

INTRODUCTION TO

HTML5 & CSS3 FUNDAMENTALS

A Guide to HTML5 and CSS3 Fundamentals



TABLE OF CONTENTS

- Introduction
- HTML Page Basics
- CSS Basics
- Link Basics
- In-Page Links
- Link Styles
- Navigation Menus
- Arranging Elements
- Block Spacing
- Positioning Elements
- Tables
- Table Styles
- Form Basics
- HTML Form Elements
- HTML5 Form Elements
- Form Styles
- Streaming Contents

INTRODUCTION

HTML5 technologies are the latest version of Hyper Text Mark Up Language (HTML), which has been made official by the World Wide Web Consortium (W3C) to be the new standards for the Internet and the future of both personal computing and mobile computing.

The core reasons as to why HTML5 has been official standard for new both personal and mobile computing include:

- It supports the multimedia content with the provision of features that permits the scriptable rendering of 2D shapes. The HTML5 technologies aim at substituting Rich Internet Application (RIA) technologies, such as the Adobe Flash technology.
- You don't have to install the plugins or player applications, as was the case with previous HTML technologies, to access and view multimedia content on your browser.
- HTML5 content can run on any type computer system. This technology can help you run content on any mobile device—whether it is an iPhone, Android devices, an iPad, a tablet or any smartphones— which cannot run Flash applications.
- It is easy to use as it provides an interactive platform to help in the creation of presentations and websites that can be designed with minimal effort.

What about CSS3?

Well, CSS3 is the latest version of Cascading Style Sheets (CSS). CSS is a style sheet language that describes the presentation of a document written which is written in HTML. The CSS3 technology helps in the development of more visually engaging web pages and applications that can run on modern browsers.

So, do you want to learn more about HTML5 and CSS3? Let's dive in and find out more about these technologies.

HTML PAGE BASICS

Any HTML document that you'll create must begin with some text. It can be a new text—that you enter when creating the new document—or, it can be an existing test—for which you are converting to HTML format.

The basic HTML document has the following structure:

```
1  <!DOCTYPE html>
2
3  <HTML>
4
5  <HEAD>
6
7  <TITLE>Your Page Title Goes Here </TITLE>
8
9  </HEAD>
10
11 <BODY>
12
13 <H1> An example of a tag</H1>
14
15 <P>An example of a tag</P>
16
17 </BODY>
18
19 </HTML>
```

Here is what you need to know about the significance of each of the above structure:

- The DOCTYPE declaration shows that the document type we are creating is an HTML document. This declaration will help your web page to be displayed correctly by the web browser.

- Your content that you wish to display in a web browser is inserted between the <HTML> and </HTML>.
- The content that is inserted between <HEAD> and </HEAD> gives information about the web page. This content is not displayed in a web browser.
- The content that is inserted between <TITLE> and </TITLE> indicates the title of the page.
- The text that is inserted between the <BODY> and </BODY> is what will be shown in a web page.
- The text that is inserted between <H1> and </H1> indicates a heading

The main function of HTML transforms a simple document that can be viewed in a web browser. This is achieved by the use of tags. The HTML tags indicate to the browser where to begin and end the special formatting. In other words, the tags are used to provide “mark ups” on the web browser.

HTML tags are bracketed commands that uses the “less than” and “greater than” signs (< and >). For instance the tag to begin a paragraph is <P>. To show or markup the end of paragraph formatting, the same tag is used with a slash (/) after the first bracket. The end of the paragraph will be </P>. The start tag is often referred to as the opening tag while the end tag is referred to as the closing tag.

So, how do you write HTML pages? Well, HTML pages can be created and edited using an editor. Example of Editors that you can use includes:

- Notepad (For Windows OS)
- Text Edit (For Mac OS)
- Microsoft WebMatrix
- Sublime Text

Notepad (For Windows OS) and Text Edit (For Mac OS) are the most commonly used Editors for creating and editing HTML contents for learning purposes. To use Notepad or Text Edit, here are the steps that you need to follow:

- Open your Notepad or Text Edit application.
- Write your HTML content in the Notepad or Text Edit.
- Save the HTML page using the .html file extension.
- View the HTML page in your browser.

Text Basics

Now that you've learnt the basic structure of all HTML documents, it's now time to dig in and find out more about HTML tags. The main function of HTML tags is to indicate to the browser where to begin and end the special formatting. In other words, the tags are used to provide "mark ups" on the web browser.

What is written between the starting and ending tag is called HTML element. For instance, the code:

<p>This is a paragraph.</p> has “This is a paragraph” as an HTML element.

Sometimes HTML elements can be nested. Nested HTML elements are elements that contain other elements. It is important to note that all the HTML documents always contain nested HTML elements.

For instance the HTML code below contains nested HTML Element:

```
1  <!DOCTYPE HTML>
2
3  <HTML>
4
5  <BODY>
6
7  <H1>This is Heading 1</H1>
8
9  <P>This is a paragraph.</P>
10
11 </BODY>
12
13 </HTML>
```

The table below shows the most commonly used tags and their functions:

Tag	Description
<!---->	Describes a comment
<!doctype>	Describes the document type
<a>	Describes a hyperlink
<abbr>	Describes an abbreviation or an acronym
<address>	Describes the contact information for the author when used in an HTML document

<u><area></u>	Describes the area inside the image-map
<u><article></u>	It provides a description of the article when used in HTML page
<u><aside></u>	It provides a description of the content from the page when used for the content in HTML page
<u><audio></u>	Describes the sound content
<u></u>	Describes a bold text
<u><bdo></u>	It overrides the current text direction
<u><blockquote></u>	Describes a section which is quoted from another source
<u><body></u>	Describes the document's body
<u>
</u>	Describes the single line break
<u><button></u>	Describes a clickable button
<u><canvas></u>	It is used to draw graphics, on the fly, via scripting (usually JavaScript)
<u><caption></u>	It describes a framework for the table caption
<u><cite></u>	Describes the title of a given work
<u><code></u>	Describes a piece of the computer code
<u><div></u>	Describes a section in a document
<u><dl></u>	Describes a description list
<u><dt></u>	Describes a term/name in a description list
<u></u>	Describes emphasized text
<u><embed></u>	Describes a container for an external (non-HTML) application
<u><fieldset></u>	It describes the groups for the related elements in a form
<u><figcaption></u>	Describes a caption for a <figure> element
<u><figure></u>	Specifies self-contained content
<u><footer></u>	Describes the footer for a document or section
<u><form></u>	Describes an HTML form for user input
<u><h1> to <h6></u>	Describes HTML headings
<u><head></u>	Describes information about the document
<u><header></u>	Describes a header for a document or section
<u><hr></u>	Describes a thematic change in the content
<u><html></u>	Describes the root of an HTML document
<u><i></u>	Describes the part of a text using an alternate voice
<u><iframe></u>	Describes an inline frame
<u></u>	Describes an image
<u><input></u>	Describes an input control
<u><ins></u>	Describes a text that has been inserted into a document
<u><kbd></u>	Describes keyboard input
<u><keygen></u>	Describes a key-pair generator field (for forms)
<u><label></u>	Describes a label for an <input> element
<u><legend></u>	Describes a caption for the <fieldset> element
<u></u>	Describes a list item
<u><link></u>	Describes the relationship between a document and an external resource

Text Formatting

All HTML text elements have a default style. This can range from a text color, to text size or font. Changing the HTML element from a default style to a style of your choice can be done using two methods. These methods are:

- Using HTML text formatting
- Using the style attribute

Using HTML text formatting

There are special elements for formatting the text in HTML. These can be achieved by the following:

- Bold text
- Important text
- Italic text
- Emphasized text
- Marked text
- Small text
- Deleted text
- Inserted text
- Subscripts
- Superscripts

For instance, the HTML *<i>* element is used to define an italic text. The example below illustrates the use of Style attribute with font-size property:

```
1 | <p><i>This text in this paragragh has been italicised</i></p>
```

