A GUIDE TO

HTML

A Beginner's Guide to the Hyper Text Markup Language

STONE RIVER
eLearning

TABLE OF CONTENTS

INTRODUCTION

HTML or Hyper Text Markup Language is a standard markup language that is absolutely essential for creating web pages. If you aspire to be a web designer or web developer, HTML is where you start; this is where you take your first step towards building your own, fully functional websites. In this eBook, we are going to look at the very basics of creating an HTML document, then move on to how the various aspects of HTML code can be used to build entire sites. To get started, we are going to walk through some of the terminology that a beginner in HTML must be familiar with.

If you're interested in learning HTML or the professions of web design or web development, Stone River eLearning has individual courses and course bundles that will take you from a complete beginner to a professional designer or developer at a very affordable price.

HTML Fundamentals

Become a Professional Web DesignÅÒ Bundle

Become a Professional Web Developer Bundle

HTML TAGS

What are HTML tags?

HTML tags are the keywords in an HTML document that are not displayed on the browser window but define how contents are formatted and displayed on the browser window. An HTML tag usually has an opening and a closing tag. Eg: <head> is an HTML tag with <head> as its opening tag and </head> is the closing tag.

Points to Ponder:

1. It must be noted that the HTML tags or keywords, as some refer them to; are included inside the angular brackets.
2. For writing the HTML script, you can use any of the text editors of your choice as the working or the use of HTML tags would not differ from one text editor to another.
3. An HTML document must start with a <html> opening tag and must end with a </html> closing tag.
4. The title of the web page is included inside the <title> tag which is again inside the <head> tag of the document. The title of the web page is the name that appears on the browser window.
5. The content to be displayed on the web browser is included between the <body> and </body> tags.

6. The formatting and styling of content to be displayed on the web browser is done using other tags available in the HTML language.

The developers of HTML have updated the language time and again to ensure that web designers find it easy and efficient to design their own web pages using HTML. The most recent update of HTML is called HTML5 and it came into existence just a year and a half ago. One of the prominent updates that HTML5 has got is the ease in format of the doctype declaration. Now only a "<!DOCTYPE html>" as declaration is sufficient as compared to the lengthy "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.xyz.com/TR/html4/loose.dtd">" in the previous versions of the language.

Eg:

```
<!DOCTYPE html>

<html>

<head>

<title> My First WebPage </title>

</head>

<body>

Hello World.!!

</body>

</html>
```

In the given example, the <html> tag begins the HTML document. The title of the page is “My First WebPage” and it is included inside the “<title>” tag of the document. The content typed inside the “<body>” tag is displayed on the web browser window.

Note:

1. The most recent HTML tag must be closed first before the other tags in the document. Eg: The “<title>”tag is closed before the “<head>” tag as it is the most recent HTML tag opened.
2. The HTML language does not include blank spaces in the display on the browser window, therefore, despite HTML will ignore all the blank spaces you give in your document.
3. There is a separate tag for giving line breaks and blank space on the browser window.

DOCUMENT OBJECT MODEL

DOM, or Document Object Model, defines a structured representation to organize the HTML documents in a tree structure. From a technical point of view, DOM is a language independent and cross platform convention for the interaction and representation through objects in HTML. The HTML document that we have written on our text editor is parsed by the browser and turned into a structured representation through Document Object Model. In simpler words, DOM is a model where the document of the Webpage, i.e objects like elements, links etc can be manipulated.

Why is DOM useful to me?

The Document Object Model provides a very versatile view of the contents in a document. DOM is useful for a web designer as it allows the data in the document to be manipulated by other routines in a way that the manipulations can be reused with other DOMs. As a programmer, you can also make use of the solutions that are already present for a particular DOM.

1. Programmers who are inclined towards learning and are ready to invest their time and resources in it, may use the DOM calls to apply the skills acquired.

2. By switching to DOM which is a cross platform convention, scripts that will work effectively in on all browsers can be written.
3. On a larger scale, a solution may be formulated for any problem with the whole server by using the different DOMs.
4. The DOM is a better representation because it makes it easier to unplug the existing connections and plug in to new components with minimal recording.

