HTML

Project Supervisor: **Muzammil Hassan | Manager Research SSRL**

Al-Khwarizmi Institute of Computer Science (KICS)

Office: +92-42-9250245 | Fax: +92-42-9250246

Al-Khawarizmi Institute of Computer Science (KICS) UET, Lahore

http://www.kics.edu.pk


بسم الله الرحمن الرحيم

About the Tutorial

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages.

HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

Audience

This tutorial is designed for the aspiring Web Designers and Developers with a need to understand the HTML in enough detail along with its simple overview, and practical examples. This tutorial will give you enough ingredients to start with HTML from where you can take yourself at higher level of expertise.

Prerequisites

Before proceeding with this tutorial you should have a basic working knowledge with Windows or Linux operating system, additionally you must be familiar with:

- Experience with any text editor like notepad, notepad++, or Edit plus etc.
- How to create directories and files on your computer.
- How to navigate through different directories.
- How to type content in a file and save them on a computer.
- Understanding about images in different formats like JPEG, PNG format.

Table of Contents

1. HTML Overview

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages.

- Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
- As its name suggests, HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

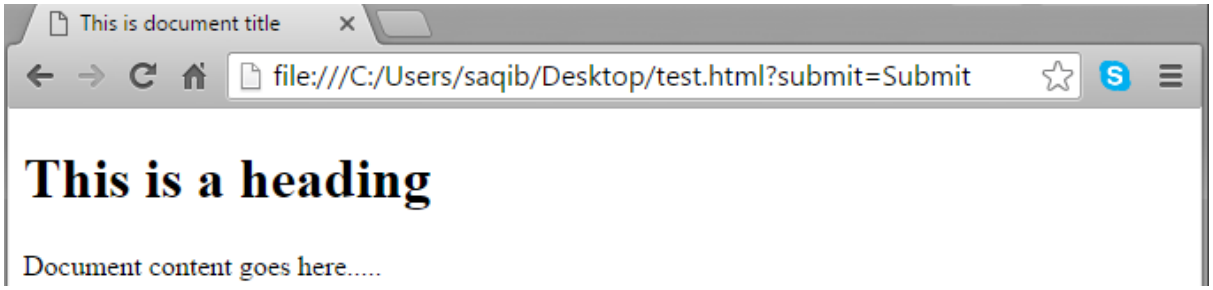
Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

Basic HTML Document

In its simplest form, following is an example of an HTML document:

```
<!DOCTYPE html>
<html>
<head>
    <title>This is document title</title>
</head>
<body>
    <h1>This is a heading</h1>
    <p>Document content goes here.....</p>
</body>
</html>
```

Either you can use Try it option available at the top right corner of the code box to check the result of this HTML code, or let's save it in an HTML file test.htm using your favorite text editor. Finally open it using a web browser like Internet Explorer or Google Chrome, or Firefox etc. It must show the following output:



HTML Tags

As told earlier, HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces <Tag Name>. Except few tags, most of the tags have their corresponding closing tags. For example, <html> has its closing tag</html> and <body> tag has its closing tag </body> tag etc.

Above example of HTML document uses the following tags:

Tag	Description
<!DOCTYPE...>	This tag defines the document type and HTML version.
<html>	This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.
<head>	This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.
<title>	The <title> tag is used inside the <head> tag to mention the document title.
<body>	This tag represents the document's body which keeps other HTML tags like <h1>, , <p> etc.
<h1>	This tag represents the heading.
<p>	This tag represents a paragraph.

To learn HTML, you will need to study various tags and understand how they behave, while formatting a textual document. Learning HTML is simple as users have to learn the usage of different tags in order to format the text or images to make a beautiful webpage.

World Wide Web Consortium (W3C) recommends to use lowercase tags starting from HTML

We will study all the header and body tags in subsequent chapters, but for now let's see what document declaration tag is.

The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration tag is used by the web browser to understand the version of the HTML used in the document. Current version of HTML is 5 and it makes use of the following declaration:

```
<!DOCTYPE html>
```

There are many other declaration types which can be used in HTML document depending on what version of HTML is being used. We will see more details on this while discussing.

<!DOCTYPE...> tag along with other HTML tags.

2. HTML – BASIC TAGS

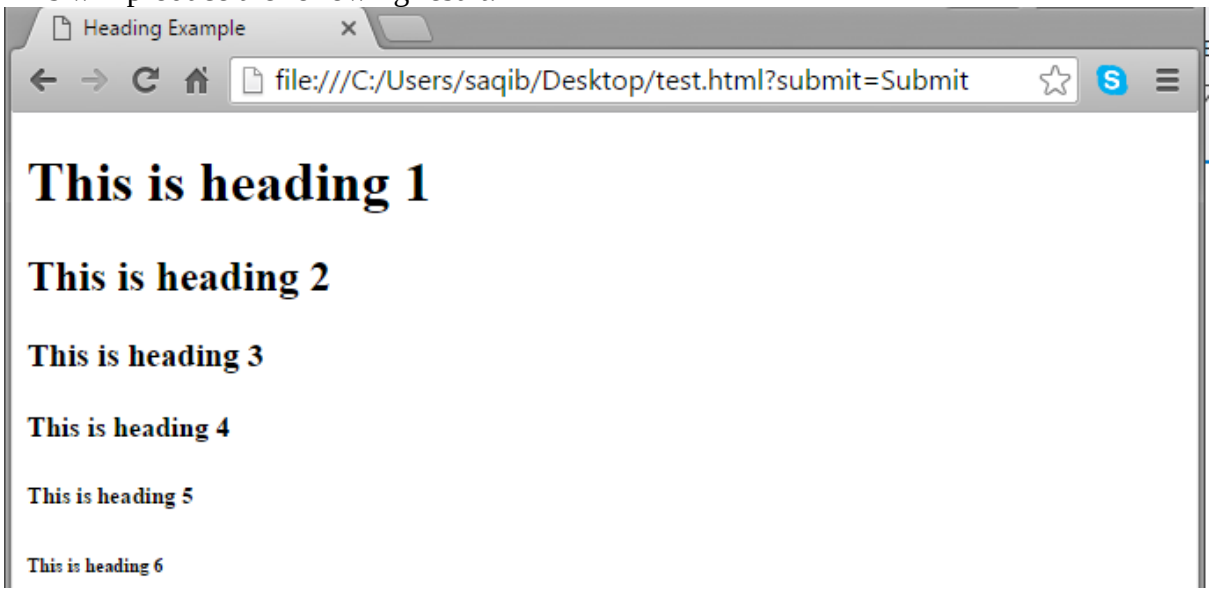
Heading Tags

Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>. While displaying any heading, browser adds one line before and one line after that heading.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Heading Example</title>
</head>
<body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3>This is heading 3</h3>
    <h4>This is heading 4</h4>
    <h5>This is heading 5</h5>
    <h6>This is heading 6</h6>
</body>
</html>
```

This will produce the following result:



Paragraph Tag

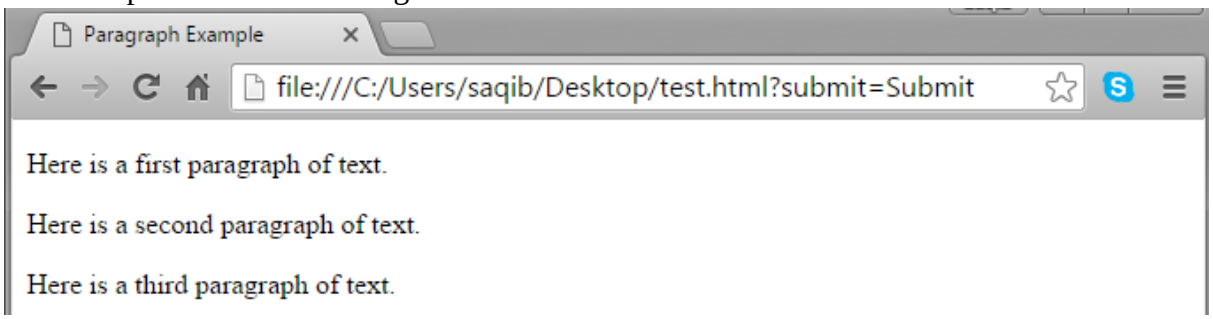
The tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening and a closing tag as shown below in the example:

Example

```
<!DOCTYPE html>
<html>
```

```
<head>
    <title>Paragraph Example</title>
</head>
<body>
    <p>Here is a first paragraph of text.</p>
    <p>Here is a second paragraph of text.</p>
    <p>Here is a third paragraph of text.</p>
</body>
</html>
```

This will produce the following result:



Line Break Tag

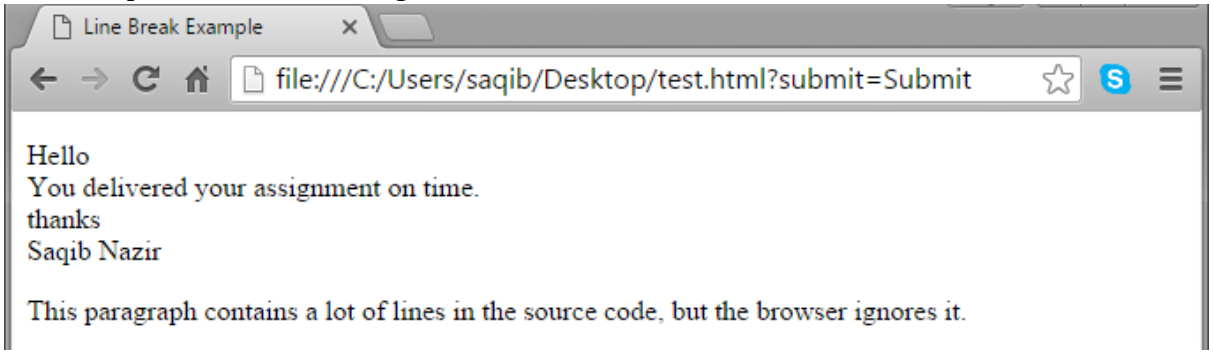
Whenever you use the
 element, anything following it starts from the next line. This tag is an example of an empty element, where you do not need opening and closing tags, as there is nothing to go in between them.

The
 tag has a space between the characters br and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use
 it is not valid in XHTML.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Line Break Example</title>
</head>
<body>
    <p>Hello<br />
        You delivered your assignment on time.<br />
        thanks<br />
        Saqib Nazir</p>
    <p>
            This paragraph
            contains a lot of lines
            in the source code,
            but the browser
            ignores it.
    </p>
</body>
</html>
```

This will produce the following result:

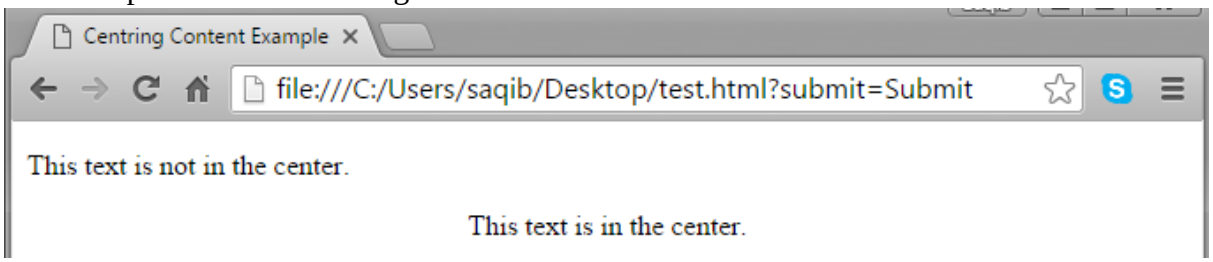


Centering Content

You can use tag to put any content in the center of the page or any table cell.

```
<!DOCTYPE html>
<html>
<head>
    <title>Centring Content Example</title>
</head>
<body>
    <p>This text is not in the center.</p>
    <center>
                <p>This text is in the center.</p>
      </center>
</body>
</html>
```

This will produce the following result:



Horizontal Lines

Horizontal lines are used to visually break-up sections of a document. The <hr> tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

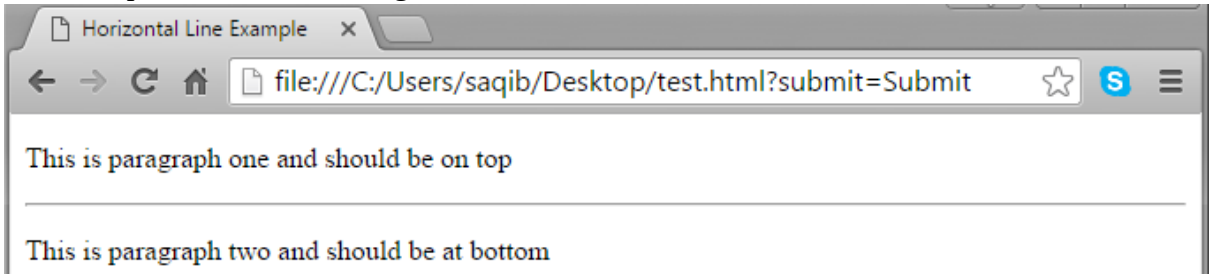
For example, you may want to give a line between two paragraphs as in the given example below:

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Horizontal Line Example</title>
</head>
<body>
    <p>This is paragraph one and should be on top</p>
```

```
    <hr />
    <p>This is paragraph two and should be at bottom</p>
</body>
</html>
```

This will produce the following result:



The <hr /> element has a space between the characters hr and the forward slash. If you omit this space, older browsers will have trouble rendering the horizontal line, while if you miss the forward slash character and just use <hr> it is not valid in XHTML.

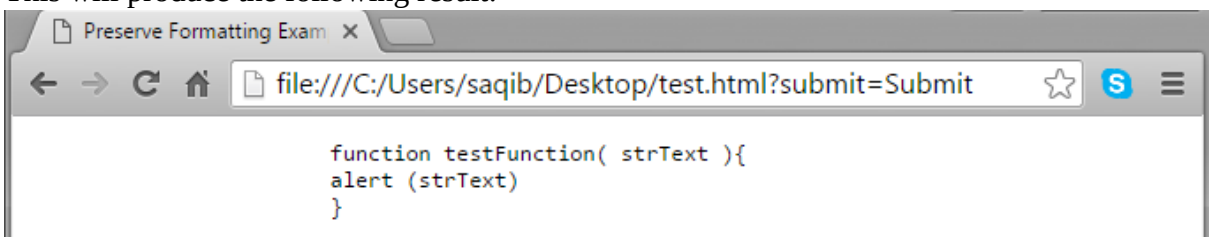
Preserve Formatting

Sometimes, you want your text to follow the exact format of how it is written in the HTML document. In these cases, you can use the preformatted tag <pre>.

Any text between the opening <pre> tag and the closing </pre> tag will preserve the formatting of the source document.

