.25 below, *<a href="#top">* links to the *<h1>* element at the top of the page whose *id* attribute has a value of *top*.



```
<html lang="en">
<head>
    <title>Links</title>
</head>
<body>

    <h1 id="top">This is the top of my page</h1>
    <p><a href="#top">Top</a></p>

</body>
</html>
```

Figure 4.25 Linking to the top of a page

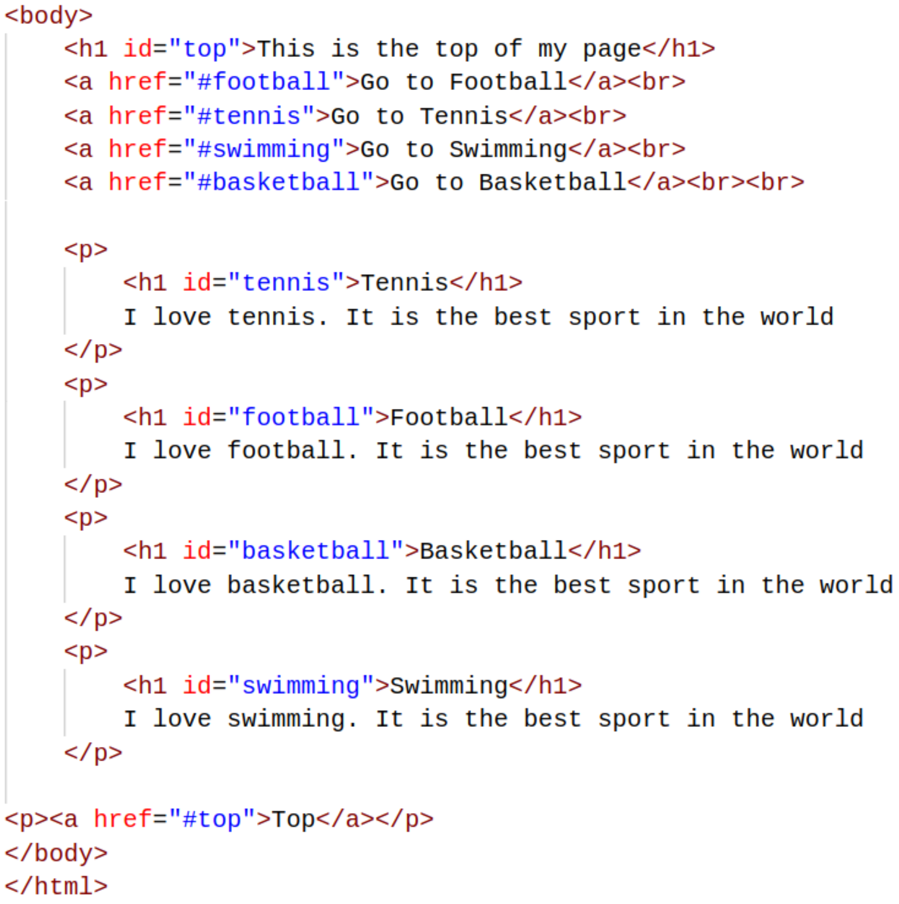


This is the top of my page

Top

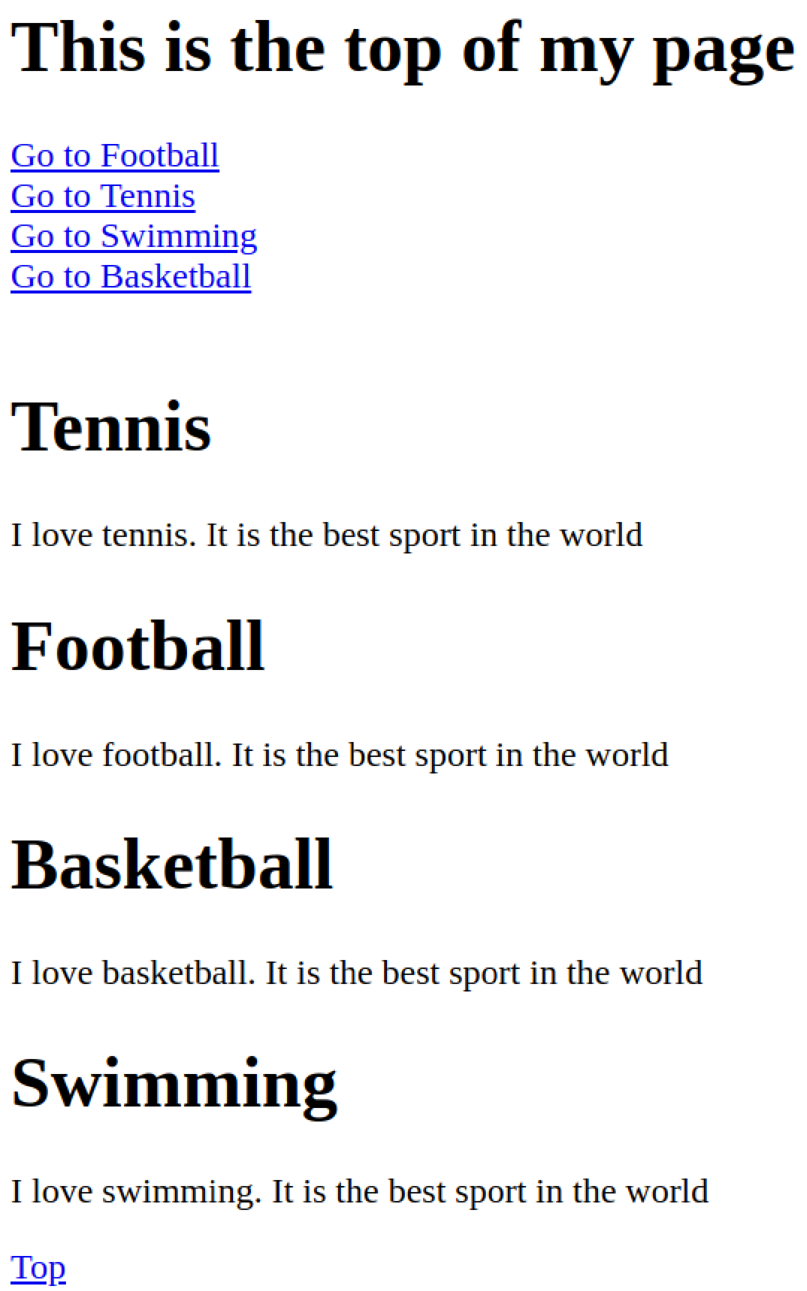
Figure 4.26 Browser results

Furthermore, you can link to different sections of your page content. As shown in figure 4.27 below, we have different sections, each containing popular sporting activities. When we click on any of the *<a>* links, it moves us to its corresponding section; for instance, clicking on the **<a href=" #football"> Go to Football </a>** moves us to the football section. You can check out the browser results in figure 4.28 below.



```
<body>
    <h1 id="top">This is the top of my page</h1>
    <a href="#football">Go to Football</a><br>
    <a href="#tennis">Go to Tennis</a><br>
    <a href="#swimming">Go to Swimming</a><br>
    <a href="#basketball">Go to Basketball</a><br><br>

    <p>
        <h1 id="tennis">Tennis</h1>
        I love tennis. It is the best sport in the world
    </p>
    <p>
        <h1 id="football">Football</h1>
        I love football. It is the best sport in the world
    </p>
    <p>
        <h1 id="basketball">Basketball</h1>
        I love basketball. It is the best sport in the world
    </p>
    <p>
        <h1 id="swimming">Swimming</h1>
        I love swimming. It is the best sport in the world
    </p>

<p><a href="#top">Top</a></p>
</body>
</html>
```

Figure 4.27 Linking to different sections of a page.



This is the top of my page

Go to Football
Go to Tennis
Go to Swimming
Go to Basketball

Tennis

I love tennis. It is the best sport in the world

Football

I love football. It is the best sport in the world

Basketball

I love basketball. It is the best sport in the world

Swimming

I love swimming. It is the best sport in the world

Top

Figure 4.28 Linking to different sections of a page; browser results.

When you click on any of the <*a*> links, you will be moved to its corresponding content section.