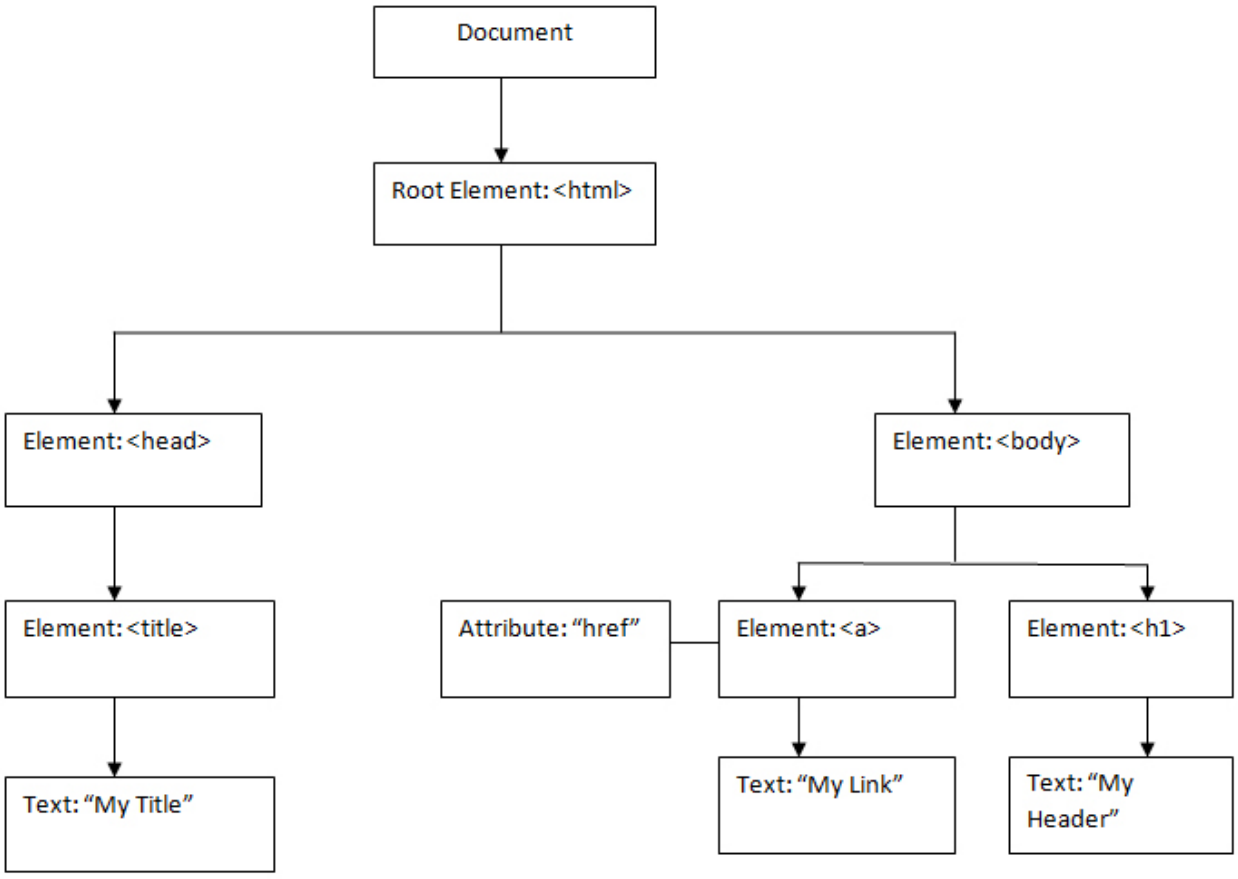
Note:

- DOM implementations may vary in size and memory demand, with a few also differing in performance of individual operations.

Eg:

```
<html>

<head>

<title> My Title </title>

</head>

<body>

<h1> My Header </h1>

<a href="http://xyz.com">My Link</a>

</body>

</html>
```

A structured representation of the DOM tree of objects in a document looks something similar to this:



TEXT EDITOR

A lot of people, especially beginners in HTML, have doubts regarding the text editors and how to use text editors for writing HTML codes. Text Editors provides the space where a user can write the codes for his HTML document. Running this document in the web browser gives the web page that the user intends to create.