Using the Style Attribute

The HTML style attribute has the following syntax:

```
1 | style="property:value"
```

In this case, the property is a CSS property while the value is a CSS value. The color property describes the text color that is to be used for an HTML element:

The example below illustrates the use of Style attribute with color property:

```
1 | <H1 style="color:blue">This is heading 1 whose color is blue</H1>
2 | <P style="color:red">Another paragragh whose color is red.</P>
```

The font-family property describes the font that is to be used for an HTML element:

The example below illustrates the use of Style attribute with font-family property:

```
1 | <H1 style="font-family:verdana"> This is heading 1 whose font is verdana</H1>
2 | <P style="font-family:courier"> Another paragragh whose font is courier.</P>
```

The font-size property describes the text-size that is to be used for an HTML element:

The example below illustrates the use of Style attribute with font-size property:

```
1 | <H1 style="font-size:300%">The font size in this heading should be 300% more than the default</H1>
2 | <P style="font-size:160%">This paragragh has 1.6 more of its font-size than the default.</P>
```

The `text-align` property describes the horizontal text alignment that is used for an HTML element:

The example below illustrates the use of `Style` attribute with `font-size` property:

```
1 | <H1 style="text-align:center">This is an example of a centered heading</H1>
```

Lists

HTML lists can be grouped into 3 categories. These are:

- Unordered lists. These lists are used to group a set of related items in unordered format.
- Ordered lists. These lists are used to group a set of related items using a specific order.
- Description lists. These are lists that describe the items in a web page.

All the unordered lists in HTML start with the `` tag. Once the `` tag has been defined, each of the list items will start with the `` tag.

The example below illustrates how unordered lists can be defined in HTML:

```
1 | <UL>
2 |   <LI>Georgia</LI>
3 |   <LI>California</LI>
4 |   <LI>New York </LI>
5 | </UL>
```

The ordered lists can be specified by using a type attribute that is added to the tag. The type attribute can assume either “A”, “a” and “1” for Upper case listing, small case listing, numbers listing respectively.

The example below illustrates how ordered lists can be defined in HTML:

```
1 | <OL type="1">
2 | <LI>Georgia</LI>
3 | <LI>California</LI>
4 | <LI>New York </LI>
5 | </OL>
```

The description lists can be specified by the tag <DL> tag and the <dt> tag which describes the term and the <dd> tag which describes each term.

The example below illustrates how description lists can be defined in HTML:

```
1 | <DL>
2 | <DT>San Diego</DT>
3 | <DD> A town in USA</DD>
4 | <DT>California</DT>
5 | <DD>A state in USA </DD>
6 | </DL>
```

Image Preparation

You've written HTML content and your website looks more convincing to the user. However, you'd like to add some images to make it more appealing. How do you proceed? Well, before you can insert any image in your HTML, you must first prepare that image.

Here is how you can prepare an image to be inserted in a web page:

- Copy the image that you'd like to use in your website to the same directory as the HTML file that you've created.
- Make sure the file format for that image is consistent with the browser that will open that web page.
- Use the tag to insert the image in your HTML page.

Inserting Images

The tag is used to insert images in a HTML page. The tag has the following attributes:

- It is a text that describes an alternate text for the image that has been inserted in a web page.
- Use-credentials. It allows images from third-party web-sites that can cross-origin access levels to be used with the canvas.
- It describes the height of the image
- It describes the image as a server side image-map
- It is the URL of the image
- It describes the width of the image

The example below illustrates how an image can be defined in HTML:

```
1 | <IMG src="peter_namisiko.gif" alt="Mr. Peter Namisiko" height="50" width="45">
```

Website Project

Create a professional website for an Online Classmate System. This website should have the following components:

- The title
- A portfolio for each user
- Links with the rollovers
- Should provide an aesthetic choices for the user
- A clear focal point
- Creativity
- Appropriateness for the purpose

Your site should have a minimum of 6 HTML pages.

CSS BASICS

Up to this point, we have learnt how HTML can help us create websites. Both HTML and CSS are essential in helping us to create websites. While HTML defines the structure of the web page, CSS defines the layout of the web page to be viewed on a variety of devices. CSS helps to separate the content from presentation when creating web pages.

This has the following advantages that include:

- It helps to keep the style in for a limited number of style sheets. This promotes the process of editing a portion or a section of a website.
- It helps to save on the bandwidth for the site owner as well as the visitor's website since it improves the performance of the website.
- It makes it easier for the site owners to reuse their content that they have developed for other purposes. This content can be re-used as RSS feeds or text to speech conversion systems.

Now that you've understood the significance of CSS in HTML, it's time to dive in and learn more about how to use CSS.

The CSS syntax set consists of a selector and a declaration block. The CSS declaration must always end in a semi-colon. For instance, all the <P> elements in the example below will be center-aligned and a red text color.

```
1 P {  
2   color: red;  
3   text-align: center;  
4 }  
5  
6  
7 }
```

The main function of a CSS selector is to enable you to select and manipulate the HTML elements in your web page. For instance, in the above example, all <P> elements will be aligned at the center with a red text color. The selectors can be grouped to minimize the code.

The example below illustrates how the selectors can be grouped.

```
1 H1 {  
2   text-align: right;  
3   color: red;  
4 }  
5  
6 H2 {  
7   text-align: left;  
8   color: blue;  
9 }  
10  
11 P {  
12   text-align: center;  
13   color: cyan;  
14 }  
15  
16  
17  
18  
19  
20  
21  
22  
23 }
```

CSS Font Styles

The CSS font properties are used to describe the font family, the boldness of the text, the size, and the style of a text.

The text font is set with the font-family property. When using the font-family property, several font names can be used as a contingency measure such that when the browser cannot support the first font, then it tries the next font. The example below illustrates how the font-family property can be defined in HTML.

```
1 H1 {  
2  
3   font-family: "Arial", Helvetica, sans serif;  
4  
5 }
```

The font-style property is used to specify an italicized text in CSS where the property can assume normal, italic or oblique. The example

Below illustrates how the font-style property can be defined in HTML:

```
1 H1.normal {  
2  
3   font-style: normal;  
4  
5 }  
6 &nbsp;  
7  
8 H1.italic {  
9  
10  font-style: italic;  
11  
12 }  
13 }
```

The font-size property is used to set the size of the text. The font-size value is either absolute or relative. In the absolute case, the font-size is set to fixed specified size while in the relative case; the font-size is set relative to other elements in HTML. The example below illustrates how the font-size property can be defined in HTML using absolute and relative methods.

```
1 H1 {  
2   font-size: 40px;  
3 }  
4  
5 H1 {  
6   font-size: 200%;  
7 }  
8  
9 }
```

Colors

In HTML, the colors are defined using the RGB system (combination of Red, Green and Blue colors). The color values for expressed in Hexadecimal numbering notation. The Hexadecimal values are written inform of 3 double digit numbers that starts with the # sign. For instance the Hexadecimal value “#oooooo” represents black color while the Hexadecimal value “#FFFFFF” denotes white color.

Based on the RGB color combinations, 16 million colors can be generated. The Hexadecimal color is specified with the Hexadecimal value: #RRGGBB, where the RR denotes the red color, GG denotes the Green color and BB denotes the blue color.

The example below illustrates how the font-size property can be defined in HTML:

```
1 H1 {background-color: #ff0000;}  
2 H2 {background-color: #00ff00;}  
3 H3 {background-color: #0000ff;}
```

The background color can be set for different elements in a HTML page.

For instance the example below illustrates how the background color for the entire page can be set, with the elements <H1> and <P> having different settings:

```
1 BODY {  
2   background-color: #23DD22;  
3 }  
4  
5 H1 {  
6   background-color: #00ff00;  
7 }  
8  
9 &nbsp;  
10  
11 P {  
12   background-color:#FFAA32;  
13 }  
14  
15
```

Link Styles

The CSS hyperlinks can be styled using property of CSS that include the color, the font-family and the background.

