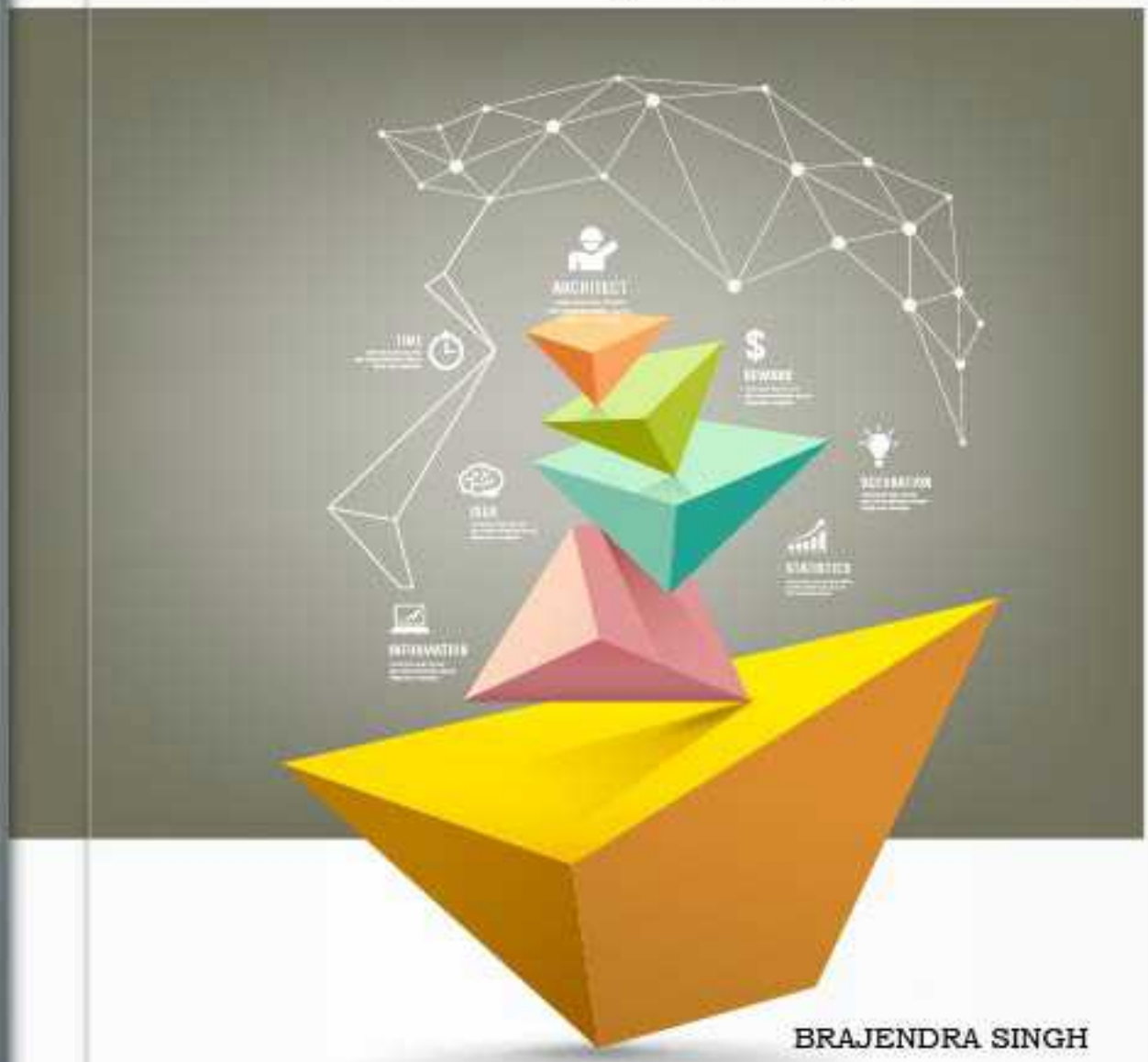
HTML for Kids

Perfect book for young curious minds



BRAJENDRA SINGH

PATHIK RAWAL

Table of Contents

Preface

One beautiful morning when I was enjoying my freshly brewed coffee and looking forward to a busy, interesting day, my son came to me and asked "*Dad! How do people create web sites? I heard they use HTML. How can I learn HTML?*" That day evening, I spent a good time searching a right book for him. It was not a quest for any HTML book, but it was a research to find the right content according to his young age and experience. It was not about finding a book which is complete for HTML, but it was about finding a book which takes him through baby steps to make him comfortable about HTML. I wanted a book which could grow his enthusiasm and curiosity to next level. Unfortunately, I could not find such book.

But it motivated us that why should we not write one such book ourselves? We then worked on this book after all our first customer was at our home only. When writing this book, we took special care to keep the learning simple and with full of examples from real world. The kids reading this book will not only learn fundamentals of HTML but also be able to visualize how these fundamentals have been used in real web sites. This book is not a complete reference to HTML but it has been written to make young minds more inquisitive about HTML and inspire them to seek advanced learnings.

After reading this book, the kids will be able to talk HTML and write web pages, but most important they would come back and ask "*Mom, Dad! I want to learn more about HTML?"*

There is no age limit for the reader of this book because curiosity has nothing to do with age. Any person interested in HTML can use this book to understand the basics.

Best Wishes!

Chapter 1 –Getting Started

Welcome Friends!

Since you have this book in your hand, I guess that you have a keen desire to learn about building websites. Your young mind wants to study and practice the fundamentals of creating a website.

Websites are created using various technologies and languages such as HTML, Scripts, styles etc. HTML is the language to create websites. It stands for **Hypertext Markup Language.** I am not going to explain why HTML has this full name because you need to know few other things before you can understand the meaning. I don't want to confuse you and trust me not knowing the name is not going to hurt you, at least as a beginner. Someone rightly said "*What's in the name?"* You can learn meaning of HTML at some later point of time when you have broader knowledge about web technologies. But we are definitely going to understand how to write HTML. After all, no web site is created without writing HTML.

Well, you have a right book in your hand. This book focuses on HTML and it is written for fundamentals. We are going to help you take those baby steps which will make you comfortable with HTML. We will learn all those things which a beginner is supposed to know. Then you can pursue advanced learnings later.

What you should know before reading this book

There are few required skills for the reader of the book. These skills will help reader learn the chapters faster. The book is not going to focus on these skills. The skills are –

- ✓ How to use computer?
- ✓ How to access website using browsers such as Internet Explorer, Chrome, and Firefox etc.?
- ✓ How to create a file in notepad and save the same?

C H A P T E R 1

The computer which you use to practice the exercises should be connected to internet.

Very basic skills and simple requirement. Isn't it?

How to read this book?

The approach taken by this book suits for beginners. Each of the chapters use simple language, explain fundamentals, show examples and finally ask you to do a small exercise. Each chapter covers one of the topic related to HTML and teaches in the following way:

- ✓ First of all the topic is introduced and its purpose is explained
- ✓ We then show example of the topic from the real web site
- ✓ The topic is then explained in detail using examples
- ✓ Finally a small exercise is given to practice the topic. The answer key for the exercise is given towards the end of the book.

We have also recommended for advanced study of HTML in the later chapters.

What is a Website?

We all know what a website is. I am sure everyone has used websites one way or another. Companies and people own and publish websites for certain purpose. The purpose could be anything – search information, sharing information, selling things, education, taking admission or booking flights.

Let's understand with few examples.

The Google website (https://www.google.co.uk/) is used for searching information on internet. It has been shown in figure1.1.

C H A P T E R 1

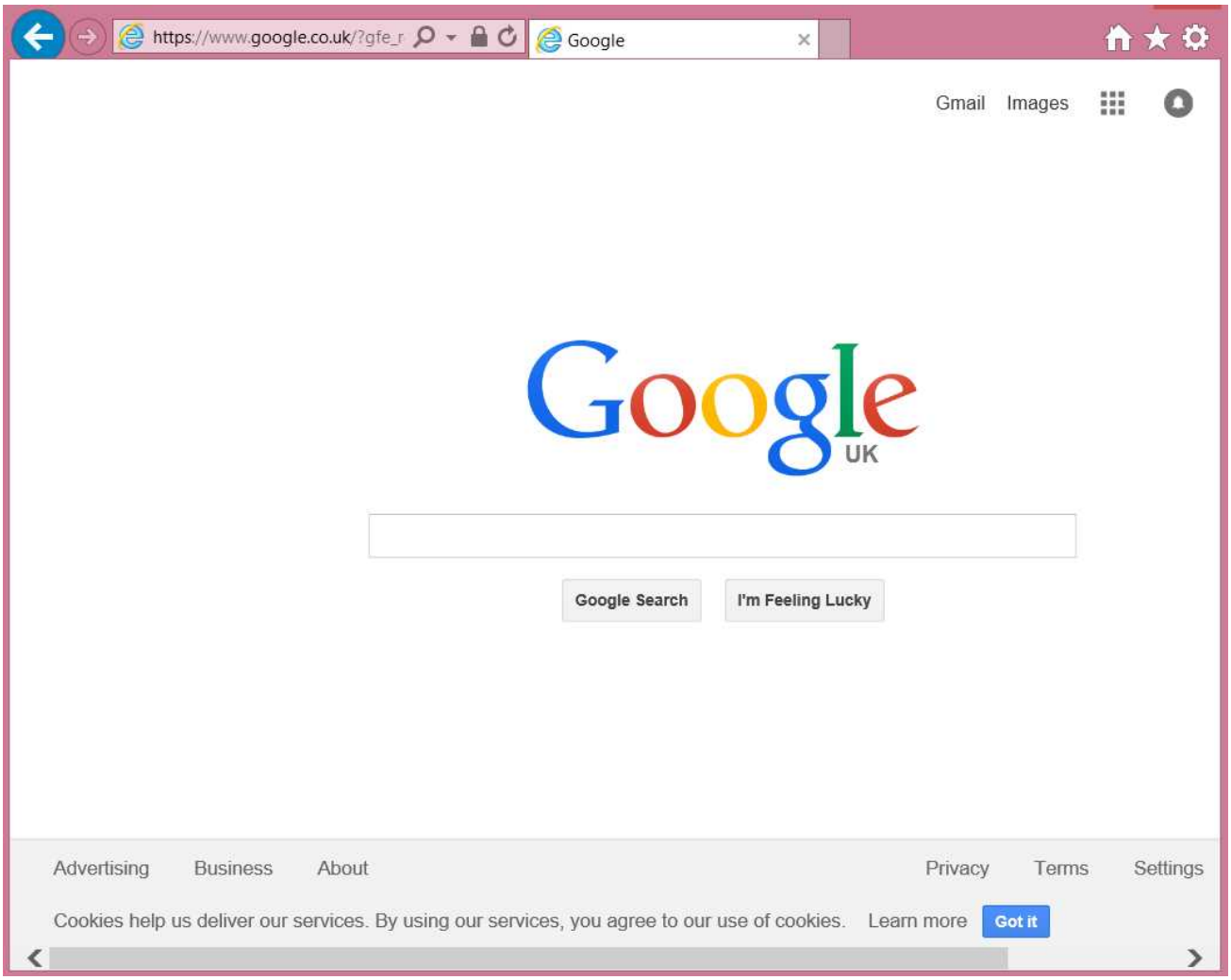


Figure 1.1: Home Page or Landing Page of Google Website

The Amazon website (http://www.amazon.co.uk/) sells things on internet. It has been shown in figure 1.2.

C H A P T E R 1

Figure 1.2: Home Page or Landing Page of Amazon Web Site

The Booking web site (http://www.booking.com/) is used to book hotels on internet. Check the figure 1.3 below.

CHAPTER 1

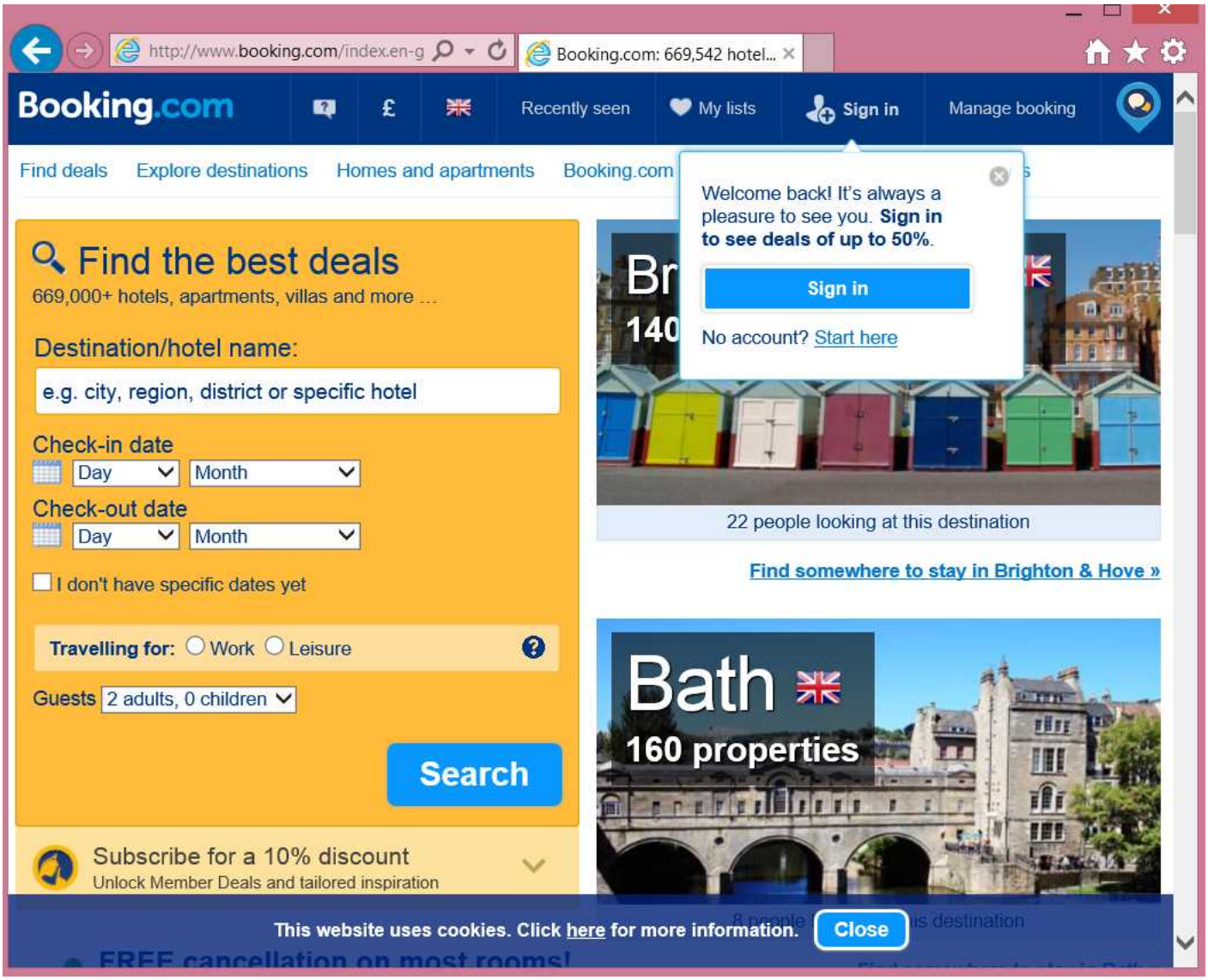


Figure 1.3: Home Page or Landing Page of Booking.com's Website

The Wikipedia website (http://www.wikipedia.org/) is used to share information on internet. It has been shown in figure 1.4.

C H A P T E R 1

Figure 1.4: Home Page or Landing Page of Wikipedia Website

You can see there is no boundary to what you want to achieve with website. Sky is the limit. Tell me, what is your idea about your own website?

C H A P T E R 1

What is a webpage?

You now have a fair understanding of what website is and what purpose it can solve. The very first question comes in our mind – what is this website made of? This question gives birth to a term called **webpage**.

Website is a collection of web pages. It means a website is made of one or more webpages.

Let's understand this concept with an example as shown in figure 1.5.

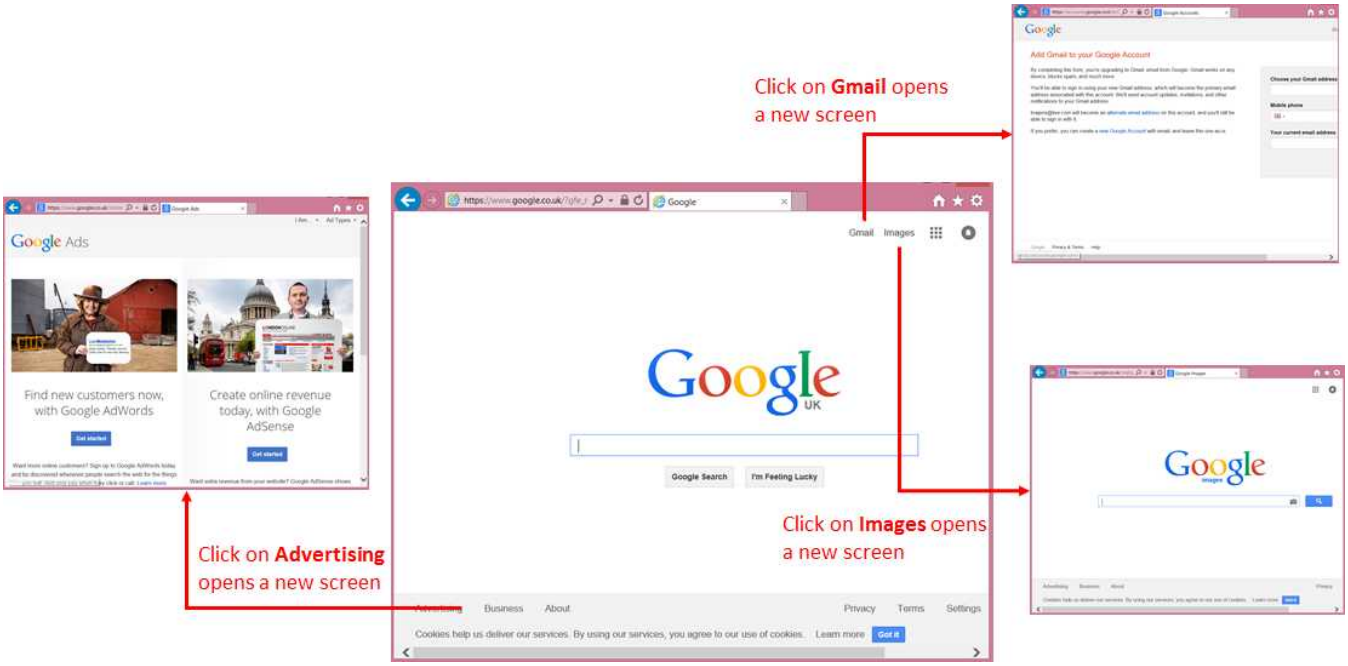


Figure 1.5: Webpages Collection

Go to Google website (https://www.google.co.uk/). When you click on **Gmail** link, it opens a new screen. Similarly when you click on **Images** or **Advertising** links, it opens respective new screens. There are many more links like these but I just gave three examples. Now each of these screens which are opening are nothing but individual web pages**. Each screen** represents **a web page**. Through these links and clicks, it is obvious that website is a collection of these screens. Since, each screen represents a web page, using this analogy we can say that **"Website is a collection of web pages"**.

To create a website, you create each of these individual screens or web pages and then place and link them together. When you open website, whatever screen or web page opens first is called **Home Page or Landing Page**. You then go to other web pages from the home page.

What is HTML?

C H A P T E R 1

We learnt that website is made of web pages. Next natural question is – what is webpage made of?

A web page is created using HTML. HTML is a language to define or create a web page.

How HTML looks like? Let's find out in an interesting way.

Open Wikipedia site (http://www.wikipedia.org/) in Internet Explorer browser as shown in figure 1.6. In the browser, **right click on the home page** using mouse. In the right click menu, click on **View Source** option as shown in the figure 1.6 below:

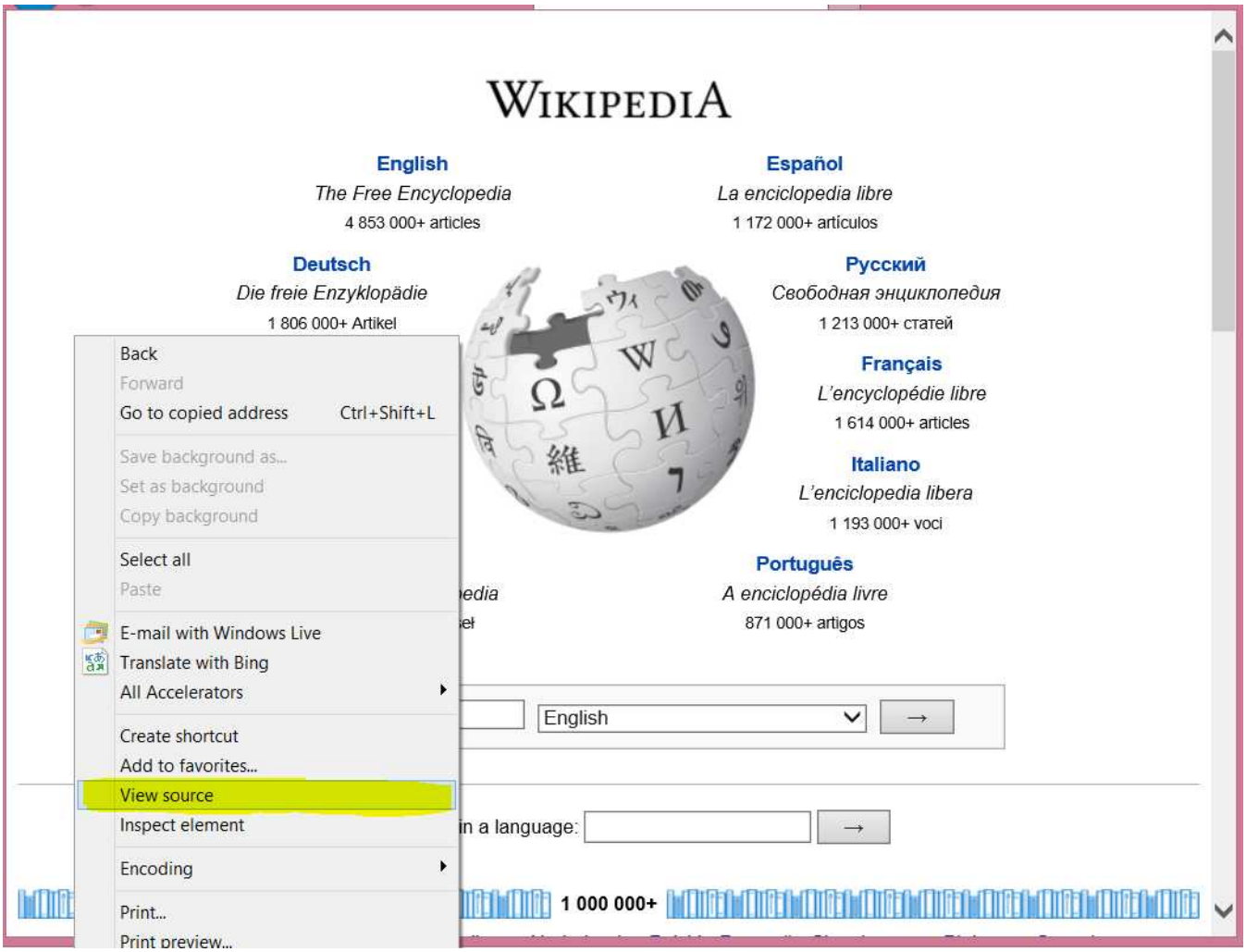


Figure 1.6: Internet Explorer's Right click to See View Source Option

Click on View Source option will open HTML behind home page in notepad. This is the same HTML which has been used to create this webpage. Please check how HTML looks like in the figure 1.7 below:

```
<!DOCTYPE html>
<html lang="mul" dir="ltr">
<head>
<!-- Sysops: Please do not edit the main template directly; update /temp and synchronise. -->
<meta charset="utf-8">
<title>Wikipedia</title>
<!--[if lt IE 7]><meta http-equiv="imagetoolbar" content="no"><![endif]-->
<meta name="viewport" content="initial-scale=1.0, user-scalable=yes">
<link rel="apple-touch-icon" href="/static/apple-touch/wikipedia.png">
<link rel="shortcut icon" href="/static/favicon/wikipedia.ico">
<link rel="license" href="//creativecommons.org/licenses/by-sa/3.0/">
<link rel="stylesheet" href="//meta.wikimedia.org/w/load.php?debug=false&amp;lang=en&amp;modules=ext.gadget.wm-portal&amp;only=styles&amp;skin=vector&amp;*">
<style type="text/css">
.central-featured-logo{background:url(//upload.wikimedia.org/wikipedia/meta/0/08/Wikipedia-logo-v2_1x.png) center center no-repeat;left:0;top:0;width:100%;height:100%
</style>
<!--[if lt IE 8]><style type="text/css">
.bookshelf-container .bookend,.otherprojects-icon{zoom:1;display:inline}
</style><![endif]-->
</head>
<body id="www-wikipedia-org">

<h1 class="central-textlogo" style="font-variant: small-caps;">
<img src="//upload.wikimedia.org/wikipedia/meta/6/6d/Wikipedia_wordmark_1x.png" srcset="//upload.wikimedia.org/wikipedia/meta/a/a9/Wikipedia_wordmark_1.5x.png 1.5x, /
</h1>

<!-- container div for the central logo and the links to the most viewed language editions -->
<div class="central-featured">

<!-- logo -->
<div class="central-featured-logo"></div>

<!-- Rankings from http://stats.wikimedia.org/EN/Sitemap.htm (data from 2015-02-28) -->
<!-- Article counts from http://meta.wikimedia.org/wiki/List_of_Wikipedias/Table -->
<!-- #1. en.wikipedia.org - 8,567,867 views/hr -->
<div class="central-featured-lang lang1" lang="en">
<a class="link-box" href="//en.wikipedia.org/" title="English – Wikipedia – The Free Encyclopedia"><strong>English</strong><br>
<em>The Free Encyclopedia</em><br>
```

Figure 1.7: Internet Explorer's View Source Showing HTML

Using the method described above, you can see HTML for any web page on any website.

The HTML at present might not be making any sense to you. But don't worry, once you have read through this book, you will be able to understand what is written here.

Summary

Great, you understand the fundamentals now. The coming chapters will describe specific topics related to HTML in detail. What are you waiting for? Continue reading and quench your thirst…

Chapter 2 – Learn the Basics

In the first chapter, we understood the concept of web site and web pages. We learnt that the HTML is the language of creating web pages. You also saw an example of HTML from a real web site. I think we are in a good position to start looking into basics of HTML. Then in coming chapters, we will develop broader understanding of HTML.

The current chapter focuses on the structure of HTML and how to view HTML in a web browser. We will also learn about tags and attributes which are the elementary building blocks of HTML. **Just like human body is made of cells; HTML is made of tags.**

"Hello World" Web Page

I know we have not learnt even a single thing about HTML, but let's be a little courageous and write down our very first HTML web page without knowing anything about it. Our very own first "Hello World" web page. Sounds great, right! Follow these steps:

1. Open **notepad** with blank screen as shown in the figure 2.1.

C H A P T E R 2

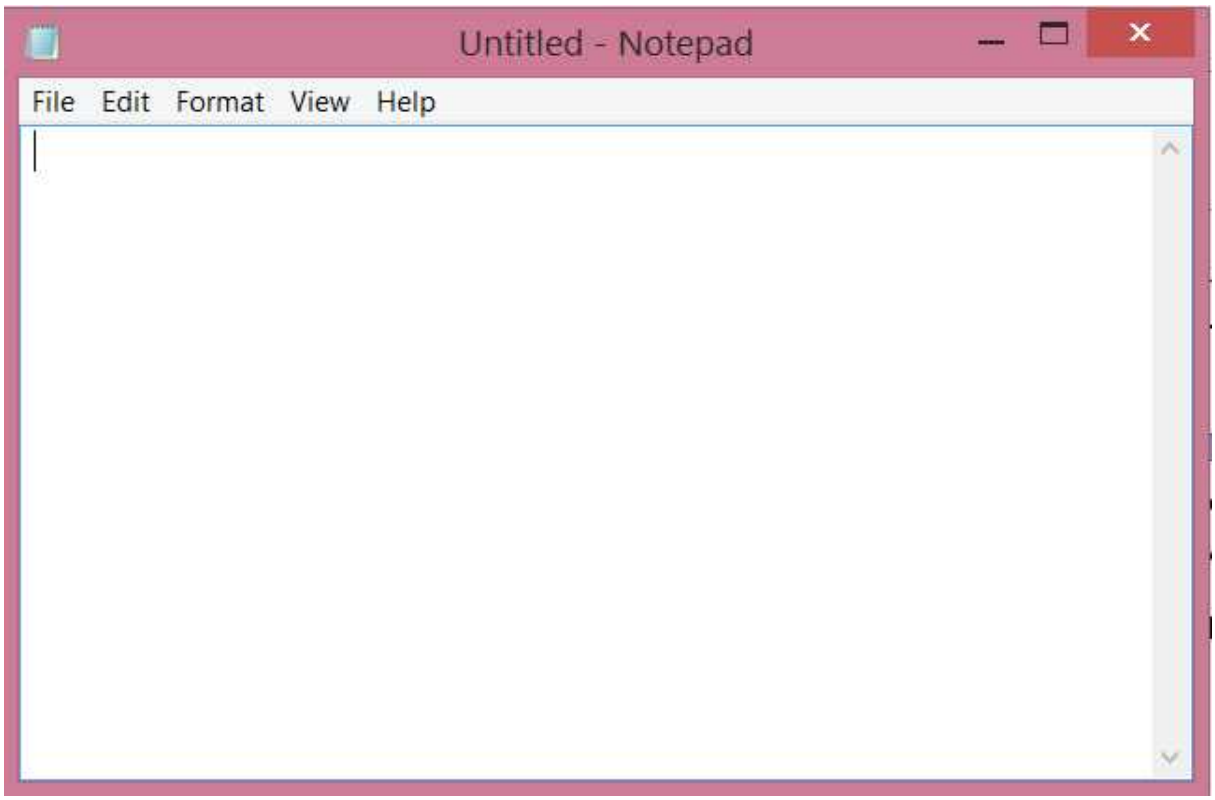


Figure 2.1: Blank Notepad

2. In this notepad, type your hello world HTML exactly as shown in the figure 2.2. Come on, it is just eight lines. It would not take long.