How to create an HTML document in a text editor?

Creating an HTML in a text editor is easy. All you need to do is, open the text editor, write the code in the space provided, save it by a ".html" extension and then open the file in the web browser. However; it is not creating an HTML document, but choosing a text editor that creates dilemma in the minds of young programmers. Therefore, in this chapter, we will discuss about text editors and choosing a text editor.

Which is the best text editor?

There was a time when Notepad was the most used text editor and beginners in web development and HTML had to bear the pain of writing their code on the plain and dull white screen of Notepad. With no numbering of lines and no in built functionalities, writing the HTML code and debugging was quite a difficult process. However; with time, text editors have seen some great changes and now text editors like Notepad++, ConText and Sublime Text are some of the text editors that have contributed in making the overall process of writing a code and building an application fun and interesting. However, classing one text editor over others would be unfair because they are all effective in their own way. Therefore, selecting a text editor depends on the choice and the comfort of the programmer.

The other major upgrade that text editors have received is the edits and modifications that you can do to the text. Now creating an HTML document is not only about writing the text on the window, but you can also customize it by selecting your own settings. You can change the background colour, font size, font style, font colour and other changes. One can customize the settings in his text editor and make it look more attractive and interesting.

The steps to customize the text editors are:

1. Go to the PREFERENCES menu
2. Select Fonts and Colours from the list of menus.
3. In the window that appears, one can make alterations to the font and other settings.

Thus, using the text editors, one can make writing HTML codes in a document all the more interesting and amusing. So select the text editor of your choice and start creating your HTML document, in a way that you enjoy.

CODING PRACTICES

In this chapter, we will be discussing about good coding practices in HTML and how to move forward during your initial days of learning HTML. HTML tags form the basic and most fundamental step of hypertext markup language. Just like any other programming language, one gets better at HTML after nurturing his/her coding skills through regular practice. However; since, through HTML you can witness your code step by step resulting into web pages, it is important to build your confidence and interest by witnessing your efforts reaping fruits on the web browser.

Good Practices for learning HTML as a beginner:

1. **Do not be lazy**

The first and the most important rule of coding is to be persistent with your work. A lot of people start on the path of web designing and start with HTML, but soon become lazy and leave it in between. If you aspire to be a good web designer and if you are really determined to attain your goals, the first roadblock that you will face is your laziness and once you have crossed that, half the battle is already won.

2. Control + C is your biggest enemy

Ctrl +C is, and probably has been your most loyal friend since you were in school, however; it is time to bid good bye to that loyal friend. If not a permanent goodbye, a temporary exile is beneficial for both parties (actually its only beneficial for you, the other party is non-existent anyways). The more you write the more you learn. Remember, the time when as a kid we were given the lesson of learning by writing things down? Well, it is time we scroll back in the memory lane and revisit that lesson. The only difference is, now you have to write code as an HTML document, and it is much more interesting than primary school lessons.

3. Have a look at your code

As a beginner in HTML, we all have this habit of blazing towards the web browser and refreshing the page to see the results of the changes in code that we have made. Even though it is not a bad habit as it saves time, but if you have just started to learn HTML, it is better to look at your code and check for possible errors before you check the results on the browser window. Not only will it help you avoiding similar mistakes in future, but a correct result on the browser window is a far more encouraging sign than writing the HTML code before time.

4. Avoid shortcuts

Though features in the text editors features like auto complete which are used to create the environment for your HTML document is a great utility when you have gained experience as a programmer; as a beginner, it is always a better option to write the opening and the closing tag by yourself.

5. Do it the hard way

It is always better to do it the hard way to ensure that when the hard times come, you are better prepared to deal with them. Paying attention to the smallest details is something that people often miss in a language like HTML, but the truth is, it is these small details that separate a good coder from an ordinary one. Now it is your choice, are you satisfied with a normal coder tag, or are you willing to put in a little extra to gain a lot more in return.