The example below illustrates how the color property of CSS can be used to style a hyperlink:

```
1 A {  
2  
3   color: #FF22CC;  
4  
5 }
```

The text-decoration property of CSS is used to eliminate the underlines from the hyperlinks as illustrated in the example below:

```
1 A:link {  
2  
3   text-decoration: underline;  
4  
5 }  
6  
7 A:visited {  
8  
9   text-decoration: none;  
10  
11 }  
12  
13 &nbsp;  
14  
15 A:hover {  
16  
17   text-decoration: none;  
18  
19 }  
20  
21 A:active {  
22  
23   text-decoration: none;  
24  
25 }
```

The background-color property of CSS is used to indicate the background color for the hyperlinks.

The example below illustrates how the background-color property can be used with hyperlinks:

```
1 A:link {  
2   background-color: #22FFDD;  
3 }  
4  
5 A:visited {  
6   background-color: #F002C5;  
7 }  
8  
9 }
```

Website Project

Create a web site using the knowledge you have learnt in CSS. Your web site should contain least five, but no more than eight content pages including the home page. Use a style sheet that configures the text, the color, and page layout for your website. Your website should capture the following:

- The title
- A portfolio for each user
- Links with the rollovers
- Should provide an aesthetic choices for the user
- A clear focal point
- Creativity
- Appropriateness for the purpose

LINK BASICS

Are you ready to learn how to develop professional websites? Fantastic! There's a lot that you need to learn about HTML links, so let's dive in.

HTML pages are hypertext pages that contain hyperlinks—that enable people to navigate between different contents on the World Wide Web. The fundamental idea in developing any website—which is at the heart of any website development—is to allow users to refer to other pages on the World Wide Web. The underlying TCP/IP application layer protocol that helps in the transfer of hypertext pages across the WWW is called Hypertext Transfer Protocol (HTTP).

For instance, when you type a URL in a web browser such as www.stoneriverelearning.com and click the enter key from your keyboard, the HTTP protocol directs the Web Server to fetch and transmit the requested page—which in this case is the Stone River eLearning home page. The requested page must be hypertext—with the links that allows you to navigate between different contents on the Stone River eLearning website such as Course Bundles and Course Demos.

Knowing how to insert these hyperlinks in your HTML page is essential in helping you understand how the World Wide Web works. Before we dive in and begin to learn the basics of HTML links, here are a couple of tips that you should bear in mind when generating links during the process of developing a good website.

- It should have leveraged links. By leveraged links, we imply that the words you choose for a link should not only be part of a key word, but also descriptive enough to make the user understand what the link is all about.
- The hyperlinks should be functional. Broken links will leave your customers exasperated lot. Try as much as possible to make your links functional so as not to disappoint your clients. But this should not only apply to hyperlinks alone. Every aspect of your website should be well-designed to make your site usable.
- There should be impeccable choice and use of color. When choosing the color for your links—whether “visited”, or “active” or “hover”—make sure you select an appropriate color scheme.
- The text you select for the hyperlinks should be easy to read. Use fonts that are not only appealing, but also available on most operating systems.

How to insert links

HTML links are defined by the `<A>` tag. This stands for an anchor—which is a text that marks the beginning and the end of a hypertext link. More often than not, the text between the opening tag and the closing tag can either be the start or the destination or even both of the hyperlinks.

The attributes of the anchor tag include the following:

- **Href** - It specifies the location (URL) of the web page to be opened. For instance the example below shows how an HTML

page named: “Index.html” that is located in the same folder as the current page can be linked.

```
1 | <A href="Index.html"> Click Here to View the Index Page</A>
```

Note that the user will click on the text: “Click Here to View the Index Page” on the web browser so that the page: “Index.html” can be opened.

- **Hreflang** - It specifies the language of the linked resource in HTML. In other words, this attribute informs the search engine the language that you are using on that specific page—to help the search engine generate the results to other users that are searching in that language. You should note that this attribute can only be used when the href attribute has already been defined.

For instance, the example below illustrates how the Hreflang attribute can be used to help the search engine generate results in English language.

```
1 | <A href="Index.html" hreflang="en"> Click Here to View the Index Page </A>
```

- **Media** - It specifies the media for which the linked resource relates to. The value of the media attribute should always be a media query. More often than not, this attribute mainly applies when linking to external style sheets which allow the user agent to select the best adapted device on which the page will run on.

The example below illustrates how the media attribute can be used to specify target URLs that are designed for desktop web browsers:

```
1 | <A href="Index.html" media="screen"> Click Here to View the Index Page </A>
```

- **Ping** - It specifies the URLs of the resources that notify the user to follow the hyperlink on a web page. This is exemplified below

```
1 | <A href="Index.html" ping="http://www.stoneriverelearning.com">Click Here to View the Index Page </A>
```

In this case, when a user clicks on the link the URL which has been specified in the ping attribute <http://www.stoneriverelearning.com> will be sent.

- **Target**. It indicates the location where the linked resource is to be displayed. The value of the target attribute can assume either “_blank”, “_top”, “_parent”, or “_self”. This is demonstrated by the example below:

```
1 | <A href="Index.html" target="_blank"> Click Here to View the Index Page </A>
```

- **Type** - It indicates the media type—usually in the form of a MIME type—for the link target. There are some instances, when the browser might want to add a small speaker icon, especially, when type is set to audio or a wav file format.

Congratulations! You've finished your first chapter about HTML link basics, but there's still a lots for you to learn about HTML links. So keep up and good luck!

IN-PAGE LINKS

How to Use In-Page Links in a Web Site

In this chapter, we'll wrap the discussion on how you can use the In-page links in your website. By the end of the chapter, you'll not only learn the significance of the In-page links in website development, but also how to create these links in your website that promotes the interactivity of your website.

To start off, let's define what the In-page links are. Well, the In-page links are hyperlinks on your web page that enables users to link to other pages on the same page. These links can be regarded as links within a page. If you're developing a website, then your main objective is to make sure that your visitors can browse as many pages as possible. That's why your website may have so many web pages.

Associating these pages requires that you use the In-page links. This is all about connecting several pages within a given web page—that's why it's called In-page link, after all. Why are these links important for a website developer?

Well, the In-page links are important to a web developer because of a number of reasons. These reasons include:

- They permit users to navigate numerous web pages on a website. If your website has numerous web pages, then the only mechanism that can allow users to navigate these pages are the In-page links.

- They help to establish the information hierarchy for a given website. The In-page links can help you organize your website according to categories. This gives your website a good structure that is easy to navigate.
- They may help you to market your website by increasing your presence on the internet. Most search engines rely on the In-page links when displaying the results for a particular query.

Types of In-Page Links

There are different types of In-page links in HTML. These are:

- Local links – They are used to link to pages that exist the same server or the folder/directory as the current page.
- External links – They are used to link to a page that exists on a different server or folder/directory
- E-mail links – They are used to open the visitor's e-mail program
- Download links – They allow users to download files on your website.

So, How Can You Use The In-Page Links In A Web Page?

Inserting an In-page link depends on whether that link is local or external or email or download. Before you can insert an in-page link in your web page, you have to establish whether the page you'd like to link to be in the same directory/folder or it's located in a different folder/directory or it's an email or a download link that you'd like to establish. Once you've established the type of link you're dealing with, you can proceed to insert that page link.

Here is the syntax for inserting different types of the in-page links in an HTML document:

1. To insert a local link

Use the syntax below to insert a page that exist the same server or the folder/directory as the current page.

```
1 | <A href="Name of the Page.html"> Click Here to View the Index Page</A>
```

For instance the example below shows how an HTML page named: “Index.html” that is located in the same folder as the current page can be linked.

```
1 | <A href="Index.html"> Click Here to View the Index Page</A>
```

2. To insert an External Link

Use the syntax below to insert a page that exist the same server or the folder/directory as the current page.

1 | Click Here to View the Index Page for pages that exists on the same computer but in different folders or directories.
In this case, you must specify the complete path to the location of the web page.