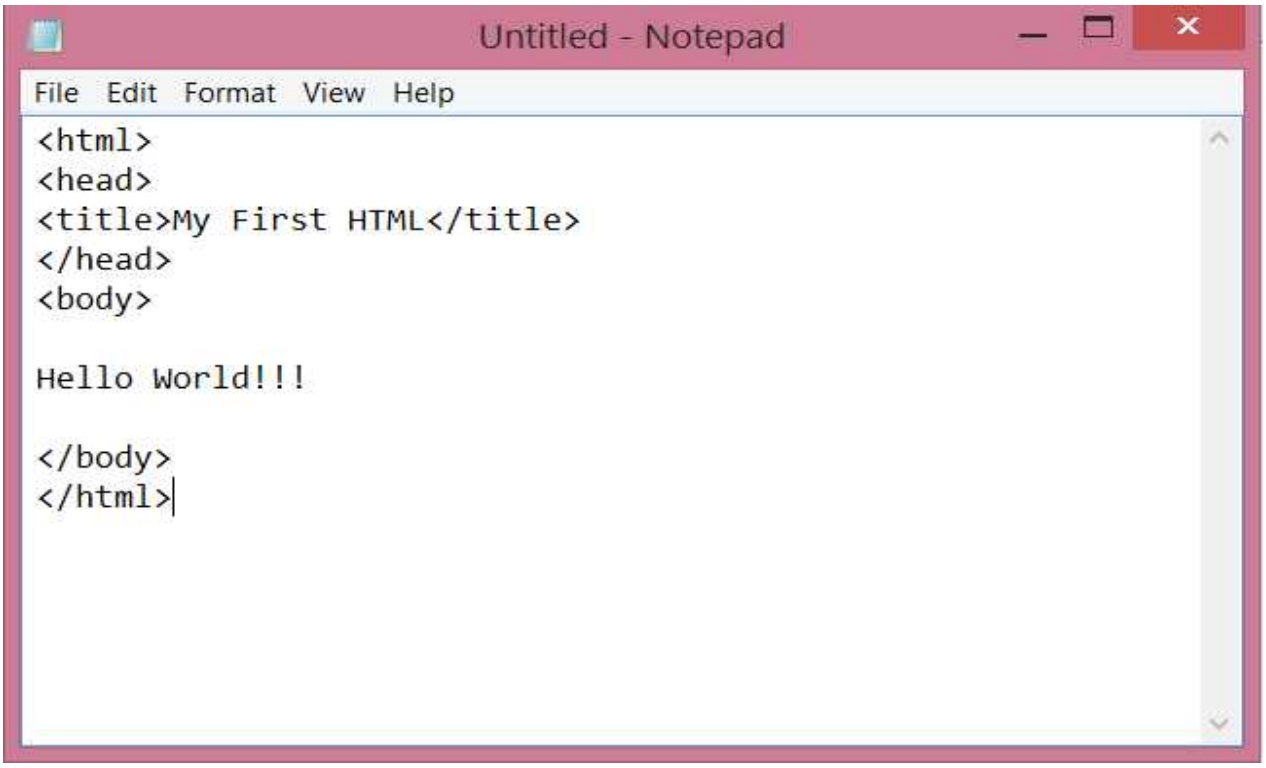


Figure 2.2: "Hello World" HTML Tags

3. Save the notepad file as "helloworld.html" in some folder location as shown in figure 2.3.

C H A P T E R 2

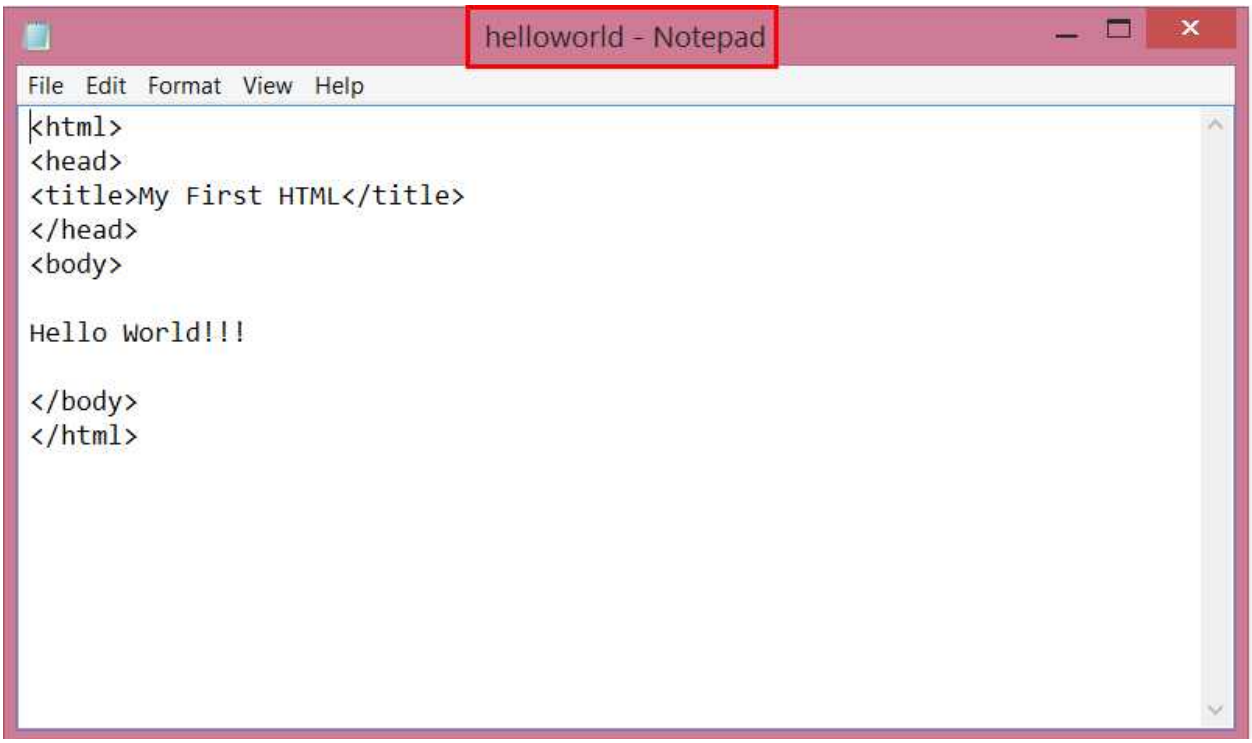


Figure 2.3: helloworld HTML

Here is the first basic lesson - all HTML files are saved with either ".html" or ".htm" extension.

4. How do we see our first HTML web page in web browser? Open Internet Explorer as shown in figure 2.4 (I am using Internet Explorer version 11. You can use other browser or version if you want.). The Internet Explorer is showing Bing map web site.

C H A P T E R 2

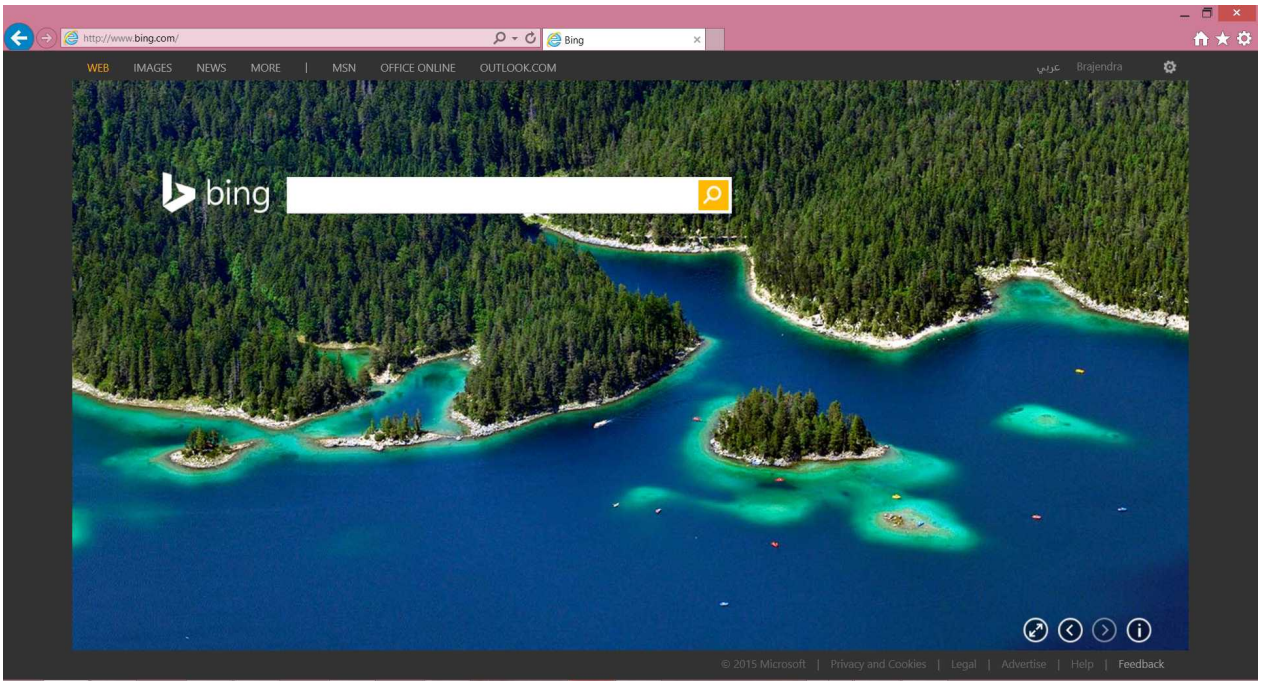


Figure 2.4: Internet Explorer

5. Go to the **Internet Explorer window** and press **Alt** button on keyboard. It will start showing **File menu** as marked in square in the figure 2.5 below.

C H A P T E R 2

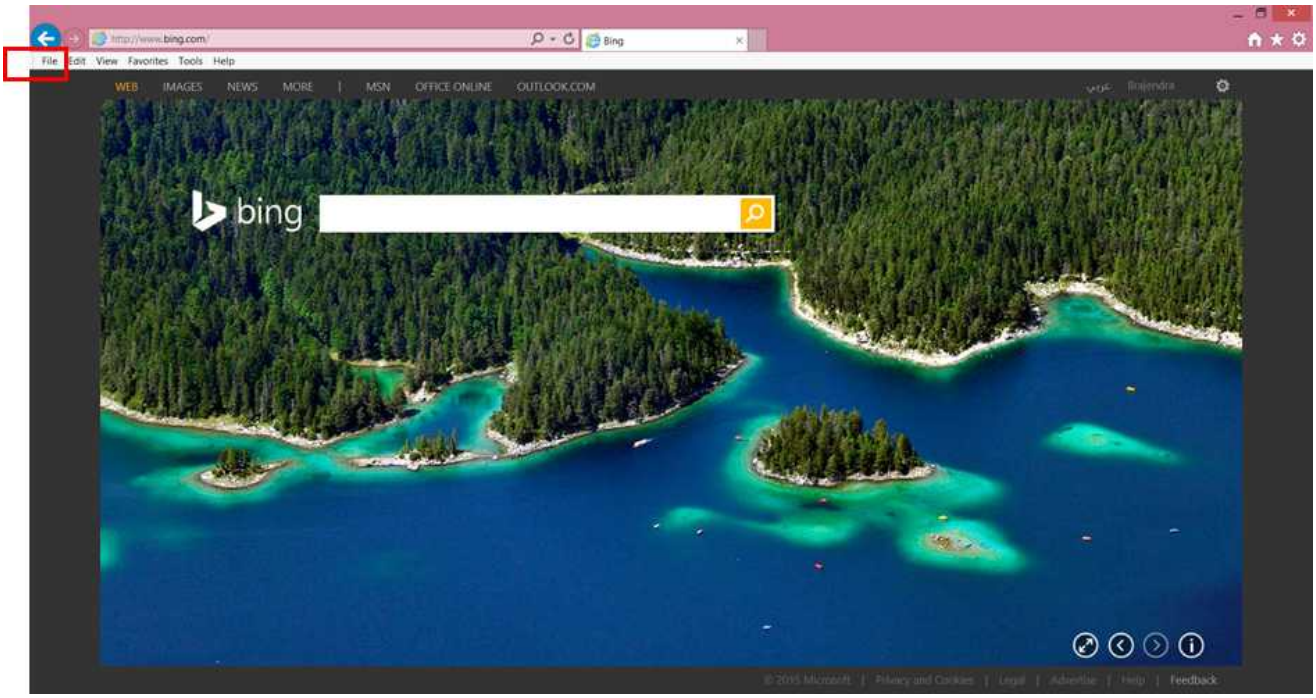


Figure 2.5: Internet Explorer File Menu

6. Click on **File** and then **Open** menu option as shown in figure 2.6 below.

C H A P T E R 2

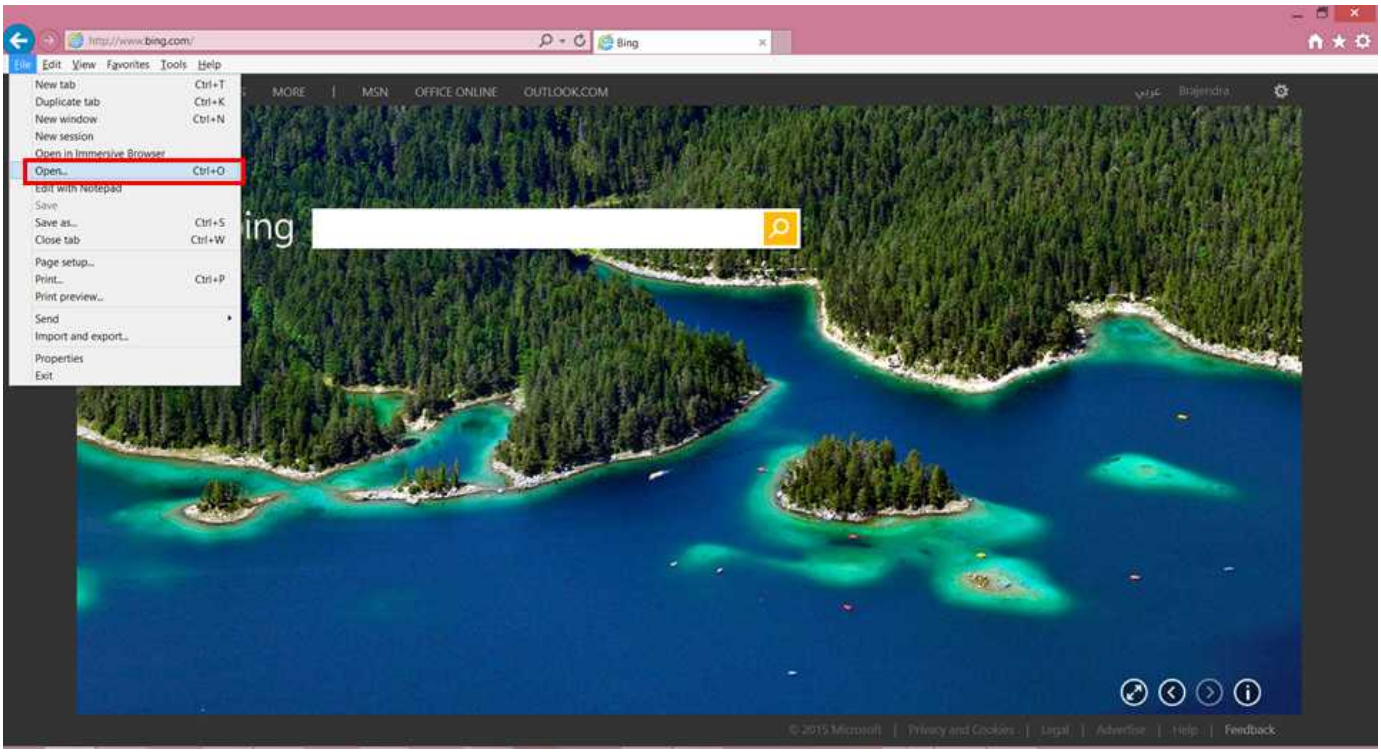


Figure 2 6: File >> Open Menu

7. It will pop up the **Open window** which is shown in figure 2.7.

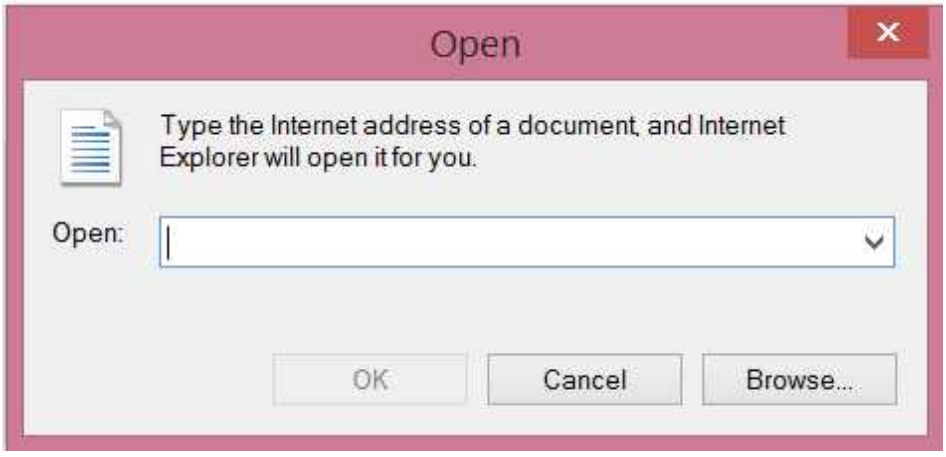


Figure 2.7: Open Window

8. Using **Browse… button** select **helloworld.html** file from the **folder** location you have saved earlier. I have used folder location "C:\books\HTML" for this demonstration. It is shown in the figure 2.8 below.

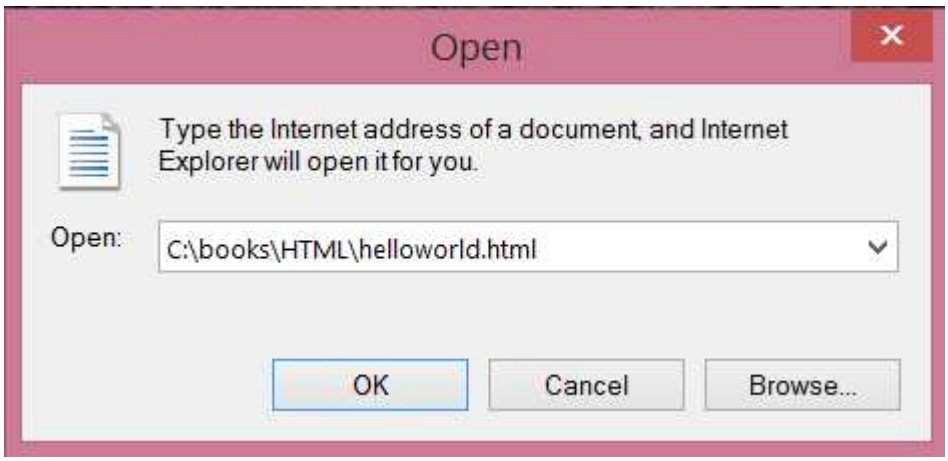


Figure 2.8: File Selected.

9. Click on OK button. It will show your first HTML web page in the Internet Explorer as shown in figure 2.9. See the Hello World message as you have typed in your HTML page.

C H A P T E R 2

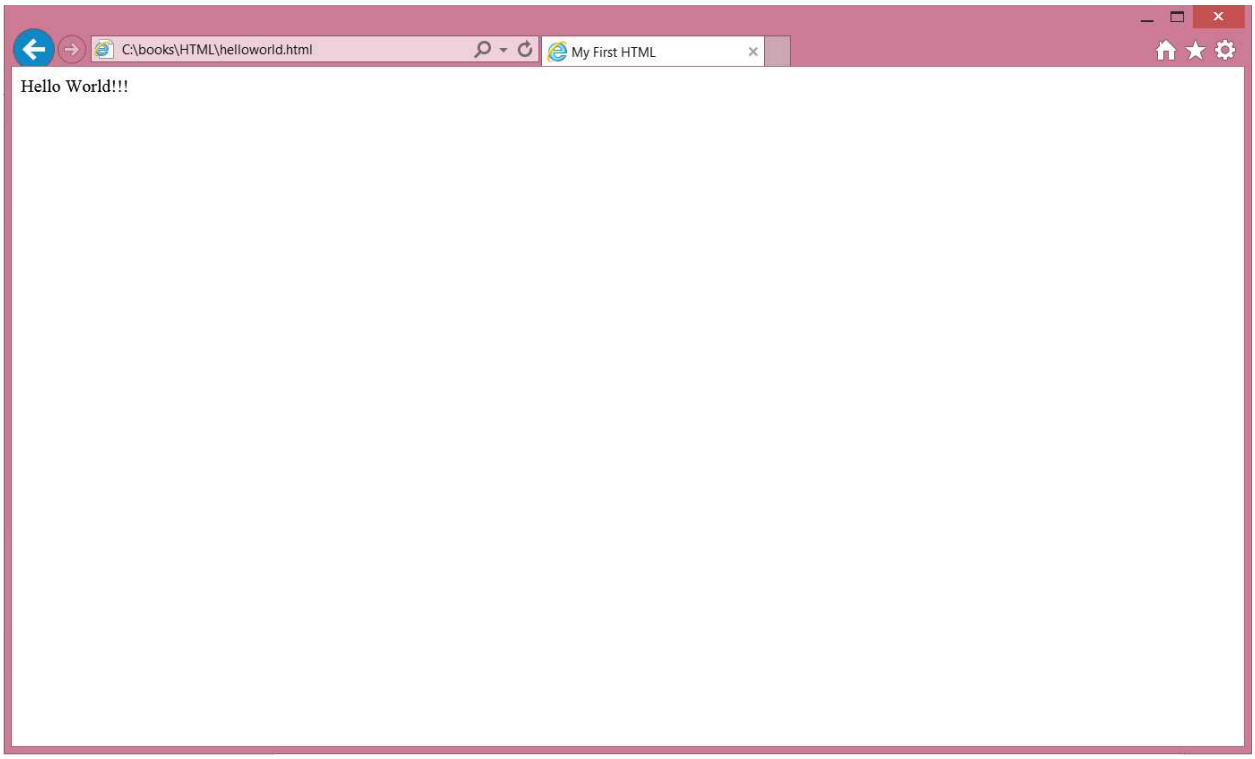


Figure 2.9: File Selected.

Congratulations! You have just written your very first HTML web page. How do you feel? Excited! There is more to come.

You should memorize and practice these steps of creating HTML in notepad and then viewing that in the web browser (Internet Explorer) because we will do this in every exercise as we move forward in other chapters.

Understanding HTML Structure

Now that you have written your first HTML web page, let's understand what it means.

As I mentioned earlier, *just like human body is made of cells; HTML is made of tags***.** Everything you see in the HTML example above is a **tag**. For instance <html>, <head>, <title>, <body>, </html>, </head>, </title>, </body>, everything is called a tag. Just like apple is called apple, you are called by your name, these things are called **tags**. Nice name, I must say. Each tag has a purpose in HTML or say on web page.

HTML is the language for web page; you learnt this in chapter one. Similar to any other language like English, HTML also has words and grammar. In this analogy, tags are like words. In English, you place words together using grammar to write some sentence that makes sense. Similarly in HTML, you place tags together using HTML grammar to write something which makes sense as a web page.

You will learn these tags or words in coming chapters. We will now study grammar of HTML in the remaining part of this chapter. Whenever we mention any particular HTML grammar rule, we will mark that with **HTML Grammar:** tip**.**

For instance, here is our very first rule.

C H A P T E R 2

HTML Grammar: Each HTML tag comes in pair. You will always have an opening tag like <tag> and a corresponding closing tag like </tag> . The opening tag is written inside < and > brackets. The closing tag is written between </ and > brackets.

Let's start with our very first tag.

<HTML> Tag

Take a look at first tag as shown below.

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

Hello World!

</Body>
</html>
```

You see an opening tag <html> and then closing tag </html>. <html> and </html> tags are like start and end points of the HTML file.

HTML Grammar: all HTML files must start with <html> tag and end with </html> file.

<HEAD> Tag

Next important tag in HTML file is <head> tag. It has been highlighted below.

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

Hello World!

</Body>
</html>
```

This tag in real world web pages is used to contain scripts, styles and <title> tag. Well, don't confuse with these new words. For now, we will keep it simple. What you need to know that it contains another important tag <title>. You will learn about <head> tag in more details when learning advanced styling and programming in HTML.

C H A P T E R 2

HTML Grammar: 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

What is <title> tag? It is an interesting and important tag. You can write some text between opening <title> and closing </title> tags. And whatever you write between these pairs will appear in the title of the browser window.

Here is the example of our very first HTML:

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

Hello World!