You can turn on/off the auto complete feature from your text editor by following the steps mentioned below:

1. Go to the PREFERENCES menu
2. Select code intelligence from the list of menus.
3. Now you can select/deselect the automatically insert end tag option as per your choice. However; we recommend not selecting it until you are familiar and comfortable with HTML.

VALIDATION

HTML validation is a program to ensure that the quality of the HTML document meets the standards set by the World Wide Web Consortium (W3C). The HTML validation process performs syntax checks on the document and ensures that the document runs smoothly and uniformly across all web browsers and different platforms. The HTML document validation is done through w3c markup validation service on the validator.w3.org website.

The steps to complete the validation of a HTML document are:

1. Create the HTML document and save it.
2. Open the validator.w3.org website on your browser.
3. You will find three options at the top of the page, 1. Validation by URI, 2. Validation by File Upload, 3. Validation by Direct Input
4. You can either directly upload the saved document file on the validator or you can copy paste the content inside the file.
5. Once you click the check option, the validator will perform a quality check on the content and give a report accordingly.
6. You can check the report and make changes to your file, if required.

What is the use of an HTML validation?

The HTML validation ensures that your content does not face any problems when it is displayed as a web page. Performing a validation also ensures that the document does not have any issues with uniformity from one browser to another. However, these are not the only benefits of using a HTML validation service and below are some more advantages of w3c validation of HTML documents.

1. An HTML validation is a useful tool for users receiving data from a number of sources, electronically.
2. Errors such as extra spaces, open tags, etc can cause major difference in the final output of the
3. Manual Validation is a time consuming and tedious work and it becomes all the more difficult when CSS validations and javascript validations are also included in the document.

Eg:

A HTML document that shows warnings during validation

```
<!DOCTYPE html>

<html>

<head>

<title> Validation </title>

</head>

<body>

This is a simple HTML document that needs validation

</body>

</html>
```

Now the same document post validation and corrections

```
<!DOCTYPE html>

<html>

<head>

<meta charset= “UTF-8”>

<title> Validation </title>

</head>

<body>

This is a simple HTML document that needs validation

</body>

</html>
```

COMMENTS

Moving ahead on good coding practices in HTML, today we are going to discuss about another important point that many programmers tend to ignore in their codes. As a beginner all of us must have been taught the lesson of putting comments in our code, however; majority ignores it, thinking that the teacher was just trying to make his job easier, only to realize in future that commenting our code only makes everyone's job easier. When you are working on an application or a big project, it is very likely that you will be working as a team with each member taking care of a certain functionality or section of the program. In such a scenario, having comments in the code allows the other programmer to understand your logic and idea behind the code that you have written. Since, ultimately the work of each programmer will be synchronized together to make the complete application, having comments gives the advantage of having the synchronization from the very start.

How do you insert comment into an HTML document?

Just like any other programming language, inserting comment inside an HTML document is extremely easy and beneficial. As explained, all you need to do is:

1. Insert the less than symbol followed by the exclamation mark and two hyphens.
2. Put in your comment after this.
3. Once you are done with the comment, close it by two hyphens and a greater than symbol.

Eg:

```
<!DOCTYPE html>

<html>

<head>

<meta charset =”UTF-8”>

<title> Comments </title>

</head>

<body>

This is a simple text

<!--This is a comment -->

</body>

</html>
```

In the above example, a comment is included in the body section of the HTML document.

Points to Ponder:

1. Small and Precise:

A comment must be as small and precise as possible. You do not want to write a long comment on a simple operation as it might distract the other programmer from the job in hand. On the flip side, the comments must not be so vague that they do not make any sense.

2. Do not pile up comments unnecessarily

Sometimes, in their quest to write comments on the code and make it self-explanatory, programmers write comments for each and every step of the code, making it unnecessarily long and inconclusive. A comment is there to aid the developer, not to make it difficult for him to find the actual code between the piles of comments that surrounds it. Only write comments for the operations that might need an explanation and not for the each and every line of the code.