For instance the example below shows how an HTML page named: “Index.html” that is located on Local Disk C.

```
1 | <A href="C:\Index.html"> Click Here to View the Index Page</A>
```

If the page exists on a different server, then use the following syntax:

```
1 | <A href="www.name of the server.com\name of the file"> Click Here to View the Index Page</A>
```

For instance the example below shows how an HTML page named: “Index.html” that is located at www.stoneriverelearning.com.

```
1 | <A href="www.stoneriverelearning.com\Index.html"> Click Here to View the Index Page</A>
```

3. To insert an Email Link

This link is used to receive feedback comments from your website visitors. To insert an email link in your page, use the syntax below:

```
1 | <A href= "mailto: someone@mail.com?Subject=Subject">Click Here To Send Me Email </A> where someone@mail.com is  
the email address of the recipient and the Subject is the message that appears in the subject line of the email program.
```

This is illustrated below:

```
1 | <A href= "mailto: namsiko@gmail.com?Subject=Greetings! ">Click Here To Send Me Email </A>
```

4. To Insert a Download Link

To insert a download link in your HTML page, use the following syntax:

```
1 | <A href="File URL.zip">Click Here to Download This File</A>
```

This is exemplified below:

```
1 | <A href="C\Notes.zip">Click Here to Download the Notes</A>
```

So there you have it—we've covered the most important concepts about the In-page links in HTML. Now, start the development of your website by using the concepts you've learned in this lesson.

LINK STYLES

How to Use Link Styles in HTML Pages

In this chapter, we are going to learn about link styles. Now, you might be thinking, “This sounds fantastic, but why are link styles important for your web page?”

Well, your readers would be probably be scanning and not reading your website. Some of the readers simply don’t have the time to read the content on your web page. So, with the little time they’ve got, you must make your web page as readable as possible.

You achieve this by making sure that your links are obvious for the reader to scan through—with the little time they’ve got at their disposal. This is why you should indicate to them on your website where they have visited and the web pages they haven’t visited. That’s where the link styles perfectly fit in.

Well, you now understand why it’s important to have link styles on your web page. So, let’s dive in and find out how you can use link styles in your web page.

How to use the Link styles in your Web page

You will apply the HTML Links in your web page by using any of the CSS property that you've learnt so far. This includes the color, the font-family and the background. *For instance, the HTML code below assigns the color "Blue-#0000FF" to all the links in your web page.*

```
1 | A {  
2 |     color: #0000FF;  
3 | }
```

Of utmost importance while designing your web page is indicating to your visitors those links they've visited and the links they haven't. You can achieve this by using the link states. They are four link states, namely:

- An A: link which is a normal link that has not been visited before.
- An A: visited link, which is a link that the user has visited before.
- An A: hover which is a link indication for the user when the user's mouse passes over while browsing.
- An A: active which is a link indication the moment it is clicked by the user.

Here's how you can use the CSS to style your link states:

```
1 A: link {  
2     color: cyan;  
3 } for normal links.  
4  
5 A: visited {  
6     color: green;  
7 } for visited links  
8 /* selected link */  
9 A: active {  
10    color: blue;  
11 } for active links
```

In some instances, you may use the text-decoration property of the CSS, which removes underlines from the links on your web page. *For instance the HTML code below shows how you can underline all the active links in your web page:*

```
1 A: active {  
2     text-decoration: underline;  
3 }
```

The background-color property of CSS is used to specify the background color for your links. For instance, the HTML code below shows how the background color for a normal link can be set to color “red-#FF0000”.

```
1 A: link {  
2     background-color: #FF0000;  
3 }
```

So, there you have it. Everything that you need to know about the HTML Link styles. In our next discussion, we will discuss how to use the navigation menus in HTML web pages.

NAVIGATION MENUS

A Definitive Guide to Using the Navigation Menus in HTML Pages

In this lesson, we look at how you can use the navigation menus to help you create a usable website. A good website is that website that is usable. One way of making this a reality, is by using the navigation menus. Having an easy-to-use navigation can help the visitors locate any information they'd like to read in a stress-free manner.

Having said that, it is important to learn how to organize your navigation menus; to help your visitors browse your site in a stress-free manner. CSS can help you transform any lackluster HTML menus into stunning navigation menus that will make your website usable.

Well, at the core of CSS navigation menus; the navigation menus can be viewed as a series of lists. So, the lists will form our background discussion on how we can use the navigation menus in HTML pages.

There are two approaches that you can use:

- CSS list styles
- CSS with NAV tags

So, let us find out how we can create navigation menus in HTML pages.

CSS List Styles and Navigation Menus

Lists in HTML are specified by the unordered list (UL) and the ordered list (LI) tags. We can use these tags to specify how we would like our menus to appear on a web page. For instance, the example below illustrates how we can create a “Home Page”, “About Us”, “Contact Us” links in a web page:

```
1 <UL>
2
3 <LI><A href="Homepage.html">Home Page</A></LI>
4
5 <LI><A href="About Us.html">About Us</A></LI>
6
7 <LI><A href="Contact Us.html">Contact Us</A></LI>
8
9 </UL>
```

We can modify the above list by removing the bullets since we've used the ordered list to specify the “Home Page”, “About Us”, “Contact Us” links on the web page (We learned that by default, the ordered lists use bullets).

So, how can achieve this? Good question.

We go back the CSS basics on how lists can be styled. The list-style-type property of CSS lists comes in handy. We will use this property to remove all the bullets in the list—since navigation menus should not have bullets.

Here's how you can remove the bullets:

```
1 UL {
2
3   list-style-type: none;
4
5 }
```

We can also style the `<A>` elements, by placing a vertical navigation bar as follows:

```
1 A {  
2   display: block;  
3   width: 60px;  
4 }  
5  
6 }  
7 }
```

In the above example, the `display: block` is used to display the links as block elements so that the whole link area becomes clickable with your mouse. The `width: 60px` indicates the width of the block.

Having used CSS lists with your navigation menus, your HTML code should look like one below:

```
1 <!DOCTYPE html>  
2 <HTML>  
3 <HEAD>  
4 <STYLE>  
5 UL {  
6   list-style-type: none;  
7 }  
8 A {  
9   display: block;  
10  width: 60px;  
11  background-color: #00EE22;  
12 }  
13 </STYLE>  
14 </HEAD>  
15 <BODY>  
16 <UL>  
17  <LI><A href="Homepage.html">Home Page</A></LI>  
18  <LI><A href="About Us.html">About Us</A></LI>  
19  <LI><A href="Contact Us.html">Contact Us</A></LI>  
20 </UL>  
21 </BODY>  
22 </HTML>
```

To use the Horizontal navigation bar, you can use the inline or floating list items of the display property for lists. For instance, the HTML code shows how you can insert a horizontal navigation bar in your navigation menus:

```
1 | LI {  
2 |  
3 |   display: inline;  
4 |  
5 | }
```

Or alternatively, you can use the <NAV> tag to create your navigation links, especially if you'd like to have a major block for your navigation links. For instance we could specify the navigation links using the HTML code below, where the <NAV> tags have replaced the tags to create blocks for our navigation links:

```
1 | <NAV>  
2 |   <A href="Homepage.html">Home Page</A>  
3 |   <A href="About Us.html">About Us</A>  
4 |   <A href="Contact Us.html">Contact Us</A>  
5 | </NAV>
```

By default, the <NAV> uses the block style, that's why in the example above the navigation links will be blocked together.

Well, go ahead and start implementing a website project by utilizing all that we've learnt about HTML links, including the in-page links, the link styles and navigation menus.

ARRANGING ELEMENTS

How to Arrange HTML Elements on a Web Page

In this chapter, we will continue with HTML page layouts. But this time round, we'll focus on how you can arrange the elements your page. By the end of the chapter, you will understand how to arrange elements in your page. So, let's dive in and start learning the basics of HTML page layouts.

Why is arranging elements best for your web page? Good question. Well, make no mistake: arranging the elements on your web page will help you to improve the page structure. This is essential because it helps you come up with a good website that looks professional.