</Body>
</html>
```

Here is the result of the <title> tag in browser (Internet Explorer in this case) window as shown in figure 2.10. You can see text written between <title> and </title> tags are shown in the title of the Internet Explorer.

C H A P T E R 2

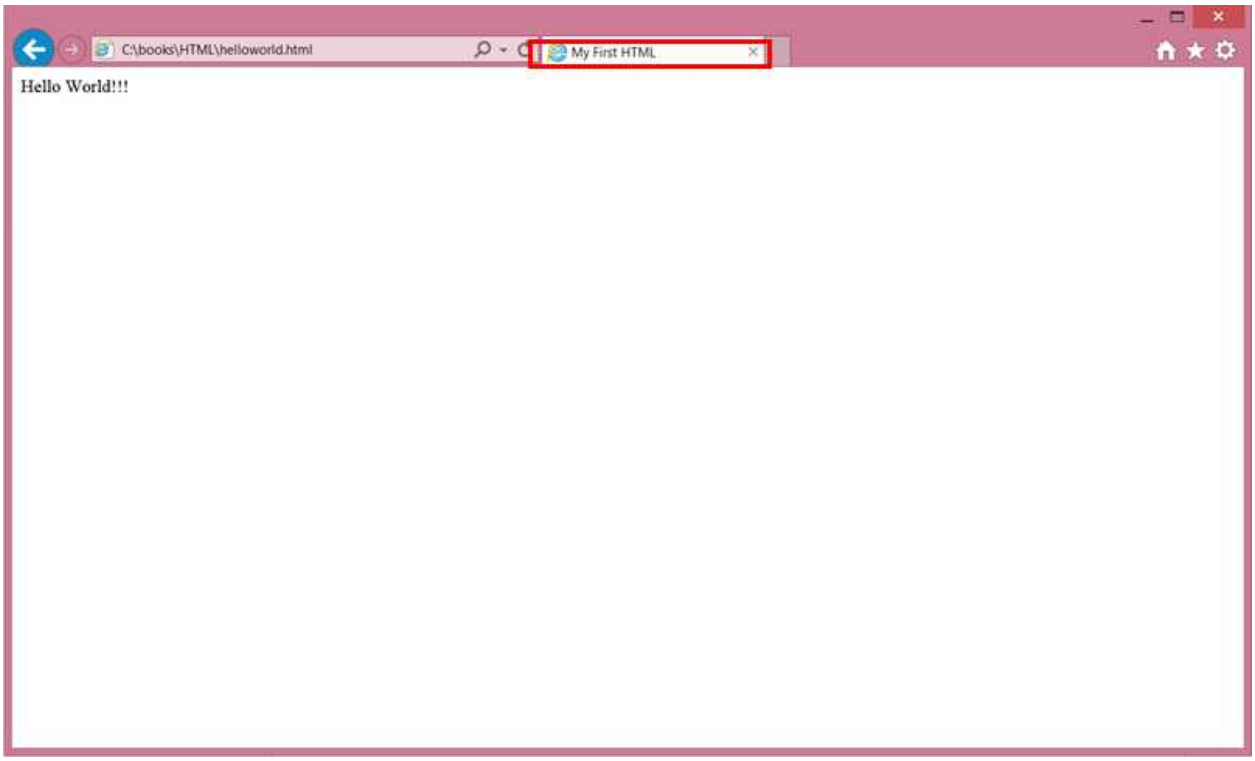


Figure 2.10: Internet Explorer Title

HTML Grammar: the <title> tags must be written inside <head> tags. It contains any text which is then shown in the title of web browser window.

HTML Grammar: The <title> and <head> tags are optional. It means, they can be inside HTML or they cannot. In coming 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.

In real life web sites, <title> tag is used to show summary or heading of the web page in the title of the browser window.

<BODY> Tag

The most important and primary tag in HTML is <body> tag. Whatever you write inside <body> tags; will be shown in the browser window. For instance, in our first page, we wrote Hello World!!! Inside <body> tag and it shown in the browser window (Internet Explorer).

The body tag has been highlighted in the HTML below.

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

<Body>

Hello World!

</Body>
</html>

The body tag content is shown in web browser as shown in the figure 2.11 below.

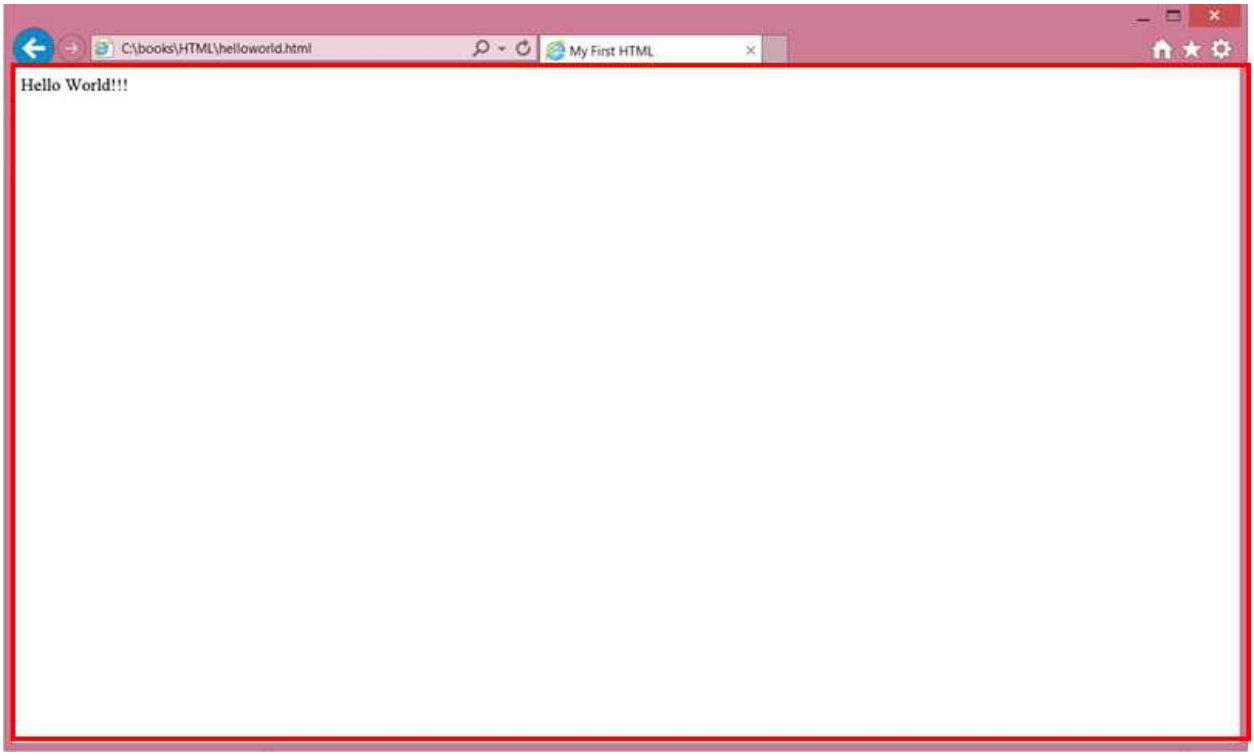


Figure 2.11: <body> Tag shown in browser

Let's take another example. Go to Google web site - https://www.google.co.uk/. You see I have squared a part of the web page as shown in figure 2.12. This squared part is made of various tags and all these tags are inside <body> and </body>.

C H A P T E R 2

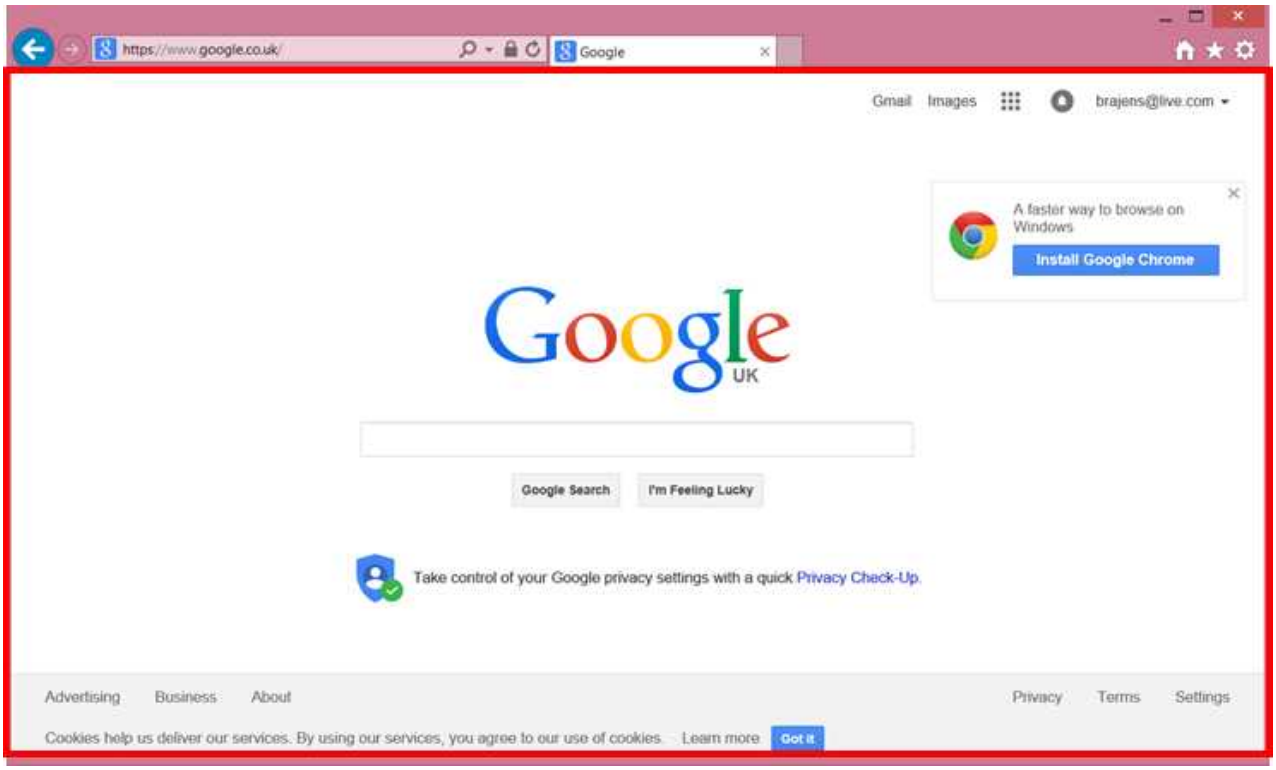


Figure 2.12: Google Site Body

Check the Figure 2.13 below. I have highlighted some of the types of tags which are used inside <body> and </body> to construct the web page.

C H A P T E R 2

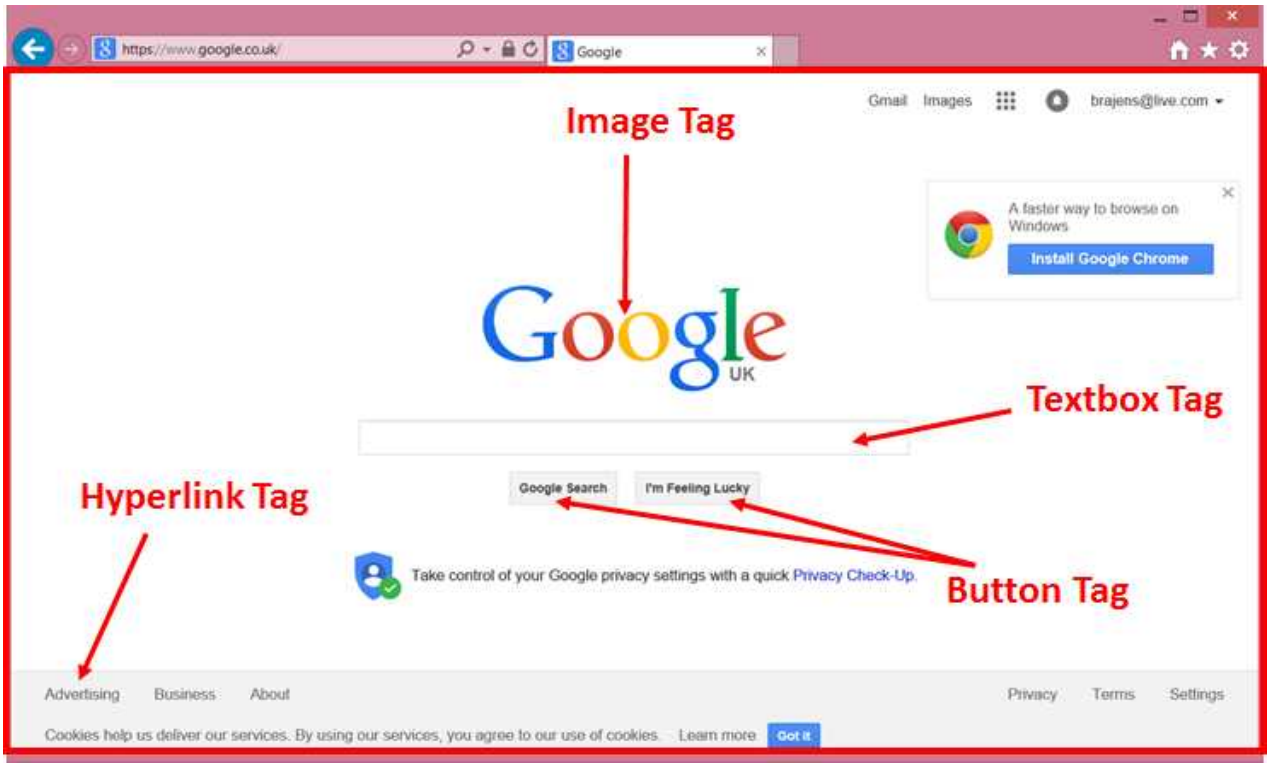


Figure 2.13: Google Site Body Tags

I have highlighted some tags but there are many more on this page. Each tag serves certain purpose on the web page. For instance, some are used to show Google logo, some to enter search text and some to be clicked to fetch search result.

You will learn about many such tags in coming chapters. The <body> grammar rule will always be applied. Every tag you learn will be written between <body> and </body> tags.

HTML Grammar: Opening <body> tag comes after <head> tag. The closing </body> tag comes before </html> tag. Every other tag you want to write in order to show your web page in the desired way is written between <body> and </body> tags.

When writing tag names, always use small English letters otherwise you might get errors. In web world, writing tags in small English letter is always appreciated as **best practice**. I am sure you want to be associated with **the best word.**

Sometimes tags are also referred as **elements** or **HTML elements**.

Attributes of the Tag

You learnt about tags and HTML structure. Now let's learn about **attributes** which are important part of the tags.

Attribute means quality. What is the purpose of attribute in real life? Let's take an example. When you see a red apple, you say that one of the quality (or say attribute) of this apple is that it is red. Similarly,

when you meet an intelligent classmate, you say that classmate has one quality or attribute and that is being intelligent.

What we realize is that the attributes provide additional details about someone or something.

Attributes play the same purpose in tags as well. They provide additional details about the tag.

Let's take an example of the <body> tag; Body is generally used for background image, background color, font size etc.

```
<body background="image of the background" bgcolor="color of the background"
id="name of the body>
```

The <Body> tag has the following attributes in this example as shown in the table 2.1 below.

Table 2.1: Body Attributes

Attribute	***What it tells about <body> tag?***
bgcolor	*Background color of entire web page body*
background	*Background image for the entire web page body*
Id	*It is used in searching a tag programmatically. You will learn more about this in coming chapters.*

Similar to <body> tag, all tags in the HTML have their own set of attributes. Some attributes like **id** are common across all tags. One tag can have more than one attributes. When you learn tags in coming chapters, you will also learn about their important attributes alongside.

Finally, let's see the structure of tag with attribute.

```
<tag attribute-name="attribute-value"></tag>
```

The attributes are presented as **attribute-name="attribute-value"** inside tags. All tags and attributes you learn in coming chapters will following the same structure.

Summary

Great, you now understand HTML structure, tags and attributes. You have fundamental concepts in place. It is time to start learning into specific tags, what purpose they solve, how to use and how they look on the web pages. Rest of the chapters in this book are focused on tags and related topics. Keep reading…

Chapter 3 – Heading and Paragraphs

We learnt about HTML fundamentals in the previous chapter. You now understand the structure of a tag in HTML web page. With this basic understanding, from now onward, we will learn about specific tags in terms of what they do on the page and how to use them.

In this chapter, we will focus on **Headings** and **Paragraphs** tags. What are "Headings" and "Paragraphs"? They are no different than what "Headings" and "Paragraphs" are used for in your books and note books. Headings are used to provide "title" of some topic. Paragraphs are used to provide written text for the topic. The purpose of "Headings" and "Paragraphs" tags in web pages is no different from how they are used in a book.

Examples of Heading and Paragraphs

Best way to learn and realize something is to "see" that thing in real life. We will now see the examples of headings and paragraphs in a web page.

Go to the following page in Wikipedia - https://en.wikipedia.org/wiki/London. I am actually looking for information on London in Wikipedia. You see examples of heading and paragraph as shown below in figure 3.1.

C H A P T E R 3

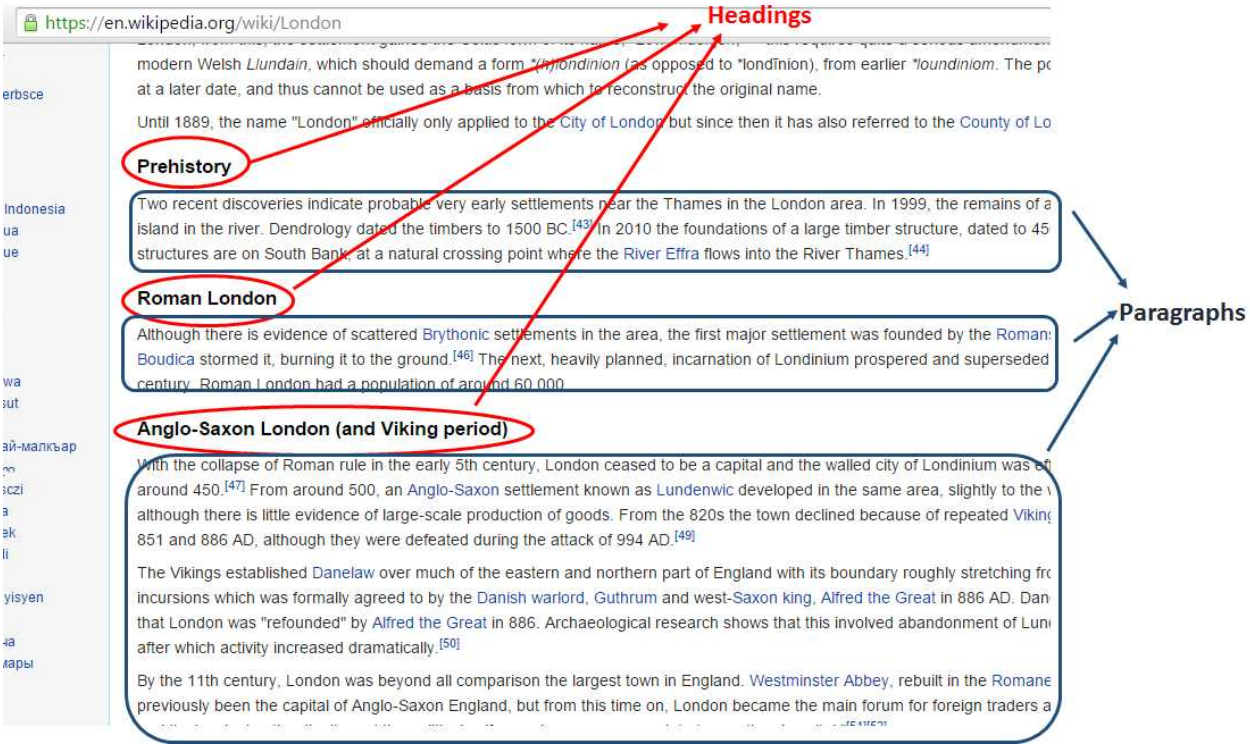


Figure 3.1: Headings and Paragraphs Example

You can see that headings has been used as title and paragraphs are used to provide long text details. For instance, Prehistory, Roman London etc. are titles which are then followed by paragraph of texts.

Let's learn how to make it happen in HTML page.

Learning the Details

Let's start with headings first. As we discussed earlier, headings are used to provide title.

You can write a heading using **<h1> to <h6>** tags. Figure 3.2 shows an example of heading using <h1> tag.

C H A P T E R 3

```
<html>
<body>

<h1>Heading for Chapter 4</h1>

</body>
</html>
```

Display in Web Page

Heading for Chapter 4

Figure 3.2: Example of h1

Similar to h1, you can also use h2, h3, h4, h5 or h6 tags. When the number increases in heading tag, the size of the text decreases. Here is the example of all heading tags as shown in figure 3.3.

```
<html>
<body>

<h1>Example of h1</h1>
<h2>Example of h2</h2>
<h3>Example of h3</h3>
<h4>Example of h4</h4>
<h5>Example of h5</h5>
<h6>Example of h6</h6>

</body>
</html>
```

Display in Web Page

Example of h1

Example of h2

Example of h3

Example of h4

Example of h5

Example of h6

Figure 3.3: Example of all heading elements

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

Is it not very simple to use? I am sure you will agree with me. Let's now move on to using paragraphs.

Paragraphs are used to provide long block of text. We use **<p>** tag to show paragraph on web page. The figure 3.4 shows a paragraph tag.

C H A P T E R 3

```
<html>
<body>

  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur
  justo lacus, elementum vel ornare quis, laoreet non diam. Duis id
  aliquam risus, sed dignissim metus. Duis feugiat ultricies tincidunt.
  Maecenas sed ultrices elit. Praesent egestas quam eros, eu suscipit
  risus  Sed aliquet metus in est feugiat egestas. Aenean bibendum,
  purus sed maximus lobortis, orci tellus porta felis, eu molestie elit
  elit a metus. Nam lobortis lectus vitae malesuada pretium. Nam laoreet
  mauris feugiat felis sodales consequat. Duis quis lacinia arcu.
  Pellentesque ultricies sem vel magna blandit ultricies. Vestibulum
  gravida nunc in quam egestas, id porttitor enim eleifend.posuere
  non.</p>

</body>
</html>
```

Display in Web Page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur justo lacus, elementum vel ornare quis, laoreet non diam. Duis id aliquam risus, sed dignissim metus. Duis feugiat ultricies tincidunt. Maecenas sed ultrices elit. Praesent egestas quam eros, eu suscipit risus Sed aliquet metus in est feugiat egestas. Aenean bibendum, purus sed maximus lobortis, orci tellus porta felis, eu molestie elit elit a metus. Nam lobortis lectus vitae malesuada pretium. Nam laoreet mauris feugiat felis sodales consequat. Duis quis lacinia arcu. Pellentesque ultricies sem vel magna blandit ultricies. Vestibulum gravida nunc in quam egestas, id porttitor enim eleifend.posuere non.

Figure 3.4: Paragraph Example

In the example above, you can see how long text block has been written between ** ** and ** ** tags to present the paragraph of text.

The real example you see when you club heading and paragraph tags together on the web page. Please see this example below as shown in figure 3.5.

```
<html>
<body>

<h2> Title1 </h2>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur
justo lacus, elementum vel ornare quis, laoreet non diam. Duis id
aliquam risus, sed dignissim metus. Duis feugiat ultricies tincidunt.
Maecenas sed ultrices elit. Praesent egestas quam eros, eu suscipit
risus  Sed aliquet metus in est feugiat egestas.</p>

<h2> Title2 </h2>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur
justo lacus, elementum vel ornare quis, laoreet non diam. Duis id
aliquam risus, sed dignissim metus. Duis feugiat ultricies tincidunt.
Maecenas sed ultrices elit. Praesent egestas quam eros, eu suscipit
risus  Sed aliquet metus in est feugiat egestas.</p>

</body>
</html>
```

Heading

Paragraph

Heading

Paragraph

Display in Web Page

Title1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur justo lacus, elementum vel ornare quis, laoreet non diam. Duis id aliquam risus, sed dignissim metus. Duis feugiat ultricies tincidunt. Maecenas sed ultrices elit. Praesent egestas quam eros, eu suscipit risus Sed aliquet metus in est feugiat egestas.

Title2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur justo lacus, elementum vel ornare quis, laoreet non diam. Duis id aliquam risus, sed dignissim metus. Duis feugiat ultricies tincidunt. Maecenas sed ultrices elit. Praesent egestas quam eros, eu suscipit risus Sed aliquet metus in est feugiat egestas.

Figure 3.5: Tile and Paragraph Together

It looks like a book chapter. Isn't it?

Finally one more thing to share. When you are writing texts and you want to start from next line. You can do so by inserting **
** tags between texts. Please see the example below as shown in figure 3.6.

C H A P T E R 3

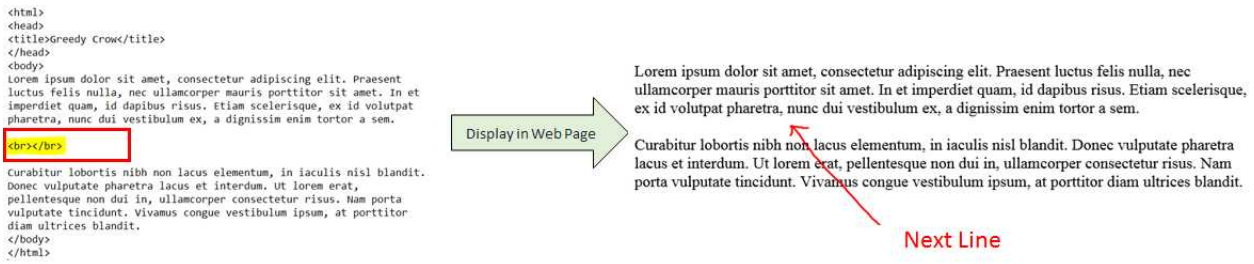


*Figure 3.6:
 Example*

You see that you can insert text directly into HTML page between <body></body> tags and then use <br> tag to insert next line.

Try it yourself

You now understand how to use heading and paragraph tags. I want to give you an exercise where you show a small story on a web page. You create a web page which shows the story in the format shown in figure 3.7.

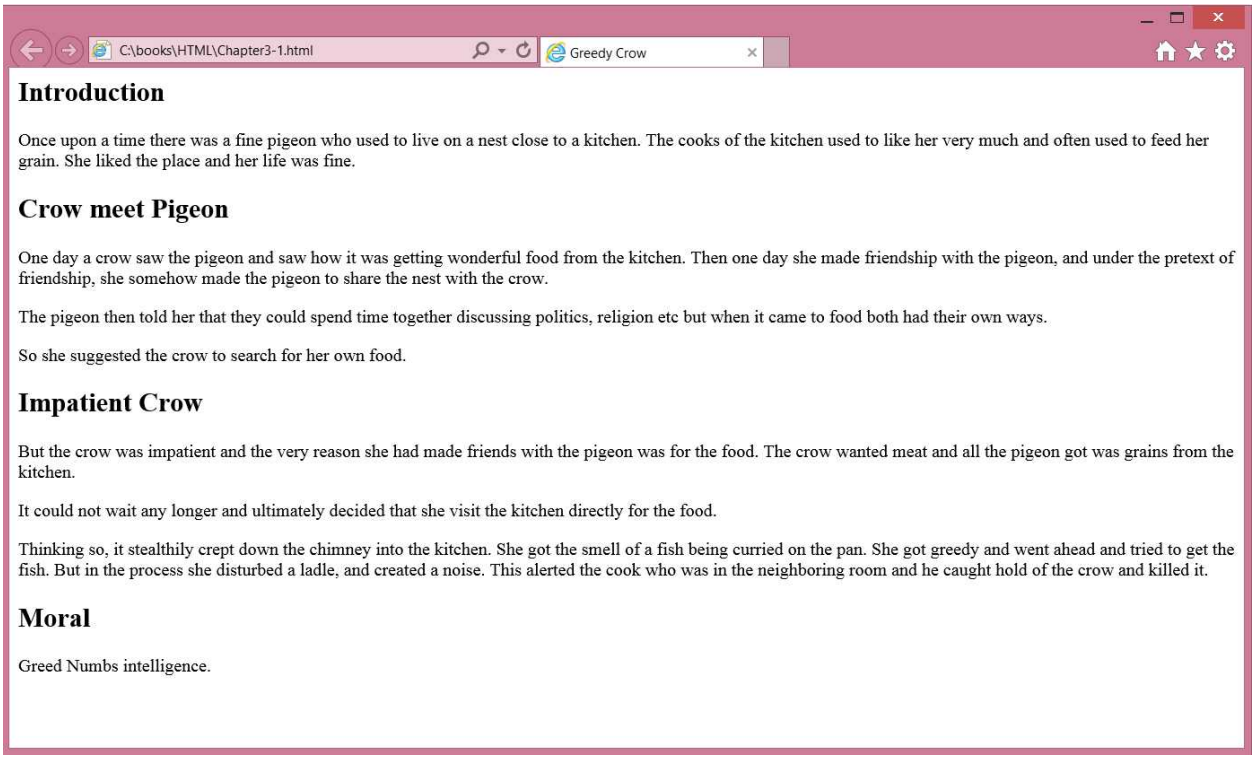


Diagram 3.7: Story Page for Exercise

Following are the guidelines to perform this exercise:

- Open Notepad
- Create HTML using heading and paragraph tags. Each heading tag will be followed by corresponding paragraph tag. Type in the text between the respective tags.
- Save the HTML file using .htm or .html extension
- Open the HTML file using web browser to see the result.

If you need any help, please check **Exercise1 – Greedy Crow** chapter to find HTML for the web page above.

Summary

Congratulations, you have just learnt using heading and paragraph tags. And not only that, you also created a web page to show a story on a web page. Are you getting ideas to build a web site with stories? Well, once you learn HTML in detail, realizing that dream is not far away. Till then, keep reading…

Chapter 4 – Formatting Text

In the previous chapter, you learnt about writing headings and paragraphs. I hope you enjoyed the exercise. We will continue with more texts in this chapter but will primarily focus on formatting the text.

What is **Formatting**? When reading any book or watching some text on any web site or reading some document in Microsoft Word, you must have observed that certain part of the text are shown as Bold (**Bold)** or as underlined (underline) or as Italic (*Italic*) to show the importance of the text. Making any text bold, underline or italic in web page is called "Formatting".

We will learn how formatting is used in HTML.

Examples of Formatting Text

Time to see samples of the formatting. Please go to the following page in Wikipedia - https://en.wikipedia.org/wiki/India . In this small paragraph shown in the figure 4.1 below, you can see examples of formatting:

C H A P T E R 4

Figure 4.1: Formatting Example

You find that certain part of the text or certain lines have been highlighted as bold and italic to show the importance of the text. Showing importance of the text is the primary use of formatting.

Learning the Details

It is time to learn details of the formatting. But before we start, let's see some common formatting example and how they look in the table below.

Table 4.1: Formatting Styles

Formatting Style	How it looks?.
Bold	**This is bold text.**
Italic	*This is Italic text.*
Underline	This is underline text.
Small	This is small text.
Subscript	This is$_{subscript}$ text.
Superscript	This is^{superscript} text.

There are more formatting methods available in HTML but we will learn these commonly used methods in this book.

Let's start with first method, making a text **bold**. To make a text bold, you write the HTML like following:

 your text here

It means, you place your texts between and tags to make them bold. for bold. It is simple. Isn't it?

So, what it takes to make text italic? Take a look here:

 your text here

How similar and again simple! Here stands for italic. Guess what, all other formatting also works in the same way in HTML. Take a look at the table below to understand how other text formatting works:

Table 4.2: Formatting Tags

Formatting	**HTML Tags**
Underline	<u> you text here </u>
Small	you text here
Subscript	_{you text here}
Superscript	^{you text here}

Interesting so far! Let's take the next step. Think about it, what it will take to make a paragraph look as shown in figure 4.2.

C H A P T E R 4

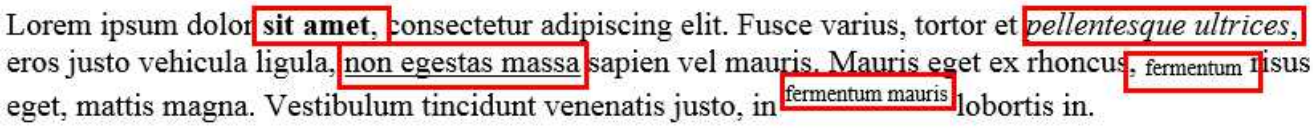


Figure 4.2: Mixed Formatting in Paragraph

You can see that different ways of formatting text has been mixed with normal text. How does it work? Look at the HTML tags below which created this kind of formatting.