Bad Practice:

```
<!DOCTYPE html>

<html>

<head>

<title> Bad Practice </title>

<body>

Name: XYZ    <!--Name of Student -->

Age: 19   <!--Ageof Student -->

Gender: Male   <!--Gender of Student -->

Hobbies: Coding, Sports  <!--Hobbies of Student -->

Future Goals: To become the best Web Designer   <!--Future Goals of Student -->

</body>

</html>
```

Better Approach:

```
<!DOCTYPE html>

<html>

<head>

<title> Better Approach </title>

<body>

Name : XYZ

Age : 19

Gender : Male

Hobbies : Coding, Sports

Future Goals : To become the best Web Designer

<!--Student Details -->

</body>

</html>
```

BLOCK LEVEL ELEMENTS

An HTML document comprises of elements that define the functionalities of the web page and decide its operations and the characteristics that it will possess. HTML elements are of two different types, namely, block level elements and inline elements. In this chapter, we will discuss about the different block level elements in HTML and understand their functioning.

What is a block level element?

A block level element occupies the entire space of the container, which is its parent body, and creates a block to display the contents on the web browser. Web browsers display the content before and after a block level element, in a different line.

Eg:

```
<!DOCTYPE html>

<html>

<head>

<meta charset =”UTF-8”>

<title>Block Level Elements </title>

</head>

<body>

This text is before a block level element. <div> This is a block level element </div> We are out of the block level element.

</body>

</html>
```

The output of the above code will look like this in the browser window:

This text is before a block level element.

This is a block level element

We are out of the block level element.

What are the different types of block level elements?

Block level elements are extensively used to arrange content on the web page and also give continuity and relevance to the content. If you need the content to show in the form of a paragraph, you will need a block level element, if you need it in the form of lists, you will need a block level element, and you will have to use a block level element even for the job of assigning footers to your webpage. You webpage cannot be completed without the proper use of block level elements in it and therefore, it is important that we have knowledge about the different block level elements and their functionality.

1. ** **

The element can be used as a container for other block level elements. In other words, element can be considered as a mini block level element inside the primary block level element.

2. **<hr>**

There are very few elements in HTML that do not require both the opening tag and the closing tag, and <hr> element is one of those few. The recent versions of HTML have allowed <hr> tag to be used without a closing tag. The <hr> element is used to get a horizontal rule, i.e dividing line in the web page.

3. <p>

The <p> element puts the content inside it in a separate paragraph. The <p> element is highly beneficial for web pages where a lot of text needs to be put up on display. Eg: The 'History' section of a company's webpage.

4. and

The ordered list <ol> and unordered list <ul> are used to display the content in the form of a list. The ordered list displays contents in the form of numbers while the unordered list displays it in the form of bullets. The <li> element (list items) is used inside the ordered/unordered list to display the content.

5. Heading Elements <h1> to <h6>

The heading tags are used to highlight a certain section of the content on the web page. There are six types of heading elements available in HTML, starting from <h1> to <h6>. The <h1> element displays the content inside it in bold and with the biggest font size. The size decreases as we move from <h1> element to <h6> element.

The block level elements are easy to use and have great utility in displaying content on the webpage. Every webpage, no matter how big or small, will use some kind of block level element in its code to make the content look organized and progressive.

LINE BREAK AND SPACING

In one of our earlier chapters, it was mentioned that there are separate HTML elements or tags to include a line break of an extra space between the texts. Well, now is the time that we introduce you to the HTML line break and spacing elements. However, before we start, it is important to understand that the web browser ignores all the extra white spacing between your texts and displays just a single space between two words.

CASE 1:

This is my text and I am checking something.

CASE 2:

This is my text and I am checking something

The above two cases will have the same result on the browser window as it will ignore all the extra space between the words "and" and "I" in CASE 2, and display it in the same way as CASE 1.

How to insert extra space between texts?

The natural question that pops up in our mind after we realize that even by multiple click of the space button on the keyboard, the browser would still display only one space is, “How can I have extra spaces in my text then?” This is where the HTML’s non-breaking space comes into the picture. Typing“&ndsp;” gives you extra bit of space between your texts as non-breaking space is actually a standard space with the browser not breaking or wrapping a line of text at its point of occurrence.