```
<!DOCTYPE html>
<html>
<head>
    <title>Preserve Formatting Example</title>
</head>
<body>
    <pre>
                function testFunction( strText ){
                alert (strText)
                }
            </pre>
</body>
</html>
```

This will produce the following result:



Try using the same code without keeping it inside <pre>...</pre> tags.

3. HTML Elimants

An HTML element is defined by a starting tag. If the element contains other content, it ends with a closing tag, where the element name is preceded by a forward slash as shown below with few tags:

Start Tag	*Content*	*End Tag*
* *	This is paragraph content.	* *
<h1>	This is heading content.	*</h1>*
* *	This is division content.	* *
* *		

So here .... is an HTML element, <h1>...</h1> is another HTML element. There are some HTML elements which don't need to be closed, such as <img.../>, <hr

/> and
 elements. These are known as void elements.

HTML documents consists of a tree of these elements and they specify how HTML documents should be built, and what kind of content should be placed in what part of an HTML document.

HTML Tag vs. Element

An HTML element is defined by a starting tag. If the element contains other content, it ends with a closing tag.

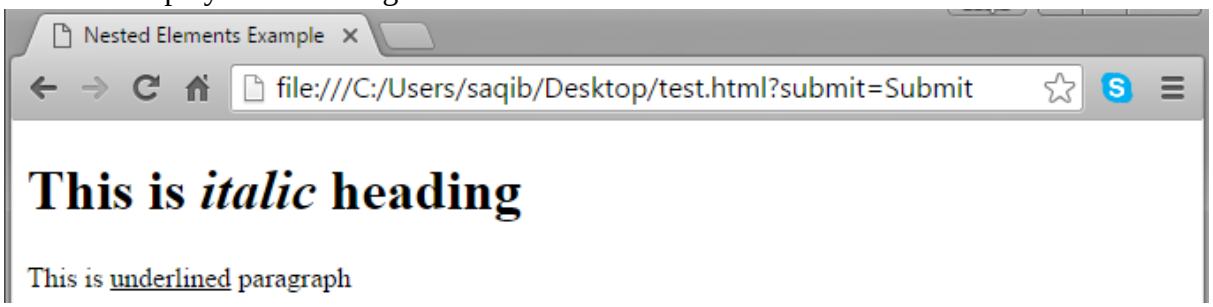
For example, is starting tag of a paragraph and is closing tag of the same paragraph but This is paragraph is a paragraph element.

Nested HTML Elements

It is very much allowed to keep one HTML element inside another HTML element:

```
<!DOCTYPE html>
<html>
<head>
    <title>Nested Elements Example</title>
</head>
<body>
    <h1>This is <i>italic</i> heading</h1>
    <p>This is <u>underlined</u> paragraph</p>
</body>
</html>
```

This will display the following result:



4. HTML Attribute

We have seen few HTML tags and their usage like heading tags <h1>, <h2>, paragraph tag <p> and other tags. We used them so far in their simplest form, but most of the HTML tags can also have attributes, which are extra bits of information.

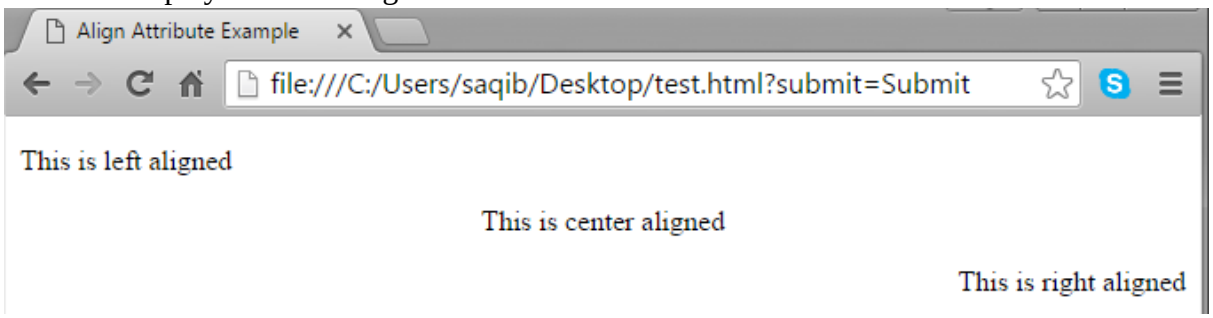
An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts: a name and a value:

- The name is the property you want to set. For example, the paragraph <p> element in the example carries an attribute whose name is align, which you can use to indicate the alignment of paragraph on the page.
- The value is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of align attribute: left, center and right.

Attribute names and attribute values are case-insensitive. However, the World Wide Web Consortium (W3C) recommends lowercase attributes/attribute values in their HTML 4 recommendation.

```
<!DOCTYPE html>
<html>
<head>
    <title>Align Attribute Example</title>
</head>
<body>
    <p align="left">This is left aligned</p>
    <p align="center">This is center aligned</p>
    <p align="right">This is right aligned</p>
</body>
</html>
```

This will display the following result:



Core Attributes

The four core attributes that can be used on the majority of HTML elements (although not all) are:

- Id
- Title
- Class
- Style

The Id Attribute

The id attribute of an HTML tag can be used to uniquely identify any element within an HTML page. There are two primary reasons that you might want to use an id attribute on an element:

- If an element carries an id attribute as a unique identifier, it is possible to identify just that element and its content.
- If you have two elements of the same name within a Web page (or style sheet), you can use the id attribute to distinguish between elements that have the same name.

We will discuss style sheet in separate tutorial. For now, let's use the id attribute to distinguish between two paragraph elements as shown below.

```
<p id="html">This para explains what is HTML</p>
<p id="css">This para explains what is Cascading Style Sheet</p>
```

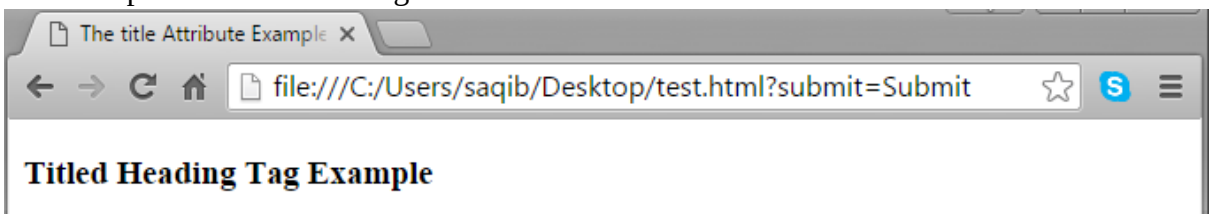
The title Attribute

The title attribute gives a suggested title for the element. They syntax for the title attribute is similar as explained for id attribute:

The behavior of this attribute will depend upon the element that carries it, although it is often displayed as a tooltip when cursor comes over the element or while the element is loading.

```
<!DOCTYPE html>
<html>
<head>
    <title>The title Attribute Example</title>
</head>
<body>
    <h3 title="Hello HTML!">Titled Heading Tag Example</h3>
</body>
</html>
```

This will produce the following result:



The class Attribute

The class attribute is used to associate an element with a style sheet, and specifies the class of element. You will learn more about the use of the class attribute when you will learn Cascading Style Sheet (CSS). So for now you can avoid it.

The value of the attribute may also be a space-separated list of class names. For example:

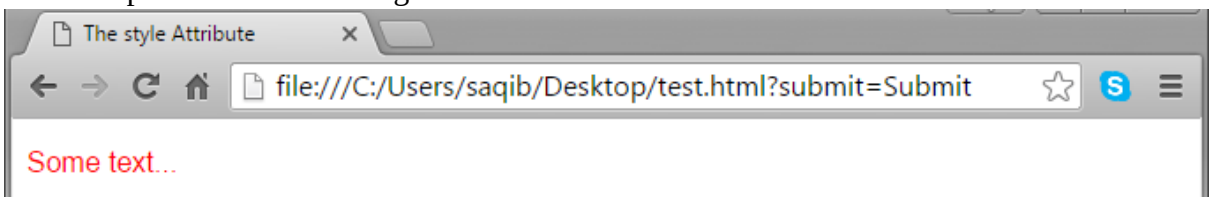
```
class="className1 className2 className3"
```

The style Attribute

The style attribute allows you to specify Cascading Style Sheet (CSS) rules within the element.

```
<!DOCTYPE html>
<html>
<head>
    <title>The style Attribute</title>
</head>
<body>
    <p style="font-family: arial; color: #FF0000;">Some text...</p>
</body>
</html>
```

This will produce the following result:



Internationalization Attributes

There are three internationalization attributes, which are available for most (although not all) XHTML elements

- dir
- lang
- xml:lang

The dir Attribute

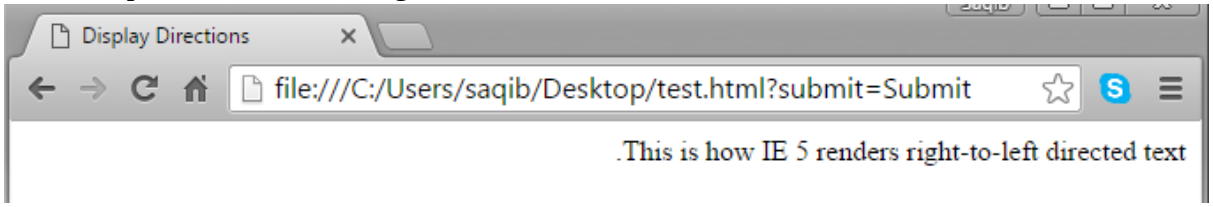
The dir attribute allows you to indicate to the browser about the direction in which the text should flow. The dir attribute can take one of two values, as you can see in the table that follows:

Value	Meaning
ltr	Left to right (the default value)
rtl	Right to left (for languages such as Hebrew or Arabic that are read right to left)

Example

```
<!DOCTYPE html>
<html dir="rtl">
<head>
    <title>Display Directions</title>
</head>
<body>
    This is how IE 5 renders right-to-left directed text.
</body>
</html>
```

This will produce the following result:



When dir attribute is used within the <html> tag, it determines how text will be presented within the entire document. When used within another tag, it controls the text's direction for just the content of that tag.

The lang Attribute

The lang attribute allows you to indicate the main language used in a document, but this attribute was kept in HTML only for backwards compatibility with earlier versions of HTML. This attribute has been replaced by the xml:lang attribute in new XHTML documents.