```
<html>
<head>
<title>/<title>
</head>
<body>
<p>
Lorem ipsum dolor <b> sit amet </b> consectetur adipiscing elit. Fusce varius tortor et <i>
pellentesque ultrices </i>, eors justo behicula ligula, <u> non egestas massa </u> sapein
vel mautis. Mauris eget ex rhoncus, <sub> fermentum </sub> risus eget, mattis magna. Vestibulum
tincidunt venenatis justo, in <sup> fermentum mauris </sup> lobortis in.
</p>
</body>
</html>
```

Formatted texts has been mixed or placed with normal text in a paragraph tags. So easy to use. You can play with the text formatting the way you want.

C H A P T E R 4

Try it yourself

Time for another exercise. We will revisit the Greedy Crow story which we created in the previous chapter exercise. We will make use of that web page. But we want the story to look like as shown in figure 4.3:

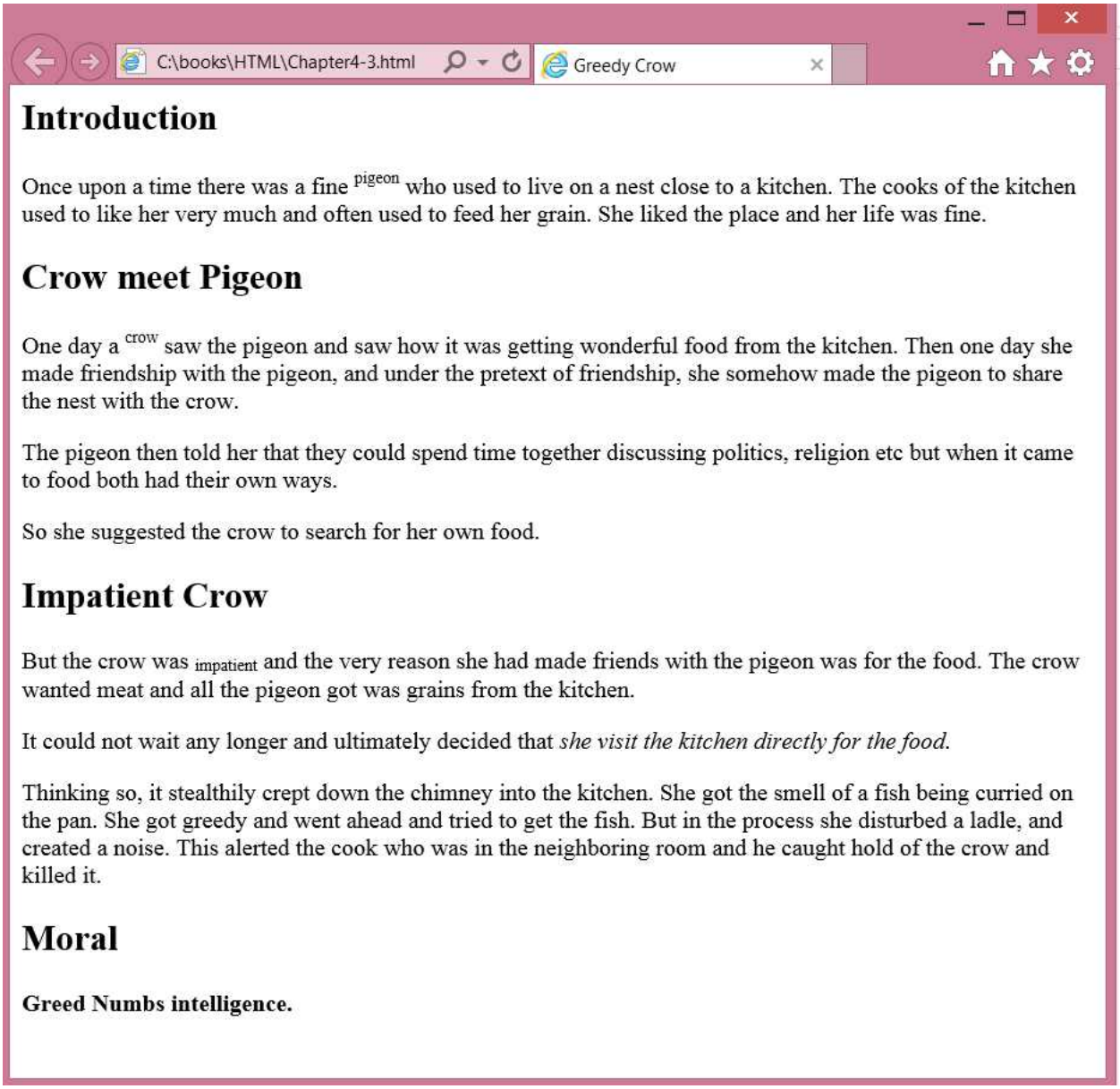


Figure 4.3: Formatted Story Page for Exercise

Following are the guidelines to perform this exercise:

- Open Greedy Crow web page in Notepad. This web page you created in the previous exercise.
- "Save As" this file with a name (say "Greedy_Crow_Formatted.html").

- In saved copy, insert your formatting tags within paragraph texts as per formatting required in the figure 4.3.
- Save the HTML file once the formatting tags are all inserted.
- Open the HTML file using web browser to see the result.

If you need any help, please check **Exercise2 – Greedy Crow Formatting** chapter to find HTML for the web page above. Best of luck!

Summary

Congratulations for another chapter. You must now be getting hang of HTML in creating web pages. HTML is incredibly simple yet very powerful in creating web sites. I hope your interest is increasing and also your appetite to learn more. So, keep reading…

Chapter 5 –Hyperlinks

Welcome back friends! In the previous chapters, you learnt about formatting the text. You must have realized how simple but powerful HTML is to create any kind of content such as story, news or any write up you want to share on web site.

We will now learn one of the most important element in HTML, it is called **Hyperlink**. In first two chapters, you learnt that any web site is a collection of web pages. How do you move from one page to another in a web site? You generally move from one page to another page in web site using Hyperlinks. You click on a hyperlink on a web page and it takes you to another web page and so on. In HTML world we say, we **navigate** from one page to another using Hyperlink.

In this chapter, we will learn how to use hyperlink. Trust me, it is very simple.

Examples of Hyperlinks

We will take an example which everyone have seen. I guess almost everyone!

Go to web site - https://www.google.co.uk/ . Please don't tell me you have not used google earlier. On this google web site, the following diagram 5.1 shows some of the examples of hyperlinks:

C H A P T E R 5

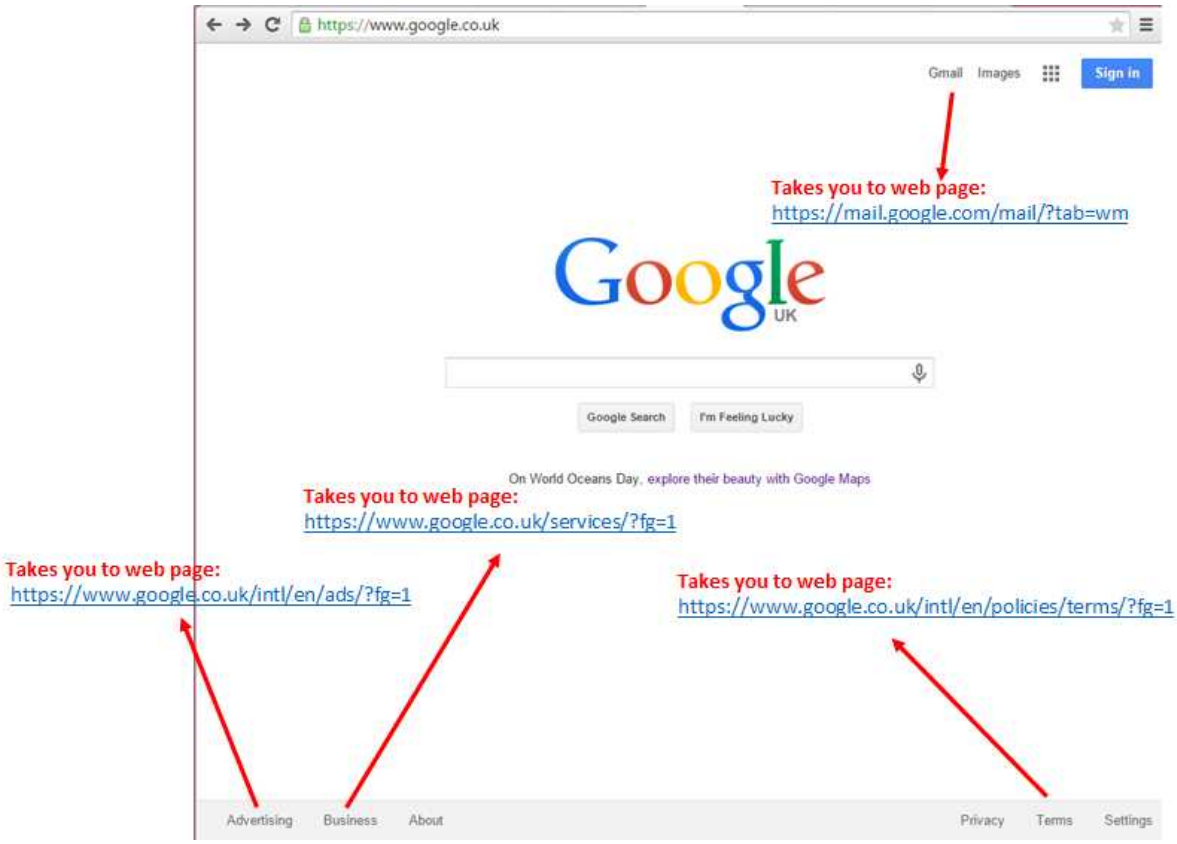


Figure 5.1: Hyperlink Example

There are more hyperlinks on this web page, but I showed some of them to make it simple to understand. When you click on a hyperlink, it takes you or say rather navigates you to the web page shown in the diagram. You can see that using hyperlink, you are able to go to various web pages which are part of Google web site.

In the example above, Advertising, Business, Terms, Gmail display are called "**Text**" to the hyperlinks. Text shows a display text for hyperlink on web page; something which people can read to know the purpose of hyperlink. The web page where hyperlink takes you when you click is called "**Address**". I suggest you memorize and understand both of these terms as they are fundamental to hyperlinks.

Time to learn how to use hyperlink.

Learning the Details

Let's start with an example. Suppose, you want to create a hyperlink which should take you to https://www.google.co.uk/ web page and should show text as "Google Search". It means, the hyperlink should display text as "Google Search" on web page (like, Advertising, Business, Terms, Gmail in example above). The address is https://www.google.co.uk/.

How do you write such hyperlink, take a look:

```
<html>
<head>
<title>/<title>
</head>
<body>
<a href="https://www.google.co.uk"> Google Search </a>
</body>
</html>
```

Hyperlinks are presented by HTML tag **<a>.** Between <a> and tags, you write **Text** of the hyperlink. **Text** is the display text of the hyperlink. You provide **address** of hyperlink in the attribute "**href**". The purpose of the **Address** is to tell the place where web page is located.

This hyperlink will look like as shown in figure 5.2 in the HTML page:

Figure 5.2: Google Search Example

It is simple. Right?

Let's take another interesting example. What will happen if I enter "#" as address in hyperlink like following?

```
<a href="#"> My Link </a>
```

You try yourself. You will find that the hyperlink will not take you anywhere but keep you on the same page where hyperlink is present. Funny, isn't it?

Try it yourself

In this exercise, we will create a web page which will be **master search page**. From this page you can navigate to Google search, Bing search, Yahoo search and Wikipedia search. All major search web sites from one web page. The web page will look like as shown in figure 5.3.

C H A P T E R 5

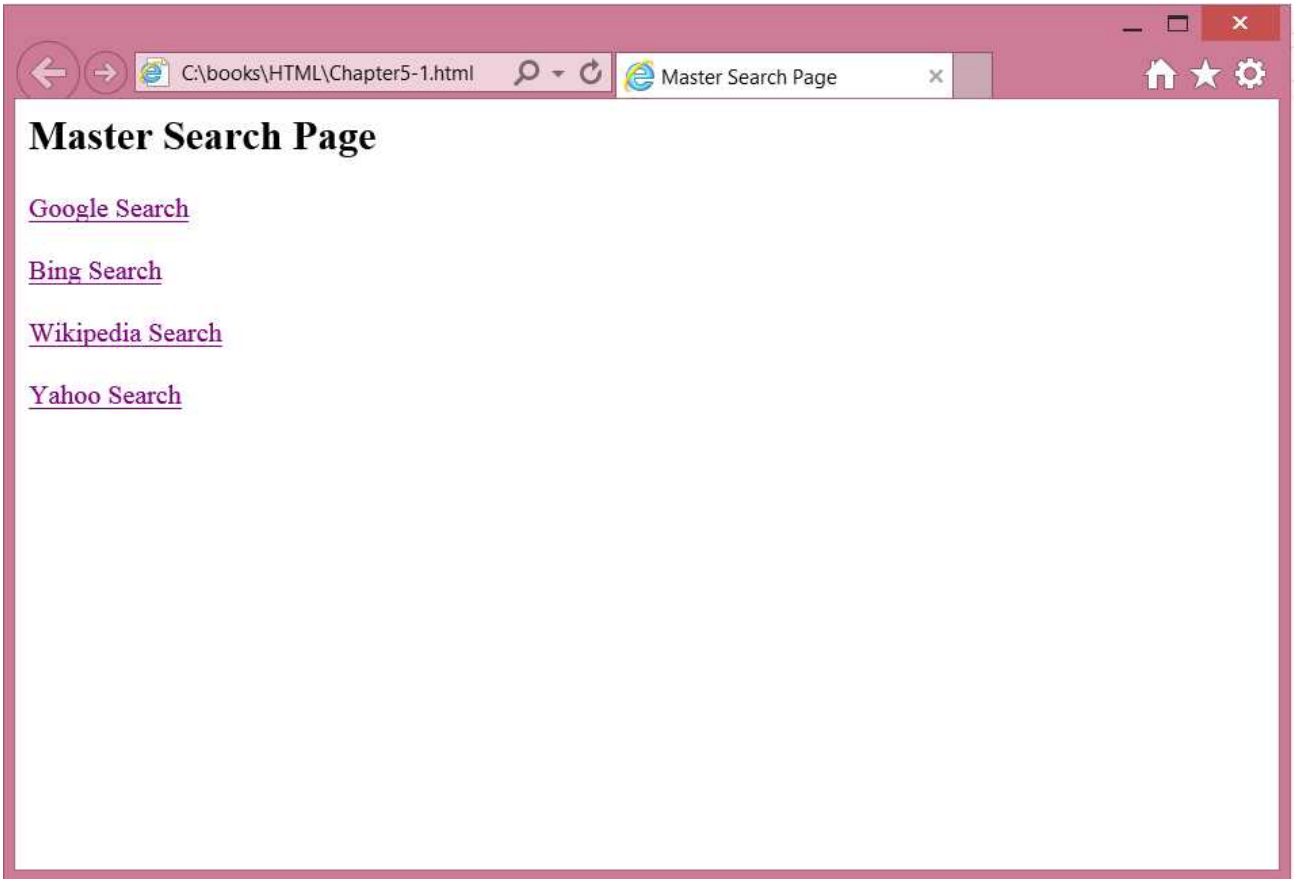


Figure 5.3: Master Search Page for Exercise

When you click on these hyperlinks, it should take you to individual search pages. The following table shows address of these hyperlinks which you can use in your web page creation.

Table 5.1: Hyperlink Addresses

Hyperlink Text	***Address***
Google Search	https://www.google.co.uk/
Bing Search	https://www.bing.com/
Wikipedia Search	http://en.wikipedia.org/wiki/Searching
Yahoo Search	https://uk.search.yahoo.com/

Following are the guidelines to perform this exercise:

- Open Notepad
- Create HTML using heading and hyperlink tags. For each hyperlink tag, use the text and address as given in the table 5.1.
- Save the HTML file using .htm or .html extension
- Open the HTML file using web browser to see the result.

If you need any help, please check **Exercise3 – Master Search Page** chapter to find HTML for the web page above. Best of luck!

Summary

Bravo, one more chapter down. Double Bravo because you finished one of the most important concept in HTML. Big web sites are not possible without hyperlinks and you have just learnt fundamentals of the same. Keep the appetite up and keep reading…

Chapter 6 –Images

You learnt about hyperlink in the previous chapter. Hyperlink is one of the most important element in HTML because it provides a way to navigate between the web pages. Let's move on to something visual now. What about **Images** or **Pictures**? There is a famous quotation that **"*picture says a thousand words***". Quotation shows how powerful pictures are in terms of delivering a message. Web sites show pictures to make web pages more exciting, optical and lively.

How do we show image or picture on any web page? This chapter will try to answer these questions.

Examples of Images

Let's see an example of picture on web page. Please go to the following page in Wikipedia - https://en.wikipedia.org/wiki/Albert_Einstein . You see images as shown below in figure 6.1:

CHAPTER 6

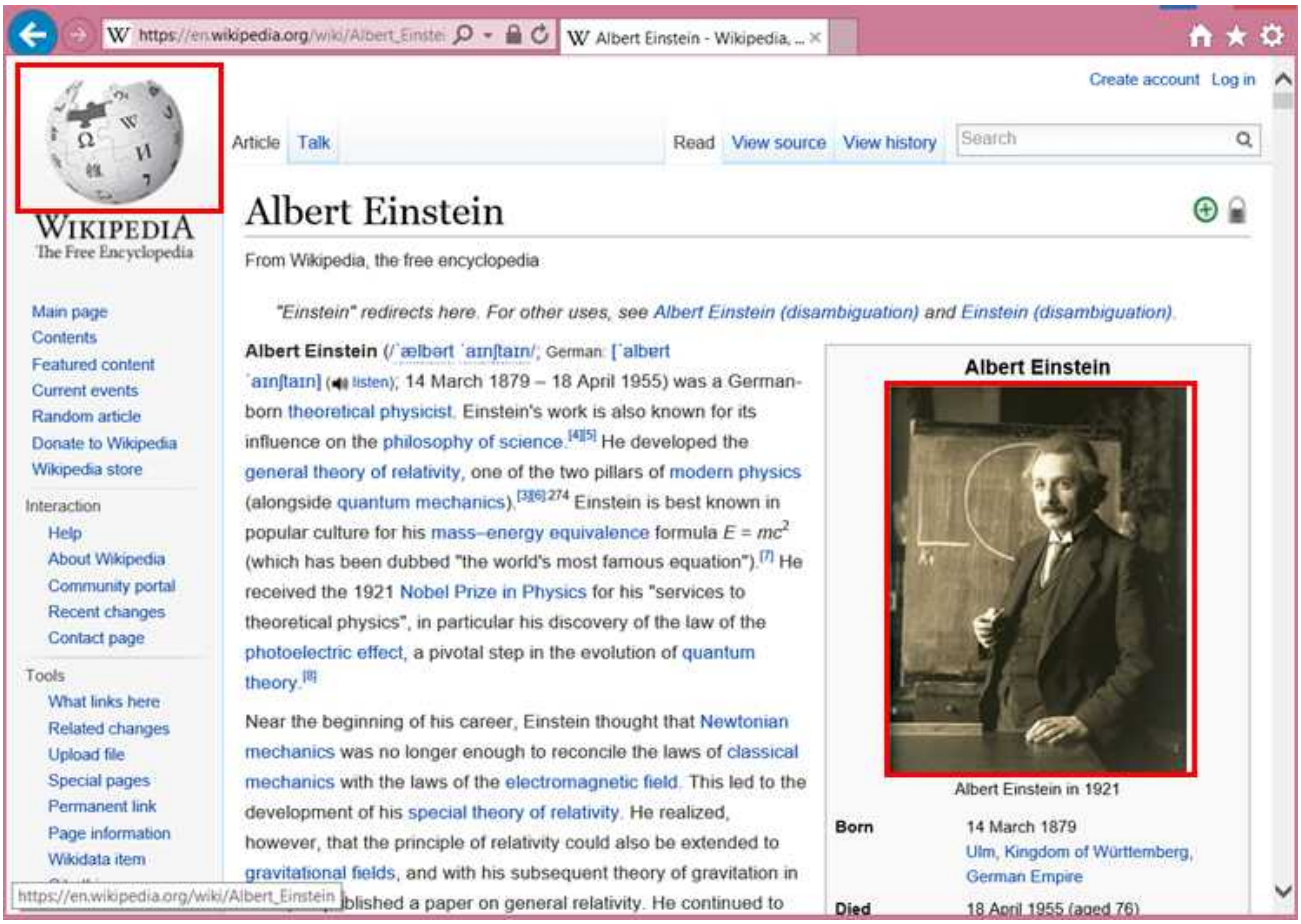


Figure 6.1: Images Example

You can see that I have marked two images – one is of **Albert Einstein** and other is of **Wikipedia Logo**. Don't you agree that these images have made this page lively?

Images on the web page can be used in two ways:

- ✓ One way is just show image. Just show; nothing else.
- ✓ Other way is to use image as hyperlink. It means when you click on the image, it will take you to some other page. How interesting!

We will learn both ways in this chapter.

Learning the Details

How do we show image on web page? We use **<img>** tag for this purpose. Please check the example below:

CHAPTER 6

```
<img src="location of the image" alt="alternate text" width="width in
pixel" height="height in pixel"></img>
```

We start <img> tag and then we close with </img> tag like we do for any HTML tag. But then we use various other attributes to make sure image is shown on the web page. Let's look into these attributes in the table below.

Table 6.1: Attributes of the <img> tag

<img> tag attribute	Purpose
src	**src** stands for **source**. Like hyperlink, you enter address of the image for this attribute. Address is the location where this image has been stored. The address could be internet based image. It means image is stored at internet somewhere like any web page. Example is Albert Einstein image stored at Wikipedia. Address is: https://upload.wikimedia.org/wikipedia/commons/thumb/3/3e/Einstein_1921_by_F_Schmutzer_-_restoration.jpg/220px-Einstein_1921_by_F_Schmutzer_-_restoration.jpg Or it could an address from your computer/laptop hard disk. It means the image is stored somewhere in your local computer hard disk. You provide folder location of the image. Example is - "**file://C:/ Books\HTMLforKids/mybook.jpg**" You can use most types of images such as **jpg**, **jpeg**, **png**, **gif** and **bmp**.
alt	**alt** stands for **alternate text**. This is the text shown on web page if image could not be shown on web page due to some reason. It is like a backup plan. You want to show some text in case image could not be shown due to some problem.
width	It is the width of the image. If you don't provide this value then it will show image in its original width. Width is measured in pixel. One pixel is like a dot on the screen. Just like you measure distance in meter or centimeter; in the same way you measure length of the image in pixel.
height	It is the height of the image. Similar to width, it is also measured in pixel. If you don't provide this value then it will show image in its original height.

Let me show you image in HTML. I am going to use Albert Einstein image address given in the table above and will show that in my HTML page. I will make height 200 pixel and width also 200 pixel; like square. The HTML looks like following:

```
<html>
<head>
<title></title>
</head>
<body>
<h2>Albert Einstein</h2>
<img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/3/3e/Einstein_1921_
by_F_Schmu tzer_-_restoration.jpg/220px-Einstein_1921_by_F_Schmutzer_-
_restoration.jpg" alt="Albert Einstein" width="200" height="200"></img>

</body>
</html>
```

The HTML page will be displayed as shown in figure 6.2 below.

Albert Einstein

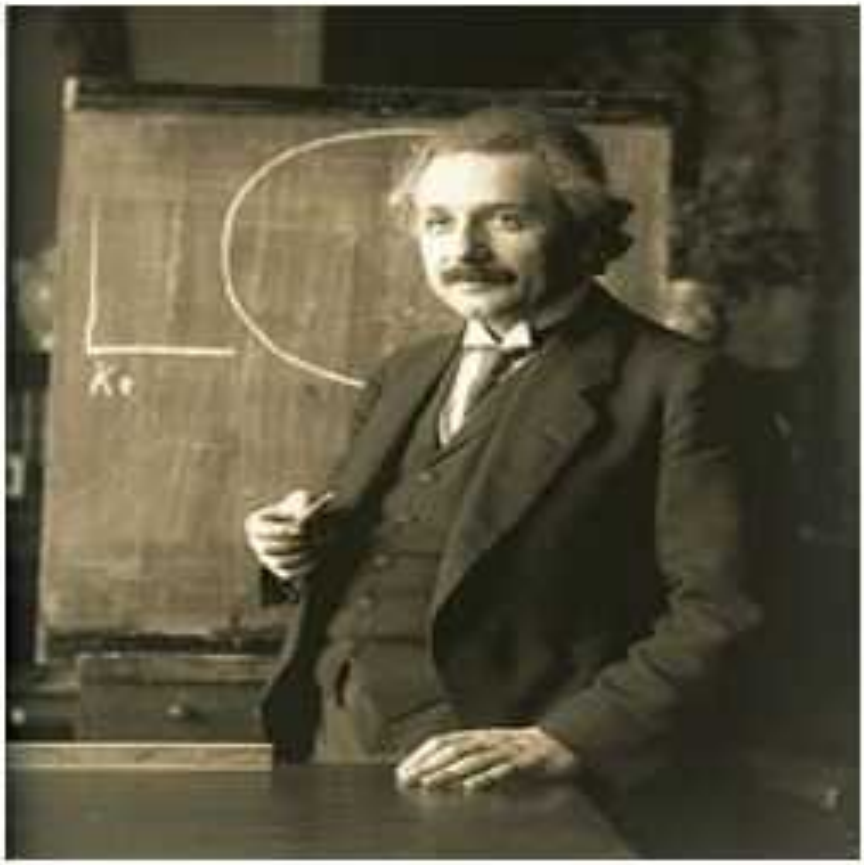


Figure 6.2: Web Page Display

You see it shows image in the square size (200 x 200) as I have mentioned in <img> tag attribute.

What will happen if some problem comes with image? For example, you give wrong address to image. In case of any problem, the web page will look like as shown in figure 6.3. I suggest you try it yourself.

Figure 6.3: Problem with Image

You observed it right. It is showing Alternate Text as defined in <img> tag attribute in place of the image.

Let's move next! What about using image as hyperlink? Sounds great! Right? Following is the structure of hyperlink tag (recall from the previous chapter):

```
<a href="Address of web page"> text of hyperlink </a>
```

To make image as hyperlink, you will replace **text of hyperlink** with <img> tag. For example, if I want to make image of Albert Einstein hyperlink which will take me to Wikipedia page, then my HTML <a> tag will look like the following:

```
<a href="https://en.wikipedia.org/wiki/Albert_Einstein"> <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/3/3e/Einstein_1921_
by_F_Schmu
tzer_-_restoration.jpg/220px-Einstein_1921_by_F_Schmutzer_-_restoration.jpg"
alt="Albert Einstein" width="200" height="200"></img> </a>
```

Is this not simple and intuitive? I am sure you will agree with me.

Try it yourself

Time to ignite web page with some image. Today, I am giving an exercise where you will use heading tag, paragraph tag and also image tag. Make **Isaac Newton web page** which looks like the figure 6.4.

C H A P T E R 6

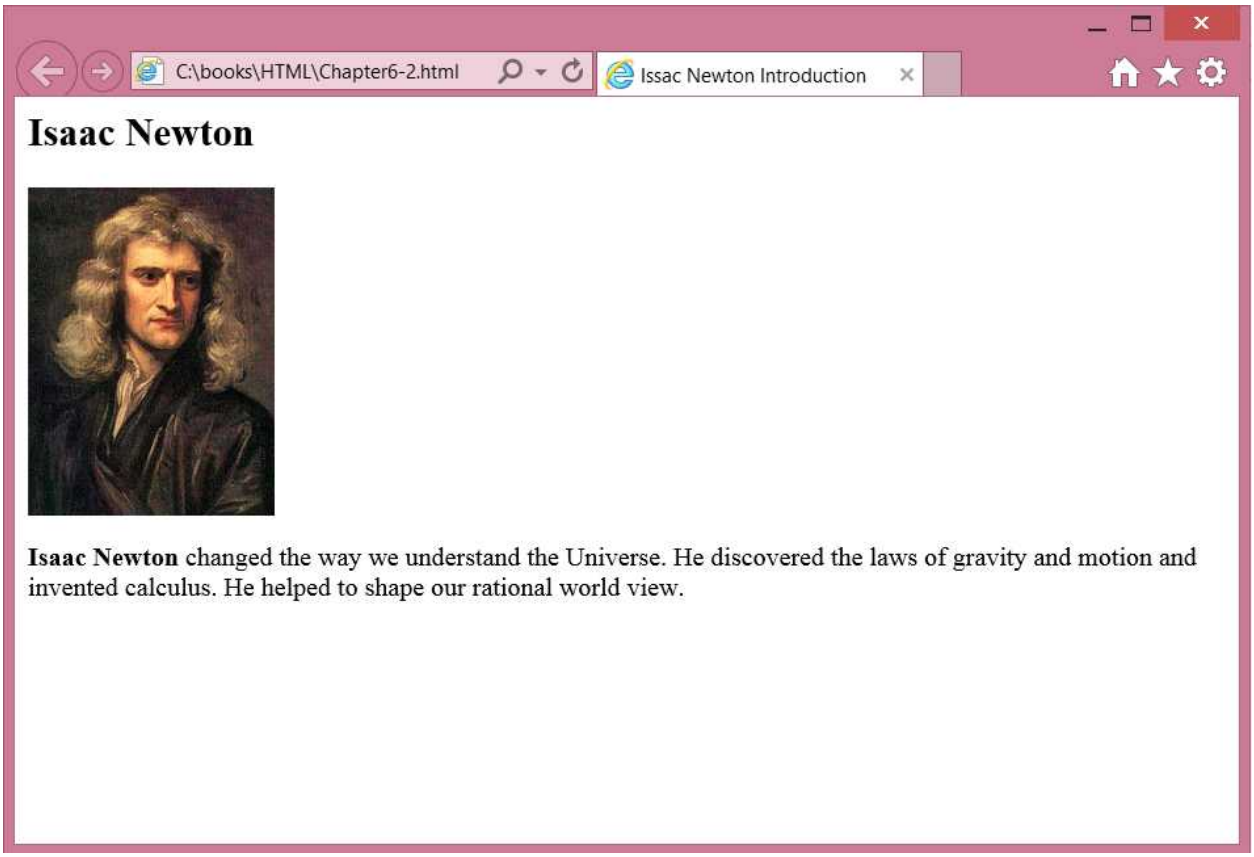


Figure 6.4: Image Page for Exercise

Please use this URL (**https://upload.wikimedia.org/wikipedia/commons/thumb/3/39/GodfreyKneller-IsaacNewton-1689.jpg/220px-GodfreyKneller-IsaacNewton-1689.jpg**) to show the image of Sir Newton in the exercise. I want the height to be 200 pixel and width to be 150 pixel.

Following are the guidelines to perform this exercise:

- Open Notepad
- Create HTML using heading, image and paragraph tags. The address or location of the image has been provided in the earlier paragraph. Provide height and width of the image in the <img> tag. Write text about Sir Newton in paragraph tag.
- Save the HTML file using .htm or .html extension
- Open the HTML file using web browser to see the result.

If you need any help, please check **Exercise4 – Isaac Newton** chapter to find HTML for the web page above. Best of luck!

C H A P T E R 6

Summary

Great, you finished one more chapter. Hope you liked it. I love images and hyperlinks on web pages because they make web sites so interactive and sparkling. I hope you echo the same. Keep reading…

Chapter 7 –Lists

Welcome! Hope you enjoyed playing with images in the previous chapter. That is fascinating! Isn't it? Let's come back to creative writing. Ok, tell me, how many times your English teacher has said that if you have more than one options to write about; use bullet points to make it emphatic and clear. Writing your options as bulleted text makes it more readable and presentable.

Lists are used in HTML to create bullet points based text on web pages. The purpose is the same, make your writing emphatic, readable and presentable.

Let's learn about it.

Examples of Lists

We will use Wikipedia again to see example of List. Please go to the following page in Wikipedia - https://en.wikipedia.org/wiki/Main_Page . You see lists as highlighted in marking in the figure 7.1 below.

C H A P T E R 7

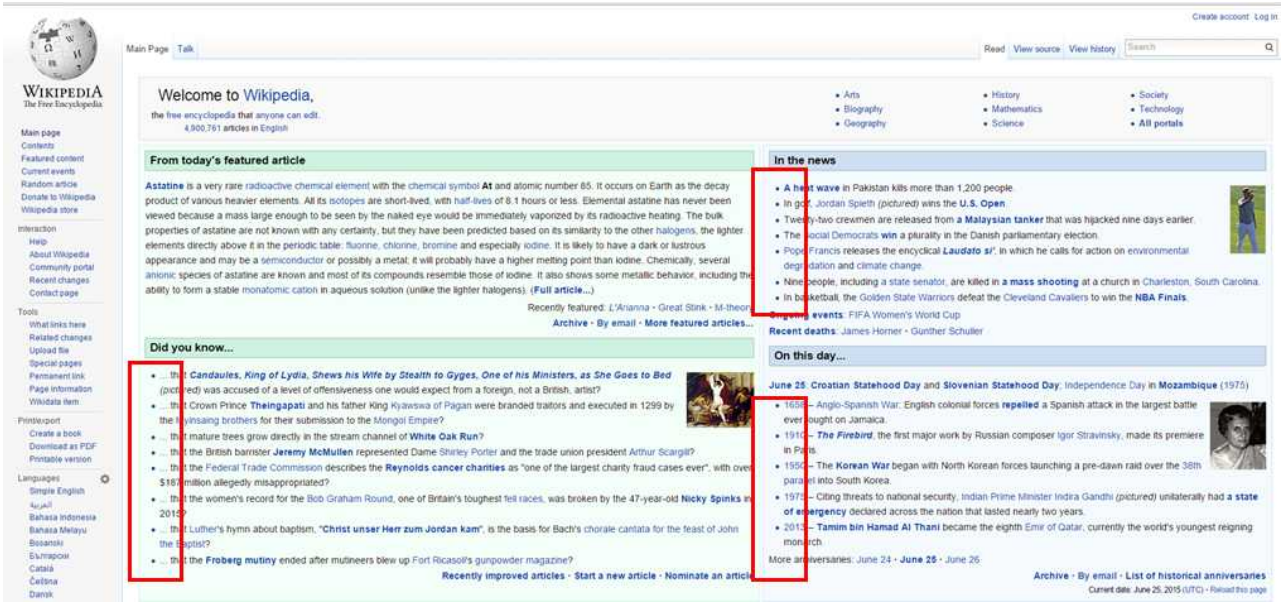


Figure 7.1: Lists Example

You can see that there are three lists or bulleted text present on the web page to show news, important events on the day and some facts. Bulleted text are making the page more readable and organized.

Time to learn the details.

Learning the Details

There are three types of lists present in HTML. The following table presents their names and shows how they look:

Table 7.1: Types of Lists in HTML

Type	Purpose	How it looks?
Unordered List	Shows text lines marked as bullets	• Chapter 1 • Chapter 2 • Chapter 3 • Chapter 4
Ordered List	Shows text lines which are marked with number or alphabets or roman numerals	1. Chapter 1 2. Chapter 2 3. Chapter 3 4. Chapter 4

Description List	Shows as ordered list and also provides description of each text line. This is particularly important when you want to provide a short text line and detailed description together in a list.	Chapter 1 - Introduction Chapter 2 - "Hello" HTML Chapter 3 - What are Elements and Attributes? Chapter 4 - Headings and Paragraphs

If you check back the Wikipedia page example given above, Wikipedia page is showing **Unordered List.**

Let's look into each type of list in detail. We will start with unordered list. We use <ul> and <li> tags to create unordered list. Here is the structure of these tags.

```
<ul>
      <li>text line one</li>
      <li> text line two</li>
      <li> text line three </li>
      <li> text line four </li>
</ul>
```

UL means **unordered list**. **LI** means **list item**. You provide individual text line items under <li> tags. The <li> tags are then listed below <ul> tag. You can add as much <li> items as you want.

What was HTML which showed the example above in the table? Here it is as shown in figure 7.2.

Figure 7.2: Unordered List Example

Simple. Isn't it?

Let's check ordered list. Here is the structure of the ordered list tags:

```
<ol>
      <li>text line one</li>
      <li> text line two</li>
      <li> text line three </li>
      <li> text line four </li>
</ol>
```

How predictable? By the way, **OL** stands for **Ordered List**. **LI** has the same name and purpose.

Let's see how HTML looks like for the ordered list example given in the table above. It is shown in figure 7.3.

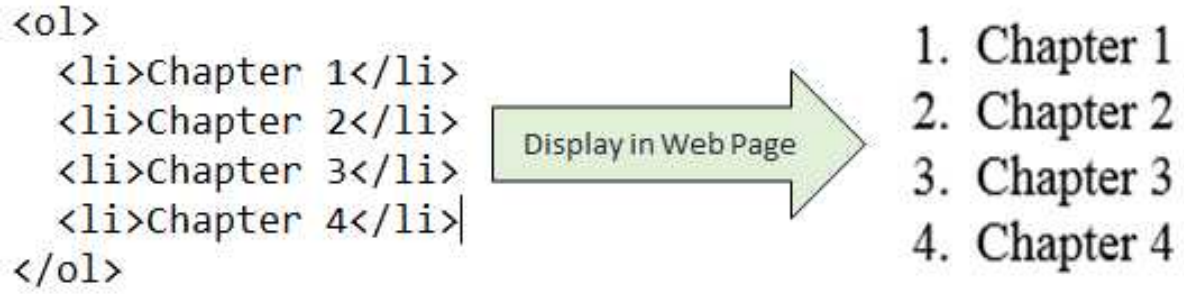


Figure 7.3: Ordered List Example

In the example above, we used numbers (1, 2, 3…) to show order. You also have choice to use alphabet or roman numerals. You have to use **type** attribute in **<ol>** tag to decide which choice to use. The following table shows examples of type attribute.

Table 7.2: Example of Type Attribute

Type example	How it looks?

`<ol type="1">`
`<li>Chapter 1</li>`
`<li>Chapter 2</li>`
`<li>Chapter 3</li>`
`<li>Chapter 4</li>`
`</ol>`
	1. Chapter 1
	2. Chapter 2
	3. Chapter 3
	4. Chapter 4

`<ol type="A">`
`  <li>Chapter 1</li>`
`  <li>Chapter 2</li>`
`  <li>Chapter 3</li>`
`  <li>Chapter 4</li>`
`</ol>`
	A. Chapter 1
	B. Chapter 2
	C. Chapter 3
	D. Chapter 4

`<ol type="a">`
`  <li>Chapter 1</li>`
`  <li>Chapter 2</li>`
`  <li>Chapter 3</li>`
`  <li>Chapter 4</li>`
`</ol>`
	a. Chapter 1
	b. Chapter 2
	c. Chapter 3
	d. Chapter 4

`<ol type="I">`
`  <li>Chapter 1</li>`
`  <li>Chapter 2</li>`
`  <li>Chapter 3</li>`
`  <li>Chapter 4</li>`
`</ol>`
	I. Chapter 1
	II. Chapter 2
	III. Chapter 3
	IV. Chapter 4

`<ol type="i">`
`  <li>Chapter 1</li>`
`  <li>Chapter 2</li>`
`  <li>Chapter 3</li>`
`  <li>Chapter 4</li>`
`</ol>`
	i. Chapter 1
	ii. Chapter 2
	iii. Chapter 3
	iv. Chapter 4

Simple but powerful.

Finally, let's learn about **Description List**. Following is the structure of description list tags:

```
<dl>
      <dt>Short text one</dt>
          <dd>Long description of text one</dd>
      <dt>Short text two</dt>
          <dd>Long description of text two</dd>
      <dt>Short text three</dt>
          <dd>Long description of text three</dd>
      <dt>Short text four</dt>
          <dd>Long description of text four</dd>
</dl>
```

DL stands for **description list**. **DT** means **data term**. DT is used to show short name of the text line. You can add as many <dt> tags as you want. **DD** stands for **data description**. DD is used to show long description for the short text line. <dd> tag comes after <dt> tag because it describes text shown in <dt> tag.

I admit it is little puzzling but you will definitely get hang of it very soon. Let's see how description list HTML will look like for the example shown in the table earlier. It is shown in figure 7.4.

```
<dl>
  <dt>Chapter 1</dt>
  <dd>- Introduction</dd>
  <dt>Chapter 2</dt>
  <dd>- "Hello" HTML</dd>
  <dt>Chapter 3</dt>
  <dd>- What are Elements and Attributes?</dd>
  <dt>Chapter 4</dt>
  <dd>- Headings and Paragraphs</dd>
</dl>
```

Chapter 1
- Introduction

Chapter 2
- "Hello" HTML

Chapter 3
- What are Elements and Attributes?

Chapter 4
- Headings and Paragraphs

Figure 7.4: Description List Example

Ok, let's do exercise now.

Try it yourself

I want you to create **table of content** web page for a HTML book. You will make use of heading and description list tags. The web page should look like as show in figure 7.5.

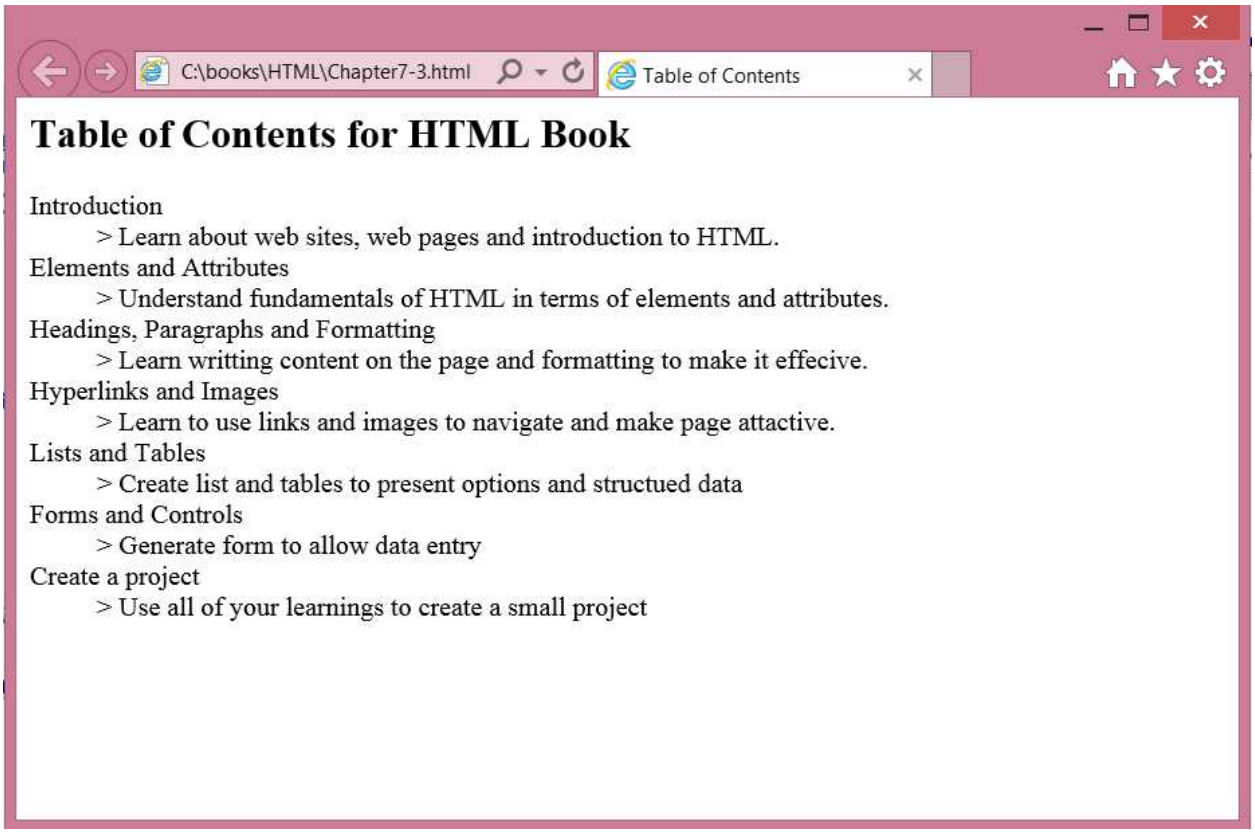


Figure 7.5: Table of Contents Example

Following are the guidelines to perform this exercise:

- Open Notepad
- Create HTML using heading and description list tags. Use <dt> and <dd> tags one by one inside <dl> tag to create table of contents.
- Save the HTML file using .htm or .html extension
- Open the HTML file using web browser to see the result.

If you need any help, please check **Exercise5 – Table of Contents** chapter to find HTML for the web page above. Best of luck!

Summary

Congratulations for making impressive progress. You have come half way in the book. From this chapter onwards, we will learn into little complex HTML tags. But we will follow simple methods to understand them. So, keep reading.

Chapter 8 –Table

In the previous chapter, you learnt about how to use list to organize your writing. List makes writing points based, more readable and more presentable. There is another way you can organize your information and writing – it is called **Table**. You must have seen tables in your book, on web sites, also in your favorite magazine. Tables help you organize information in rows and columns.

Tables are used for similar purpose on web sites as well. They help organize writing or information in rows and columns structure.

Let's learn about tables in this chapter.

Examples of Table

We will search for football world cup at Wikipedia to see an example of table. Please go to the following web page - https://en.wikipedia.org/wiki/FIFA_World_Cup . On this page, check for **Attendance**, there you find a table as shown in figure 8.1 below.

CHAPTER 8

Attendance

See also: List of sports attendance figures

	Year	Host country	Continent	Host's perf.*	Sites / towns	Total gate	Matches	Average gate	Top gate ‡	Site of best-attended game(s) ‡
1	1930	Uruguay	S. Amer.	1st	3 / 1	590,549	18	32,808	93,000	Estadio Centenario, Montevideo
2	1934	Italy	Europe	1st	8 / 8	363,000	17	21,353	55,000	Stadio Nazionale PNF, Rome
3	1938	France	Europe	QF	10 / 10	375,700	18	20,872	58,455	Olympique de Colombes, Paris
4	1950	Brazil	S. Amer.	2nd	6 / 6	1,045,246	22	47,511	171,772	Maracanã Stadium, Rio
5	1954	Switzerland	Europe	QF	6 / 6	768,607	26	29,562	65,000	St. Jakob Stadium, Basel
6	1958	Sweden	Europe	2nd	12 / 12	819,810	35	23,423	51,800	Råsunda Stadium, Solna †
7	1962	Chile	S. Amer.	3rd	4 / 4	893,172	32	27,912	68,679	Estadio Nacional, Santiago
8	1966	England	Europe	1st	8 / 7	1,563,135	32	48,848	98,270	Wembley Stadium, London
9	1970	Mexico	N. Amer.	QF	5 / 5	1,603,975	32	50,124	108,192	Estadio Azteca, Mexico City
10	1974	West Germany	Europe	1st	9 / 9	1,865,753	38	49,099	83,168	Olympiastadion, West Berlin
11	1978	Argentina	S. Amer.	1st	6 / 5	1,545,791	38	40,679	71,712	River Plate Stadium, Buenos Aires
12	1982	Spain	Europe	top 12	17 / 14	2,109,723	52	40,572	95,500	Camp Nou, Barcelona
13	1986	Mexico	N. Amer.	QF	12 / 11	2,394,031	52	46,039	114,600	Estadio Azteca, Mexico City
14	1990	Italy	Europe	3rd	12 / 12	2,516,215	52	48,389	74,765	San Siro, Milan
15	1994	United States	N. Amer.	top 16	9 / 9	3,587,538	52	68,991	94,194	Rose Bowl, Pasadena, Calif.†
16	1998	France	Europe	1st	10 / 10	2,785,100	64	43,517	80,000	Stade de France, Saint-Denis †
17	2002	South Korea & Japan	Asia	4th / top 16	10 / 10 10 / 10	2,705,197	64	42,269	69,029	International Stadium Yokohama (Japan)
18	2006	Germany	Europe	3rd	12 / 12	3,359,439	64	52,491	80,000	Olympiastadion (Berlin)
19	2010	South Africa	Africa	*Group*	10 / 9	3,178,856	64	49,670	84,490	Soccer City, Johannesburg
20	2014	Brazil	S. Amer.	4th	12 / 12	3,429,873	64	53,592	74,738	Maracanã Stadium, Rio
Overall						**37,500,710**	**836**	**44,857**	**171,772**	**Maracanã Stadium, Rio (1950)**

Figure 8.1: Table Example

You can see how information such as year, country name, continent, matches etc. have been organized nicely in rows and column structure. Does it not make it more readable? I am sure you agree and that is what the purpose of table is.

Learning the Details

Before we learn about table HTML tags, let's study some of the names or definitions related to the table.

Please refer the figure 8.2 to see various definitions related to the tables.

C H A P T E R 8

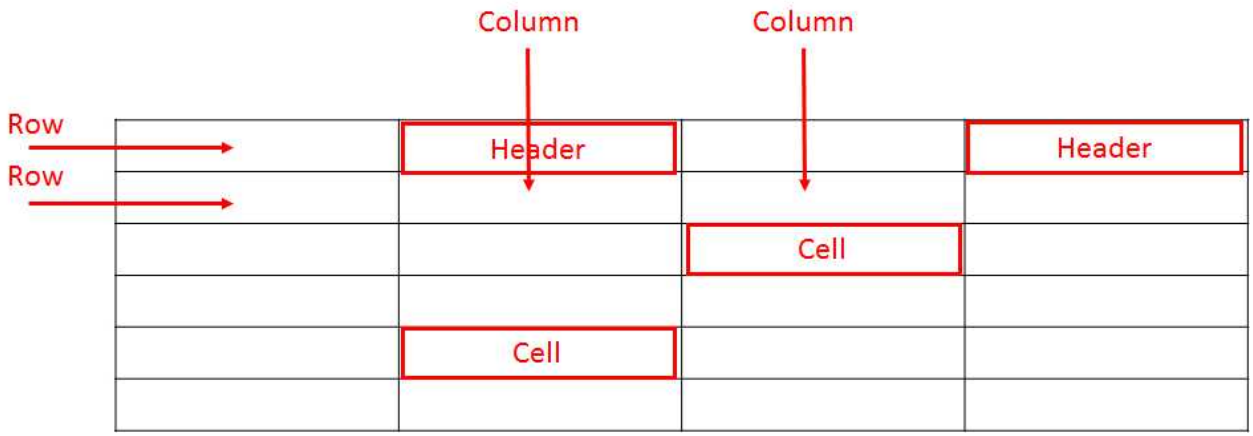


Figure 8.2: Table Definitions

Row is the horizontal line in the table. You can add as many rows as you want in the table. Row determines the height of the table.

Column is the vertical line in the table. Again you can add as many columns as you require in your table. Column decides the width of the table.

Cell is the cross section of row and column in the table. Number of cells in the table depends on the number of rows and column in the table. You write your information in table in the cell only.

Header is a type of cell and it is a cell in the **first row** of the table. Generally they describe the name of the column in the header cells. In the world cup example above, header has text like year, host country, continent, matches which shows type of information written in the particular column. Header text is generally shown as bold.

In HTML, tables are defined using **<table>**, **<th>**, **<tr>** and **<td>** tags. The **<th>** tag represents header cell. The **<tr>** tag represents row. The **<td>** tag represents cell. The **<table>** tag works as container and keeps all these tags together.

Let's map these tags to the table shown in figure 8.2 above. The mapping has been presented in the figure 8.3.

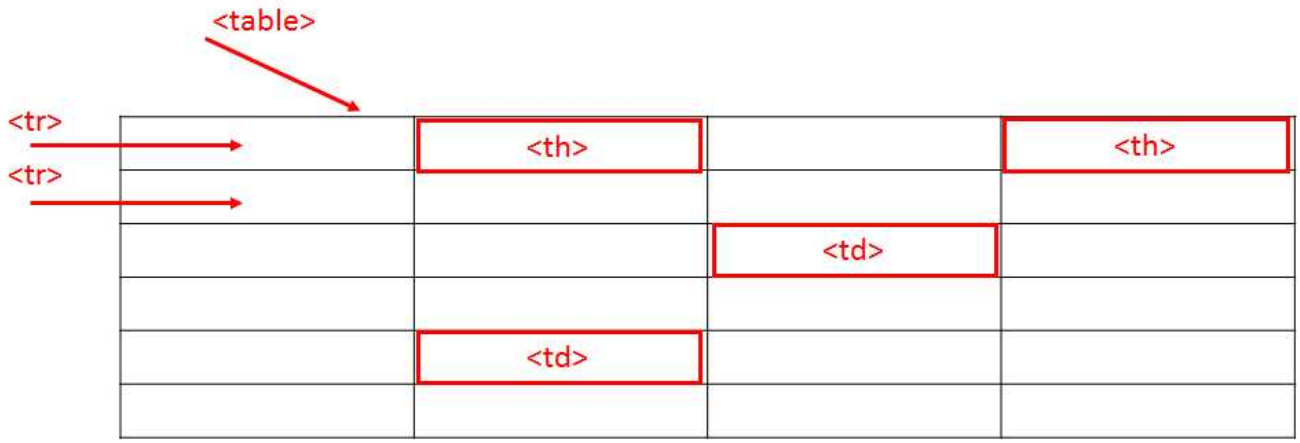


Figure 8.3: Table Tags Mapping

Number of <tr> tags within <table> tag defines number of rows in the table. If you want five rows in a table, you add five <tr> tags inside <table> tag.

Similarly, number of <td> tags within <tr> tag defines number of cells in the row. You want three 3 cells, you add three <td> tags within <tr> tag.

<th> tag works in the same way as <td> tag works.

Header text is written inside <th></th> tags. Cell text is written within <td> </td>tags.

With this understanding, let's see the structure of table HTML.

```
<table border="width in pixel">
        <tr>
                <th>Header Text</th>
                <th>Header Text</th>
        </tr>
        <tr>
                <td>Cell Text</td>
                <td>Cell Text</td>
        </tr>
        <tr>
                <td>Cell Text</td>
                <td>Cell Text</td>
        </tr>
</table>
```

If you look at the structure, you can relate what I was talking about these different table tags in the previous paragraph. The **border** attribute shows the width of the table lines in pixel. If you don't provide border value, the table will have no lines.

Let's see an example. Please check the HTML below:

```
<html>
```

```
<head>
        <title>Table Example</title>
</head>
<body>
<table border="1">
        <tr>
                <th>Name</th>
                <th>Job Title</th>
        </tr>
        <tr>
                <td>Brajendra Singh</td>
                <td>Solution Architect</td>
        </tr>
        <tr>
                <td>Pathik Rawal</td>
                <td>Practice Lead</td>
        </tr>
</table>
</body>
</html>
```

HTML Web Page looks like the following as shown in figure 8.4:

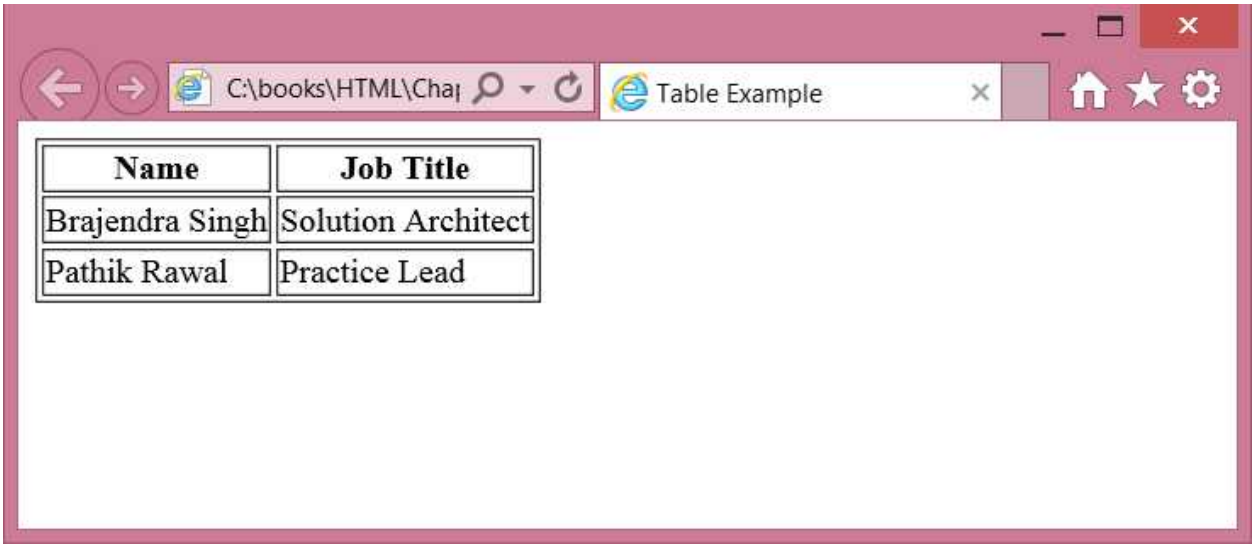


Diagram 8.4: Table Example Web Page

In table cell (<td>), you can write anything – text, image, hyperlink. In the football example shown above, you can see images and hyperlinks used as cell text.

Try it yourself

In the exercise today, we will create a Mark Sheet Table on web page. The web page should look like the figure 8.5 shown below.

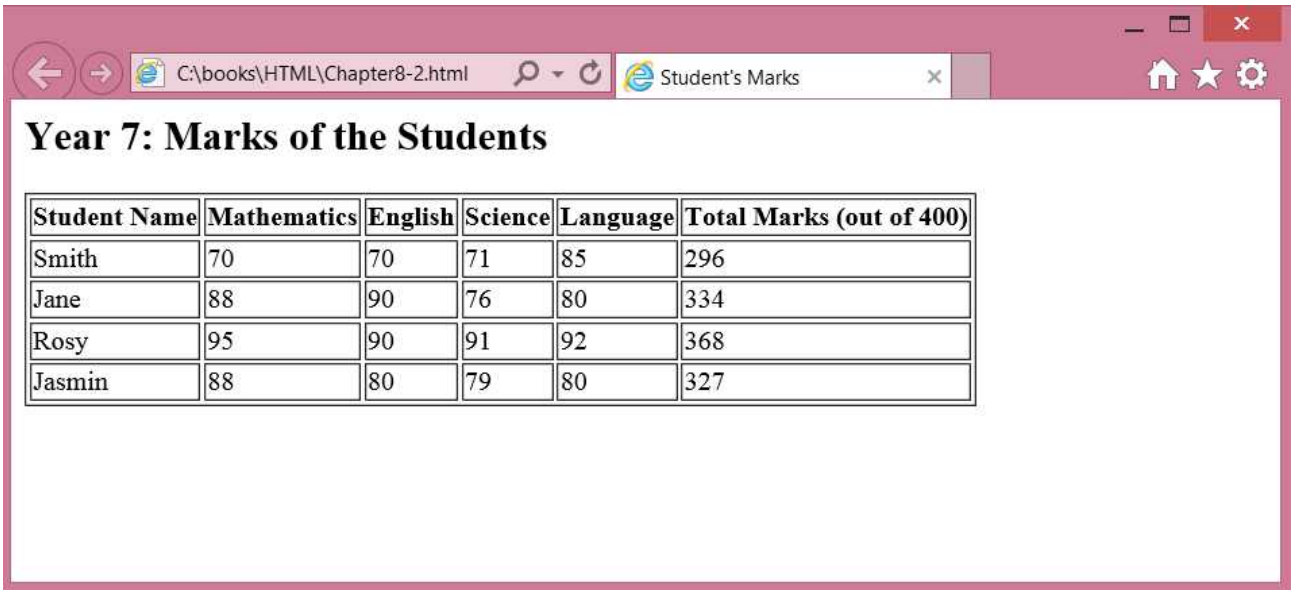


Figure 8.5: Mark Sheet Web Page

Following are the guidelines to perform this exercise:

- Open Notepad
- Create HTML using <table>, <th>, <td> and <tr> tags.
- Use six <td> tags to create columns and five <tr> tags to write rows.
- Write text in the individual cells.
- Save the HTML file using .htm or .html extension
- Open the HTML file using web browser to see the result.

If you need any help, please check **Exercise6 – Student's Marks** chapter to find HTML for the web page above. Best of luck!

Summary

Congratulations for completing this important chapter. Hope you enjoyed it. Tables add tremendous capability in organizing data on web pages. I hope you will use it very soon in some of your web sites. Till then, keep reading…

Chapter 9 – Styles on Page

You have learnt most of the fundamental tags in HTML in the previous chapters. You understand when to use which tag and for what purpose. You can put all these tags together to make a nice web page.

What we have not learnt is how to decorate the tags? How to make them colorful or of various size or of different font etc.? **Styles** are used in HTML for decoration purpose. Styles are like make-up of the web pages. You can choose your color, back ground, shades, lining to make the page more beautiful and attractive.

There are so many styling methods. But we will learn some fundamental ones in this chapter. You will get a good idea about how to use styles and then you can explore other styling methods through further studies. Ready to color your web page! Here we go.

Examples of Styles

Well, there is no exact sample of styles because every page does styles – some do more and some do less. Let's see the web page of **Butlin's** and see how they have used styles to color their page. For those who don't know what Butlin's is, they offer holiday resorts for the kids and families. Please go to the following web page - http://www.butlins.com/ and check styles as shown in figure 9.1 below.

CHAPTER 9

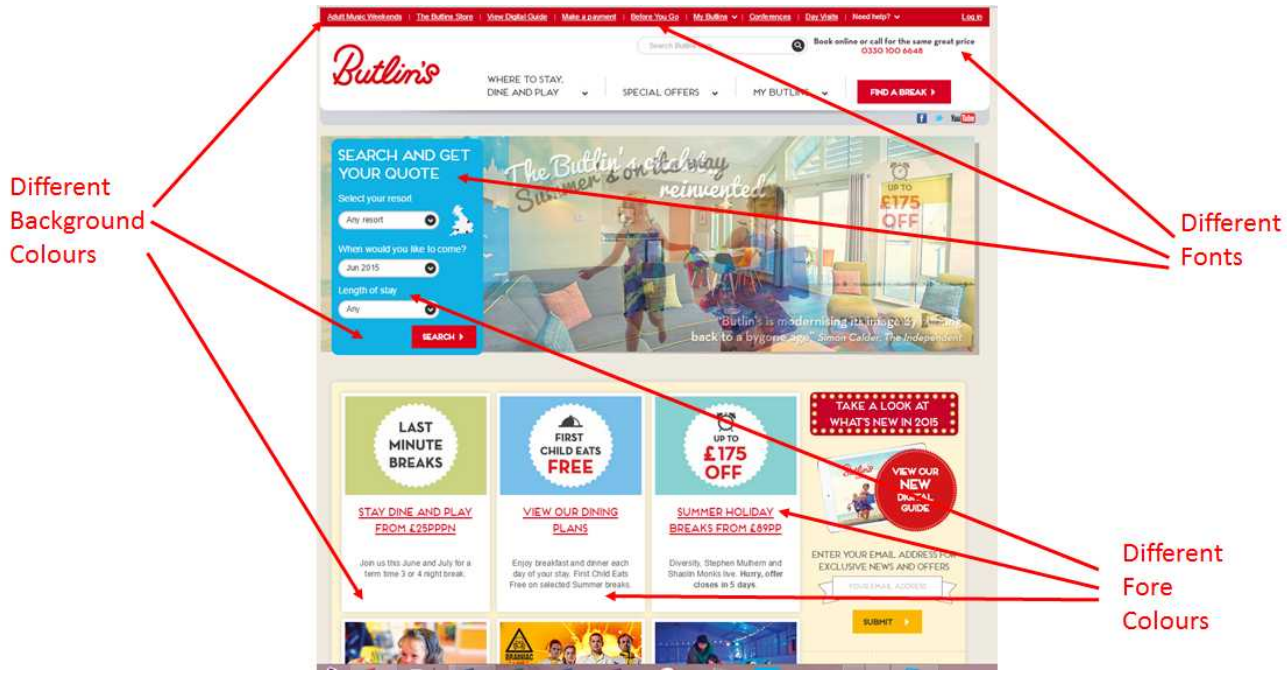


Figure 9.1: Styles Example

See, how styles have been used to decorate the web page. They look so beautiful and inviting. I think you have got an idea of what style means. Let's study them in detail.

Learning the Details

Styles are applied using a tag attribute called **style**. This attribute can be used with any tag. It has the following structure:

```
style="property: value;  property: value"
```

Property means what style you want to use. **Value** means what style value you have chosen; how you want your page or tag to look. Not making much sense. Ok, let check the table below of some common style names, values and their purpose. It will then make perfect sense to you.

Table 9.1: Common Used Styles

Style Property	Style Value	Purpose
background-color	Red or Green or Grey, any color	Used to paint background color of the web page
color	Red or Green or Grey, any color	Used to paint color the text or hyperlink

font-family	Verdana, Courier, Arial etc.	Decides which text font to use.
Font-size	10 / 5/ 20 any value	Set the size of text or hyperlink in pixel
text-align	Left, Center, Right	Align text or hyperlink or image to left or to center or to right.

You can use more than one style properties but they should be separated by a **semicolon [;].** Property and Value have **colons [:]** between them.

Let's see an example as shown in figure 9.2. Let's make a page with heading tag but no style.

```
<html>
<head>
<title>Style Example</title>
</head>
<body>
        <h2>I am a heading.</h2>
</body>
</html>
```

Display in Web Page

I am a heading.

Figure 9.2: Heading Tag No Style

Now, we will add style to make the heading tag look red as shown in figure 9.3.

```
<html>
<head>
<title>Style Example</title>
</head>
<body>
        <h2 style="color:red">I am a heading.</h2>
</body>
</html>
```

Display in Web Page

I am a heading.

Figure 9.3: Heading Tag in Red Color

You can see that I have added **style** attribute in **<h2>** tag with property as **color** and value as **red**. And when I did that it changed the text of <h2> tag from black to red. This is the beauty of style.

Let's change the font of <h2> text to **Arial** as shown in figure 9.4.

```
<html>
<head>
<title>Style Example</title>
</head>
<body>
        <h2 style="color:red;font-family:arial">I am a heading.</h2>
</body>
</html>
```

Figure 9.4: Heading Tag text in Arial font

You see the font has changed now. The text style looks different. I added **font-family** property with value **arial**. I used semicolon to separate both properties color and font-family.

Let's change style of another tag to add back-ground color in the web page. I will use **background-color** property with value **skyblue** in the **body** tag of the HTML. It is shown in figure 9.5.

```
<html>
<head>
<title>Style Example</title>
</head>
<body style="background-color:skyblue">
        <h2 style="color:red;font-family:arial">I am a heading.</h2>
</body>
</html>
```

Figure 9.5: Background Style

You see how various combination of styles and properties can be used to make your web page colorful. All you need is knowledge of these different styles and the power of imagination. Isn't it?

Try it yourself

For exercise, we will revisit Greedy Crow web page you created in the exercise 1. We will add color and style to this page. Please do necessary changes in web page HTML to make it look like as shown in figure 9.6.

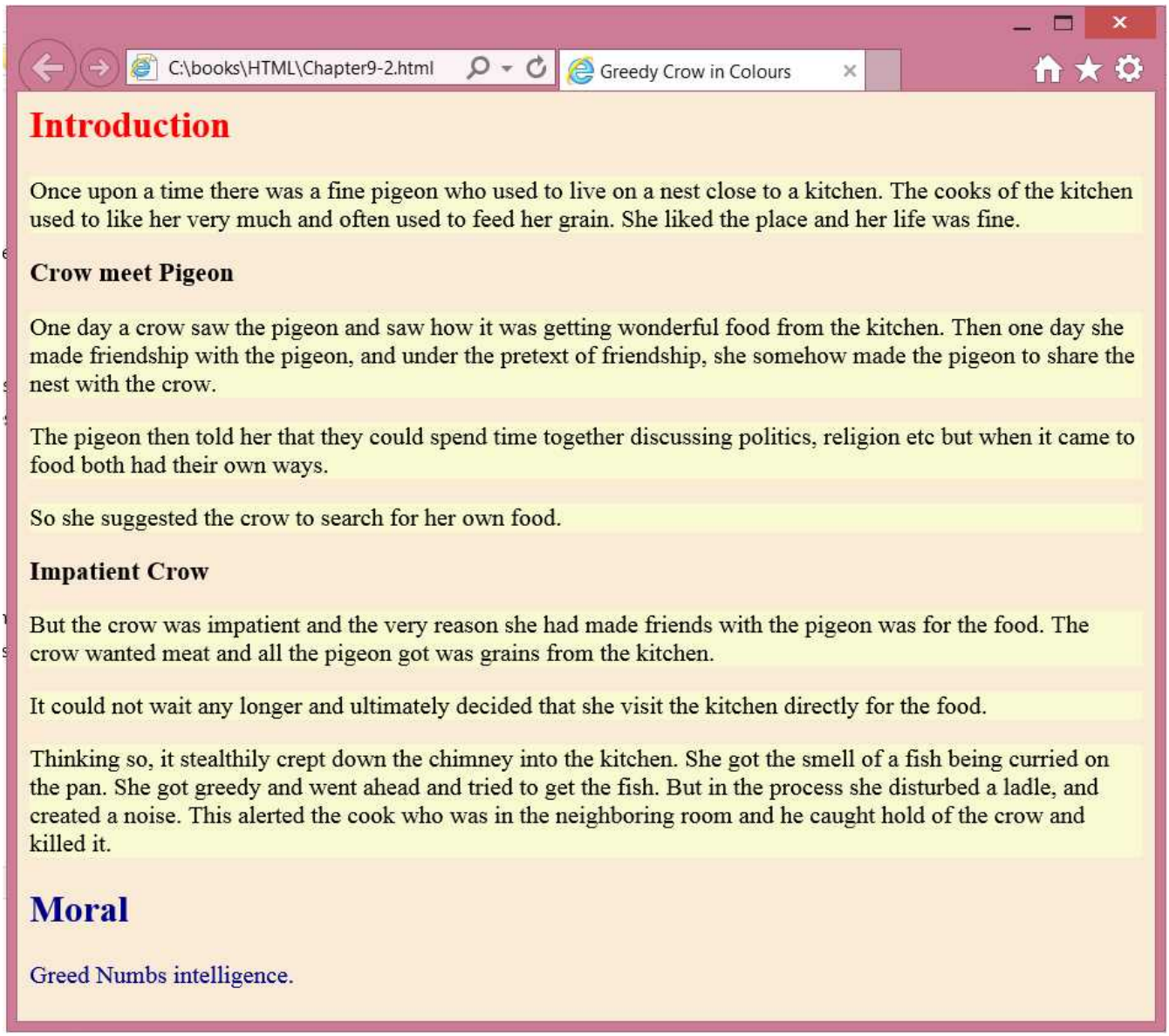


Figure 9.6: Story in Colors Web Page

For your help, please use the following web page to learn names of various colors used in this exercise: http://www.w3schools.com/html/html_colornames.asp

Following are the guidelines to perform this exercise:

- Create a copy of **Exercise1 – Greedy Crow** HTML file with the same extension .htm or .html. Give the copied file a new name say **Exercise7 – Greedy Crow in Colors**.
- Open **Exercise7 – Greedy Crow in Colors** file in Notepad.

- Apply different styles to the texts, heading and body of the web page as asked in the exercise. You will need to add style attribute in the respective tags.
- Save the changes to the HTML file.
- Open the HTML file using web browser to see the result.

If you need any help, please check **Exercise7 – Greedy Crow in Colors** chapter to find HTML for the web page above. Best of luck!

Summary

Congratulations for coloring a simple web page. Styles are for the right part of the brain – the creative part. You can use your imagination and power of style to create a spectacular piece of art in your web page. There are web designers who specialize in styling and make good fortune out of it. I really wish you use these techniques someday. Till then, keep reading…

C H A P T E R 10

Chapter 10 –Using Iframe

You learnt about styles in the previous chapter. You now know how to change background, color, font or font size to make your web page look elegant. Just pour in your creativity and create a master piece web pages using styles.

In this chapter, we are going to discuss a very interesting topic. What if you want to show Wikipedia web page inside your own web page? Yes, you read it right. You are not going to create a Wikipedia web page, but will show that page inside your web page. You can achieve this using **Iframe** tags.

Showing other web pages inside your page is sometimes very useful when you simply want to make use of some other web site inside your web page. You don't want to recreate the other's pages but reuse it for your purpose.

Interesting, right?

Examples of Iframe

We will go to University of Michigan web page to see an example of Iframe. Please go to the web site - http://sitemaker.umich.edu/iframe.example/the__iframe__lives_here . You see Iframe as shown in figure 10.1 below.

C H A P T E R 10

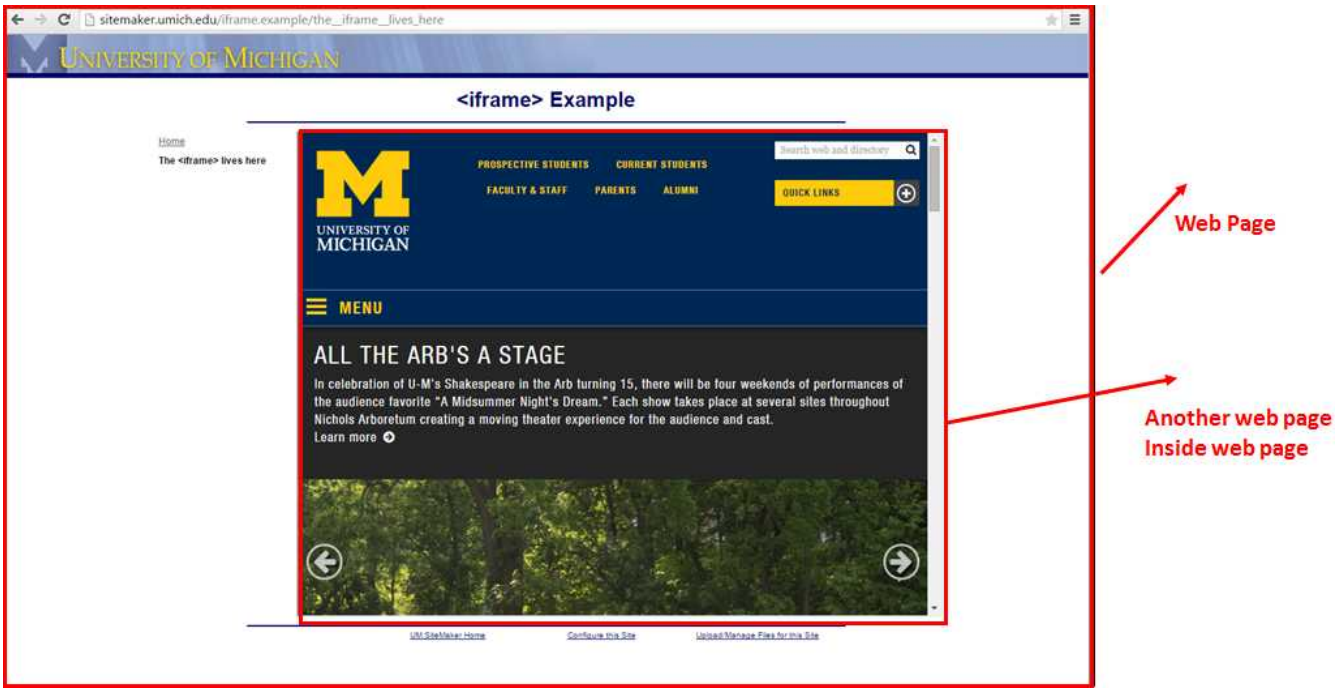


Figure 10.1: Iframe Example

You can see that there is a web page open in browser. And then you have another web page open inside the browser web page. You can perform any action on this inside web page like click, navigate, enter data etc. The inside web page is real and live.

Learning the Details

How we achieve what has been shown in the example above? The **<iframe>** tag is used for this purpose. Please check the structure of <iframe> tag below:

```
<iframe src="address of inside web page" width="width in
pixel" height="height in pixel"></iframe>
```

We start and close <iframe> tags and use attribute to provide address of the inside web page.

Src stands for **source**. We provide the address of the inside web page in this attribute. Please what recall we discussed providing address in **href** attribute for **<a>** tag and in **src** attribute in **<img>** tag? This is also similar to that. You provide the address of the web page or web site which you want to put inside Iframe.

Width is the width of the inside web page when shown. It is measured in pixel.

Height is the height of the inside web page and it is also measured in pixel.

Let's create a web page which shows Wikipedia page inside. The HTML of the web page will look like following:

```
<html>
<head>
<title>Learning Iframe</title>
</head>
<body>
<h1>Learn IFrame in HTML </h1>

<iframe src="https://en.wikipedia.org/wiki/Main_page" width="600"
height="600"></iframe>

</body>
</html>
```

The web page will be shown as in the figure 10.2 below.

C H A P T E R 10

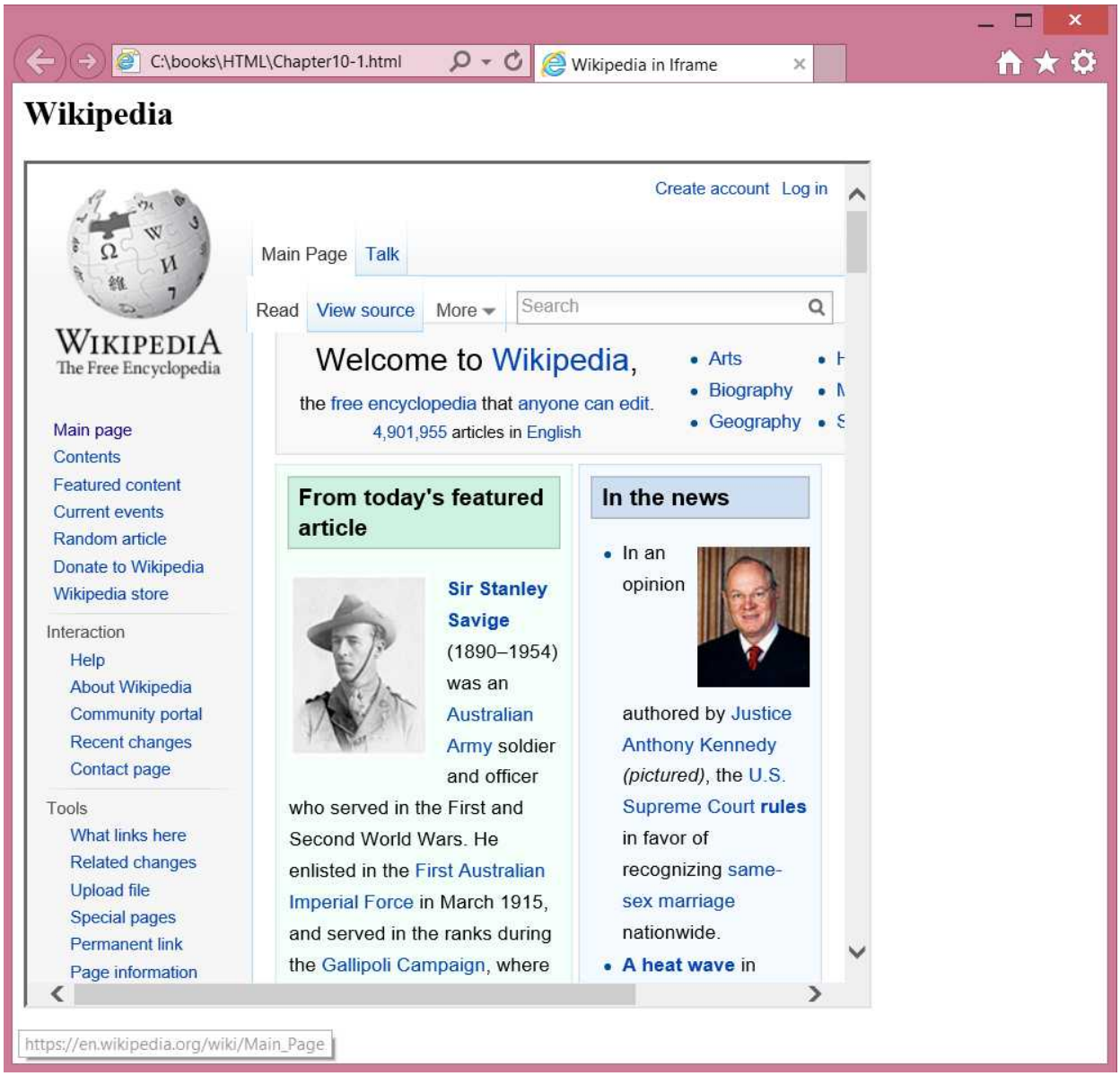


Figure 10.2: Wikipedia in Iframe HTML Display

Wow, you have just shown Wikipedia page inside your page. Simple and Awesome!

Many web sites don't like and don't allow themselves to be put inside Iframe. Google web site is an example of that. We should respect their decision and try not to use their web sites in Iframe.

C H A P T E R 10

Try it yourself

Exercise time. We will create a web page named **My Search** which will show Bing search and Wikipedia search both in web sites inside My Search web page. Well, I wanted to use Google and Yahoo search as well but they don't support Iframe and I totally respect that.

Your web page will look like as shown in figure 10.3.

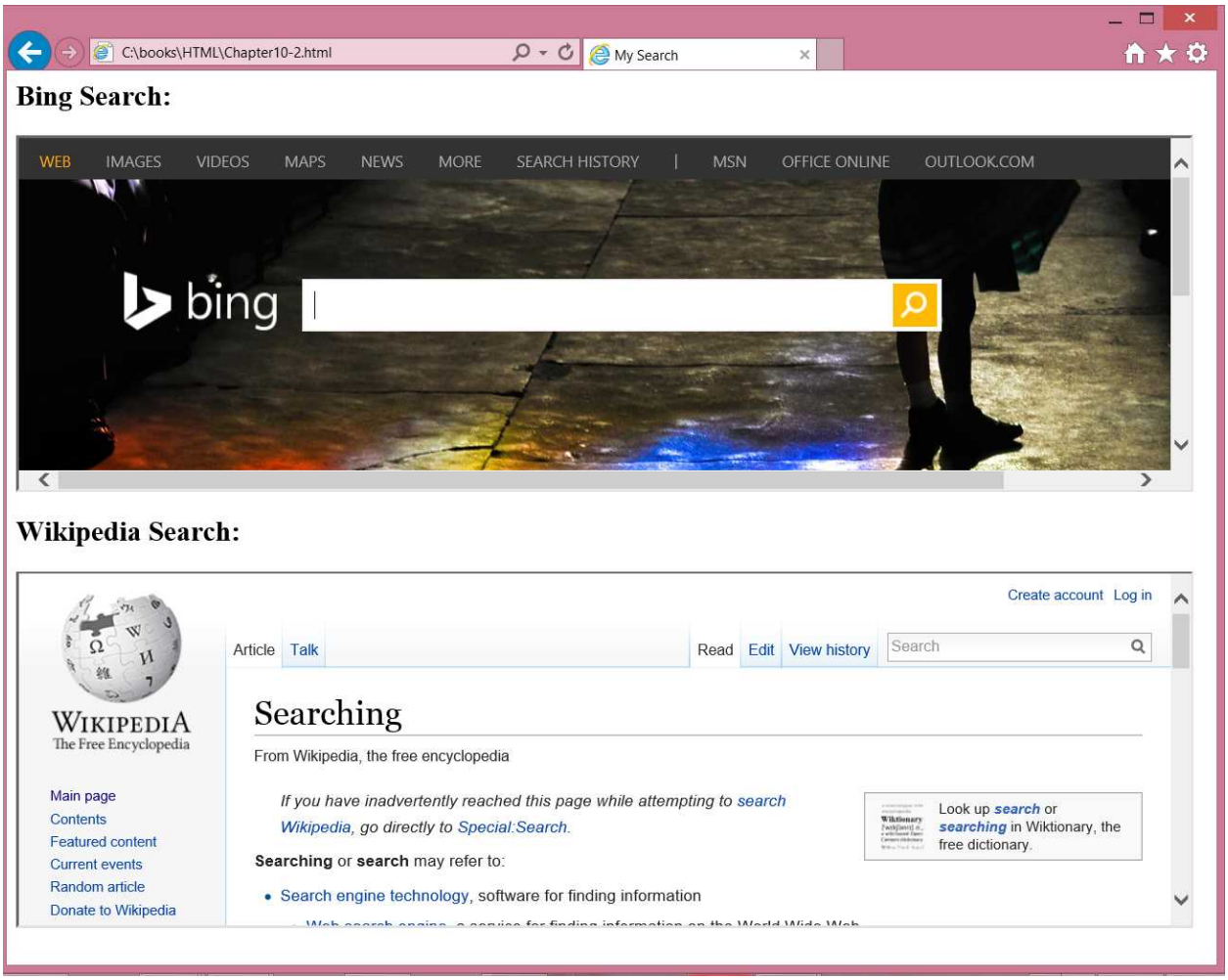


Figure 10.3: My Search Exercise Page

Please go ahead and play with both inside web pages. They will allow you to do everything.

Here are the details to create the web page.

Address of Bing search is: https://www.bing.com/

Address of Wikipedia search is: http://en.wikipedia.org/wiki/Searching

Height is 300 and width is 1000 for both Iframes.

Following are the guidelines to perform this exercise:

- Open Notepad
- Create HTML using heading and Iframe tags. You will need two Iframe tags to show two web sites. The address of the web sites has been given in the earlier paragraph.
- Save the HTML file using .htm or .html extension
- Open the HTML file using web browser to see the result.

If you need any help, please check **Exercise 8 – My Search** chapter to find HTML for the web page above. Best of luck!

Summary

That finishes using Iframe. You have learnt so many tags and styling methods in the previous chapters. Are you getting some ideas to put them together to make a fantastic web page? If so, I would say who is stopping you? Rock on! Keep reading…

Chapter 11 –Forms and Controls

In the chapters so far, we mostly read about how to show information on web page or how to use links or how to make pages look beautiful. We will now focus on building a web page where people can type some information or able to select some choices using their keyword and mouse on the web pages. We generally call such activities of the people as **user inputs**.

We all have seen examples of user inputs on various web sites. For example, on Google site when you type search text and press search button, that is a type of user input. Similarly when you or your parents go to online booking web sites and book for hotel or flights by typing start date, end date, number of people etc. that is also a type of user input.

Forms and Controls are used to create such user inputs. **Form** is like a container (just like table was container earlier for <tr> and <td> tags) and **Controls** are individual user inputs. When I say container, it is similar to a book which is container for pages. A book contains one or more pages. Similarly, a form contains one or more controls. Not making much sense? Right? No worries, lets understand it again in the examples in the next section.

This chapter and next chapter are closely related to each other. In current chapter you learn about designing forms while in next chapter you learn how to make use of user inputs to perform some business actions. For instance, on Google site, when I enter search text and press search button, the site responds back with search results. Providing these search results is business action for the Google site. Similarly, booking hotel or flight tickets based on people user inputs is another example of business action.

Examples of Forms and Controls

Let's start with the first example we discussed. Please go to Google site (https://www.google.co.uk/) as shown in figure 11.1.

C H A P T E R 11

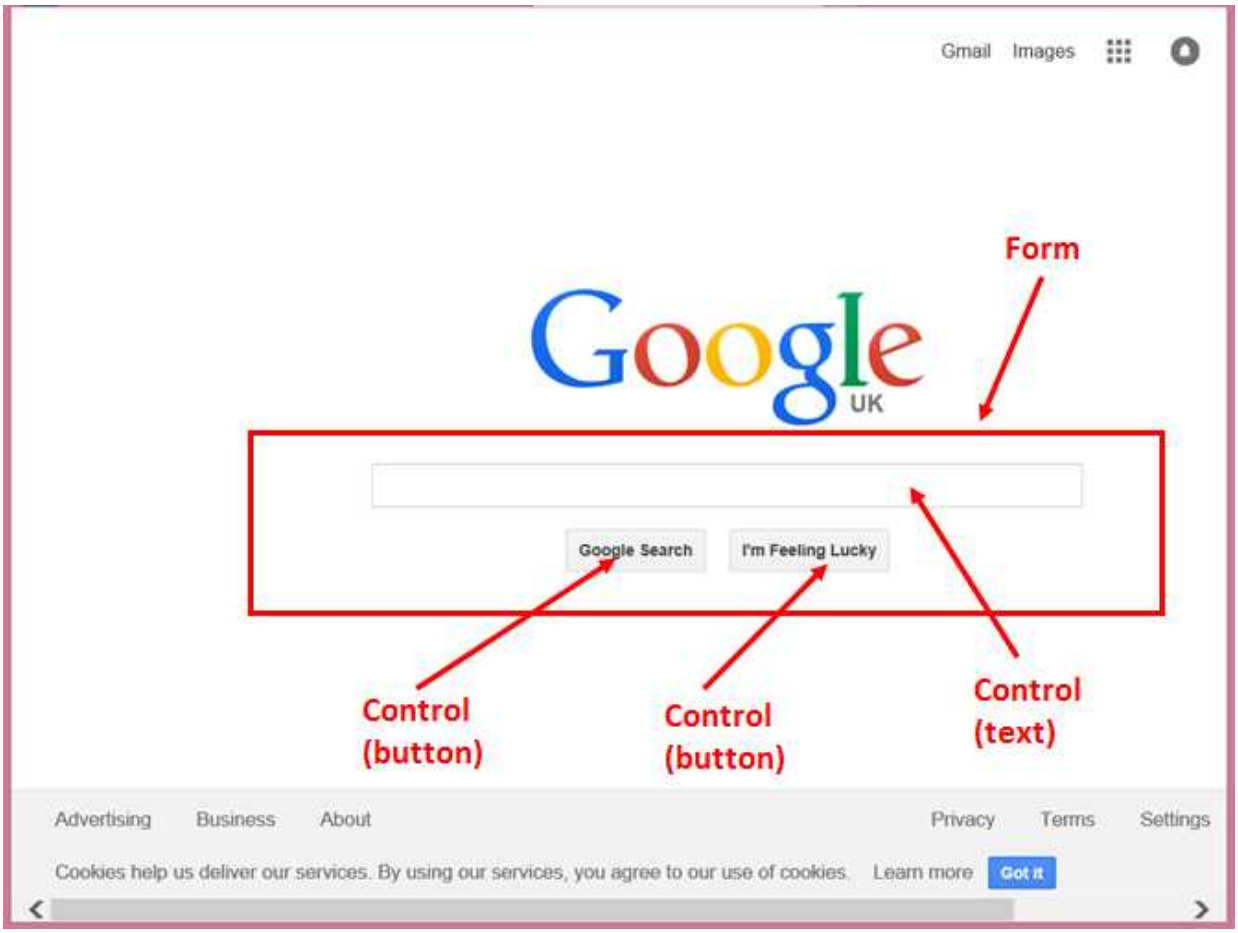


Figure 11.1: Google Site Form Example

You can see in this simple web page, there are three controls – a place to enter search text, two buttons to click. And all three controls are inside a form. Didn't I say that Form is container which contains one or more controls? People provide user input to these controls such as typing a text, selecting from choices or clicking on a button. Each control is a way for people to provide some kind of input.

To emphasize this idea, let's see another example. Please open http://www.booking.com/ web site as shown in figure 11.2.

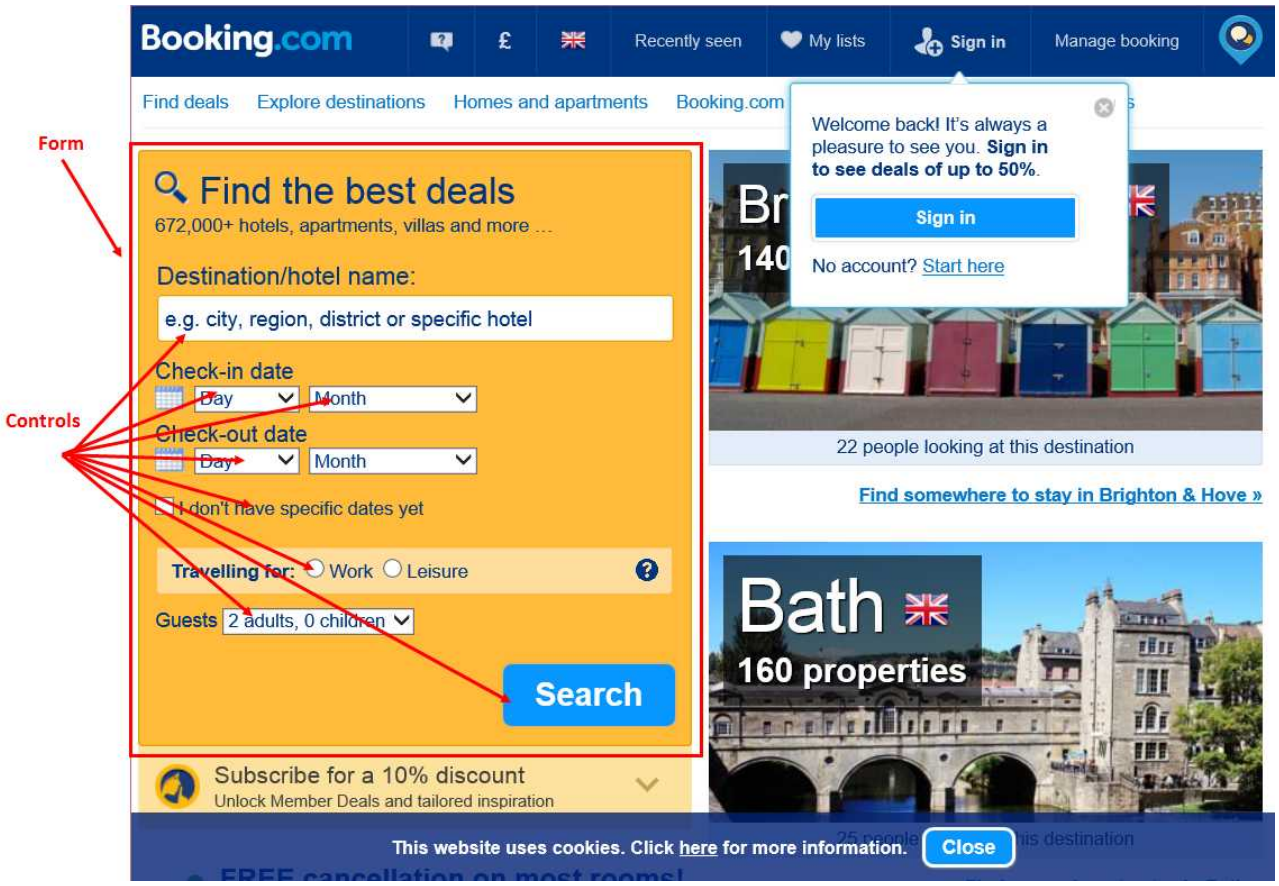


Figure 11.2: Booking Site Form Example

This is another example where you have one form containing more than 5-6 controls. You see some more types of control in the figure 11.2 above such as ability to select dates of stay, selecting choices between Work and Leisure.

HTML comes with various types of controls. We will learn about them in the remaining chapter.

On a single web page, you can have more than one forms. And each form can have their own set of controls.

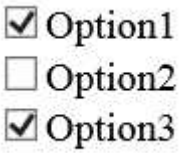
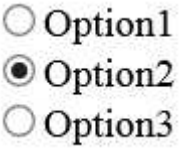
Learning the Details

You now understand the concept of form and controls. There are various types of control available in HTML. But for this book, we will focus on some important ones which are generally used in designing web pages.

The table below shows different types of control in HTML, the purpose of the controls and how they look.

C H A P T E R 11

Table 11.1: Types of Control

Control Type	Purpose	How it looks?
Textbox	Allows to type single line of text	
Text Area	Allows to type multiple lines of text	
Checkbox	Allows to select multiple options from the given choices.	
Radio	Allows to select only one option from the given choices.	
Button	Allows to click on a button to perform some action	
Select	Allows to select one option from multiple drop down choices.	

Time to learn how each of these individual controls work – what tags they use and what attributes are useful?

Most of the controls use the same tag **<input>**. It is the **type** attribute value of <input> tag which is different based on what control you are using. Simple Enough.

C H A P T E R 11

Textbox Control

Let's start with **textbox** type control. It is used to type single line of text or sentence. Following is the example of textbox type control tag:

```
<input type="text" id="control name"></input>
```

You can see it uses **<input>** tag and **type** attribute value is **"text"** to tell it is textbox type control. **Id** stands for **identifier** and this attribute is used to keep a name for the control. Just like, teacher takes your name among other students in the class to find you. The Id attribute value can be used to find the control uniquely among other controls. We will learn the use of Id attribute more in the next chapter.

Checkbox and Radio Controls

The next type of controls are **checkbox** and **radio**. They are used to select options from multiple choices given. The following are the tags of these controls:

```
<input type="checkbox" id="control name" >Option1</input>

<input type="radio" id="control name" >Option1</input>
```

Both again use <input> tags but their type attribute values are **checkbox** and **radio**. **Option1** is the display text of the choice shown on the web page. Since, using these controls, you provide multiple choices on the web page, you generally add more than one checkbox or radio controls to show multiple choices. You will see an example of this very soon.

Button Control

Moving on, the following is the tag structure for **button** type control.

```
<input type="button" value="Button" id="control name" ></input>
```

The type attribute value is **button**. The text in the **value** attribute is the display text of the button shown on the web page.

Textarea Control

Let's see controls which do not use <input> tag. We will start with the **textarea** control. Here is how the tag looks like:

```
<textarea id="control name" rows="10" cols="30"></textarea>
```

We use **<textarea>** tag for this control. Since the textarea can take multiple lines of text, you also have choice to set height and width of the control using **rows** and **cols** attributes. Rows attribute determines the height of the control. Cols stands for columns and it determines width of the control. They both take integer values.

Select Control

We will next talk about **select** control which allows to select one option from multiple available choices. Here is the tag structure of select control:

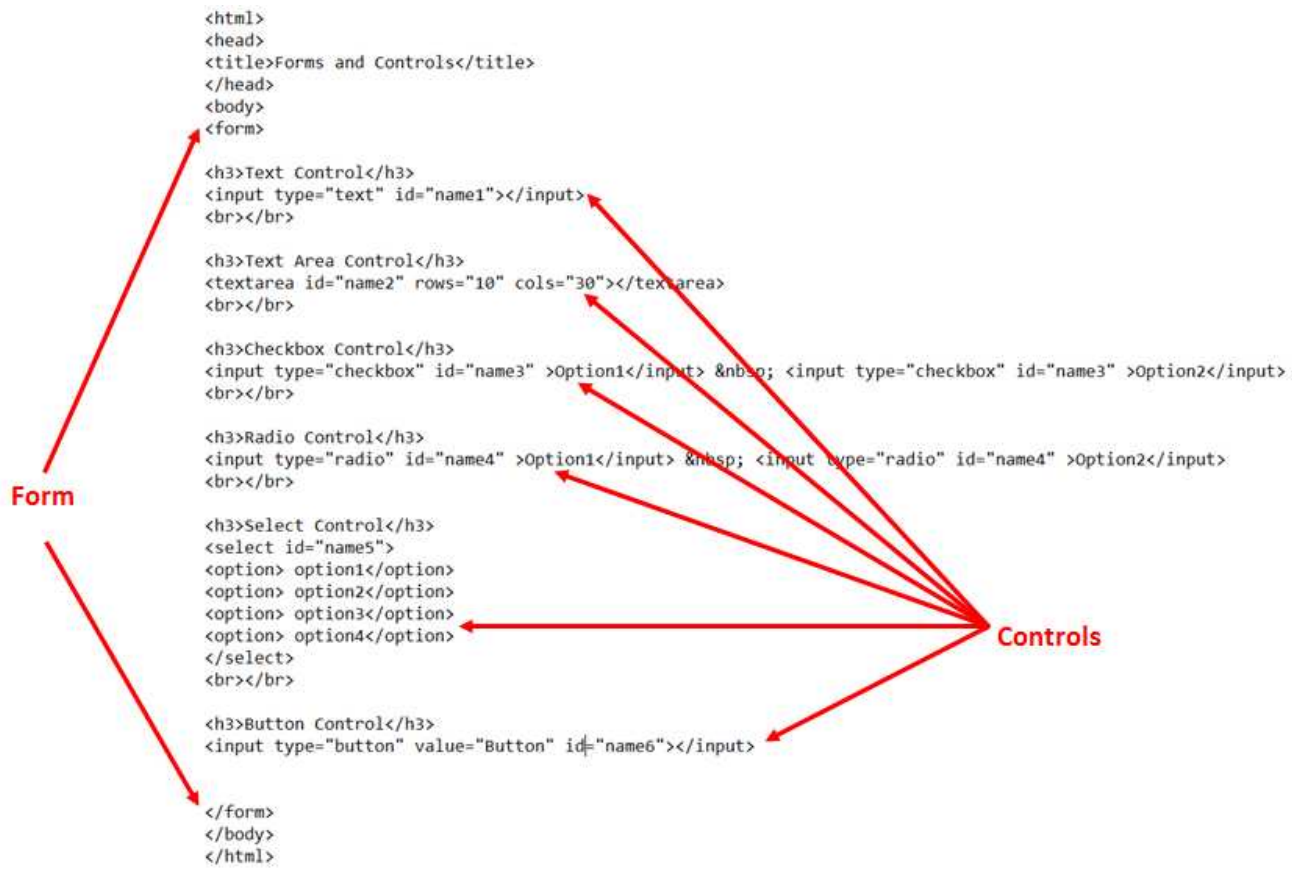
```
<select id="control name">
<option> option1</option>
<option> option2</option>
<option> option3</option>
<option> option4</option>
</select>
```

The tag starts with **<select>** and ends with **</select>.** And between these two tags, you can place multiple **<option></option>** tags. Each <option> tag represents one choice which can be selected by people. Text between <option> and </option> tags presents text of the choice on the web page. You add two or more <option> tags between <select> tags depending on number of choices to be provided on web page.

We now have discussed all important controls. I suggest that you read about each control one more time and then look back in the table 11.1 to see how the control tags are shown on the web page.

Form

We now understand all types of controls and their tags. Let's place all these controls under **<form>** tag in HTML. After all, form is container for all the controls. Here is how the HTML with form and controls will look like as shown in the figure 11.3.




```
<html>
<head>
<title>Forms and Controls</title>
</head>
<body>
<form>

<h3>Text Control</h3>
<input type="text" id="name1"></input>
<br></br>

<h3>Text Area Control</h3>
<textarea id="name2" rows="10" cols="30"></textarea>
<br></br>

<h3>Checkbox Control</h3>
<input type="checkbox" id="name3" >Option1</input>   <input type="checkbox" id="name3" >Option2</input>
<br></br>

<h3>Radio Control</h3>
<input type="radio" id="name4" >Option1</input>   <input type="radio" id="name4" >Option2</input>
<br></br>

<h3>Select Control</h3>
<select id="name5">
<option> option1</option>
<option> option2</option>
<option> option3</option>
<option> option4</option>
</select>
<br></br>

<h3>Button Control</h3>
<input type="button" value="Button" id="name6"></input>


</form>
</body>
</html>
```

Figure 11.3: Form HTML Tags

You can see, I have placed all the controls one by one inside **<form>** tag. I have separated each control by a new line tag **
** so that each control come next to each other upside-down in vertical line. It makes form more readable.

If you look carefully, I have added a new element "** **". This element is used to place a space anywhere you want on the web page. I have added them between checkbox and radio controls to have some space between them. On web page, it will show one space between choices given.

For each control, I have used heading tag **<h3>** to show the name of the control.

Finally, I have given some ids (identifiers) to the controls such as **name1, name2** etc. We will talk more about this attribute in the next chapter.

So, how does this HTML look like? It is shown in figure 11.4.

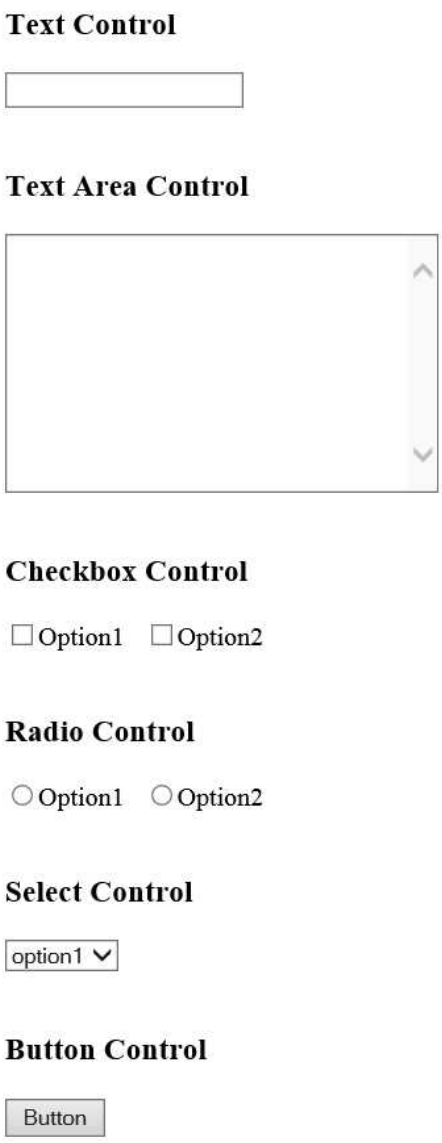


Figure 11.4: Form HTML Display

Easy, right? I suggest you look at each HTML controls one by one as shown in the figure 11.3 and check how it has been shown on the web page in the figure 11.4. You should be comfortable with each control tag before we start the exercise.

Let's do some exercise.

C H A P T E R 11

Try it yourself

We will try to create a **Flight Booking Form** which can be used to book flight tickets. The form will look like as shown in figure 11.5.

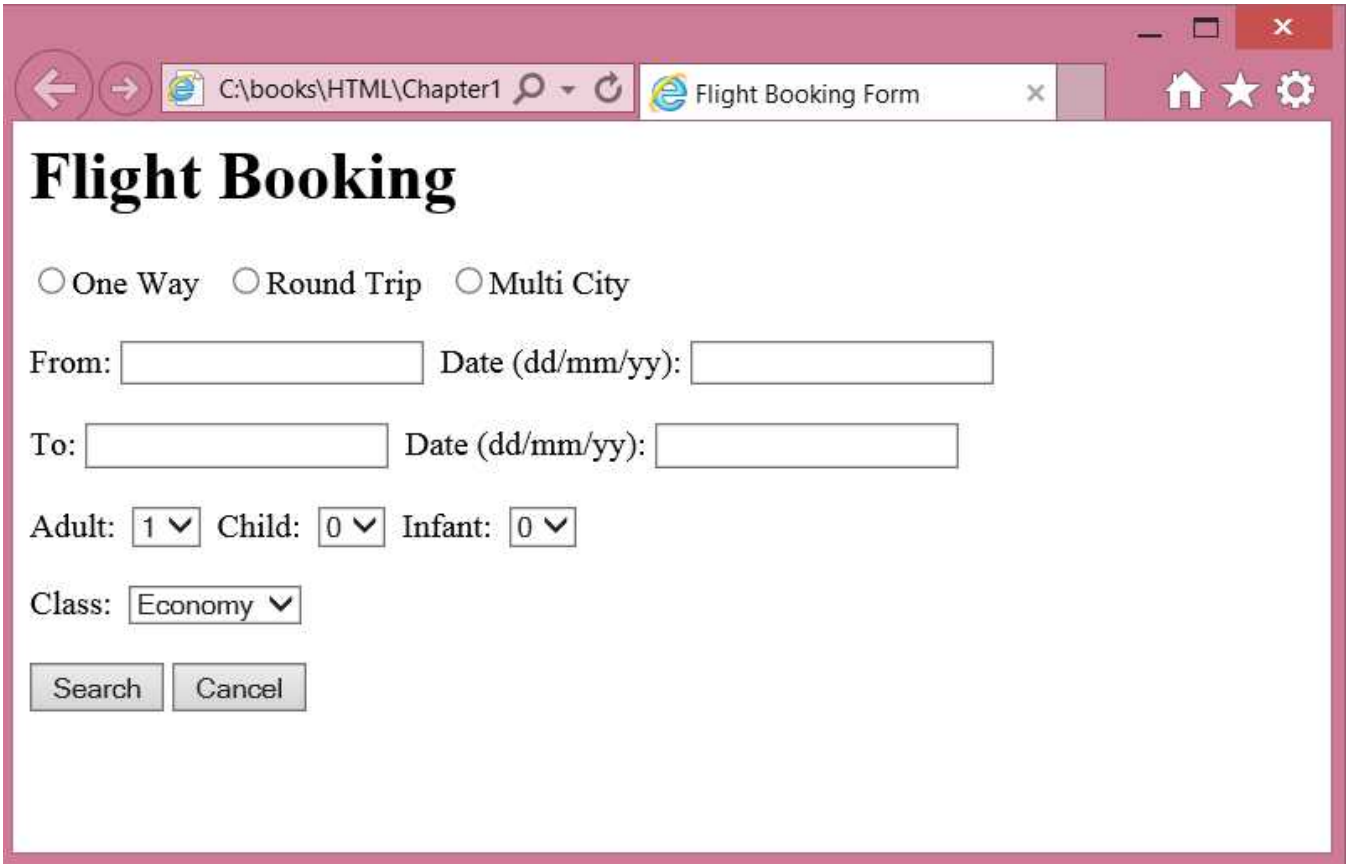


Figure 11.5: Flight Booking Form Exercise

The following table shows choices for each select type control.

Table 11.2: Choices for Select Type Controls

Select Type Control	Choices
Adult	1, 2, 3, 4, 5
Child	0, 1, 2, 3, 4
Infant	0, 1, 2
Class	First, Business, Economy

Following are the guidelines to perform this exercise:

- Open Notepad
- Create HTML using a single form and various types of controls. Refer the previous part of the chapter to learn how to use each type of control tag.
- Save the HTML file using .htm or .html extension
- Open the HTML file using web browser to see the result.

If you need any help, please check **Exercise 9 – Flight Booking** chapter to find HTML for the web page above. Best of luck!

Summary

Congratulations, you just finished one of the important concept in HTML. There are so many business applications or programs on internet and most of them use form based web pages. In next chapter, we will learn how to use these control values to perform some business action and calculations. That will make forms more interesting. Till then, keep reading…

Chapter 12 –Brushing with JavaScript

We are very close to finish basic understanding of HTML. You did a great job in finishing all the previous chapters. In the earlier chapter, we learnt about forms and controls and we primarily focused on designing the form on the web page. In real world, forms are used to solve real life business problems such as managing student admission, manage your income tax or book hotel and flight for some nice vacation.

In order to solve business problems, it needs something called **programming**. Through programming you tell your form to perform some business action like the examples stated above. Just as HTML is a language for creating web pages, **JavaScript** is a language for the programming.

Programming is a big topic and will require a separate book for itself. Keeping such magnitude in mind, we have kept programming out of the objectives of this book. But we will do a minimal JavaScript programming in this chapter where we will read values from the controls and try to do some calculation to get a result. This minimal JavaScript will give you a sense of programming and will definitely help in your next level of studies.

Examples of Programming

There is a website http://www.xe.com/ which is used for the currency conversion. For instance, you can enter 1 US dollar and come to know the equivalent value in Pounds or Euros or any other currency. Please go to this web site. On the web page form, enter **10** as value of the currency**,** select **USD - US Dollar** as first currency, select **GBP – British Pound** as second currency and then click on the  image. The steps have been shown in the figure 12.1 below.

C H A P T E R 12

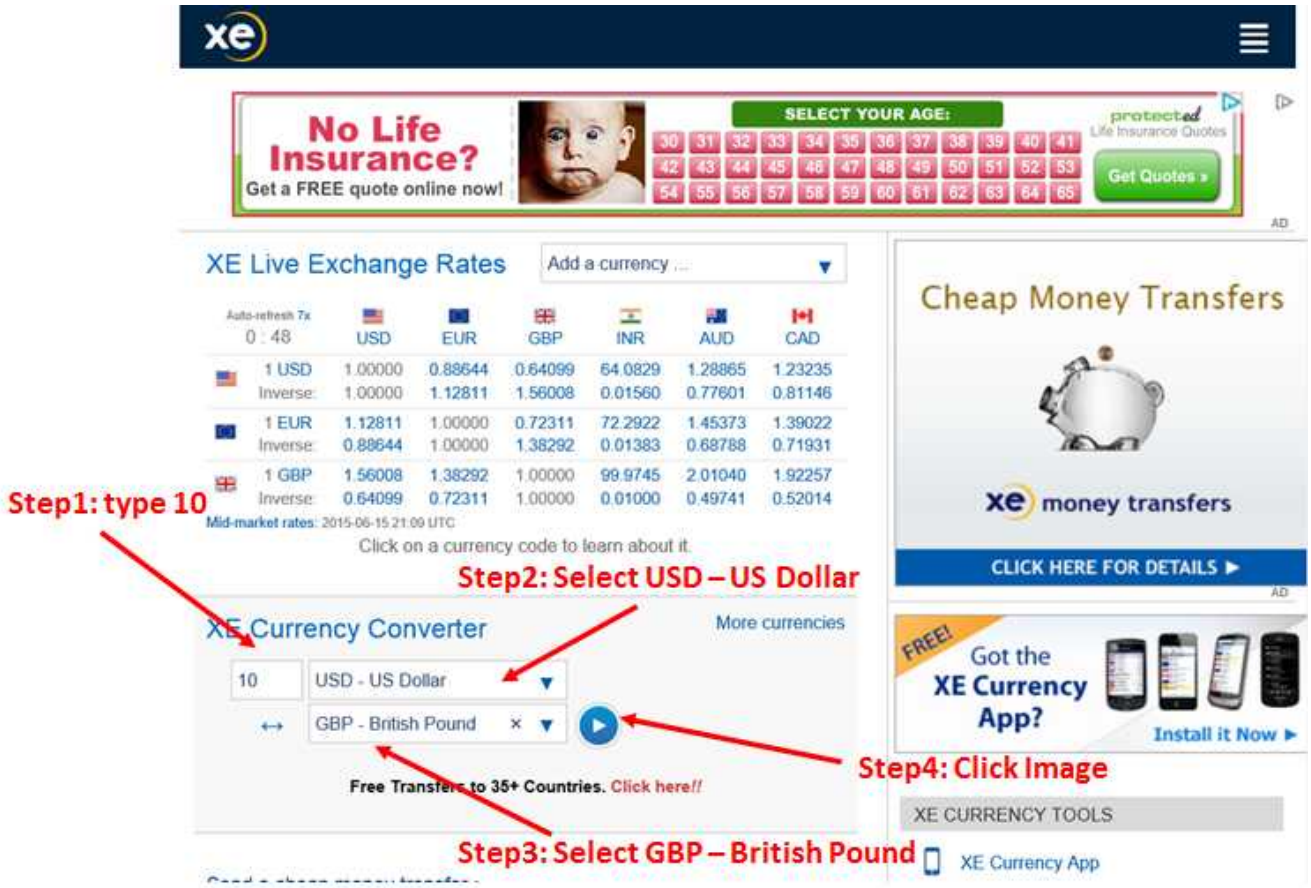


Figure 12.1: Currency Conversion

When you click on the  image, it takes you to a second web page where it shows GBP equivalent for the USD amount as shown in figure 12.2.

C H A P T E R 12

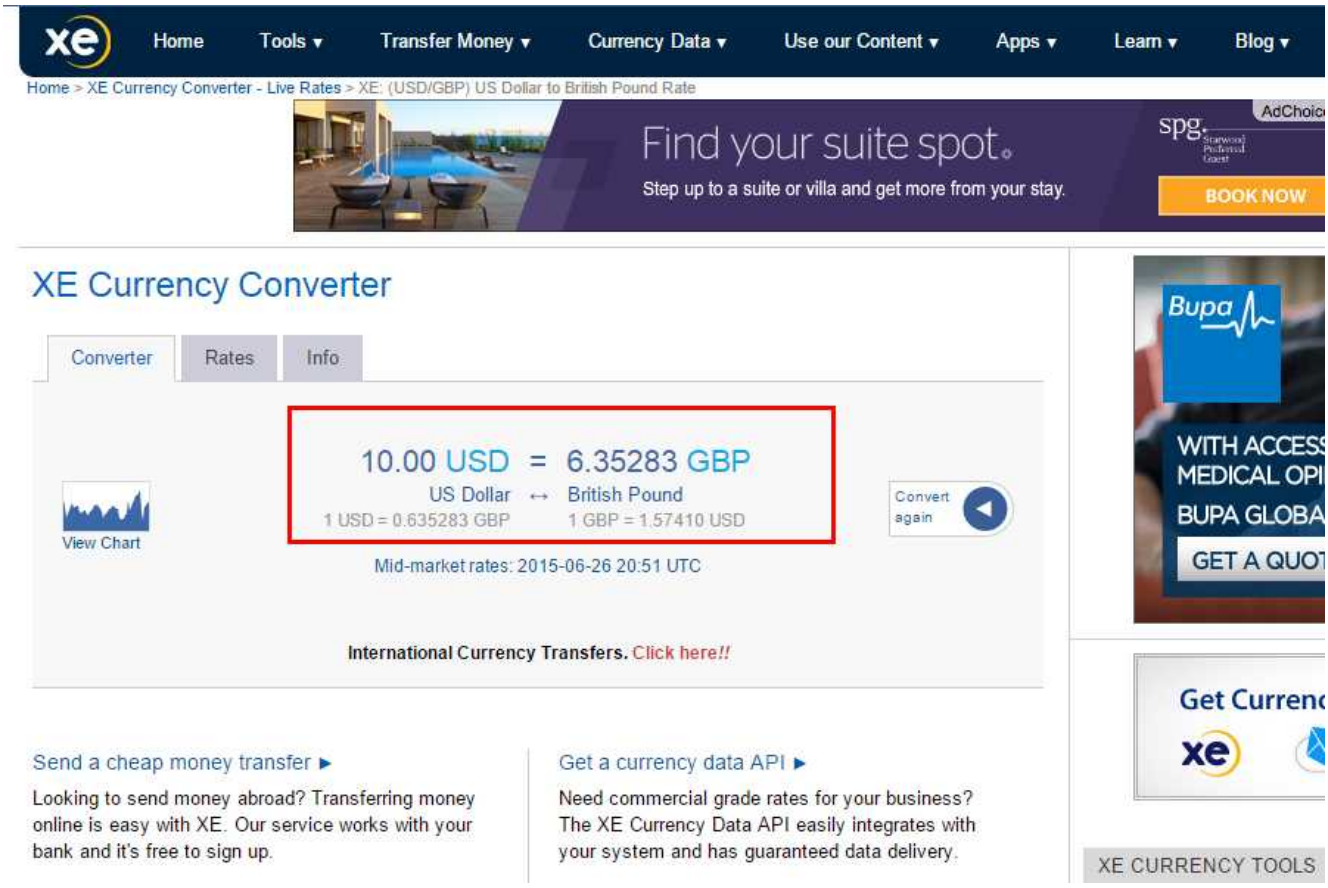


Figure 12.2: Currency Conversion Result

The business problem (i.e. converting one currency value to another) which this website solved has been possible due to programming conducted over the form and controls. The programming read the value of USD currency and then used some calculation to convert that into GBP equivalent and show as result.

We will try to do something similar in this chapter but simple one.

Learning the Details

To learn JavaScript programming, we will create a simple form on a web page and try out simple programming on that.

Here is the HTML for the form as shown in figure 12.3 below.