Summary of Chapter Four

In this chapter, you learned that attributes can be applied to all HTML elements; you can add attributes to HTML tags by entering their attribute's name to the opening tag. You also saw some common examples of HTML attributes. You learned to use the title attribute to include additional relevant information on your web page. You now know that values for the attributes should be written between double or single quotes.

Later on in the chapter, you further learned that links are created using the *<a>* element. *<a>* element use the *href* attribute to indicate the page you are linking to. If you are linking to a page within your own website, it is recommended that you should use relative links. You also learned that you can create links to open email programs with an email address in the *"mailto"* field. You can also use the *id* attribute to target elements within a page that can be linked to.

CHAPTER FIVE
IMAGES

Chapter Objectives:

At the end of this chapter, the students should be able to:

- **identify the usefulness of images to a website.**
- **save images on their computer using different image formats like jpg, png, jpeg.**
- **mention image attributes and their functions.**
- **change the height and width of images on their web page.**
- **add images to their portfolio website.**

In the previous chapter, you learned about hyperlink. The hyperlink is one of the most important elements in HTML because it provides a way to move or navigate between web pages. Now, let us move on to something visual, i.e. images or pictures. Websites show pictures to make web pages look more exciting, optical and lively. How do we show an image or picture on our web page? This chapter will answer that question.

Images

Images include photos, drawings, diagrams, charts, and other graphics that carry visual information. Images can be used as a way of also communicating messages to users visiting a website. On a webpage, they help breakdown texts and contribute to the design and beauty of a website. However, rather than merely decorate a webpage, images should support the purpose of the webpage. For instance, if a website contains information about animals, it should also display images about animals; an animal website should not display images of a car because cars do not fit into the purpose of the website.

Images can also provide visual representations of a company’s products and services. When determining the kind of image to use on your website, choose the kind that relates directly to the content. Images that do not support the content can be confusing or distracting.

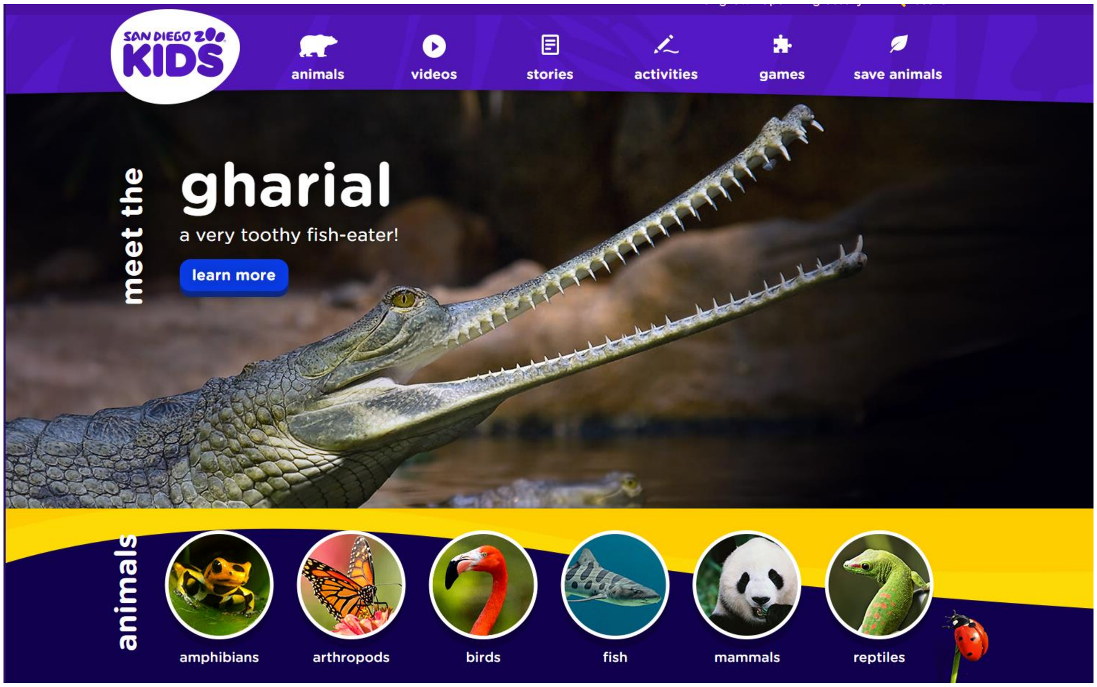


Figure 5.0 Images on a random Animal website showing relatable images.