```
<!DOCTYPE html>
<html lang="en">
        <head>
                <title>English Language Page</title>
        </head>
        <body>
                This page is using English Language
        </body>
</html>
```

Generic Attributes

Here's a table of some other attributes that are readily usable with many of the HTML tags.

Attribute	Options	Function
Align	right, left, center	Horizontally aligns tags
valign	top, middle, bottom	Vertically aligns tags within an HTML element.
bgcolor	numeric, hexidecimal, RGB values	Places a background color behind an element
background	URL	Places a background image behind an element
id	User Defined	Names an element for use with Cascading Style Sheets.
class	User Defined	Classifies an element for use with Cascading Style Sheets.
height	Numeric Value	Specifies the height of tables, images, or table cells.
title	User Defined	"Pop-up" title of the elements.

We will see related examples as we will proceed to study other HTML tags.

5. HTML Formatting

If you use a word processor, you must be familiar with the ability to make text bold, italicized, or underlined; these are just three of the ten options available to indicate how text can appear in HTML and XHTML.

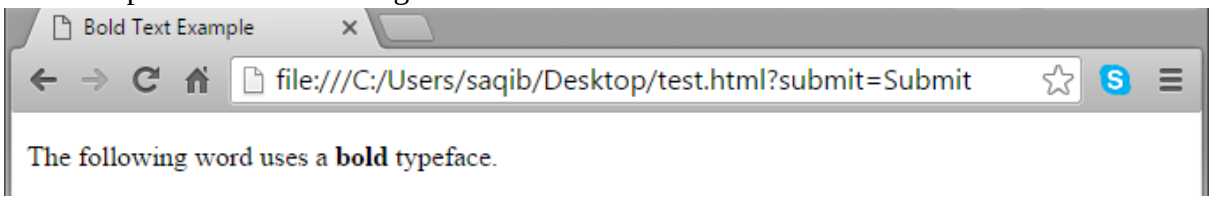
Bold Text

Anything that appears within ... element, is displayed in bold as shown below:

Example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Bold Text Example</title>
</head>
<body>
    <p>The following word uses a <b>bold</b> typeface.</p>
</body>
</html>
```

This will produce the following result:



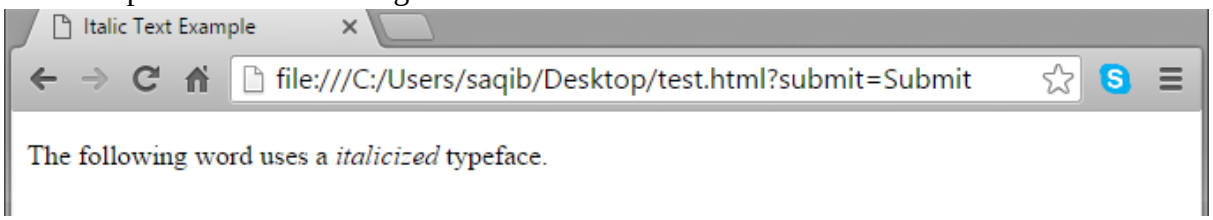
Italic Text

Anything that appears within ... element is displayed in italicized as shown below:

Example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Italic Text Example</title>
</head>
<body>
    <p>The following word uses a <i>italicized</i> typeface.</p>
</body>
</html>
```

This will produce the following result:



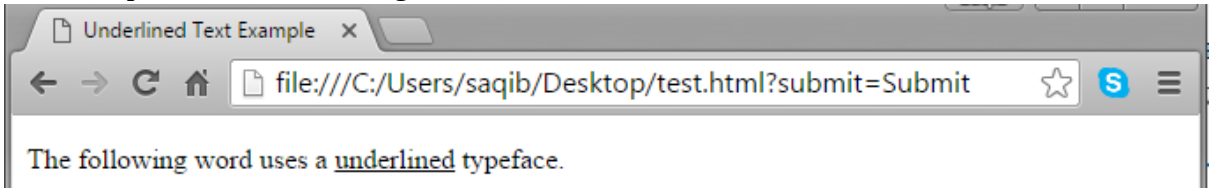
Underlined Text

Anything that appears within <u>...</u> element, is displayed with underline as shown below:

Example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Underlined Text Example</title>
</head>
<body>
    <p>The following word uses a <u>underlined</u> typeface.</p>
</body>
</html>
```

This will produce the following result:



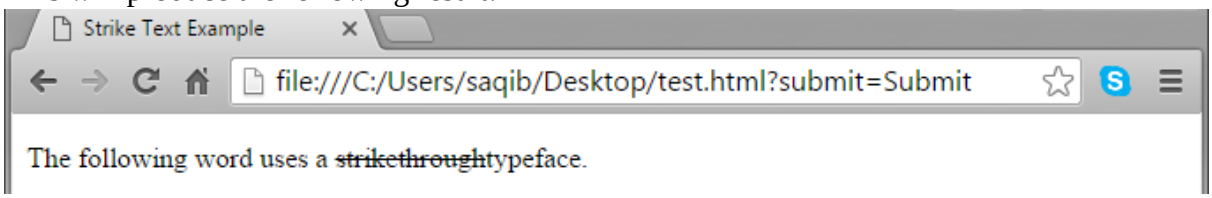
Strike Text

Anything that appears within <strike>...</strike> element is displayed with strikethrough, which is a thin line through the text as shown below:

Example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Strike Text Example</title>
</head>
<body>
    <p>The following word uses a
<strike>strikethrough</strike>typeface.</p>
</body>
</html>
```

This will produce the following result:

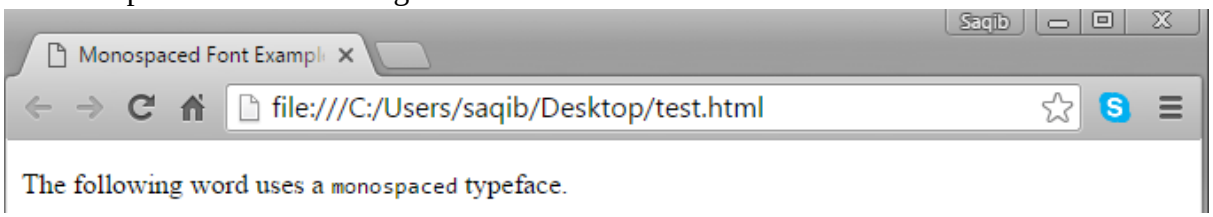


Monospaced Font

The content of a <tt>...</tt> element is written in monospaced font. Most of the fonts are known as variable-width fonts because different letters are of different widths (for example, the letter 'm' is wider than the letter 'i'). In a monospaced font, however, each letter has the same width.

```
<!DOCTYPE html>
<html>
	<head>
		<title>Monospaced Font Example</title>
	</head>
	<body>
		<p>The following word uses a <tt>monospaced</tt> typeface.</p>
	</body>
</html>
```

This will produce the following result:



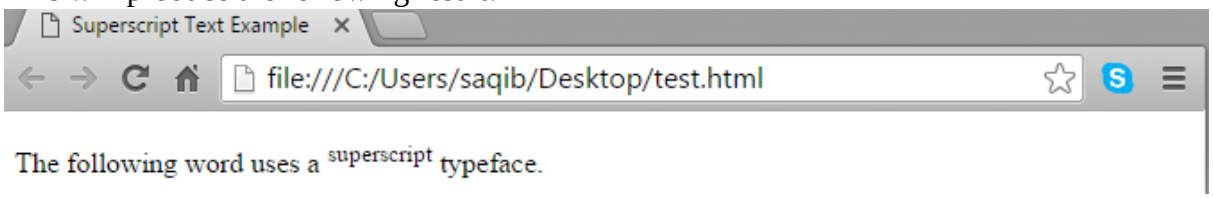
Superscript Text

The content of a ^{...} element is written in superscript; the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

Example:

```
<!DOCTYPE html>
<html>
	<head>
		<title>Superscript Text Example</title>
	</head>
	<body>
		<p>The following word uses a <sup>superscript</sup> typeface.</p>
	</body>
</html>
```

This will produce the following result:



Subscript Text

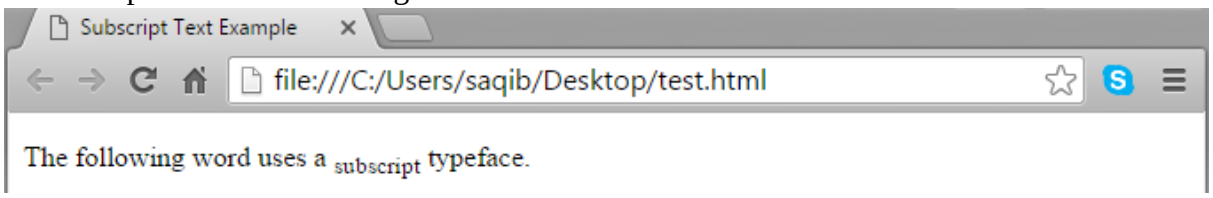
The content of a _{...} element is written in subscript; the font size used is the same as the characters surrounding it, but is displayed half a character's height beneath the other characters.

Example:

```
<!DOCTYPE html>
<html>
	<head>
		<title>Subscript Text Example</title>
```

```
        </head>
        <body>
            <p>The following word uses a <sub>subscript</sub> typeface.</p>
        </body>
</html>
```

This will produce the following result:



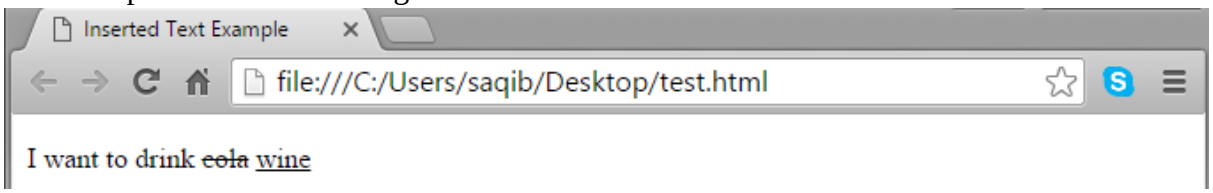
Inserted Text

Anything that appears within <ins>...</ins> element is displayed as inserted text.

Example:

```
<!DOCTYPE html>
<html>
        <head>
            <title>Inserted Text Example</title>
        </head>
        <body>
            <p>I want to drink <del>cola</del> <ins>wine</ins></p>
        </body>
</html>
```

This will produce the following result:



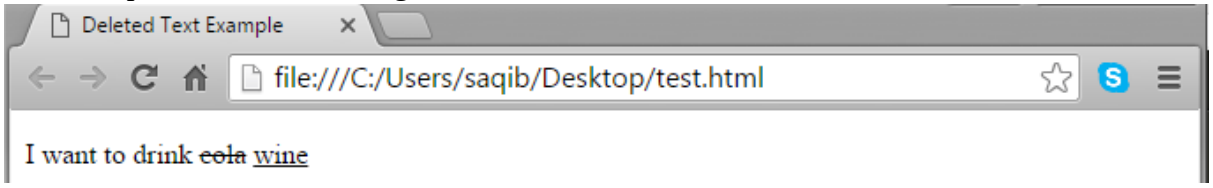
Deleted Text

Anything that appears within ~~...~~ element, is displayed as deleted text.

Example:

```
<!DOCTYPE html>
<html>
        <head>
            <title>Deleted Text Example</title>
        </head>
        <body>
            <p>I want to drink <del>cola</del> <ins>wine</ins></p>
        </body>
</html>
```

This will produce the following result:

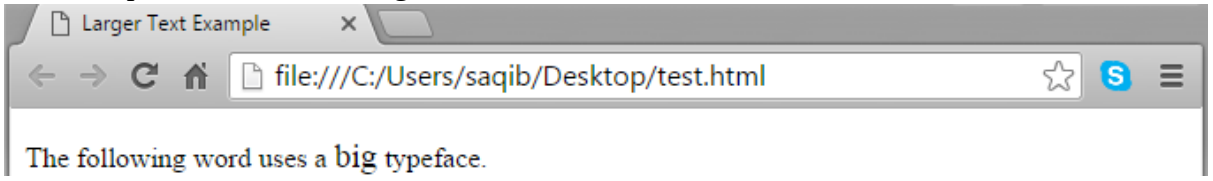


Larger Text

The content of the <big>...</big> element is displayed one font size larger than the rest of the text surrounding it as shown below:

```
<!DOCTYPE html>
<html>
        <head>
                <title>Larger Text Example</title>
        </head>
        <body>
                <p>The following word uses a <big>big</big> typeface.</p>
        </body>
</html>
```

This will produce the following result:

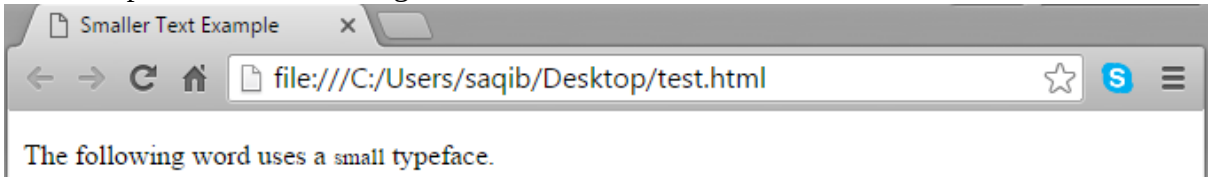


Smaller Text

The content of the <small>...</small> element is displayed one font size smaller than the rest of the text surrounding it as shown below:

```
<!DOCTYPE html>
<html>
        <head>
                <title>Smaller Text Example</title>
        </head>
        <body>
                <p>The following word uses a <small>small</small> typeface.</p>
        </body>
</html>
```

This will produce the following result:



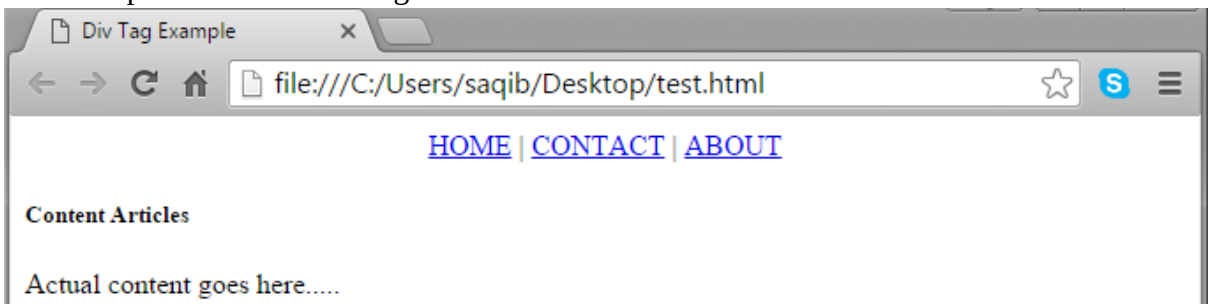
Grouping Content

The and elements allow you to group together several elements to create sections or subsections of a page.

For example, you might want to put all of the footnotes on a page within a element to indicate that all of the elements within that element relate to the footnotes. You might then attach a style to this element so that they appear using a special set of style rules.