I'm sure that we all love our work to be professional. That's why it's absolutely necessary that you grasp the concept behind arranging elements on your web page. So, let's dive in and learn how you can arrange the elements in your web page.

Basics of Arranging Elements

To arrange elements in your web page, you'll have use to the <DIV> tag and CSS. In a previous chapter, we learnt that the <DIV> tag specifies the division or a section of an HTML page. Well, we shall continue using this important tag in HTML, together with CSS to arrange HTML elements in a web page.

We will begin with margins and paddings in web pages. A margin on a web page is the space that's outside of the elements in your web page while padding is the space that's around the elements of

the text to the border of your web page. How you specify the margins and padding determines how you will arrange the elements in your web page.

The example below illustrates how you can specify the margin and padding settings for your page that has been divided into two sections using block1 and block2 as the ID property of the <DIV> tag.

```
1 DIV {  
2     Width: 40px;  
3     Height: 40px;  
4 }  
5 #block1 {  
6     Margin: 20px;  
7     Padding: 15px;  
8     Background-color: red;  
9 }  
10 #block2 {  
11     Margin: 15px;  
12     Padding: 15px;  
13     Background-color: green;  
14 }  
15 </STYLE>
```

In the above example, the margin will be uniform for the left, the right, bottom and top margins. Similarly, the padding spacing will have a uniform spacing for the left, the right, bottom and top.

The example below shows how you can specify different setting for the left, the right, bottom and top margins:

```
1 #block1 {  
2     Margin: 40px, 30px, 20px, 10px;  
3     Padding: 30px, 25px, 20px, 15px;  
4     Background-color: red;  
5 }  
6  
7 }
```

Images can also be arranged in your document by use of the <DIV> tags and CSS. There are two types of images in HTML, namely: the aligned images and floating images. While aligned images are those images that position the images using the left, center and right formats, the floated images are those images that wrap around the text.

The example below shows how images can be floated horizontally:

```
1 <IMG style="float: right; margin: 2px 3px 15px 25px;" src="/support/image/your-image.jpeg" width="150" />
```

What about overlapping elements?

Overlapping elements in your HTML page can be set by the z-index property. The z-index property is used to define the stack order of that element which is overlapped. Such an element can be placed in front of others or behind the others.

The example below shows how the z-index property can be used to specify the positioning of overlapping elements:

```
1  IMG {  
2      position: relative;  
3      left: 1.5px;  
4      top: 1px;  
5      z-index: -2;  
6  }  
7  
8  }
```

We can also use the float property in CSS to arrange images on a web page. This property can help you wrap around your images in your web page. *For instance, the HTML code below illustrates how you can achieve this:*

```
1  IMG {  
2      float: left;  
3      margin: 10px,20px 10px 15px;  
4  }  
5  
6  }
```

So, there you have it. Stick to the use of <DIV> tag and CSS when arranging elements—whether images, margins, paragraphs—on your web page and you'll be on your way to becoming a professional website developer. Isn't that your dream? Well, in the next chapter, we critically examine how you can position your elements in a web page.

BLOCK SPACING

In this chapter, we will learn how to use page layouts. After this, you will know how to set your web page layout and why it's essential in web development. So, let's dive in and start learning the basics of HTML page layouts.

The content on your webpage may be of different topics and subjects. This requires you to use multiple columns so that each content is distinct from each other. A good web developer should know how to differentiate different sections of a web page. It makes your website to be usable. Remember, users would like to visit a website that has content which is well arranged.

That's why knowing how to arrange web pages using HTML layouts becomes necessary for any website developer—who would like to create a usable website. So, do you now understand why it's imperative that you learn how to use HTML layouts? Fantastic! How do we do page layouts in HTML? Let's dig in and find out.

Using the <DIV> Elements

The <DIV> HTML tag is a basic layout tool that specifies the division or a section of an HTML page. It's basically used to group related content together. Once these content has been grouped together, we can go ahead and assign different formatting for each content using CSS.

For instance, the HTML code below shows creates two sections or blocks of text that will be placed in 2 different paragraphs:

```
1 <DIV ID="block1">
2 <P>
3 Welcome to Stone River eLearning.
4 </P>
5 <DIV ID="block2">
6 </DIV>
7 <P>
8 Stone River eLearning is an Online Platform that avails you any course at your doorstep
9 </P>
10 </DIV>
```

In the above example, we have used the <DIV> element to create 2 blocks of text that are written in 2 different paragraphs. Inside each block of text (In this case the paragraph, we can apply different formatting features such as headings and spacing).

With this background information about the <DIV> tag, let's dig in and find out how CSS can help you format your blocks of text.

How to use <DIV> and CSS tags to format an HTML Page Layout

CSS can help you specify the height and width of each block of text. In the earlier example, we learnt how to use the <DIV> tag to group related content together. Suppose you'd like to specify the width and height of each paragraph, here's is how you can achieve this in CSS.

```
1 <STYLE>
2
3 DIV {
4
5 Width: 40px;
6
7 Height: 40px;
8
9 }
10
11 </STYLE>
```

In the above example, each block of text or paragraph in our case will have a width of 20 pixels (Of course, you can specify the measurements in centimeters, inches or you can use percentages to indicate the height and width of each section of your content, just the same way you do with the font sizes).

Now, suppose you'd like each paragraph or block to have a different width and height, how can you achieve this? Well, it's pretty simple. You'll use the id property of the <DIV> tag to specify each width and height for the different sections of your content. In our case we used "block1" and "block2".

Here's how you can specify the different width and height for your content:

```
1 <STYLE>
2
3 DIV {
4
5 Width: 40px;
6
7 Height: 40px;
8
9 }
10
11 #block1 {
12
13 Width: 30px;
14
15 Height: 45px;
16
17 Background-color: red;
18
19 }
20
21 #block2 {
22
23 Width: 50px;
24
25 Height: 30px;
26
27 Background-color: green;
28
29 </STYLE>
```

In the above example, we have set our first block of text to have a width of 40 pixels and height of 40 pixels. Notice that we have also set its background color to red. For the second block of text, the width is 50 pixels and the height is 30 pixels. Its background color is green.

You can also specify the different margins and paddings for each section of content that has been defined using the <DIV> tag of utmost importance to make a distinction between margins and padding as used in CSS.

Well, a margin is usually the space that's outside of or a space that's between the elements in your web page while padding is the space that's around or inside the elements from the text to the border of your web page.

To specify different margins and paddings for your web page, here's how you will write the code in CSS:

```
1 DIV {  
2     width: 40px;  
3     height: 40px;  
4 }  
5  
6 #block1 {  
7     margin: 20px;  
8     padding: 15px;  
9     background-color: red;  
10 }  
11  
12 #block2 {  
13     margin: 15px;  
14     padding: 15px;  
15     background-color: green;  
16 }  
17  
18 </STYLE>
```

Well, there you have it. We've come to the end of the chapter on HTML page layouts. You should have a pretty good idea on how you can arrange the content on your web page. In the next lesson, we are going to learn how to use block spacing in your web pages.

POSITIONING ELEMENTS

A Guide to Positioning HTML Elements in Your Website

In this chapter, we will continue with HTML page layouts—with special focus on positioning HTML elements in your page. By the end of the chapter, you should be in a position to position any block or section of content on your web page.

There is no point dipping your feet in on this topic, so let's dive right in and learn how you can position HTML elements.

How to position HTML Elements

We'll still use the <DIV> tag and CSS, but this time round we'll add in the CSS position property. The CSS position property is used to define the type of positioning method that can be employed on an element. Basically, there are four types of positioning methods. These are static, relative, fixed and absolute.

The default method is usually the static. For instance, an element with the “position: static” will not be positioned in any special way, since the static position is the default method.

All the elements that have “position: relative” will be positioned relative to the viewport locations on the web page. You can set your top margin, right margin, bottom margin, and your left properties of a positioned element which will make the HTML element to be adjusted from its normal position.