```
<html>
<head>
<title>Calculation on Form</title>
</head>
<body>
<form>
```

```
Number 1:  ;<input type="text" id="number1"></input>
Number 2:  ;<input type="text" id="number2"></input>
<br></br>
Result:  <input type="text" id="result"></input>
<br></br>
<input type="button" value="multiply" id="multiplybutton"></input> 
<input type="button" value="clear" id="clearbutton"></input> 
</form>
</body>
</html>
```

Figure 12.3: Simple Form HTML

The form has three textbox controls identified with name – **number1**, **number2** and **result**. There are two buttons identified with name – **multiplybutton** and **cleanbutton**. The display text for the buttons are **Multiply** and **Clear.**

The HTML form looks like the following figure 12.4:

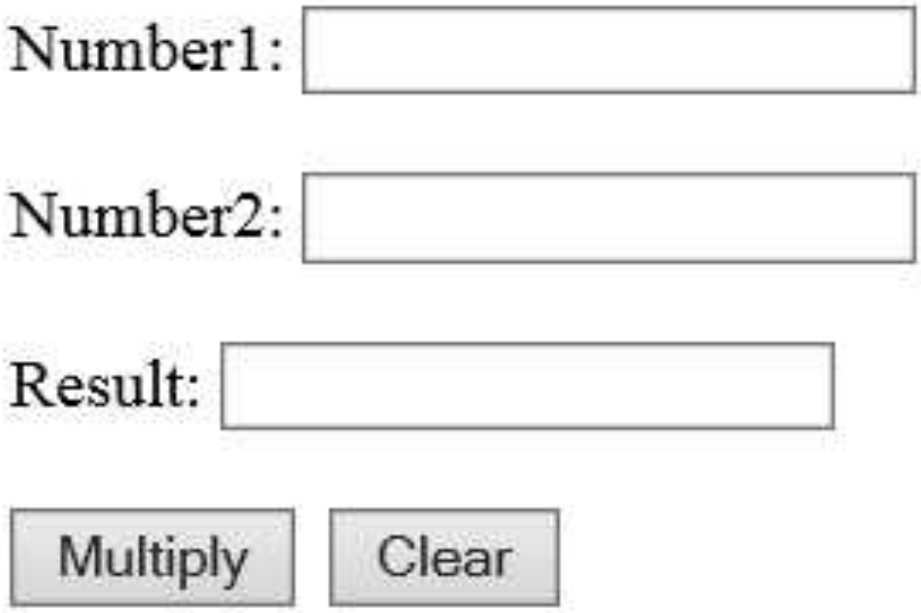


Figure 12.4: Simple Form

In this simple form, we will perform simple programming to meet the following objectives:

- ✓ **Objective1:** When you type a number in **Number1** text box, type another number in **Number2** text box and then click on **Multiply**

button, it will show **multiplication** of both the numbers in **Result** text box.

- ✓ **Objective2:** When you click on **Clear** button, it will **cleanup** or say **remove** any value given in **Number1**, **Number2** and **Result** textboxes**.**

Ready! Excited! Here we go.

JavaScript to Read Value from Controls

To be able to do the multiplication, we should have some way to read number written or typed in the controls. Then only we can apply some mathematical operation on the numbers. How do we read text or value from the control? The following is the JavaScript sentence to read value from the control:

```
document.getElementById ('control name').value
```

I am sure this JavaScript sentence is not making much sense. Let's dissect this statement as shown in figure 12.5 to learn more about it.

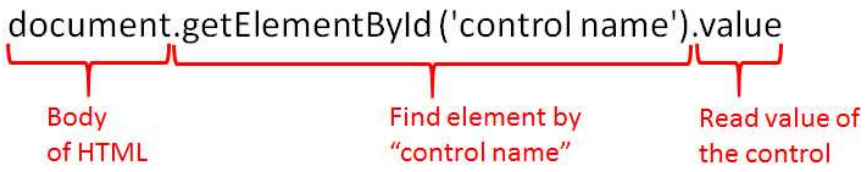


Figure 12.5: Parts of the Statement

The first part of the sentence is the **body of the HTML**. It represents everything in the HTML **between <body>** and **</body>** tags. All HTML tags, elements and controls actually are placed in the body of the HTML. We learnt about this in chapter 2.

The second part is **finding element by control name**. Controls are a type of HTML tags only which are used for user input. You learnt about it in the previous chapter. You must have seen that we are providing **control name** in **Id** attribute. This attribute is used to individually identify the control inside HTML body. Just like your name or roll number is a unique identifier in your class. In this second part, we are saying that find a control with the name "control name" inside HTML body.

The last part says, since you have found the control inside HTML body, go **read value** from this control.

In simple English, this JavaScript statement is saying ***"Find a control with the name 'control name' inside HTML body and read value of it".*** Wow that complex looking JavaScript has such a simple meaning.

JavaScript language is case sensitive. It means you need to write the statement taking care of each capital and small letter as shown in the statement here. If you write all in small letter or capital letters or mix other way around, then it will not work and throw error.

So, if I have to read value of the textbox control named **number1** in the HTML shown earlier, your JavaScript statement will look like following:

```
document.getElementById ('number1').value
```

JavaScript to Write Value to Controls

Now that we know how to read value from the controls, let's also learn, how to write value to a control? After all, the multiplication of number1 and number2 has to be written inside the control named result. It means multiplication value has to be written to the result control.

We will take help of **mathematic algebra** for this. You all must have learnt **expression** in algebra. Look at this expression problem here:

"Write a value of 5 to a term **x** and present as an expression"

You must have solved this kind of problem many times. It is kid's job right? How would the expression look like?

```
x = 5
```

Simple. Let me now change the problem statement in our HTML way.

"Write a value of 5 to textbox control result and present as an expression"

The new expression will look very similar:

```
document.getElementById ('result').value = 5
```

Simple again. All I am doing that I am finding the result control by its identifier name and then writing a value of 5 to it. Exactly similar to term expression above. **This is another JavaScript statement which is used to write some value to a control.**

C H A P T E R 12

JavaScript to Perform Mathematical Operations

We now know both read and write statements in JavaScript. We would now focus on multiplication of number1 and number2. Algebra expressions will again come to our help.

Look at another expression problem here:

"There are three terms – **x, y and z**. Write an expression which multiplies terms **x, y** and assigns the result of multiplication to term **z**"

Little lengthy problem but again simple one. How would the expression look now?

```
z = x * y
```

(*) is sign of multiplication in computer. Similarly, (+), (-) and (/) are signs of addition, subtraction and division respectively.

Let me again change the problem statement in our HTML way:

"There are three textbox controls – **number1, number2 and result**. Write an expression which multiplies values of **number1, number2** controls and assigns the result of multiplication to control **result** as value"

Using the same statement as above, the expression in HTML Controls context will look like following:

```
document.getElementById ('result').value = document.getElementById
('number1').value * document.getElementById ('number2').value
```

This is the JavaScript statement to multiply values of two controls and assign to a third control. Similarly, you can write JavaScript statements for subtraction, addition and division. Only the sign of operation will change from (*) to (+), (-) or (/).

Let's move next. What if there are more than one JavaScript statements? How do we write them? If there are more than one JavaScript statements, then we separate them using **semicolon (;)**. Take a look at the example below:

```
 document.getElementById ('number1').value = 5; document.getElementById
('number2').value = 5
```

There are two JavaScript statements next to each other but separated by semicolon (;).

C H A P T E R 12

JavaScript Events

Now one last thing to learn before we put everything together in a web page. Let me show you again the first objective with the HTML which was shared in the beginning of this section. It is presented in figure 12.6 below.

- **Objective1:** When you type a number in **Number1** text box, type another number in **Number2** text box and then click on **Multiply** button, it will show **multiplication** of both numbers in **Result** text box.

Figure 12.6: Objective 1

I have highlighted the area which I want to discuss now. The objective is to show multiplication of number1 and number2 in result control **when someone clicks on Multiply button**. How would JavaScript programming come to know when someone has clicked the button?

Such things in JavaScript programming are managed by a term called "**Event**". There are various types of events and they raise or occur when something special happens. For instance, in our case, there will be an event called **onclick** for button control which is raised when someone clicks on the button. Similar to onclick, there are many other events such as onfocus, onblur etc. I am not going to go in details because you would need some more knowledge about programming in order to understand it completely.

Focusing back on **onclick** event. Here is how onclick JavaScript statement looks like:

```
onclick="JavaScript Programming Statements"
```

As mentioned earlier, this event occurs when someone clicks on a button. And when this event occurs, it will perform the JavaScript programming statements provided to it. The JavaScript statements could be anything – reading, writing or performing operation as we learnt earlier.

For instance, take look into following example:

```
onclick = "document.getElementById ('number1').value = 5;
document.getElementById ('number2').value = 5"
```

In the JavaScript statement above, when onclick event occurs, it will perform both JavaScript statements which write a value of 5 to both number1 and number2 controls.

How do I tell HTML that this onclick event is for this button? You do that by adding onclick event in the button control tag. Here is the example:

```
<input type="button" id="control name" onclick=" JavaScript Programming
Statements" ></input>
```

Event goes as one of the attribute inside the tag.

Putting Everything Together

Time to put everything together. But before I do that let me show the objective of the HTML form again to refresh memories:

- ✓ **Objective1:** When you type a number in **Number1** text box, type another number in **Number2** text box and then click on **multiply** button, it will show **multiplication** of both numbers in **Result** text box.
- ✓ **Objective2:** When you click on **Clear** button, it will **cleanup** or say **remove** any value given in **Number1**, **Number2** and **Result** textboxes**.**

To meet these objectives, we will perform the following JavaScript programming:

- ✓ Place **onclick event** on **Multiply button**. This event will have one JavaScript programming statement which multiplies values of **number1** and **number2** textbox controls and **writes** to **result** text box control. This will meet objective 1.
- ✓ To meet objective 2, we will place **onclick** event on **Clear button**. The event will have three JavaScript statements back to back separated by semicolon (;). First statement will **write blank value** (two single quotes together '') to **number1** control, second and third statements will write blank value to **number2** and **result** controls respectively. Purpose is to clean these control values. In JavaScript, two single quotes together ('') means cleaning control value.

I have modified the previous HTML with above JavaScript programming and it looks like as shown in figure 12.7.

```
<html>
<head>
<title>Calculation on Form</title>
</head>
<body>
<form>

Number1: <input type="text" id="number1"></input>
<br></br>
Number2: <input type="text" id="number2"></input>
<br></br>
Result: <input type="text" id="result"></input>
<br></br>
<input type="button" value="Multiply" id="multiplybutton" onclick="document.getElementById('result').value=
document.getElementById('number1').value * document.getElementById('number2').value" ></input> 

<input type="button" value="Clear" id="clearbutton" onclick="document.getElementById('number1').value='';
document.getElementById('number2').value=''; document.getElementById('result').value=''"></input>

</form>
</body>
</html>
```

Figure 12.7: JavaScript Programming

The changes are highlighted in the figure 12.7 above. In the HTML above, I have inserted JavaScript programming on onclick event for both buttons. Nothing new, whatever we learnt earlier has been put together to meet some business requirement or calculation.

You can now play with the modified web page and click on Multiply/Clear buttons to see result as shown in figure 12.8.

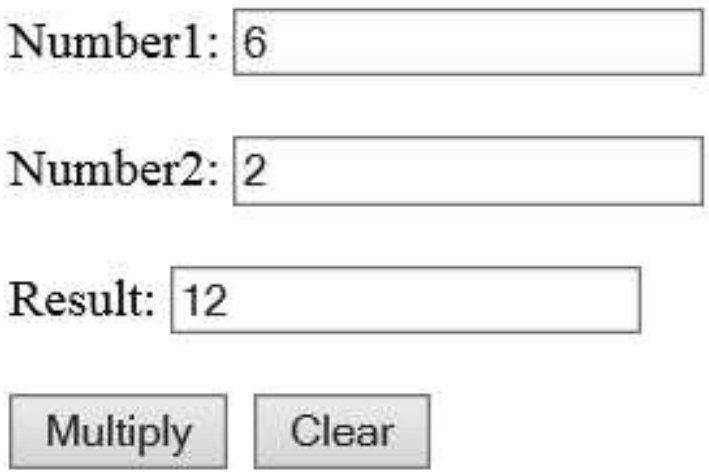


Figure 12.8: Functions in Web Page

Time for another exercise.

Try it yourself

This exercise is very simple. Take the above HTML page and make changes as following:

- Add one more button between Multiply and Clear buttons. This new button should have display text “Subtract”.
- When you click on Subtract button, it will subtract the value of number2 from number1 control and show result in result control.
- Copy the existing JavaScript and make changes such as changing the operation from multiplication (*) to subtraction (-).

The updated HTML should look like as shown in figure 12.9.

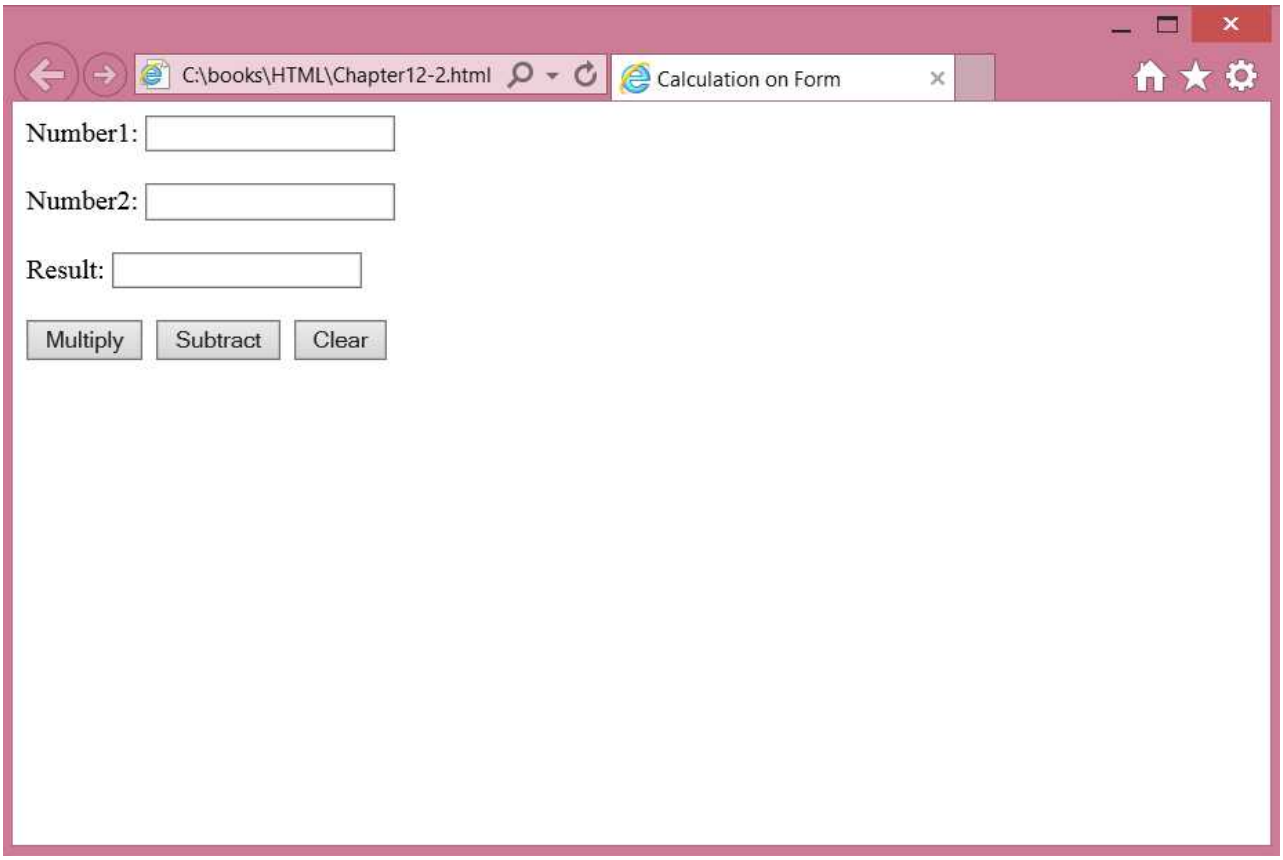


Figure 12. 9: Exercise Web Page

If you need any help, please check **Exercise 10 – Calculation on Form** chapter to find HTML for the web page above. Best of luck!

Summary

This concludes the introductory chapter on JavaScript programming. There are lots more things to learn about programming but you should refer next level of books for the same. The current book is primarily focused on making you take baby steps in learning HTML and use of it.

In the next chapter, you will find a future study reference table which you can use to continue advance topics in HTML and JavaScript programming.

As I always say and believe, keep reading…

Chapter 13 – References for future Study

I am sure you now understand the fundamentals of HTML. Since this book covers basics, you will need further studies to learn advanced topics in HTML. We are recommending the following references for future studies:

Name	Type	Link
HTML Tutorial	Online free tutorial	http://www.w3schools.com/html/default.asp
HTML & CSS: Design and Build Web Sites	Book	http://www.amazon.co.uk/HTML-CSS-Design-Build-Sites/dp/1118008189/ref=sr_1_1?ie=UTF8&qid=1434477419&sr=8-1&keywords=HTML+%26+CSS%3A+Design+and+Build+Web+Sites
HTML references from W3	Online free references	https://docs.webplatform.org/wiki/html
JavaScript Tutorial	Online free tutorial	http://www.w3schools.com/js/default.asp
JavaScript: The Good Parts	Book	http://www.amazon.co.uk/JavaScript-Good-Parts-Douglas-Crockford/dp/0596517742/ref=sr_1_1?ie=UTF8&qid=1434479012&sr=8-1&keywords=JavaScript%3A+The+Good+Parts
JavaScript references from W3	Online free references	http://www.w3.org/standards/webdesign/script

Best of luck with future studies.

Exercise 1 –Greedy Crow

This is the HTML for the exercise given in **Chapter 3 -Headings and Paragraphs.** Use this as answer key or guidance if you face any problem in performing the exercise.

```
<html>
<head>
<title>Greedy Crow</title>
</head>
<body>

<h2>Introduction</h2>

<p>Once upon a time there was a fine pigeon who used to live on a nest

close to a kitchen. The cooks of the kitchen used to like her very

much and often used to feed her grain. She liked the place and her

life was fine.</p>

<h2>Crow meet Pigeon</h2>

<p>One day a crow saw the pigeon and saw how it was getting wonderful

food from the kitchen. Then one day she made friendship with the

pigeon, and under the pretext of friendship, she somehow made the

pigeon to share the nest with the crow.</p>

<p>The pigeon then told her that they could spend time together

discussing politics, religion etc but when it came to food both had

their own ways.</p>
```

E X E R C I S E 1

```
<p>So she suggested the crow to search for her own food.</p>

<h2>Impatient Crow</h2>

<p>But the crow was impatient and the very reason she had made friends
with the pigeon was for the food. The crow wanted meat and all the
pigeon got was grains from the kitchen. </p>

<p>It could not wait any longer and ultimately decided that she visit
the kitchen directly for the food. </p>

<p>Thinking so, it stealthily crept down the chimney into the kitchen.
She got the smell of a fish being curried on the pan. She got greedy
and went ahead and tried to get the fish. But in the process she
disturbed a ladle, and created a noise. This alerted the cook who was
in the neighboring room and he caught hold of the crow and killed
it.</p>

<h2>Moral</h2>
<p>Greed Numbs intelligence. </p>

</body>
</html>
```

Exercise 2 –Greedy Crow Formatting

This is the HTML for the exercise given in **Chapter 4 - Formatting Text**. Use this as answer key or guidance if you face any problem in performing the exercise.

```
<html>
<head>
<title>Greedy Crow</title>
</head>
<body>

<h2>Introduction</h2>

<p>Once upon a time there was a fine <sup>pigeon</sup> who used to live on a
nest close to a kitchen. The cooks of the kitchen used to like her very much
and often used to feed her grain. She liked the place and her life was
fine.</p>

<h2>Crow meet Pigeon</h2>

<p>One day a <sup>crow</sup> saw the pigeon and saw how it was getting
wonderful food from the kitchen. Then one day she made friendship with the
pigeon, and under the pretext of friendship, she somehow made the pigeon to
share the nest with the crow.</p>

<p>The pigeon then told her that they could spend time together discussing
politics, religion etc but when it came to food both had their own ways.</p>

<p>So she suggested the crow to search for her own food.</p>

<h2>Impatient Crow</h2>
```

E X E R C I S E 2

```
<p>But the crow was <sub>impatient</sub> and the very reason she had made
friends with the pigeon was for the food. The crow wanted meat and all the
pigeon got was grains from the kitchen. </p>

<p>It could not wait any longer and ultimately decided that <i>she visit the
kitchen directly for the food</i>. </p>

<p>Thinking so, it stealthily crept down the chimney into the kitchen. She
got the smell of a fish being curried on the pan. She got greedy and went
ahead and tried to get the fish. But in the process she disturbed a ladle,
and created a noise. This alerted the cook who was in the neighbouring room
and he caught hold of the crow and killed it.</p>

<h2>Moral</h2>
<p><b>Greed Numbs intelligence.</b></p>

</body>
</html>
```

Exercise 3 – Master Search Page

This is the HTML for the exercise given in **Chapter 5 –Hyperlinks**. Use this as answer key or guidance if you face any problem in performing the exercise.

```
<html>
<head>
<title>Master Search Page</title>
</head>
<body>
<h2>Master Search Page</h2>

<a href="https://www.google.co.uk/"> Google Search </a>
<br></br>
<a href="https://www.bing.com/"> Bing Search </a>
<br></br>
<a href="http://en.wikipedia.org/wiki/Searching"> Wikipedia Search </a>
<br></br>
<a href="https://uk.search.yahoo.com/"> Yahoo Search </a>

</body>
</html>
```

Exercise 4 – Isaac Newton

This is the HTML for the exercise given in **Chapter 6 –Images**. Use this as answer key or guidance if you face any problem in performing the exercise.

```
<html>
<head>
<title>Issac Newton Introduction</title>
</head>
<body>
<h2>Isaac Newton</h2>

<img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/3/39/GodfreyKneller
-
IsaacNewton-1689.jpg/220px-GodfreyKneller-IsaacNewton-1689.jpg" alt="Isaac
Newton"
width="150" height="200"></img>

<p>
<b>Isaac Newton</b> changed the way we understand the Universe. He discovered
the laws
of gravity and motion and invented calculus. He helped to shape our rational
world
view.
</p>

</body>
</html>
```

Exercise 5–Table of Contents

This is the HTML for the exercise given in **Chapter 7 –Lists**. Use this as answer key or guidance if you face any problem in performing the exercise.

```
<html>
<head>
<title>Table of Contents</title>
</head>
<body>
<h2>Table of Content for HTML Book</h2>
<dl>
  <dt>Introduction</dt>
  <dd>> Learn about web sites, web pages and introduction to HTML.</dd>
  <dt>Elements and Attributes</dt>
  <dd>> Understand fundamentals of HTML in terms of elements and attributes.</dd>
  <dt>Headings, Paragraphs and Formatting</dt>
  <dd>> Learn writting content on the page and formatting to make it effecive.</dd>
  <dt>Hyperlinks and Images</dt>
  <dd>> Learn to use links and images to navigate and make page attactive.</dd>
  <dt>Lists and Tables</dt>
  <dd>> Create list and tables to present options and structued data</dd>
```

EXERCISE 5

```
  <dt>Forms and Controls</dt>
  <dd>> Generate form to allow data entry</dd>
  <dt>Create a project</dt>
  <dd>> Use all of your learnings to create a small project</dd>
</dl>

</body>
</html>
```

Exercise 6 – Student's Marks

This is the HTML for the exercise given in **Chapter 8 –Tables**. Use this as answer key or guidance if you face any problem in performing the exercise.

```
<html>
<head>
<title>Student's Marks</title>
</head>
<body>
<h2>Year 7: Marks of the Students<h2>
<table border="1">
        <tr>
                <th>Student Name</th>
                <th>Mathematics</th>
                <th>English</th>
                <th>Science</th>
                <th>Language</th>
                <th>Total Marks (out of 400)</th>
        </tr>
        <tr>
                <td>Smith</td>
                <td>70</td>
                <td>70</td>
                <td>71</td>
                <td>85</td>
                <td>296</td>
        </tr>
        <tr>
                <td>Jane</td>
                <td>88</td>
                <td>90</td>
                <td>76</td>
                <td>80</td>
                <td>334</td>
        </tr>
```

E X E R C I S E 6

```
        <tr>
                <td>Rosy</td>
                <td>95</td>
                <td>90</td>
                <td>91</td>
                <td>92</td>
                <td>368</td>
        </tr>
        <tr>
                <td>Jasmin</td>
                <td>88</td>
                <td>80</td>
                <td>79</td>
                <td>80</td>
                <td>327</td>
        </tr>
</table>
</body>
</html>
```

Exercise 7 –Greedy Crow in Colors

This is the HTML for the exercise given in **Chapter 9 –Styles on Page**. Use this as answer key or guidance if you face any problem in performing the exercise.

```
<html>

<head>

<title>Greedy Crow in Colors</title>

</head>

<body style="background-color:AntiqueWhite">

<h2 style="color:red">Introduction</h2>

<p style="background-color:LightGoldenRodYellow">Once upon a time there was a fine pigeon who used to live on a nest close to a kitchen. The cooks of the kitchen used to like her very much and often used to feed her grain. She liked the place and her life was fine.</p>

<h2 style="font-size:17" >Crow meet Pigeon</h2>

<p style="background-color:LightGoldenRodYellow">One day a crow saw the pigeon and saw how it was getting wonderful food from the kitchen. Then one day she made friendship with the pigeon, and under the pretext of friendship, she somehow made the pigeon to share the nest with the crow.</p>
```

E X E R C I S E 7

```
<p style="background-color:LightGoldenRodYellow ">The pigeon then told her that they could spend time together discussing politics, religion etc but when it came to food both had their own ways.</p>

<p style="background-color:LightGoldenRodYellow ">So she suggested the crow to search for her own food.</p>

<h2 style="font-size:17">Impatient Crow</h2>

<p style="background-color:LightGoldenRodYellow ">But the crow was impatient and the very reason she had made friends with the pigeon was for the food. The crow wanted meat and all the pigeon got was grains from the kitchen. </p>

<p style="background-color:LightGoldenRodYellow ">It could not wait any longer and ultimately decided that she visit the kitchen directly for the food. </p>

<p style="background-color:LightGoldenRodYellow ">Thinking so, it stealthily crept down the chimney into the kitchen. She got the smell of a fish being curried on the pan. She got greedy and went ahead and tried to get the fish. But in the process she disturbed a ladle, and created a noise. This alerted the cook who was in the neighboring room and he caught hold of the crow and killed it.</p>

<h2 style="color:DarkBlue ">Moral</h2>

<p style="color:DarkBlue">Greed Numbs intelligence.</p>

</body>

</html>
```

Exercise 8 –My Search

This is the HTML for the exercise given in **Chapter 10 -Using Iframe**. Use this as answer key or guidance if you face any problem in performing the exercise.

```
<html>
<head>
<title>My Search</title>
</head>
<body>
<h2>Bing Search:</h2>
<iframe src="https://www.bing.com/" width="1000" height="300"></iframe>
<h2>Wikipedia Search:</h2>
<iframe src="http://en.wikipedia.org/wiki/Searching" width="1000"
height="300"></iframe>
</body>
</html>
```

Exercise 9 – Flight Booking

This is the HTML for the exercise given in **Chapter 11 -Form and Controls**. Use this as answer key or guidance if you face any problem in performing the exercise.

```
<html>
<head>
<title>Flight Booking Form</title>
</head>
<body>
<form>

<h1>Flight Booking</h1>

<input type="radio" id="triptype" >One Way</input>  
<input type="radio" id="triptype" >Round Trip</input>  
<input type="radio" id="triptype" >Multi City</input>
<br></br>

From: <input type="text" id="txtfrom"></input> 
Date (dd/mm/yy): <input type="text" id="txtfromdate"></input>
<br></br>

To: <input type="text" id="txtto"></input> 
Date (dd/mm/yy): <input type="text" id="txttodate"></input>
```

E X E R C I S E 9

```
<br></br>

Adult: 
<select id="adult">
<option> 1</option>
<option> 2</option>
<option> 3</option>
<option> 4</option>
<option> 5</option>
</select>

 Child: 
<select id="child">
<option> 0</option>
<option> 1</option>
<option> 2</option>
<option> 3</option>
<option> 4</option>
</select>

 Infant: 
<select id="infant">
<option> 0</option>
<option> 1</option>
<option> 2</option>
</select>
<br></br>

Class: 
<select id="class">
<option> Economy</option>
<option> Business</option>
<option> First</option>
</select>
<br></br>
```

EXERCISE 9

```
<input type="button" value="Search" id="searchbutton"></input> <input
type="button" value="Cancel" id="cancelbutton"></input>


</form>

</body>

</html>
```

Exercise 10 –Calculation on Form

This is the HTML for the exercise given in **Chapter 12 -Brushing with JavaScript**. Use this as answer key or guidance if you face any problem in performing the exercise.

```
<html>
<head>
<title>Calculation on Form</title>
</head>
<body>
<form>
Number1: <input type="text" id="number1"></input>
<br></br>
Number2: <input type="text" id="number2"></input>
<br></br>
Result: <input type="text" id="result"></input>
<br></br>
<input type="button" value="Multiply" id="multiplybutton"
onclick="document.getElementById('result').value=
document.getElementById('number1').value *
document.getElementById('number2').value" ></input> 

<input type="button" value="Subtract" id="subtractbutton"
onclick="document.getElementById('result').value=
document.getElementById('number1').value -
document.getElementById('number2').value" ></input> 

<input type="button" value="Clear" id="clearbutton"
onclick="document.getElementById('number1').value='';
document.getElementById('number2').value='';
document.getElementById('result').value=''"></input>
</form>
</body>
</html>
```