When to use “ ”? When to avoid it?

The problem with “ ” is that many browsers ignore the consecutive appearance on the document and output a single or a couple of extra spaces. It depends on how your browser interprets the document, however; to be on the safer side, it is advisable to use “ ” only when you need one or two spaces between the texts in your document.

How to insert line break in HTML?

Now that we have discussed about inserting extra spaces between texts, the next hurdle is inserting a line break in the text displayed on the web browser. Since web browsers ignore all the extra spacing between texts, they will surely ignore if we try to give line break by pressing the “Enter button”. So how should be get a line

break between texts? The answer is by using HTML elements, the
 element to be precise.

The
 HTML element falls under the inline elements category and is quite similar to the <hr> element discussed during the chapter in block level elements. Just like the <hr> tag, the
 tag does not need a closing tag and it gives a line break between the texts written just before and just after it.

Eg:

```
<!DOCTYPE html>

<html>

<head>

<meta charset =”UTF-8”>

<title> Line Break </title>

</head>

<body>

This text is before the line break

<br>

This text is after the line break.

</body>

</html>
```

There is another way of inserting line breaks between texts in HTML, one that we have already discussed before. We discussed in length about the paragraph tag “<p>” in the chapter on the block level elements, however; there is one added advantage of the paragraph tag and it is the line breaks that it inserts between the texts. The “<p>” element arranges the text between it in the form of a paragraph and also introduces line spacing and breaks similar to that of a paragraph on the browser window. In simpler words, it means that two set of texts inside separate “<p>” tags will have a line break similar to that of a paragraph between them.

Eg:

```
<!DOCTYPE html>

<html>

<head>

<meta charset =”UTF-8”>

<title> Line Break by Paragraph </title>

</head>

<body>

<p>This is the first paragraph </p>

<p>This is the second paragraph. </p>

</body>

</html>
```

ID ELEMENT

Have you ever visited a webpage that has large chunk of text on display with all the content grouped into different sections? How did you feel about scrolling through all that pile of text before you reach the section that you wanted to read? Surely, it is quite frustrating and time consuming. However; what if you could directly jump to the section of your interest by just a single click of your mouse button? Well, some web pages already have this functionality and you can add it on your web page as well. Having discussed a lot about good coding practices and a few important elements and tags in HTML, it is now time that we shift our attention from the programming part and look towards a topic that deals more with how HTML works on the outside.

How can it be achieved?

HTML allows users to create links in their webpage which can be used to visit other pages. However; have you ever wondered if these links could be used to visit a different section on the same page? You must be thinking, how is this possible? We need to give the name of the html document with the link, what will we give in case of the same page? This is a very common and fair doubt and we will introduce a new HTML entity before we solve it.

What is an id in HTML?

An id element is used to specify a unique id to other elements in an HTML document. In other words, you can assign different settings for a paragraph tag “<p>” and an ordered list “<ol>” by using the ids.

Thus, by using the ids, we can differentiate between the different sections in an HTML document. However; it still does not solve our problem. How will we link a section on the same page in HTML? Let us try to find an answer with an example.

Eg:

```
<!DOCTYPE html>

<html>

<head>

<meta charset =”UTF-8”>

<title> ID Example </title>

</head>

<body>

<h2  id=”top”>These are my details </h2>

<p> My name is XYZ </p>

<p> My am 19 years old </p>

<p> My love HTML. </p>

<p> My hobbies are watching movies and playing sports.</p>

<p> I want to get better at HTML. </p>

<a href = “#top”>Go to my details! </a>

</body>

</html>
```

In the given example, the level two heading is given an id “top”. This id is thus associated with this particular element on the HTML document. The anchor tag is used to create a link at the end of the body and the reference of the link is set as “top”. Thus, the link will lead to the level two heading with the id of “top”, if it is clicked by the user.

Note:

1. The id is unique for a single element and therefore it is different from a class, which can be used for a number of elements.
2. A “#” sign must be used before the name of the id in the anchor tag. The “#” sign is used to tell the browser that the link is referring to an id or an anchor.
3. Using ids in the HTML document is a good practice is it makes styling of the document using CSS a much easier and effective process.
4. Creating a link on the same webpage finds great utility in pages where there is a lot of text and the same is effectively used in e-books available on the internet.