Overusing Images

Be cautious about overusing images on your webpage. Using too many images may give your webpage a disorderly look, or distract users visiting your webpage. An image should have a purpose, such as, to make web contents easier to understand or comprehend; to visually organize a page; to provide a hyperlink; or to serve some other functions.

Image Types and Uses

Images can be used in many ways to enhance the look of a Web page, and make it look more lively and colourful. You can use images for the following reasons:

- To add a background colour to a Web page.
- To help organize a Web page.
- To help clarify a point being made in the text content.
- To serve as links to other Web pages (we will discuss image links later on in this chapter).

Images are also used to break Web pages into sections, or as a directional element which allows a user visiting your website to easily navigate through your Website. Web pages use three types of image files which are: **GIF**, **JPEG**, and **PNG**.

GIF (Graphics Interchange Format)

Graphics Interchange Format (GIF) files have an extension of ***.gif***. Remember we discussed file extensions in chapter two, where we created an HTML file by saving it on our computer using a ***.html*** file extension.

A graphic image saved as a GIF (pronounced "giff") uses compression techniques, called LZW compression, to make it smaller for download on the Web. Standard GIF images are displayed one line at a time when loading. Interlaced GIF images load all at once, starting with a blurry look and then, becoming sharper as they load. Using interlaced GIFs for large images is a good technique because visitors of your website can see a blurred outline of the image as it loads. See an example of a GIF image in figure 5.4 below.

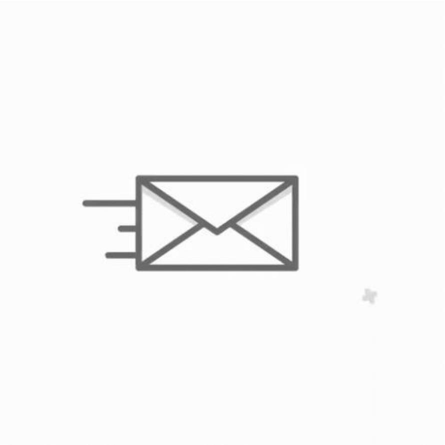


Figure 5.1 Example of a GIF image, gotten from tenor.com

PNG (PORTABLE NETWORK GRAPHICS)

The second type of image file is Portable Network Graphics (PNG), which has a *.png* extension. The PNG (pronounced "*ping"*) format supports multiple colours and resolutions. Resolutions refer to the image details, that is, the quality of an image. Higher resolution images mean higher quality while lower resolution images mean lower quality. The PNG format is a free alternative to the GIF format. Most newer browsers including the latest versions of Google Chrome support PNG images.

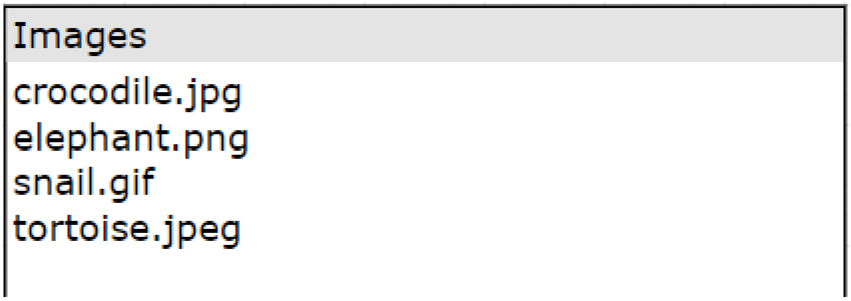
Figure 5.2 A sample PNG image from pixabay.com

JPEG (JOINT PHOTOGRAPHIC EXPERTS GROUP)

Finally, JPEG (Joint Photographic Experts Group) files have an extension of *.jpg*, *.jpe*, or *.jpeg*. A JPEG (pronounced *"JAY-peg"*) is a graphic image saved using a lossy compression technique. It is best suited for images with smooth variations of tone and colour. JPEG files are often used for more complex images, such as photographs because the JPEG file format supports more colours and resolutions than the other image file types.