The HTML code below illustrates this:

```
1 DIV {  
2  
3   position: absolute;  
4  
5   left: 10px;  
6  
7   border: 1.5px;  
8  
9 }
```

All the elements that have “position: fixed” will be positioned relative to their normal positions. You can set your top margin, right margin, bottom margin, and your left properties of a positioned element which will make the HTML element to be adjusted from its normal position.

The HTML code below illustrates this:

```
1 DIV{  
2  
3   position: fixed;  
4  
5   bottom: 2;  
6  
7   right: 2;  
8  
9   width: 160px;  
10  
11  border: 1px;  
12  
13 }
```

What about overlapping elements?

Well, overlapping elements in your HTML page can be set by the z-index property. The z-index property is used to define the stack order of that element which is overlapped. Such an element can be placed in front of others or behind the others.

The example below shows how the z-index property can be used to specify the positioning of overlapping elements:

```
1  IMG {  
2      position: relative;  
3      left: 2px;  
4      top: 1.5px;  
5      z-index: -2;  
6  }  
7  
8
```

We've come to the end of the HTML page layout. By now, you should be ready to take on any website development assignment by laying out your web pages in a professional manner.

TABLES

Introduction to HTML Tables

In this chapter, we will learn how to create tables. But before we begin creating tables, why are tables important in your web page? Good question?

Well, tables used to be and are still very important components of any web development process. Way back in time before the arrival of CSS, tables were used to structure page layouts. Tables were actually the de-facto standards for definition of HTML page layouts.

Of course that has changed—because of CSS. But there remains one fundamental aspect about tables that we can't just wish away: how to present tabular information. As a matter of fact, there is no other way you can avoid tables if you'd like to present your information in tabular format apart from using tables.

For instance, if you are a web hosting company data readable, and you're interested in presenting your users with various application's pricing pages so that users can compare them, there is no way you'll avoid using tables. This is because tables will help you to present information that's easily scan-able and comparable. So, now that you understand the importance of tables, let's get to the basics of creating HTML tables.

How to Define HTML Tables

An HTML table is basically a collection of rows and columns. The intersection between the rows and columns is called a cell. What can be inserted into a cell is data which can range from images, text, links or even other tables.

We use the <TABLE> tag to define an HTML table.

We use the <TR> tag to define the table rows within a given table.

We use the <TD> tag to define the data that's inserted in the cells.

Have a look at the following HTML code:

```
1  <TABLE>
2
3  <TR>
4
5  <TD>John</TD>
6
7  <TD>Janet</TD>
8
9  <TD>50</TD>
10
11 </TR>
12
13 <TR>
14
15 <TD>Samson</TD>
16
17 <TD>Jackson</TD>
18
19 <TD>31</TD>
20
21 </TR>
22
23 </TABLE>
```

The above HTML code produces table with 2 rows and 3 columns. Note that the table produced by the above code lacks the borders, title and other important information that tables should have.

Other attributes of the <TABLE> in include the following:

- Sortable. It defines whether a given table can be sorted or not. This attribute is supported in HTML 5.
- Align. It defines the alignment of a table. However, this attribute is not supported in HTML5.
- Bgcolor. It defines the background color of a table. It is not supported in HTML5.
- Border. It defines the borders of a table. It is not supported in HTML5.
- Cell padding. It's measured in pixels and indicates the space that exists between the cell wall in a table and the cell content. It is not supported in HTML5.
- Cell spacing. It's the spacing between the cells in a table. It is not supported in HTML5.
- Width. It defines the width of a table. It is not supported in HTML5.
- Caption. It defines the title of the table.

There are sometimes when you'd like to specify the number of rows or columns that a particular cell should span. In such a case, you'll have to use the <TD> tag attributes such as:

- Colspan. The Colspan attribute specifies the number of columns that a given column should extend.
- Rowspan. It defines the number of rows that a given row should extend.

The HTML code below shows how the two attributes of the <TD> can be used:

```
1  <TABLE>
2
3  <TR>
4
5  <TH>PETER</TH>
6
7  <TH>KEN</TH>
8
9  </TR>
10
11 <TR>
12
13 <TD>MAY</TD>
14
15 <TD>$450</TD>
16
17 </TR>
18
19 <TR>
20
21 <TD>JUNE</TD>
22
23 <TD>$500</TD>
24
25 </TR>
26
27 <TR>
28
29 <TD COLSPAN="2">$950</TD>
30
31 </TR>
32
33 </TABLE>
```

Well, there you have it. Keep practicing. You're on your way to becoming a professional web developer! In our next class, we'll learn how to use CSS with tables.

TABLE STYLES

A Definitive Guide to Using HTML TABLE STYLES

In this chapter, we continue our discussion on tables by looking at how you can use styles on your tables. Specifically, we'll learn how to use CSS with our tables. You are now asking yourself this question: "why must we use CSS to format the table?" Well, that's a good question.

The fundamental concepts upon which CSS was invented to solve remain the same. If you are a web developer, then you should have more control over any page layout and display that you are creating. Having more control means separating the content of your web page from the style.

The same principle applies to HTML tables. As a web developer, you shouldn't focus much of your energy on the layouts and styles for your tables. CSS can handle that. Once you separate the content that's placed on the table with the layouts and style used, you'll have more leverage to your web page.

Having said that, it's important to refresh our minds again on how tables are generated in HTML. This is because we will use the same concepts, but this time round, we will focus on the layout and not the content.

Do you remember some of the attributes of the tables that we discussed which were not supported in HTML5? Well, attributes such as "align", "border", "width", "cellspacing" and "cellpadding" have been taken up by CSS.

In this lesson, we will look at how we can format our tables with some of the attributes that I have mentioned above. So, let's not dip so much and dive in to find out how to use an HTML table styles.

HTML Table Borders

The borders of a table can be specified by the “border” attribute of a table. In CSS, it’s possible to specify the borders for <TABLE>, <TH> and <TD>. For instance, the example below shows how a border of “2px” can be set for <TABLE>, <TH> and <TD> tags.

The border-collapse attribute of a table specifies the table borders when they are joined into a single border.

For instance, the HTML code below illustrates this:

```
1 TABLE {  
2  
3 border-collapse: collapse;  
4  
5 }
```

Width and Height of the Table

The width and height of a table are specified in pixels or percentages. They are defined by the “width” and “height” properties respectively in CSS.

For instance, in the example below, a width of 60 pixels and height of 40 pixels has been specified:

```
1 TABLE {  
2  
3 width: 60px;  
4  
5 height: 40px;  
6  
7 }
```

Cell Padding

It indicates the space that exists between the cell wall in a table and the cell content. Here's how it can be specified in CSS:

```
1 | TD {  
2 |   padding: 15px;  
3 | }  
4 |  
5 |
```

Table's Color

Here's how the table's background color can be set using CSS:

```
1 | TABLE, TD, TH {  
2 |   border: 2px solid blue;  
3 | }  
4 |  
5 | TH {  
6 |   background-color: cyan;  
7 | }  
8 |  
9 |   color: red;  
10| }  
11|  
12| }  
13|
```

So there you have it—the basics of HTML table styles. Now, you can use this knowledge to get you started on working with tables.

FORM BASICS

An Introduction to Html Form Basics

Let's start out with a bigger picture in this chapter. Why do we need forms in HTML?

The arrival of new standards in technology which have led to the growth of the internet has dramatically changed the way we use our websites. It's no longer the case that you'll just use your website for sharing information that you have with your end users. For you to get the most out of your website, you must consider user experiences.

That means, you must have a mechanism to collect user input experiences. It's the only way that you can connect with your website visitors. That's where the HTML forms come in. An HTML form can act as a communication channel between your company and your visitors and vice versa.

Therefore, for you to make a successful website, you must include the forms on such a website to help you collect information from your clients or visitors. Having said that, let's dive in and learn how we can construct HTML forms.