```
<!DOCTYPE html>
<html>
	<head>
		<title>Div Tag Example</title>
	</head>
	<body>
		<div id="menu" align="middle" >
			<a href="/index.htm">HOME</a> |
			<a href="/about/contact_us.htm">CONTACT</a> |
			<a href="/about/index.htm">ABOUT</a>
		</div>
		<div id="content" align="left" bgcolor="white">
			<h5>Content Articles</h5>
			<p>Actual content goes here.....</p>
		</div>
	</body>
</html>
```

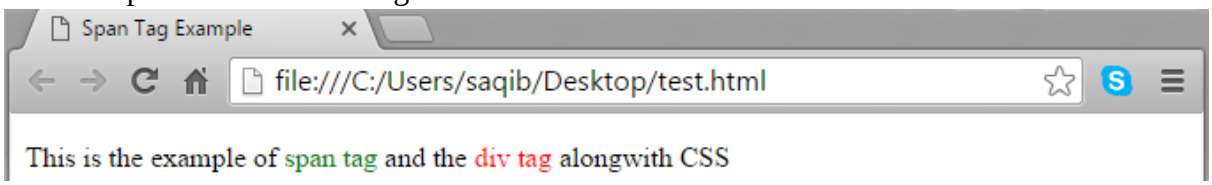
This will produce the following result:



The element, on the other hand, can be used to group inline elements only. So, if you have a part of a sentence or paragraph which you want to group together, you could use the element as follows Example:

```
<!DOCTYPE html>
<html>
	<head>
		<title>Span Tag Example</title>
	</head>
	<body>
		<p>This is the example of <span style="color:green">span tag</span> and the
	<span style="color:red">div tag</span> alongwith CSS</p>
</body>
</html>
```

This will produce the following result:



6. HTML Phrase Tag

The phrase tags have been desicolgned for specific purposes, though they are displayed in a similar way as other basic tags like <b>, <i>, <pre>, and <tt>, you have seen in previous chapter. This chapter will take you through all the important phrase tags, so let's start seeing them one by one.

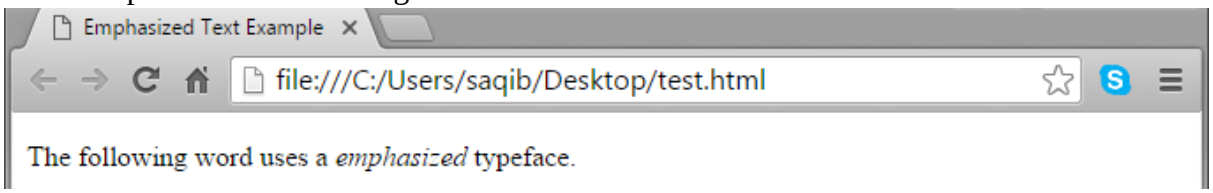
Emphasized Text

Anything that appears within ... element is displayed as emphasized text.

Example

```
<!DOCTYPE html>
<html>
        <head>
                <title>Emphasized Text Example</title>
        </head>
        <body>
                <p>The following word uses a <em>emphasized</em> typeface.</p>
        </body>
</html>
```

This will produce the following result:



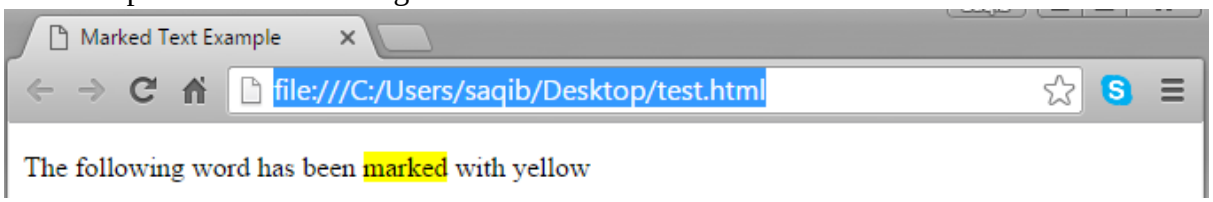
Marked Text

Anything that appears with-in <mark>...</mark> element, is displayed as marked with yellow ink.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Marked Text Example</title>
</head>
<body>
    <p>The following word has been <mark>marked</mark> with yellow</p>
</body>
</html>
```

This will produce the following result:



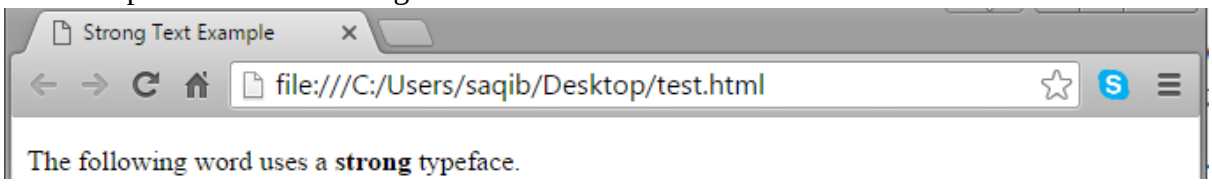
Strong Text

Anything that appears within ... element is displayed as important text.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Strong Text Example</title>
</head>
<body>
    <p>The following word uses a <strong>strong</strong> typeface.</p>
</body>
</html>
```

This will produce the following result:



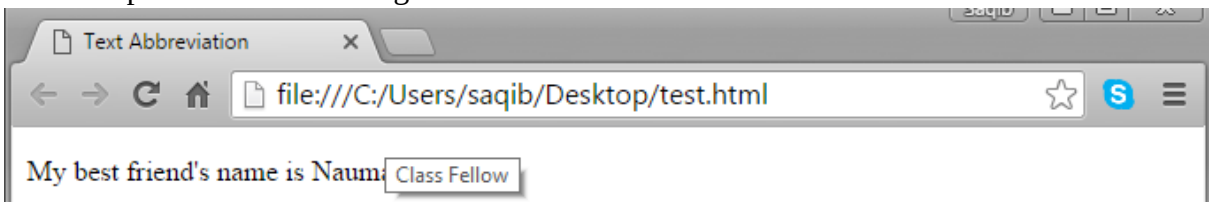
Text Abbreviation

You can abbreviate a text by putting it inside opening <abbr> and closing </abbr> tags. If present, the title attribute must contain this full description and nothing else.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Text Abbreviation</title>
</head>
<body>
    <p>My best friend's name is
        <abbr title="Class Fellow">Nauman</abbr>.</p>
</body>
</html>
```

This will produce the following result:



Acronym Element

The <acronym> element allows you to indicate that the text between <acronym> and

</acronym> tags is an acronym.

At present, the major browsers do not change the appearance of the content of the

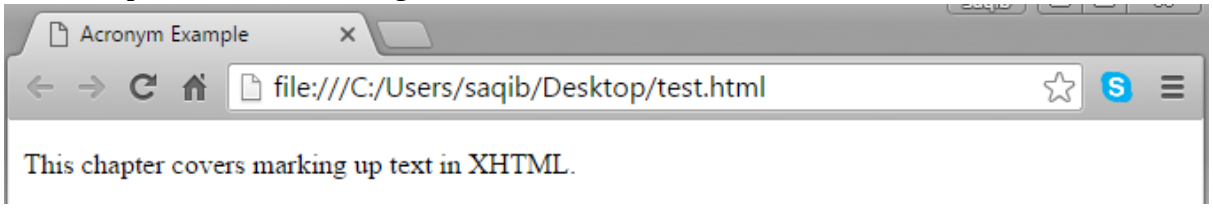
<acronym> element.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Acronym Example</title>
</head>
```

```
<body>
    <p>This chapter covers marking up text in <acronym>XHTML</acronym>.</p>
</body>
</html>
```

This will produce the following result:



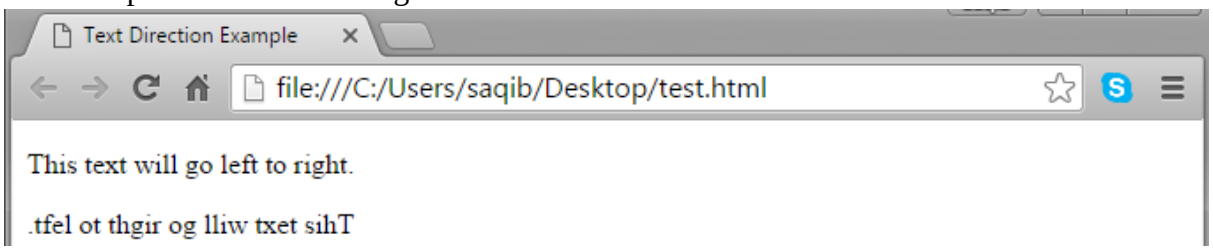
Text Direction

The <bdo>...</bdo> element stands for Bi-Directional Override and it is used to override the current text direction.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Text Direction Example</title>
</head>
<body>
    <p>This text will go left to right.</p>
    <p><bdo dir="rtl">This text will go right to left.</bdo></p>
</body>
</html>
```

This will produce the following result:



Special Terms

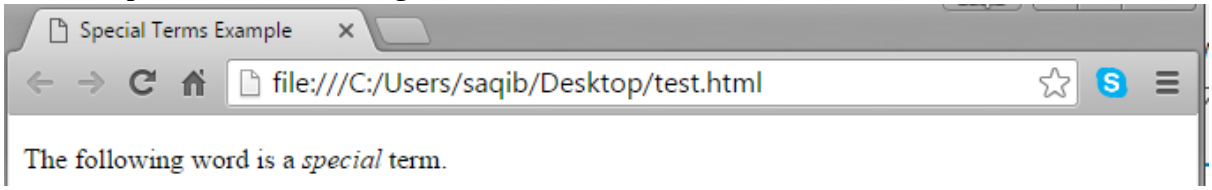
The <dfn>...</dfn> element (or HTML Definition Element) allows you to specify that you are introducing a special term. It's usage is similar to italic words in the midst of a paragraph.