Storing images on your computer

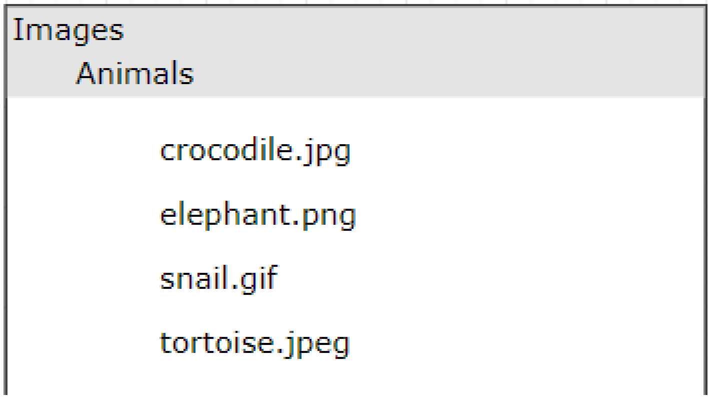
While building your web site from scratch, it is advisable to always create a folder for all the images your website uses. As the content and images on your website increases, keeping images in a separate folder will help you to understand how your site is organized. Here, you can see an example of the folder structure for a website; all of the images are stored in a folder called **Images**.



```
Images
crocodile.jpg
elephant.png
snail.gif
tortoise.jpeg
```

Figure 5.3 An image folder with animal images

For a big website, you might have to add subfolders inside the image folder. For example, all animal images can be put inside a sub-folder called **Animals**.



```
Images
    Animals

            crocodile.jpg

            elephant.png

            snail.gif

            tortoise.jpeg
```

Figure 5.4 An animal sub-folder.

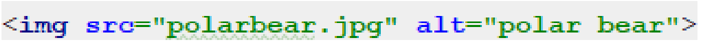
Adding images to your website

To add an image to a webpage, you need to use an element which is the image tag, written as *<img>*. This is an empty element (which means there is no closing tag). It must carry the following two attributes:

- *src,* and

- *alt*

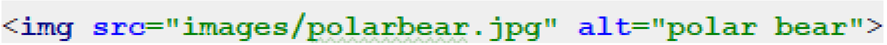
src
This tells the browser where it can find the image file. This is usually a *relative URL* (we discussed *relative URL* paths in the previous chapter) pointing to an image on your site. Here, the image is in an absolute path (which we also discussed in the previous chapter). See figure 5.14 below.



```
<img src="polarbear.jpg" alt="polar bear">
```

Figure 5.14 Image with absolute path

Here, you can see that the image is in a subfolder called images. See figure 5.15 below.



```
<img src="images/polarbear.jpg" alt="polar bear">
```