INLINE ELEMENT

Having talked in length about the block level elements in a previous chapter, we will now discuss about the other types of elements in HTML, namely, the inline elements. The biggest characteristic of an inline HTML element is that, unlike block level elements it starts from the same line in the web browser. Inline elements are also phase level elements and 'text-level –elements', however; inline elements is the name most popularly known. The best and the most used example of block level elements are "" and "". We will discuss these two elements in depth later in this chapter.

What are the major properties of inline elements?

Just like block level elements, inline elements also have some specific properties that make it different and useful at the same time.

1. Inline elements do not clear the previous content as it flows along with the text content.
2. Inline elements ignore the width and height properties of the HTML web browser and will also ignore the top and bottom margin settings. However; the left and right margin setting and the padding are not ignored by inline elements.

How are inline elements different from block level elements?

The biggest and the most fundamental difference between block level elements and inline elements is the one that while block level elements display content before or after it in different lines, inline elements are known to display contents on the same. However; this is not the only difference between block level elements and inline elements.

1. Block level elements create larger structures than inline elements.
2. Block level elements may contain other blocks, inline elements or data elements in it.

If there are block level elements, 'big' structures and inline elements are present in a document, it is always a good approach to establish the block level elements first and include the big structures inside them; and then shift attention towards the inline elements.

**** element: Probably the most widely used inline element, span is a generic inline element and is used for grouping elements together on a document. It is usually a great practice to group the elements together so that styling of these elements can be done easily.

**** element: No matter how your webpage looks like or what operation it performs, an inline element that will surely find great

utility is the strong element. The strong element is used to perform the same operation that the bold button on the word document is used for. If you want your text in bold, all you have to do is include it inside the strong elements.

**** element: Just like the element we discussed above, the element also has great utility and application in web pages as it converts the text within it into italics. "" element is called emphasized element and is used in the same way as the "" element.

Eg:

```
<!DOCTYPE html>

<html>

<head>

<meta charset ="UTF-8">

<title> Inline Elements </title>

</head>

<body>

<span> I am inside the inline element. </span> Now I am outside the inline element.

<br>

<strong> This is bold and beautiful. </strong>

<em> This is italics and stylish </em>

</body>

</html>
```

ANCHOR ELEMENT

Talking of inline elements, there is one specific HTML tag that is used extensively in applications no matter big or small, to connect the different pages to each other. No prizes for guessing that we are referring to the anchor tag which is denoted by the “<a>” opening and closing brackets. Now it is important to note that while anchor tag is denoted by “<a>” and is used to link one page to another, this anchor tag holds little significance without the “href” attribute which helps browser get signals towards the links destination. We have already discussed about how the anchor element can be utilized to jump to different locations on the same page, therefore, in this chapter, we will be discussing about the ways to use anchor element to jump to a different page, or a link on the internet. Before we start, let us learn a bit more about the anchor tags.

- A page linked to an HTML document using the anchor tag is normally displayed in the same browser window, unless otherwise specified.

- To display the link in a different window, you must include the “target = _blank” attricbute inside your anchor element.
- The default properties of the links appearing on the web browser are:
  - Unvisited links appear underlined and blue in colour.
  - Visited links appear underlined and purple on the browser screen.
  - An active link appears underlined and red in colour.

Now before we discuss about linking different pages to an HTML document using the anchor element and the “href” attribute, it is important to understand that there can be three ways in which these links can be established.

1. When the document to be linked is in the same folder as the root file.
2. When the document to be linked is in a different folder of the same system, as the root file.
3. When the page to be linked is on the internet.

We will take one case at a time and try to explain the ways of creating the HTML link to the other document in each case. It is important to note that there are only a few, very small changes required to execute the link in the three cases; however, missing these changes might lead to an error in the link.