The <FORM> element is used to define an HTML form in a web page. There are two basic attributes for <FORM> element. These are:

- Action. It specifies where the data that has been collected is sent when the submit button is clicked.
- Method. The method indicates how the data that has been collected on the form is sent. There are two types of methods. The “GET” value attaches the form data input to the URL when the submit button has been clicked. The “POST” value of the method property is used to send the form data input as an HTTP transaction which is more secure.

Here's an example:

```
1 | <FORM action="index.php" method="post">
2 |
3 | </FORM>
```

So, what's placed between the <FORM> and </FORM> tags? Different types of input data elements such as text boxes, checkboxes, radio buttons, submit buttons, drop down boxes are placed between the <FORM> and </FORM> tags to help you collect data from your website visitors.

The <INPUT> Element

It has many variations that are applied depending on the type of data that you'd like to collect. Here are examples of how you can use the <INPUT> element:

The <INPUT TYPE="text"> is used to define a one line input field for collecting a text input from a user.

The `<INPUT TYPE="radio">` is used to specify the radio button—where a user has the option of selecting one out of many alternative options.

The `<INPUT TYPE="submit">` is used to specify a button that users will click on so that their form data can be sent to the server.

The example below shows how the three `<INPUT>` elements can be used:

```
1 <FORM ACTION="action.php">
2
3 First Name :<BR>
4
5 <INPUT TYPE="text" name="firstname" value="namisiko">
6
7 <BR>
8
9 Last Name :<BR>
10
11 <INPUT TYPE="text" name="lastname" value="last">
12
13 <INPUT TYPE="radio" name="sex" value="Male">Male
14
15 <BR>
16
17 <INPUT TYPE="radio" name="sex" value="female">Female
18
19 <BR><BR>
20
21 <INPUT TYPE="submit" value="Send">
22
23 </FORM>
```

This marks the end of part one of our discussion on forms. Keep practicing so that you improve your skills in forms creation.

HTML FORM ELEMENTS

How to Use HTML Form Elements

We started this discussion by defining what HTML forms are and their main functions in web development process. We saw how the `<FORM>` element is used to define HTML form in a web page with the `2` attributes namely the “action” and the “method”.

We also saw how you could create a basic form for collecting data where we used a text box, a radio button and a submit button. In this chapter, we continue the discussion by focusing on more the form elements.

Here are the form elements that you'll frequently use when working with forms:

The `<INPUT>` Element

It has many variations that are applied depending on the type of data that you'd like to collect. Here are examples of how you can use the `<INPUT>` element:

The `<INPUT TYPE="text">` is used to define a one line input field for collecting a text input from a user.

The `<INPUT TYPE="radio">` is used to specify the radio button—where a user has the option of selecting one out of many alternative options.

The `<INPUT TYPE="submit">` is used to specify a button that users will click on so that their form data can be sent to the server.

The `<INPUT TYPE="file">` is used to specify a file to be uploaded to a server.

The example below shows how the three <INPUT> elements can be used:

```
1 <FORM ACTION="action.php">
2
3 First Name:
4
5 <INPUT TYPE="text" name="firstname" value="namisiko">
6
7 <BR>
8
9 Last Name:
10
11 <INPUT TYPE="text" name="lastname" value="last">
12
13 <INPUT TYPE="radio" name="sex" value="Male">Male
14
15 <BR>
16
17 <INPUT TYPE="radio" name="sex" value="female">Female
18
19 <BR>
20
21 <INPUT TYPE="file">: Upload your CV
22
23 <BR><BR>
24
25 <INPUT TYPE="submit" value="Send">
26
27 </FORM>
```

The <SELECT> Element

It is used to define a drop down list. Here's an example that shows a drop down list for months:

```
1 <SELECT name="months">
2
3 <option value="jan">January</option>
4
5 <option value="feb">February</option>
6
7 <option value="March">March</option>
8
9 <option value="april">April</option>
10
11 </SELECT>
```

The <TEXTAREA> Element

It is used to define a text area—where multiple lines of text can be entered in a form by users. *Here's an example:*

```
1 | <TEXTAREA name="message" rows="20" cols="40">
2 |
3 | The multiple lines of text will be entered here
4 |
5 | </TEXTAREA>
```

This marks the end of part one of our discussion on forms. Keep practicing so that you improve your skills in forms creation.

The <BUTTON> Element

It's used to define a button that can be clicked. *Here's an example:*

```
1 | <BUTTON TYPE="button">Click Me</BUTTON>
```

The <Datalist> Element

It's used to define a list of pre-defined options when you use the <INPUT> element. When you use this element, your users will see a drop down menu for the pre-defined options that you have specified by the <DATALIST> element.

Here's an example of how to use this element:

```
1 | <input list="months" name="months">
2 |
3 | <datalist id="browsers">
4 |
5 | <option value="January">
6 |
7 | <option value="February">
8 |
9 | <option value="March">
10 |
11 | <option value="April">
12 |
13 | <option value="May">
14 |
15 | </datalist>
```

HTML5 FORM ELEMENTS

How to Use HTML5 Form Elements

We started this discussion by defining what HTML forms are and their main functions in web development process. We discussed the most common form elements. In this discussion, we look at new HTML 5 elements and attributes.

HTML 5 has introduced new elements together with new attributes. These new elements are meant to give hints to your browser about what type of the keyboard layouts to display for the on-screen keyboards. This is all meant to give you an impeccable experience while filling out the forms.

It's important to note that web browser support for HTML5 elements will vary from one browser to another. In case the browser does not support HTML5, then it defaults to HTML4—which is supported by virtually all browsers.

So, let's find out what's new in HTML5 form elements.

New Input Types of Forms in HTML 5

Here are the new input types of forms in HTML 5

1. Color

The color input type is used to provide the user with a native color picker.

For instance the HTML code below illustrates this:

```
1 | <input type="color" value="#00ff22">
```

2. Date

It's used to indicate the date picker. This provides users with mechanisms to easily select the date.

For instance the HTML code below illustrates this:

```
1 | <input type="date" value="2015-05-05">
```

3. Datetime

It's an element that's used to select both the date and the time for a particular day. *The HTML code below illustrates this:*

```
1 | <input type="datetime" value="2015-05-05 T12:30:22.10">
```

4. Datetime-local

This HTML element is used to select both the date and the time, which captures the local settings for your system.

The HTML code below illustrates this:

```
1 | <input type="datetime-local" value="2015-05-05 T10:20">
```

5. Email

It provides a field for entering the e-mail address.

For instance, the code below illustrates this:

```
1 | <input type="email" value="someone@email.com">
```

6. Month

It's used to elect a full month.

For instance, the code below illustrates how a full month can be selected by this HTML element:

```
1 | <input type="month" value="2015-05">
```

7. Number

It's used for selecting numbers only.

For instance, the HTML code below shows how you can define a field for picking numbers only:

```
1 | <input type="number" value="200">
```

New Attributes for Forms in HTML 5

Here's a list of new attributes that have been added to HTML5 to complement the new input types:

1. autofocus

It's used to set to the field when the element has been loaded. For instance, the code below sets an autofocus for the text field:

```
1 | <input type="text" autofocus>
```

2. formaction

This new attribute overrides the JavaScript's validation in HTML 4 and previous versions of HTML. It's used to override the action attribute of the form.

For instance, the code below illustrates this:

```
1 | <input type="submit" formaction="http://www.stoneriverelearning.com" value="Submit">
```

3. formenctype

It specifies the form's encoding type when submitting data.

For instance, the HTML code below illustrates this:

```
1 | <input type="submit" formenctype=" http://www.stoneriverelearning.com" value="Submit with enctype">
```

4. **formmethod**

It's used to override the form's method attribute.

The code below illustrates this:

```
1 | <input type="submit" formmethod="GET" value="Submit">
```

5. **max**

It specifies the maximum value for the number that has been entered in a text field.

The code below illustrates this:

```
1 | <input type="range" max="100">
```

6. **min**

It specifies the minimum value for the number that has been entered in a text field. *The code below illustrates this:*

```
1 | <input type="range" min="0">
```

Well, that's all that we can discuss about HTML 5 form elements. Try to experiment with these new HTML5 form elements and notice the difference between HTML4 and HTML5 form elements.