Figure 5.15 Image with relative path

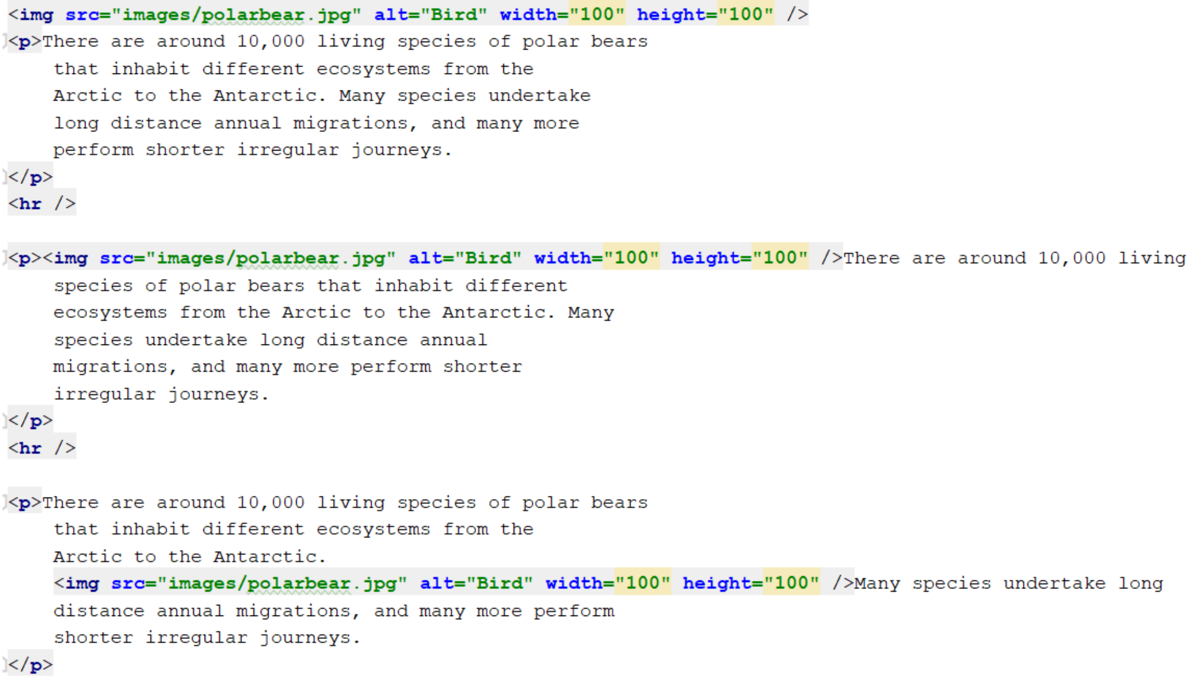
alt
This provides a text description of the image; it describes the image if it fails to load due to an error or loss of internet connection.

Placing Images in Codes

Where an image is placed in your code will affect how it will be displayed on your browser. Here are some examples of image placement that will produce different results:

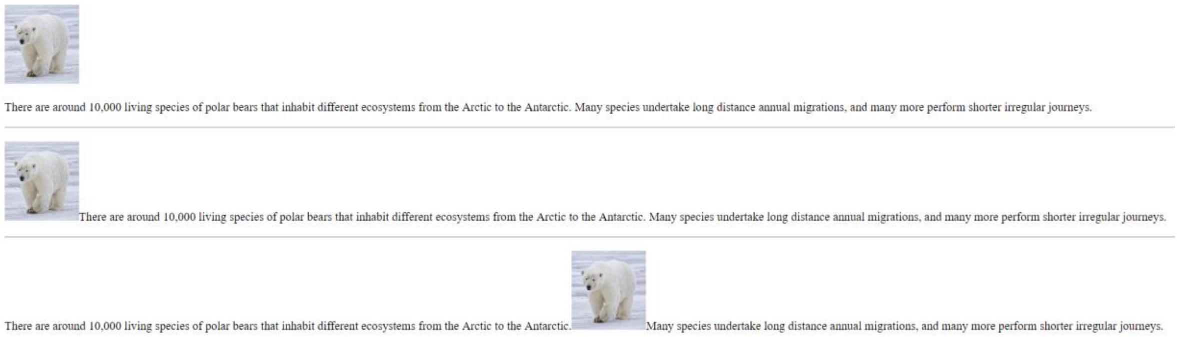
- Before a paragraph- In this case, the paragraph starts on a new line after the image.
- Inside the start of a paragraph- In this case, the first row of text aligns with the bottom of the image.
- In the middle of a paragraph- In this case, the image is placed between the words of the paragraph that it appears in.

See figure 5.16 and 5.17 below for examples.



```
<img src="images/polarbear.jpg" alt="Bird" width="100" height="100" />
<p>There are around 10,000 living species of polar bears
    that inhabit different ecosystems from the
    Arctic to the Antarctic. Many species undertake
    long distance annual migrations, and many more
    perform shorter irregular journeys.
</p>
<hr />

<p><img src="images/polarbear.jpg" alt="Bird" width="100" height="100" />There are around 10,000 living
    species of polar bears that inhabit different
    ecosystems from the Arctic to the Antarctic. Many
    species undertake long distance annual
    migrations, and many more perform shorter
    irregular journeys.
</p>
<hr />

<p>There are around 10,000 living species of polar bears
    that inhabit different ecosystems from the
    Arctic to the Antarctic.
    <img src="images/polarbear.jpg" alt="Bird" width="100" height="100" />Many species undertake long
    distance annual migrations, and many more perform
    shorter irregular journeys.
</p>
```

Figure 5.16



There are around 10,000 living species of polar bears that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.