Case 1: Document in the same folder

Eg:

```
<!DOCTYPE html>

<html>

<head>

<meta charset =”UTF-8”>

<title> Link Case 1 </title>

</head>

<body>

<a href = “Page2.html” > Link to the Next Page </a>

</body>

</html>
```

Case 2: The HTML page is in some other folder on the same system

Eg:

```
<!DOCTYPE html>

<html>

<head>

<meta charset =”UTF-8”>

<title> Link Case 2 </title>

</head>

<body>

<a href = “. ./ home/index.html” > Link to a page in a different folder </a>

</body>

</html>
```

Case 3: The HTML page links to a page on the internet

Eg:

```
<!DOCTYPE html>

<html>

<head>

<meta charset =”UTF-8”>

<title> Link Case 3 </title>

</head>

<body>

<a href = “http://google.com” > Link to a page on the internet </a>

</body>

</html>
```

Points to Ponder:

- The Case 1 when you need to link an HTML document in the same folder does not need anything other than the file name in the href attribute to perform a successful link.
- The Case 2 needs the user to mention the complete url to complete link. It is important to note that a complete url includes the “http” attribute is included.
- The Case 3 needs you to move out of the current folder where the root document is saved and to the folder where the document to be linked is saved. Only when you reach the document you can create a link to it.

IMAGES

When we are talking about web pages, one of the first things that come to our mind is all the images we see on the internet. Be it a sports star, a Hollywood celebrity or a big politician, pictures of them circulate day in and day out in the social media and across online news portals. Have you ever wondered how these images are posted on the internet? How are images actually uploaded on a webpage or an HTML document?

Uploading images on a webpage requires a very simple command, however; the important thing is to arrange and adjust that image on the webpage such that it appears beautiful and relevant. The <img> is used to put up an image in the HTML document. However; just like it was in the case of the anchor tag discussed in the chapter on the links between HTML documents, the "<img>" tag holds little significance without the "src" attribute used alongside.

While we have already discussed the “img” element, the “src” element is used to specify the url of the image. Before we go ahead, let us look at a simple example of an image being displayed on an HTML page.

Eg:

```
<!DOCTYPE html>

<html>

<head>

<meta charset =”UTF-8”>

<title> Image Display </title>

</head>

<body>

<img src = “abcd.jpg” >

</body>

</html>
```

Note:

The name of the image can be directly specified if it is present in the same folder as the HTML document. However; in case the image lies in a different folder, we must specify the complete location of the file. The complete location is available in the properties of the image, from where it can be copied to the HTML document.

That's it? We just have to use the "<img src>" tag and we can put any image on the HTML webpage in whatever way we want? Well, the answer is yes and no. Yes, we can put any image on the HTML document and subsequently on the web browser by using the "<img src>" tag, however; it might not be in the way we want it to look like. To ensure that the image looks exactly of the size we want, we can use some other attributes along with the "<img src>" element.

Eg:

```
<!DOCTYPE html>

<html>

<head>

<meta charset ="UTF-8">

<title> Image Display </title>

</head>

<body>

<img src = "abcd.jpg" height ="300" width="300" alt="The picture of a Lion" >

</body>

</html>
```

Note:

1. In the above example, the width and the height attributes are used to specify the width and height of the image to be displayed.
2. It is important to note that the width and height attribute are used inside the “<img src>” element and the input given to these two attributes are in pixels.
3. You can also give the input to the width and the height attribute in terms of percentage. Giving the input in terms of percentage, commands the web browser to multiply the size of the image to that percent and then display it on the screen. Thus you can both decrease and increase the size of the image by this function.
4. The “alt” attribute used in the end is of great utility as sometimes the web browsers fail to load the image from the HTML document.
5. The “alt” attribute, or the alternate text, displays the text mentioned in it in place of the image, in case the image fails to load on the browser. If the image loads successfully on the browser, the alternate text is not displayed. Therefore, using alt text is a good practice as it keeps your web page prepared for any adverse situations.
6. Since the alternate text gets displayed when the image is not loaded, it is important to keep the text short, but more importantly highly relevant and explanatory.

Thank you for reading!

We invite you to share your thoughts and reactions

STONE RIVER
eLearning