Typically, you would use the <dfn> element the first time you introduce a key term. Most recent browsers render the content of a <dfn> element in an italic font.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Special Terms Example</title>
</head>
<body>
    <p>The following word is a <dfn>special</dfn> term.</p>
</body>
</html>
```

This will produce the following result:



Quoting Text

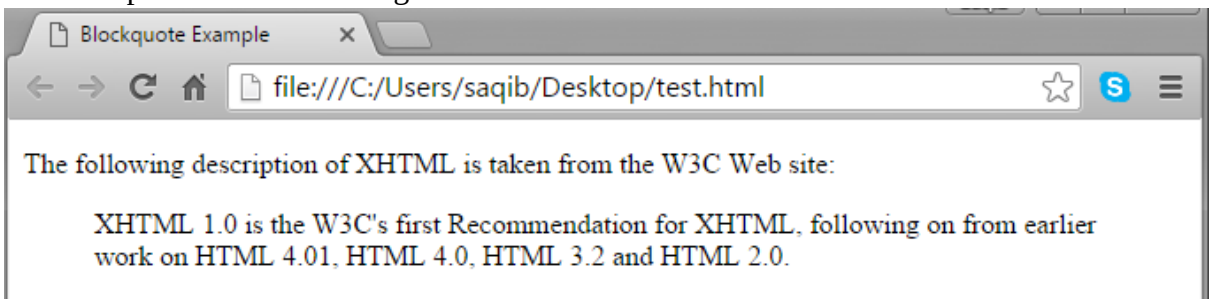
When you want to quote a passage from another source, you should put it in between<blockquote>...</blockquote> tags.

Text inside a <blockquote> element is usually indented from the left and right edges of the surrounding text, and sometimes uses an italicized font.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Blockquote Example</title>
</head>
<body>
    <p>The following description of XHTML is taken from the W3C Web site:</p>
    <blockquote>XHTML 1.0 is the W3C's first Recommendation for XHTML, following on
from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML 2.0.</blockquote>
</body>
</html>
```

This will produce the following result:



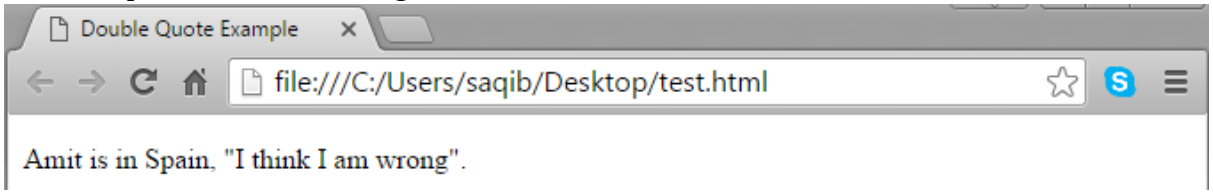
Short Quotations

The <q>...</q> element is used when you want to add a double quote within a sentence.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Double Quote Example</title>
</head>
<body>
    <p>Amit is in Spain, <q>I think I am wrong</q>.</p>
</body>
</html>
```

This will produce the following result:



Text Citations

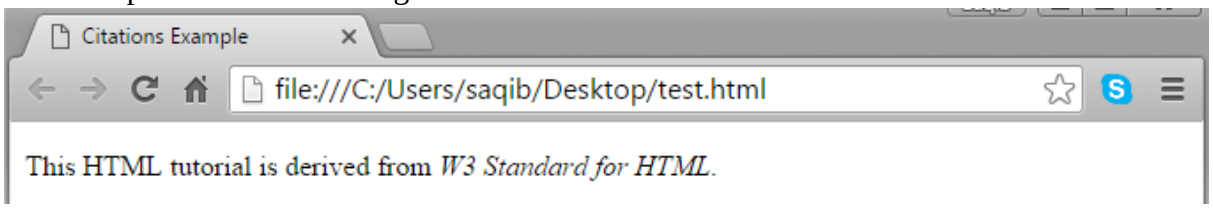
If you are quoting a text, you can indicate the source placing it between an opening <cite>tag and closing </cite> tag

As you would expect in a print publication, the content of the <cite> element is rendered in italicized text by default.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Citations Example</title>
</head>
<body>
    <p>This HTML tutorial is derived from <cite>W3 Standard for HTML</cite>.</p>
</body>
</html>
```

This will produce the following result:



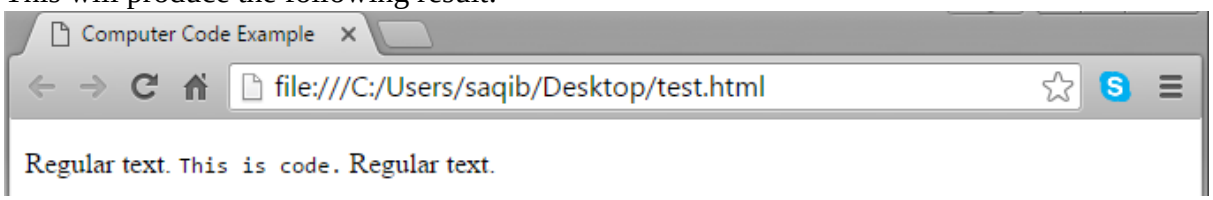
Computer Code

Any programming code to appear on a Web page should be placed inside <code>...</code>tags. Usually the content of the <code> element is presented in a monospaced font, just like the code in most programming books.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Computer Code Example</title>
</head>
<body>
    <p>Regular text. <code>This is code.</code> Regular text.</p>
</body>
</html>
```

This will produce the following result:



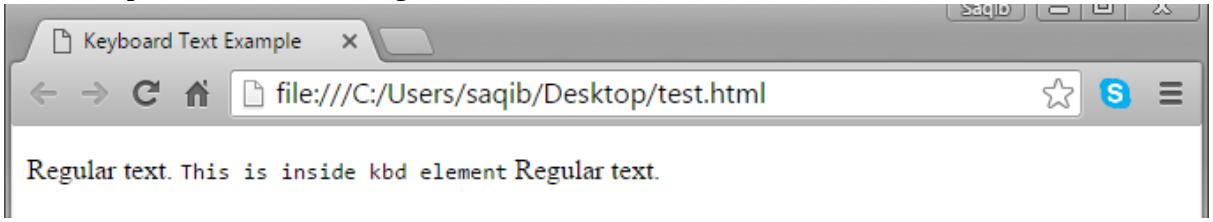
Keyboard Text

When you are talking about computers, if you want to tell a reader to enter some text, you can use the <kbd>...</kbd> element to indicate what should be typed in, as in this example.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Keyboard Text Example</title>
</head>
<body>
    <p>Regular text. <kbd>This is inside kbd element</kbd> Regular text.</p>
</body>
</html>
```

This will produce the following result:



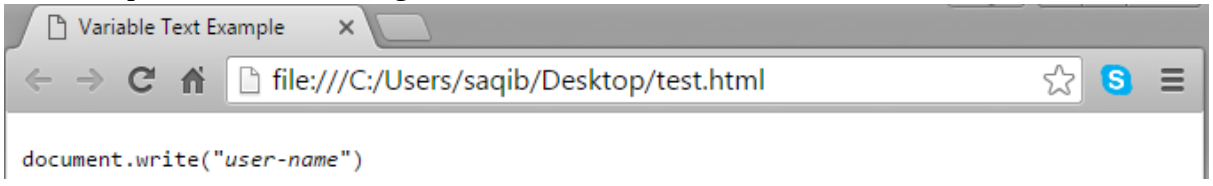
Programming Variables

This element is usually used in conjunction with the <pre> and <code> elements to indicate that the content of that element is a variable.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Variable Text Example</title>
</head>
<body>
    <p><code>document.write("<var>user-name</var>")</code></p>
</body>
</html>
```

This will produce the following result:



Program Output

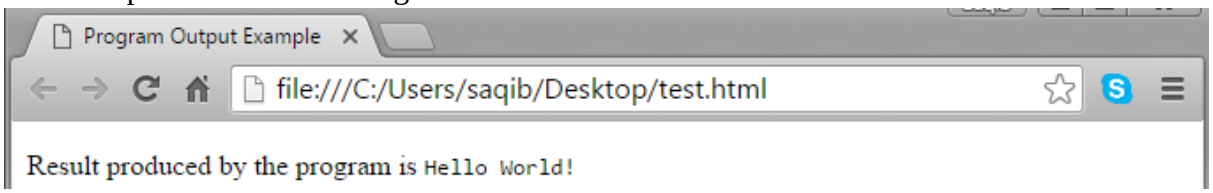
The <samp>...</samp> element indicates sample output from a program, and script etc. Again, it is mainly used when documenting programming or coding concepts.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Program Output Example</title>
</head>
```

```
<body>
    <p>Result produced by the program is
        <samp>Hello World!</samp></p>
</body>
</html>
```

This will produce the following result:



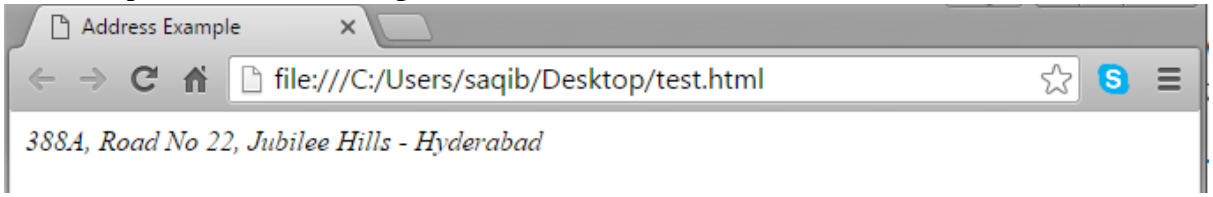
Address Text

The <address>...</address> element is used to contain any address.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Address Example</title>
</head>
<body>
    <address>388A, Road No 22, Jubilee Hills - Hyderabad</address>
</body>
</html>
```

This will produce the following result:



7. HTML – META TAGS

HTML lets you specify metadata - additional important information about a document in a variety of ways. The META elements can be used to include name/value pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author etc.

The <meta> tag is used to provide such additional information. This tag is an empty element and so does not have a closing tag but it carries information within its attributes.

You can include one or more meta tags in your document based on what information you want to keep in your document but in general, meta tags do not impact physical appearance of the document so from appearance point of view, it does not matter if you include them or not.

Adding Meta Tags to Your Documents

You can add metadata to your web pages by placing <meta> tags inside the header of the document which is represented by <head> and </head> tags. A meta tag can have following attributes in addition to core attributes:

Attribute	Description
Name	Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc.
content	Specifies the property's value.
scheme	Specifies a scheme to interpret the property's value (as declared in the content attribute).
http- equiv	Used for http response message headers. For example, http-equiv can be used to refresh the page or to set a cookie. Values include content-type, expires, refresh and set-cookie.

Specifying Keywords

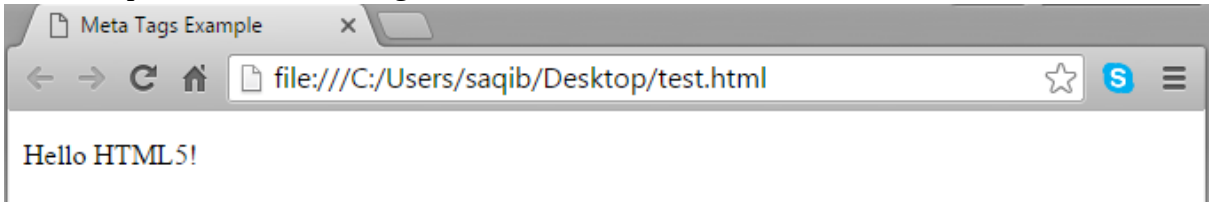
You can use <meta> tag to specify important keywords related to the document and later these keywords are used by the search engines while indexing your webpage for searching purpose.

Example

Following is an example, where we are adding HTML, Meta Tags, Metadata as important keywords about the document.

```
<!DOCTYPE html>
<html>
<head>
    <title>Meta Tags Example</title>
    <meta name="keywords" content="HTML, Meta Tags, Metadata" />
</head>
<body>
    <p>Hello HTML5!</p>
</body>
</html>
```


This will produce the following result:



Document Description

You can use <meta> tag to give a short description about the document. This again can be used by various search engines while indexing your webpage for searching purpose.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Meta Tags Example</title>
    <meta name="keywords" content="HTML, Meta Tags, Metadata" />
    <meta name="description" content="Learning about Meta Tags." />
</head>
<body>
    <p>Hello HTML5!</p>
</body>
</html>
```

Document Revision Date

You can use <meta> tag to give information about when last time the document was updated. This information can be used by various web browsers while refreshing your webpage.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Meta Tags Example</title>
    <meta name="keywords" content="HTML, Meta Tags, Metadata" />
    <meta name="description" content="Learning about Meta Tags." />
    <meta name="revised" content="Kics, 28/12/2015" />
</head>
<body>
    <p>Hello HTML5!</p>
</body>
</html>
```

Document Refreshing

A <meta> tag can be used to specify a duration after which your web page will keep refreshing automatically.

Example

If you want your page keep refreshing after every 5 seconds then use the following syntax.

```
<!DOCTYPE html>
<html>
<head>
    <title>Meta Tags Example</title>
    <meta name="keywords" content="HTML, Meta Tags, Metadata" />
    <meta name="description" content="Learning about Meta Tags." />
```

```
    <meta name="revised" content="Kics, 28/12/2015" />
    <meta http-equiv="refresh" content="5" />
</head>
<body>
    <p>Hello HTML5!</p>
</body>
</html>
```

Page Redirection

You can use <meta> tag to redirect your page to any other webpage. You can also specify a duration if you want to redirect the page after a certain number of seconds.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Meta Tags Example</title>
    <meta name="keywords" content="HTML, Meta Tags, Metadata" />
    <meta name="description" content="Learning about Meta Tags." />
    <meta name="revised" content="Kics, 28/12/2015" />
    <meta http-equiv="refresh" content="5; url=http://www.kics.edu.pk" />
</head>
<body>
    <p>Hello HTML5!</p>
</body>
</html>
```

Following is an example of redirecting current page to another page after 5 seconds. If you want to redirect page immediately then do not specify content attribute.

Setting Cookies

Cookies are data, stored in small text files on your computer and it is exchanged between web browser and web server to keep track of various information based on your web application need.

You can use <meta> tag to store cookies on client side and later this information can be used by the Web Server to track a site visitor.