There are around 10,000 living species of polar bears that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.

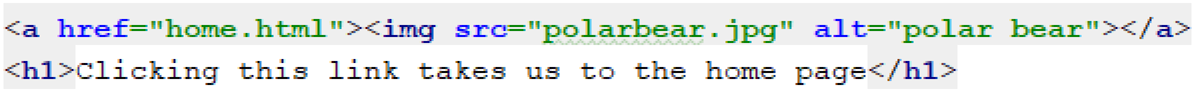
There are around 10,000 living species of polar bears that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.

Figure 5.17

Image Links

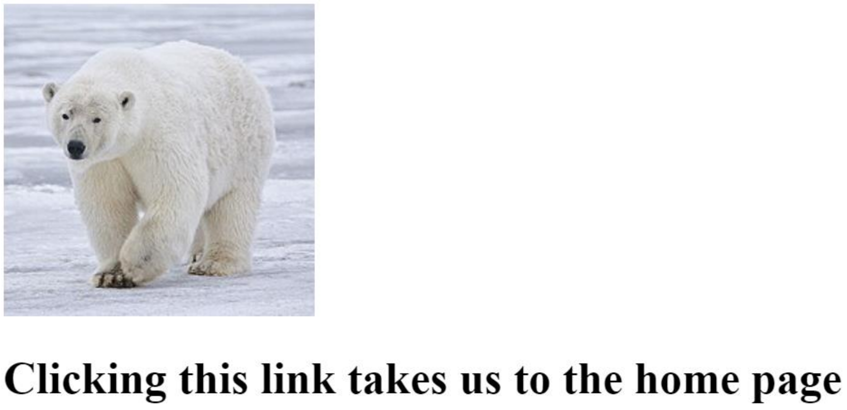
Just like link texts, images can also link to another web page, another website, an email address, or a telephone number. To configure an image with a link, place the starting anchor tag ***<a>*** before the image element and place the ending anchor tag ***</a>*** after the image element.

Figure 5.16 shows an example of an image with a relative link to the website's home page. This code means that when you click on the polar bear image on your browser, you will be redirected to the home page of your website.



```
<a href="home.html"><img src="polarbear.jpg" alt="polar bear"></a>
<h1>Clicking this link takes us to the home page</h1>
```

. **Figure 5.16 Adding hyperlink to an image**



Clicking this link takes us to the home page

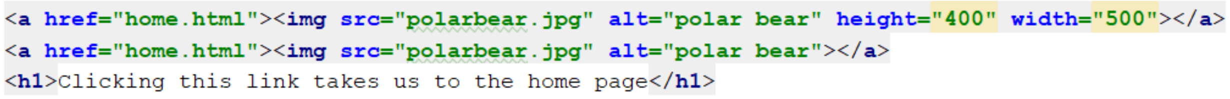
Figure 5.17 Browser results

Height & Width of Images

You will also often see an element use two other attributes that specify its size, these attributes are: height and width.