FORM STYLES

How to Design HTML Forms with CSS

So far, we've discussed how we can create forms in HTML using the <FORM> element along with its attributes and how to use them. We've also looked at different form elements such as text fields and text area; multiple choice input menus such as radio and checkboxes, drop down list menus and form button such as submit and reset buttons.

It's now time to wrap up this discussion by focusing on how to layout the form. When designing a form, it's essential that the layout of your form is enhanced. This is because bad form layouts have the tendency to confound your visitors therefore driving them away from your website.

It's a fact that shouldn't be belabored a lot when designing HTML forms. So, in this chapter, we'll explore how you can augment your website by focusing on CSS styles. The main aim of using CSS to style up your forms to separate the content—what the forms will capture—and the layout.

This provides you with the flexibility and the control you need when creating the HTML forms. Having said that, let's dive in and learn how you can create layouts for your HTML forms.

Grouping different fields using the <FIELDSET> Element

In our last discussion, we saw how you can create a form together with different form elements such as text fields and text area; multiple choice input menus such as radio and checkboxes, drop down list menus and form button such as submit and reset buttons.

However, when you view the output of your work in a web browser, we realized that the different fields were disorganized. Well, that should no longer be the case with the <FIELDSET> tag. This tag has been included to help you group the related form elements together.

For instance, the HTML code below shows how you can use this tag:

```
1  <FORM>
2
3
4  <FIELDSET>
5
6
7  <LEGEND>Personal Information</LEGEND>
8
9
10 Full Name: <input type="text">
11
12 Email Address: <input type="text">
13
14 Date of Birth: <input type="text">
15
16 </FIELDSET>
17
18
19 </FORM>
```

In the above example, the <LEGEND> tag has been used to provide the caption (Personal Information) for our <FIELDSET> element.

How about using CSS with the <FIELDSET> tag?

Let's dig deeper and elaborate more about this tag. The purpose of this tag to help you group related elements such as personal information (As illustrated in the above example). On a more positive note, we can provide different formatting features for each group of items that have been defined by the <FIELDSET> tag.

For instance, we can now define how we would like the grouped HTML elements to be appearing by using the following code in CSS:

```
1 FIELDSET {  
2   display: block;  
3   margin-left: 2px;  
4   margin-right: 3px;  
5   padding-top: 0.45em;  
6   padding-bottom: 0.65em;  
7   padding-left: 0.43em;  
8   padding-right: 0.75em;  
9   border: 2px groove (internal value);  
10 }  
11 }  
12 }  
13 }  
14 }  
15 }  
16 }  
17 }  
18 }  
19 }
```

The display property “block” helps the grouped elements to be displayed as a block. The margin property helps us to set the left, top, right and bottom of the block. The padding property helps us to set the top, bottom, left and right padding space for our blocks.

Other CSS styles for the different form elements

So far, we've looked at how the <FIELDSET> property can help us to format our grouped items. However, how can we make the individual form elements look better? This is the trickiest part when creating form layouts.

For us to format the individual form elements, we must specify the <LABEL> tag for each form element that we would like to format.

The <LABEL> tag is used to define the label for the <INPUT> element, such as a text field or a text area. It has the following attributes:

- Id. The <LABEL>'s "id" property is used to define an identifier for the associated element of the label.
- Class. The <LABEL>'s "class" property is used to define the class for the associated element of the label.

Having said that, let's see how we can create a form that uses not only the <FIELDSET> property, for grouping related form elements, but also the <LABEL> tag to help in formatting of individual form elements.

We can now specify our form as follows:

```
1 <FORM>
2
3 <FIELDSET>
4
5
6 <LEGEND>Personal Information</LEGEND>
7
8
9
10 <P>
11 <LABEL class="field">Full Name :< /LABEL> <input type="text">
12 <LABEL class="field">Email Address :< /LABEL> <input type="text">
13 <LABEL class="field">Date of Birth:< /LABEL><input type="text">
14
15 </P>
16
17 </FIELDSET>
18
19
20
21
22 </FORM>
```

Now, having defined the label for each form element (In the above example, we've defined labels for "Full Name", "Email Address" and "Date of Birth"), let's go ahead and define the layout formats for these form elements.

Here's is how we will define the CSS for our created labels:

```
1 FIELDSET
2 {
3
4     display: block;
5
6     width: 450px;
7
8     padding-bottom: 0.65em;
9
10    padding-left: 0.43em;
11
12 }
13
14 LEGEND
15
16 {
17
18     font-size:18px;
19
20 }
21
22 Label.field
23
24 {
25
26     text-align: right;
27
28     width:80px;
29
30     float:left;
31
32     font-weight: bold;
33
34 }
35
```

Applying the above CSS features distinguishes your forms by making them usable—a feature that will drive visitors to your website. This marks the end of form styles.

STREAMING CONTENT

How to Set up Streaming Video Content in HTML5

Now that you've learnt all the basic components of HTML5, It's time to wrap up this discussion by looking at how you can set up streaming video content. You'll also learn how to add a streaming video in a web page.

First and foremost, it's important to understand how video content streaming works. Whenever, you're watching for a video from a web browser, the information comes in smaller chunks. So, when you're watching these smaller chunks of video, what's actually happening is that video being displayed has been encoded using multiple bitrates and resolutions that creates the smaller chunks of multiple sizes that you're watching on your web browser.

This is how your browser is able to display these chunks of information that you watch on your web browser. It's also important to define the video standards that are accepted in HTML 5. Modern browsers and even mobile browsers support can play videos directly. Note that the video file formats supported in HTML5 are .mp4, .ogv, and .webm. So, if you are considering streaming video content on a web browser, then the only file formats that are supported are .mp4, .ogv, and .webm.

But this doesn't mean you can't embed your video into any web browser. You can still convert that video content that you'd like to stream into either of the file formats using "video converter" applications that can easily be downloaded from the internet. If

you want to convert your videos, just search these apps on a search engine.

The search engine will display quite a number of them. Download and install the app so that you can convert any video into a file format of your choice.

Preparation of the Video Content

So, how do you prepare a video content to be streamed to a browser? Here's is what you should do before adding any video content on your HTML page:

- Convert the video that you would like to add to a web page into a file format that's easily recognized on most devices and browsers. The most common video file formats that are supported include: the .mp4, the .ogv, and the .webm.
- If you're developing your website for people who may have the non-Flash browsers, then it's important to help the browsers to automatically switch to HTML5 fallback mode when a non-Flash browser is encountered while browsing.
- You can use a couple of applications that can help you achieve this functionality. One such application is JW Player. Download and install the application.
- Follow the instructions of the application to help you configure browsers that are likely to switch automatically to HTML5 fallback mode.

Adding the Video Streaming Content to HTML Page

Well, now that you've prepared the video that you'd like to add to a webpage, it's now time to add that video content on an HTML page. Addition of video to a webpage is done by the <VIDEO>

element. The basic <VIDEO> element has a couple of attributes that specify other characteristics of video that's to be streamed to a web browser.

One such attribute is the “src” attribute that specifies the URL where the video to be streamed to a web browser is located.

For instance, the HTML code below illustrates how a video that's located on C: can be streamed on a webpage:

```
1 | <VIDEO src="C:\Music.webm" controls>
2 |
3 | </VIDEO>
```

Other attributes of the <VIDEO> element include:

- Autoplay. The Autoplay attribute is used to notify the browser to begin downloading the video and playing it the moment the webpage is loaded.
- Preload. It's used to notify the browser about whether the optimistic downloading of the video the video's metadata is important.
- Poster. It's used to notify the image to be displayed before the video loads on a web browser.
- Controls. They display the default video controls such as play and pause.
- Height. It defines the height of the video in pixels when displayed on a web browser.
- Width. It defines the width of the video in pixels when displayed on a web browser.
- Loop. It automatically loops the video, when the video has finished playing.