Example

Following is an example of redirecting current page to another page after 5 seconds. If you want to redirect page immediately then do not specify content attribute.

```
<!DOCTYPE html>
<html>
<head>
    <title>Meta Tags Example</title>
    <meta name="keywords" content="HTML, Meta Tags, Metadata" />
    <meta name="description" content="Learning about Meta Tags." />
    <meta name="revised" content="Kics, 28/12/2015" />
    <meta http-equiv="refresh" content="5; url=http://www.kics.edu.pk" />
    <meta http-equiv="cookie" content="userid=xyz; expires=Wednesday, 08-Aug-15
    23:59:59 GMT;" />
</head>
<body>
    <p>Hello HTML5!</p>
</body>
</html>
```

If you do not include the expiration date and time, the cookie is considered a session cookie and will be deleted when the user exits the browser.

Note: You can check PHP and Cookies tutorial for a complete detail on Cookies.

Setting Author Name

You can set an author name in a web page using meta tag. See an example below:

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Meta Tags Example</title>
    <meta name="keywords" content="HTML, Meta Tags, Metadata" />
    <meta name="description" content="Learning about Meta Tags." />
    <meta name="revised" content="Kics, 28/12/2015" />
    <meta http-equiv="refresh" content="5; url=http://www.kics.edu.pk" />
    <meta http-equiv="cookie" content="userid=xyz; expires=Wednesday, 08-Aug-15
    23:59:59 GMT;" />
    <meta name="author" content="Saqib Nazir" />
</head>
<body>
    <p>Hello HTML5!</p>
</body>
</html>
```

Specify Character Set

You can use <meta> tag to specify character set used within the webpage.

Example

By default, Web servers and Web browsers use ISO-8859-1 (Latin1) encoding to process

Web pages. Following is an example to set UTF-8 encoding:

```
<!DOCTYPE html>
<html>
<head>
    <title>Meta Tags Example</title>
    <meta name="keywords" content="HTML, Meta Tags, Metadata" />
    <meta name="description" content="Learning about Meta Tags." />
    <meta name="revised" content="Kics, 28/12/2015" />
    <meta http-equiv="refresh" content="5; url=http://www.kics.edu.pk" />
    <meta http-equiv="cookie" content="userid=xyz; expires=Wednesday, 08-Aug-15
    23:59:59 GMT;" />
    <meta name="author" content="Saqib Nazir" />
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
    <p>Hello HTML5!</p>
</body>
</html>
```

To serve the static page with traditional Chinese characters, the webpage must contain a

<meta> tag to set Big5 encoding:

```
<!DOCTYPE html>
<html>
<head>
```

```
    <title>Meta Tags Example</title>
    <meta name="keywords" content="HTML, Meta Tags, Metadata" />
    <meta name="description" content="Learning about Meta Tags." />
    <meta name="revised" content="Kics, 28/12/2015" />
    <meta http-equiv="refresh" content="5; url=http://www.kics.edu.pk" />
    <meta http-equiv="cookie" content="userid=xyz; expires=Wednesday, 08-Aug-15
    23:59:59 GMT;" />
    <meta name="author" content="Saqib Nazir" />
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
    <p>Hello HTML5!</p>
</body>
</html>
```

8. HTML – COMMENTS

Comment is a piece of code which is ignored by any web browser. It is a good practice to add comments into your HTML code, especially in complex documents, to indicate sections of a document, and any other notes to anyone looking at the code. Comments help you and others understand your code and increases code readability.

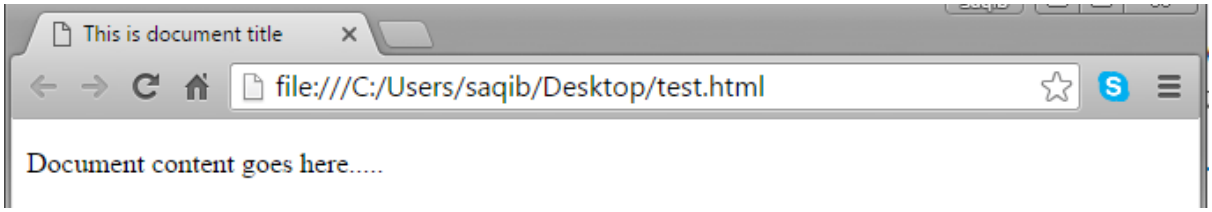
HTML comments are placed in between <!-- ... --> tags. So, any content placed with-in

<!-- ... --> tags will be treated as comment and will be completely ignored by the browser.

Example

```
<!DOCTYPE html>
<html>
<head>
    <!-- Document Header Starts -->
    <title>This is document title</title>
</head>
<!-- Document Header Ends -->
<body>
    <p>Document content goes here.....</p>
</body>
</html>
```

This will produce the following result without displaying the content given as a part of comments:



Valid vs Invalid Comments

Comments do not nest which means a comment cannot be put inside another comment. Second the double-dash sequence "--" may not appear inside a comment except as part of the closing --> tag. You must also make sure that there are no spaces in the start-of- comment string.

Example

Here, the given comment is a valid comment and will be wiped off by the browser.

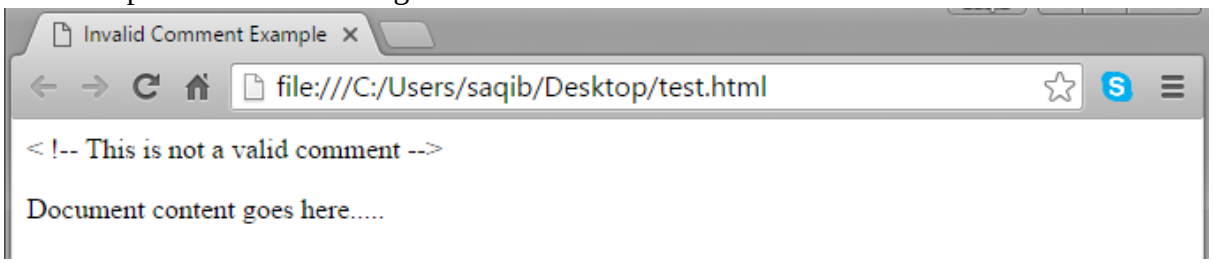
```
<!DOCTYPE html>
<html>
<head>
    <title>Valid Comment Example</title>
</head>
<body>
    <!--  This is valid comment -->
    <p>Document content goes here.....</p>
</body>
</html>
```

But, following line is not a valid comment and will be displayed by the browser. This is because there is a space between the left angle bracket and the exclamation mark.

```
<!DOCTYPE html>
```

```
<html>
<head>
    <title>Invalid Comment Example</title>
</head>
<body>
    < !--  This is not a valid comment -->
    <p>Document content goes here.....</p>
</body>
</html>
```

This will produce the following result:



Multiline Comments

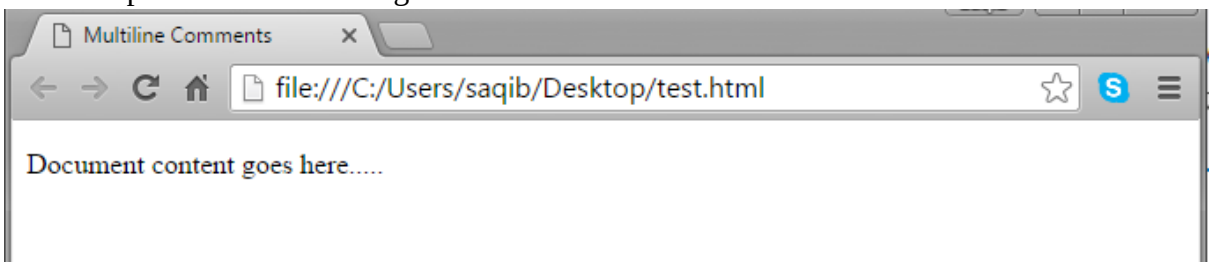
So far we have seen single line comments, but HTML supports multi-line comments as well.

You can comment multiple lines by the special beginning tag <!-- and ending tag --> placed before the first line and end of the last line as shown in the given example below.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Multiline Comments</title>
</head>
<body>
    <!--
This is a multiline comment and it can span
through as many as lines you like.
-->
    <p>Document content goes here.....</p>
</body>
</html>
```

This will produce the following result:



Conditional comments only work in Internet Explorer (IE) on Windows but they are ignored by other browsers. They are supported from Explorer 5 onwards, and you can use them to give conditional instructions to different versions of IE.

Example

```
<!DOCTYPE html>
<html>
<head>
```

```
    <title>Conditional Comments</title>
    <!--[if IE 6]>
Special instructions for IE 6 here
<![endif]-->
</head>
<body>
    <p>Document content goes here.....</p>
</body>
</html>
```

You will come across a situation where you will need to apply a different style sheet based on different versions of Internet Explorer, in such situation conditional comments will be helpful.

9. HTML – IMAGES

Images are very important to beautify as well as to depict many complex concepts in simple way on your web page. This tutorial will take you through simple steps to use images in your web pages.

Insert Image

You can insert any image in your web page by using <img> tag. Following is the simple syntax to use this tag.

```
<img src="Image URL" ... attributes-list/>
```

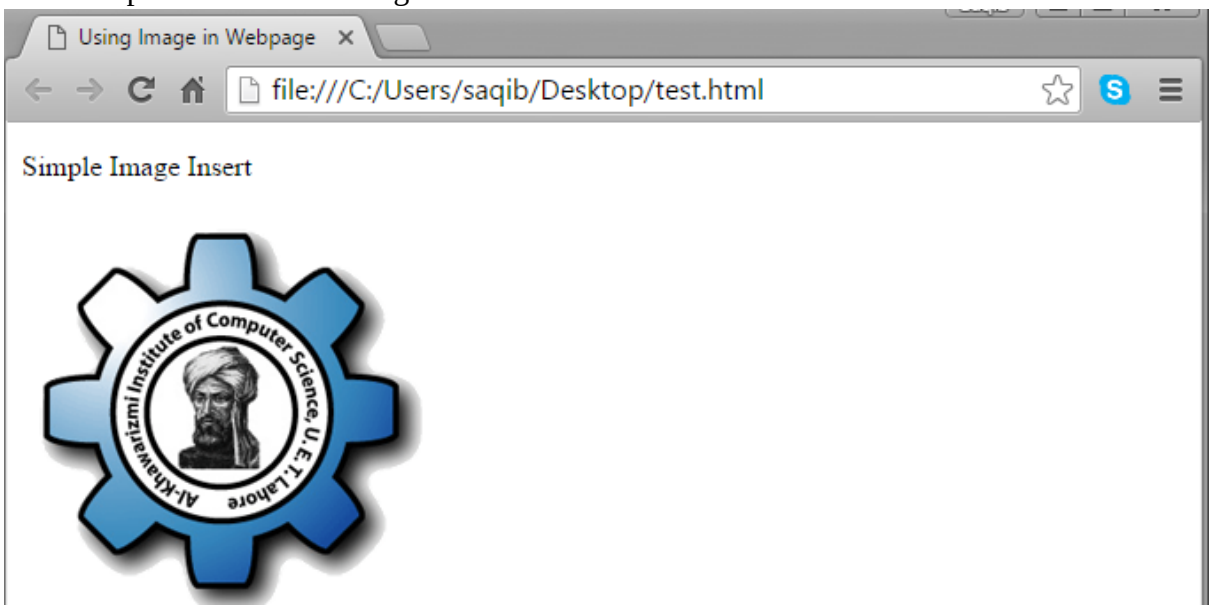
The <img> tag is an empty tag, which means that, it can contain only list of attributes and it has no closing tag.

Example

To try following example, let's keep our HTML file test.htm and image file test.png in the same directory:

```
<!DOCTYPE html>
<html>
<head>
    <title>Using Image in Webpage</title>
</head>
<body>
    <p>Simple Image Insert</p>
    <img src="test.png" alt="Test Image" />
</body>
</html>
```

This will produce the following result:



You can use PNG, JPEG or GIF image file based on your comfort but make sure you specify correct image file name in src attribute. Image name is always case sensitive.

The alt attribute is a mandatory attribute which specifies an alternate text for an image, if the image cannot be displayed.

Set Image Location

Usually we keep all the images in a separate directory. So let's keep HTML file test.htm in our home directory and create a subdirectory images inside the home directory where we will keep our image test.png.