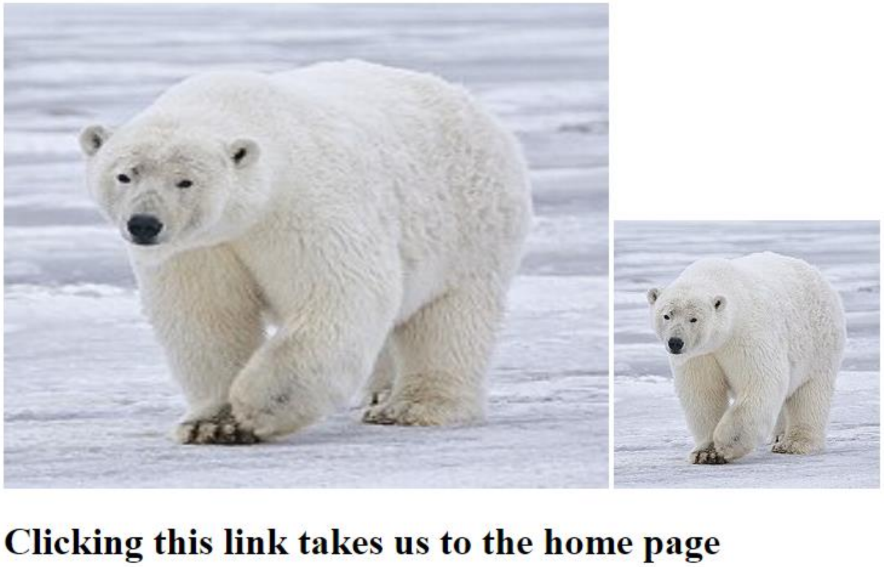
Height specifies the height of the image, and it stretches vertically (i.e. from top to button). Width specifies the width of the image, and it stretches horizontally (i.e. from left to right).

Let us set a height and a width for our polar bear.



```
<a href="home.html"><img src="polarbear.jpg" alt="polar bear" height="400" width="500"></a>
<a href="home.html"><img src="polarbear.jpg" alt="polar bear"></a>
<h1>Clicking this link takes us to the home page</h1>
```

Figure 5.18 Setting image height and width



Clicking this link takes us to the home page

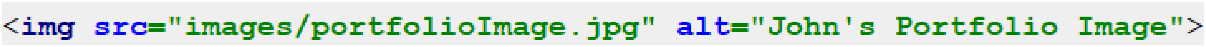
Figure 5.19 Setting image height and width browser results

In figure 5.19 above, the image on the right is the original image size while the image on the left is the new image size after setting a height of 700pixels and a width of 500pixels.

Adding an Image to your portfolio website

Remember our friend, John? We are still in the process of building John's portfolio website. Let us add an image to the home page of his website.

We have created a file called ***home.html*** which will be John's home page. We will be adding an image called ***portfolioImage.jpg*** to his webpage (take note of the upper and lower cases when naming images). We will be adding a fallback text, in case John's portfolio image fails to load. See figure 5.20 below.



```
<img src="images/portfolioImage.jpg" alt="John's Portfolio Image">
```

Figure 5.20 Adding image to the portfolio website

After adding the image, we have something like this. See figure 5.21 below

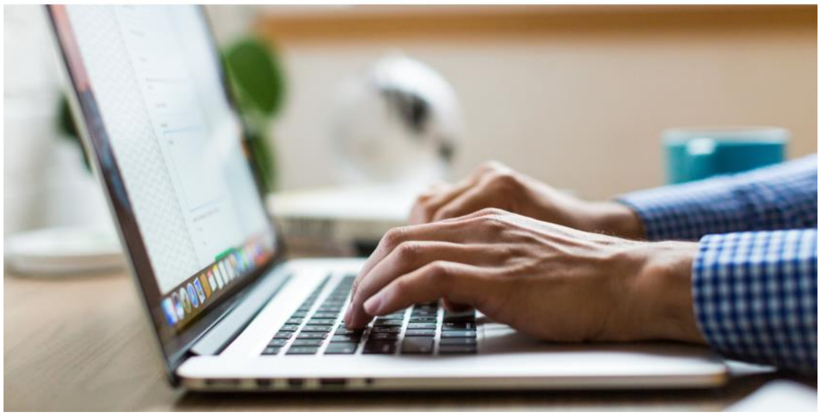


Figure 5.21

Next, let us set a height and a width for John's portfolio image. We set a height of 1500pixels and a width of 2000pixels. See figure 5.22

Figure 5.22

Great! You now know how to add an image to a webpage. Start warming up for upcoming chapters.

Summary of Chapter Five

In this chapter, you learned about images and their usefulness on a website, you also learned how to save images to your computer using different image formats. Congratulations! You now have a good understanding of how to add images to a website. HTML is getting more exciting, right? Let us explore other awesome things we can do with HTML.