Example

Assuming our image location is "image/test.png", try the following example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Using Image in Webpage</title>
</head>
<body>
    <p>Simple Image Insert</p>
    <img src="images/test.png" alt="Test Image" />
</body>
</html>
```

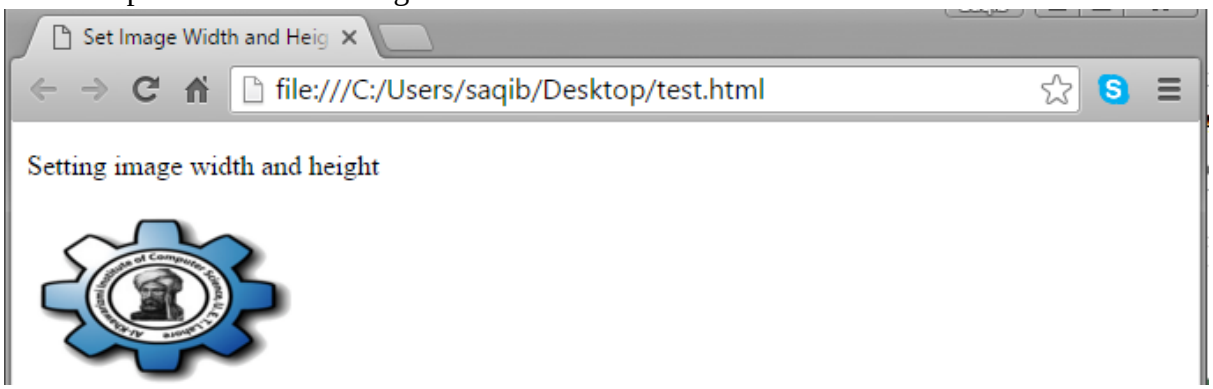
Set Image Width/Height

You can set image width and height based on your requirement using width and height attributes. You can specify width and height of the image in terms of either pixels or percentage of its actual size.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Set Image Width and Height</title>
</head>
<body>
    <p>Setting image width and height</p>
    <img src="test.png" alt="Test Image" width="150" height="100" />
</body>
</html>
```

This will produce the following result:



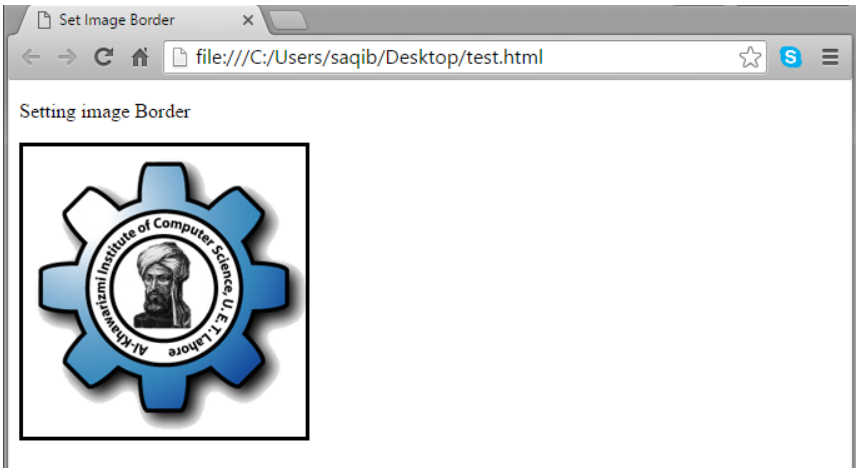
Set Image Border

By default, image will have a border around it, you can specify border thickness in terms of pixels using border attribute. A thickness of 0 means, no border around the picture.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Set Image Border</title>
</head>
<body>
    <p>Setting image Border</p>
    <img src="test.png" alt="Test Image" border="3" />
</body>
</html>
```

This will produce the following result:



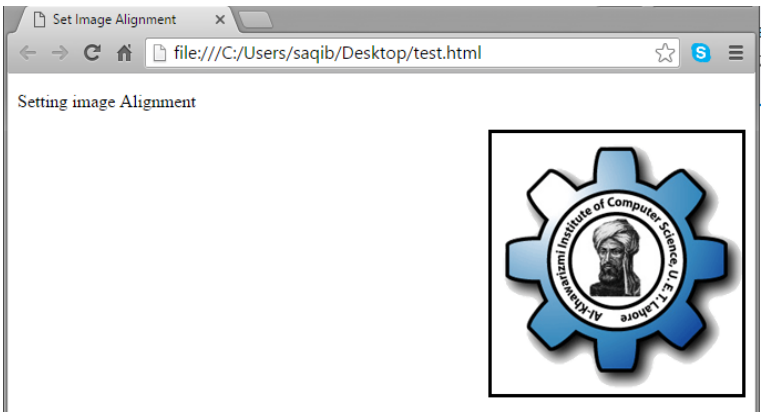
Set Image Alignment

By default, image will align at the left side of the page, but you can use align attribute to set it in the center or right.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>Set Image Alignment</title>
</head>
<body>
    <p>Setting image Alignment</p>
    <img src="test.png" alt="Test Image" border="3" align="right" />
</body>
</html>
```

This will produce the following result:



10. HTML – TABLES

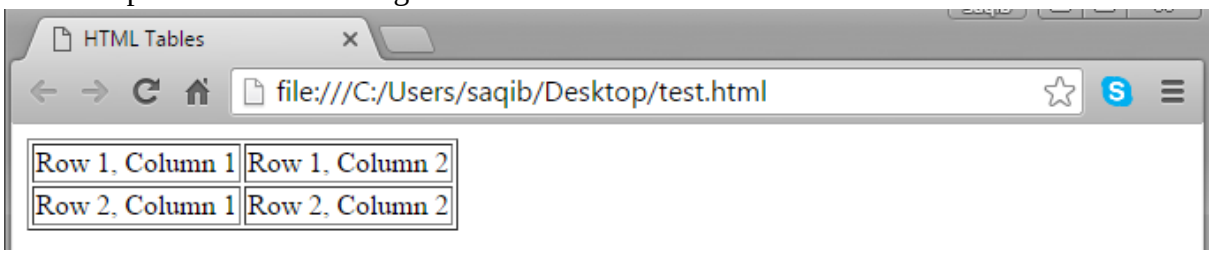
The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.

The HTML tables are created using the <table> tag in which the <tr> tag is used to create table rows and <td> tag is used to create data cells.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Tables</title>
</head>
<body>
    <table border="1">
        <tr>
            <td>Row 1, Column 1</td>
            <td>Row 1, Column 2</td>
        </tr>
        <tr>
            <td>Row 2, Column 1</td>
            <td>Row 2, Column 2</td>
        </tr>
    </table>
</body>
</html>
```

This will produce the following result:



Here, the border is an attribute of <table> tag and it is used to put a border across all the cells. If you do not need a border, then you can use border="0".

Table Heading

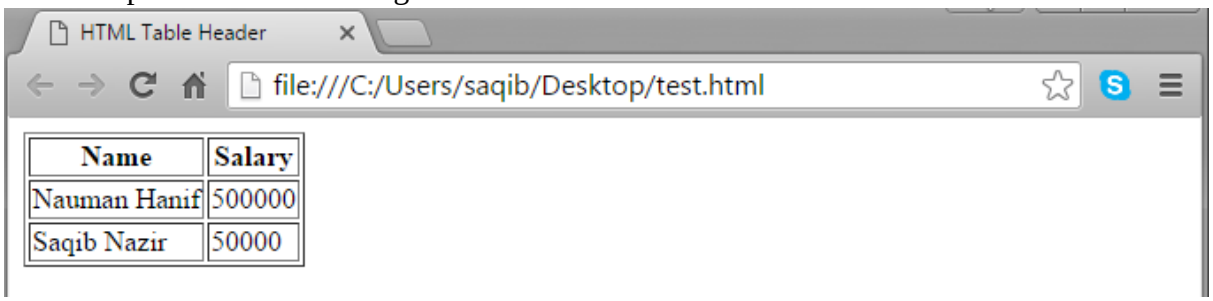
Table heading can be defined using <th> tag. This tag will be put to replace <td> tag, which is used to represent actual data cell. Normally you will put your top row as table heading as shown below, otherwise you can use <th> element in any row.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Table Header</title>
</head>
<body>
    <table border="1">
        <tr>
            <th>Name</th>
            <th>Salary</th>
        </tr>
```

```
        <tr>
            <td>Nauman Hanif</td>
            <td>500000</td>
        </tr>
        <tr>
            <td>Saqib Nazir</td>
            <td>50000</td>
        </tr>
    </table>
</body>
</html>
```

This will produce the following result:



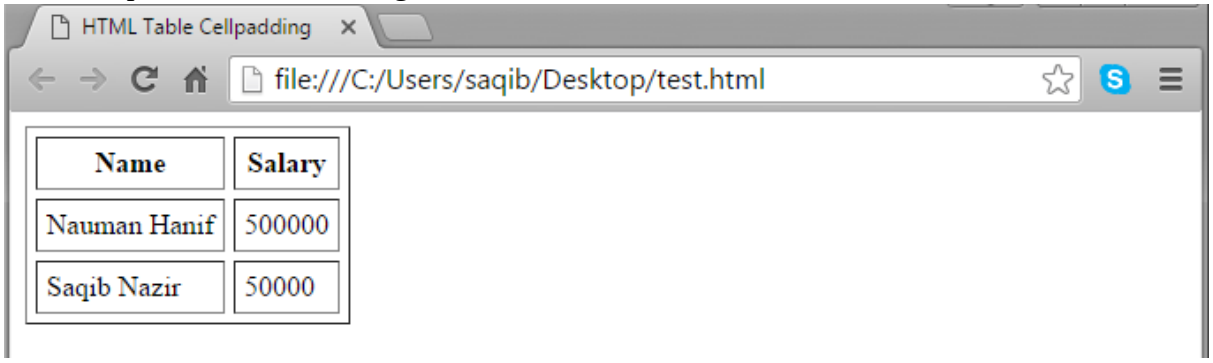
Cellpadding and Cellspacing Attributes

There are two attributes called cellpadding and cellspacing which you will use to adjust the white space in your table cells. The cellspacing attribute defines the width of the border, while cellpadding represents the distance between cell borders and the content within a cell.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Table Cellpadding</title>
</head>
<body>
    <table border="1" cellpadding="5" cellspacing="5">
        <tr>
            <th>Name</th>
            <th>Salary</th>
        </tr>
        <tr>
            <td>Nauman Hanif</td>
            <td>500000</td>
        </tr>
        <tr>
            <td>Saqib Nazir</td>
            <td>50000</td>
        </tr>
    </table>
</body>
</html>
```

This will produce the following result:



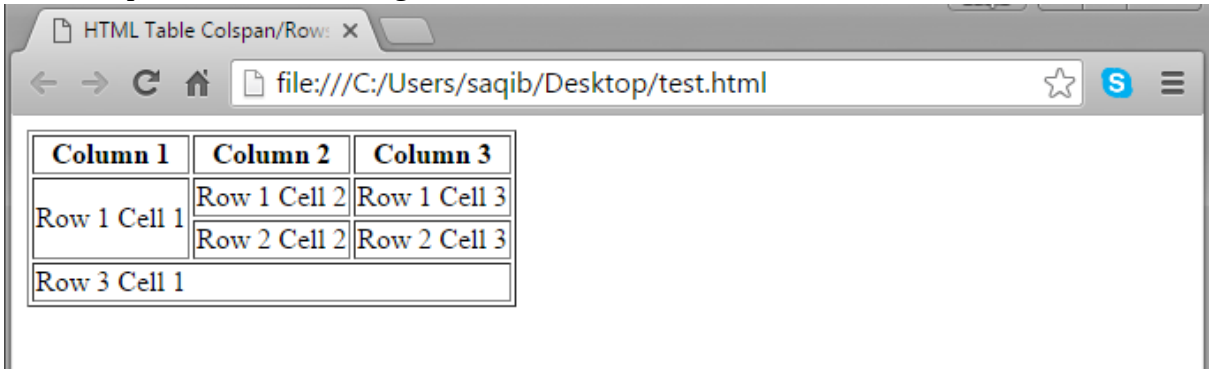
Colspan and Rowspan Attributes

You will use colspan attribute if you want to merge two or more columns into a single column. Similar way you will use rowspan if you want to merge two or more rows.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Table Colspan/Rowspan</title>
</head>
<body>
    <table border="1">
        <tr>
            <th>Column 1</th>
            <th>Column 2</th>
            <th>Column 3</th>
        </tr>
        <tr>
            <td rowspan="2">Row 1 Cell 1</td>
            <td>Row 1 Cell 2</td>
            <td>Row 1 Cell
3</td>
        </tr>
        <tr>
            <td>Row 2 Cell 2</td>
            <td>Row 2 Cell 3</td>
        </tr>
        <tr>
            <td colspan="3">Row 3 Cell 1</td>
        </tr>
    </table>
</body>
</html>
```

This will produce the following result:



Tables Backgrounds

You can set table background using one of the following two ways:

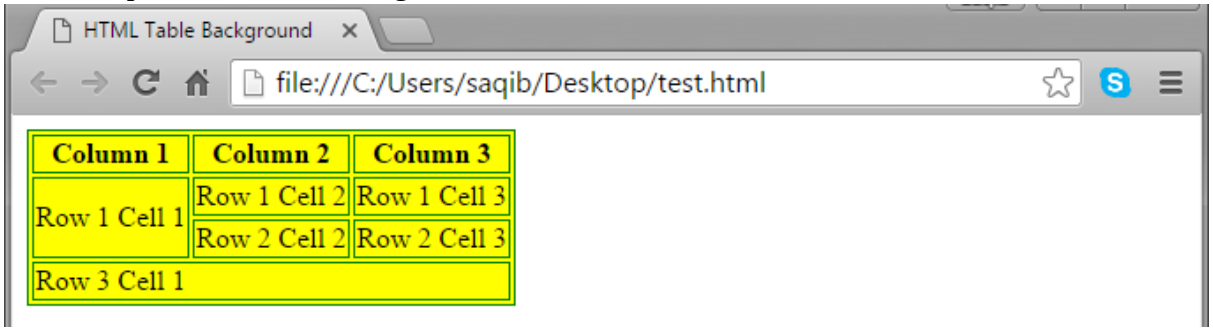
- bgcolor attribute - You can set background color for whole table or just for one cell.
- background attribute - You can set background image for whole table or just for one cell.

You can also set border color also using bordercolor attribute.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Table Background</title>
</head>
<body>
    <table border="1" bordercolor="green" bgcolor="yellow">
        <tr>
            <th>Column 1</th>
            <th>Column 2</th>
            <th>Column 3</th>
        </tr>
        <tr>
            <td rowspan="2">Row 1 Cell 1</td>
            <td>Row 1 Cell 2</td>
            <td>Row 1 Cell
3</td>
        </tr>
        <tr>
            <td>Row 2 Cell 2</td>
            <td>Row 2 Cell 3</td>
        </tr>
        <tr>
            <td colspan="3">Row 3 Cell 1</td>
        </tr>
    </table>
</body>
</html>
```

This will produce the following result:



Here is an example of using background attribute. Here we will use an image available in /images directory.

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Table Background</title>
</head>
<body>
    <table border="1" bordercolor="green" background="test.png">
        <tr>
            <th>Column 1</th>
            <th>Column 2</th>
            <th>Column 3</th>
        </tr>
        <tr>
            <td rowspan="2">Row 1 Cell 1</td>
            <td>Row 1 Cell 2</td>
            <td>Row 1 Cell 3</td>
        </tr>
        <tr>
            <td>Row 2 Cell 2</td>
            <td>Row 2 Cell 3</td>
        </tr>
        <tr>
            <td colspan="3">Row 3 Cell 1</td>
        </tr>
    </table>
</body>
</html>
```

This will produce the following result.

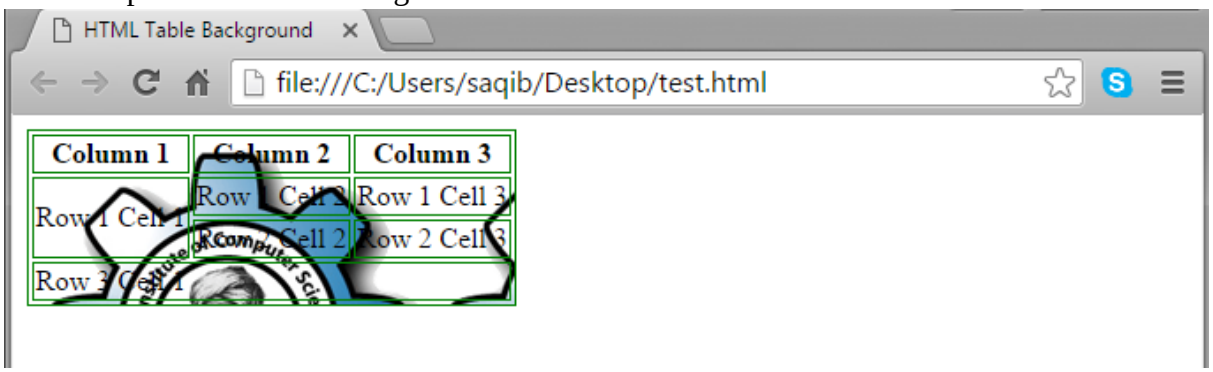


Table Height and Width

You can set a table width and height using width and height attributes. You can specify table width or height in terms of pixels or in terms of percentage of available screen area.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Table Background</title>
</head>
<body>
    <table border="1" width="500" height="150">
        <tr>
            <th>Column 1</th>
            <th>Column 2</th>
            <th>Column 3</th>
        </tr>
        <tr>
            <td rowspan="2">Row 1 Cell 1</td>
            <td>Row 1 Cell 2</td>
            <td>Row 1 Cell 3</td>
        </tr>
        <tr>
            <td>Row 2 Cell 2</td>
            <td>Row 2 Cell 3</td>
        </tr>
        <tr>
            <td colspan="3">Row 3 Cell 1</td>
        </tr>
    </table>
</body>
</html>
```

This will produce the following result:

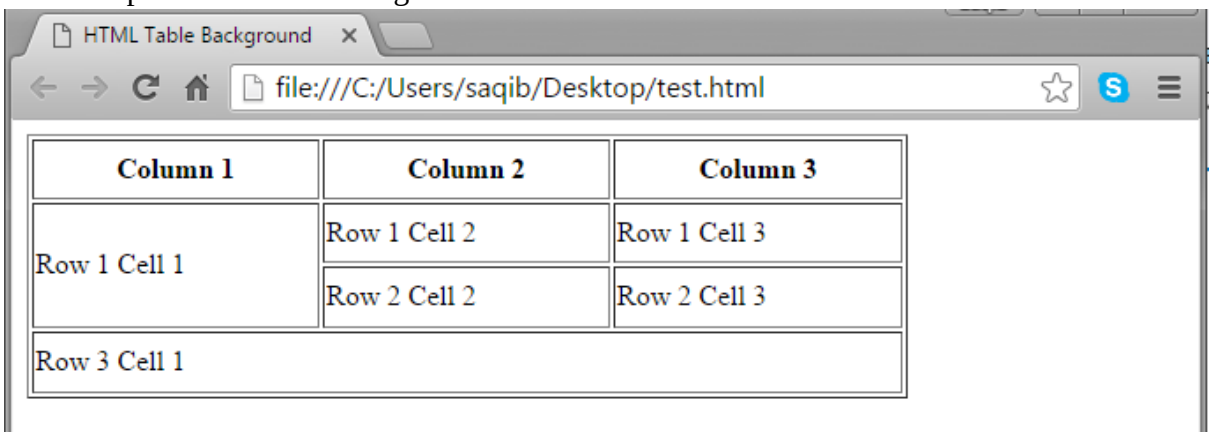


Table Caption

The caption tag will serve as a title or explanation for the table and it shows up at the top of the table. This tag is deprecated in newer version of HTML/XHTML.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Table Background</title>
```

```
</head>
<body>
    <table border="1" width="500" height="150">

        <caption>This is the caption</caption>
        <tr>
            <td rowspan="2">Row 1 Cell 1</td>
            <td>Row 1 Cell 2</td>
            <td>Row 1 Cell 3</td>
        </tr>

        <tr>
            <td colspan="3">Row 3 Cell 1</td>
        </tr>
    </table>
</body>
</html>
```

This will produce the following result:

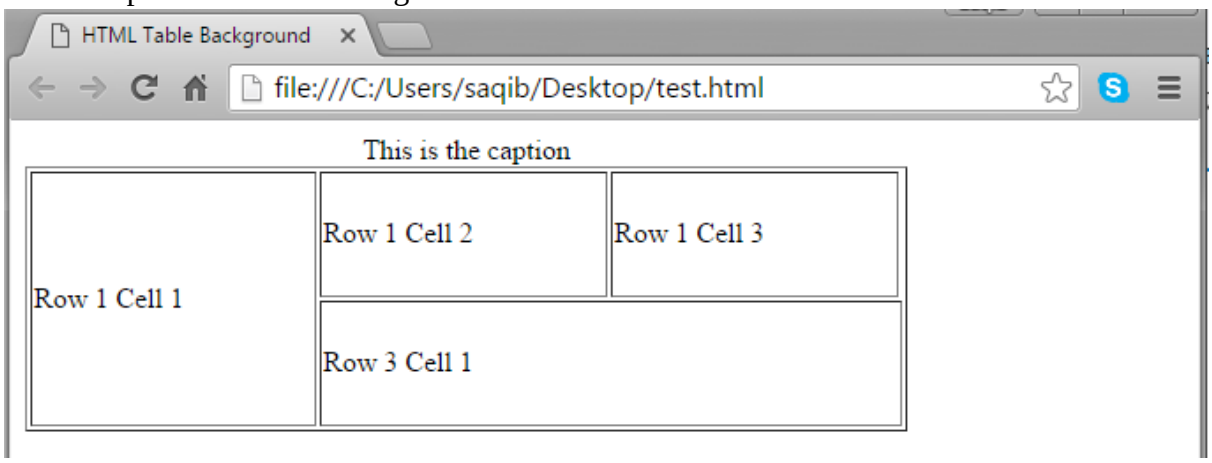


Table Header, Body, and Footer

Tables can be divided into three portions: a header, a body, and a foot. The head and foot are rather similar to headers and footers in a word-processed document that remain the same for every page, while the body is the main content holder of the table.

The three elements for separating the head, body, and foot of a table are:

- <thead> - to create a separate table header.
- <tbody> - to indicate the main body of the table.
- <tfoot> - to create a separate table footer.

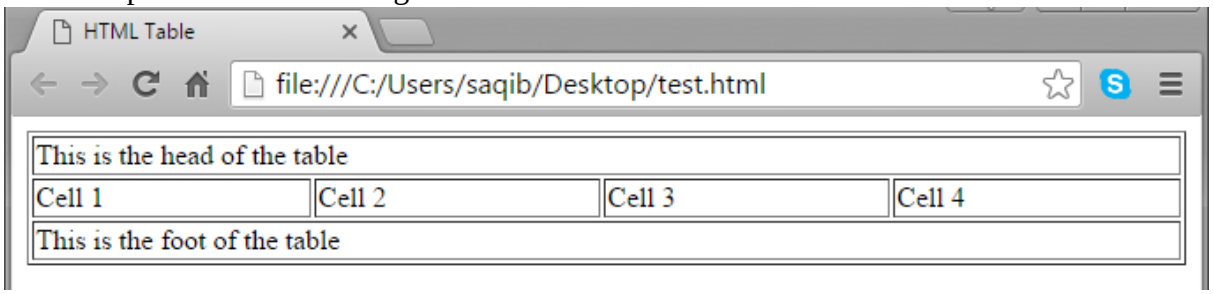
A table may contain several <tbody> elements to indicate different pages or groups of data. But it is notable that <thead> and <tfoot> tags should appear before <tbody>

Example

```
PE html>
<html>
<head>
    <title>HTML Table</title>
</head>
<body>
    <table border="1" width="100%">
        <thead>
            <tr>
```

```
                <td colspan="4">This is the head of the table</td>
            </tr>
        </thead>
        <tfoot>
            <tr>
                <td colspan="4">This is the foot of the table</td>
            </tr>
        </tfoot>
        <tbody>
            <tr>
                <td>Cell 1</td>
                <td>Cell 2</td>
                <td>Cell 3</td>
                <td>Cell 4</td>
            </tr>
        </tbody>
    </table>
</body>
</html>
```

This will produce the following result:



Nested Tables

You can use one table inside another table. Not only tables you can use almost all the tags inside table data tag <td>.

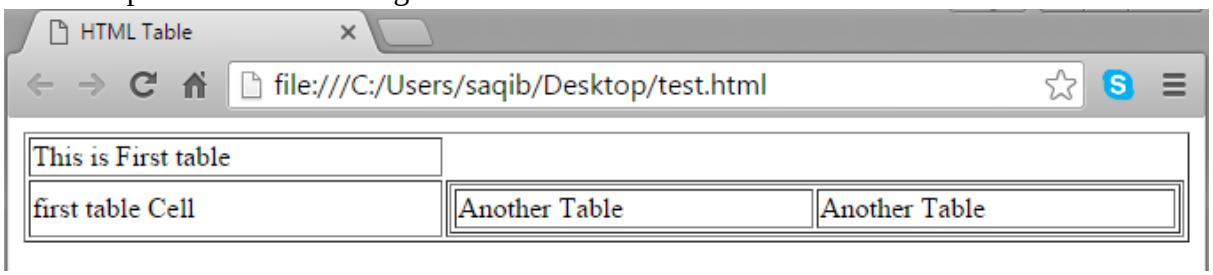
Example

Following is the example of using another table and other tags inside a table cell.

```
<!DOCTY<!DOCTYPE html>
<html>
<head>
    <title>HTML Table</title>
</head>
<body>
    <table border="1" width="100%">
        <tr>
            <td>This is First table</td>
        </tr>
        <tr>
            <td>first table Cell</td>
            <td>
                <table border="1" width="100%">
                    <tr>
                        <td>Another Table</td>
                        <td>Another Table</td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>
```

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

This will produce the following result:



11. HTML – LISTS

HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain:

- - An unordered list. This will list items using plain bullets.
- - An ordered list. This will use different schemes of numbers to list your items.
- <dl> - A definition list. This arranges your items in the same way as they are arranged in a dictionary.

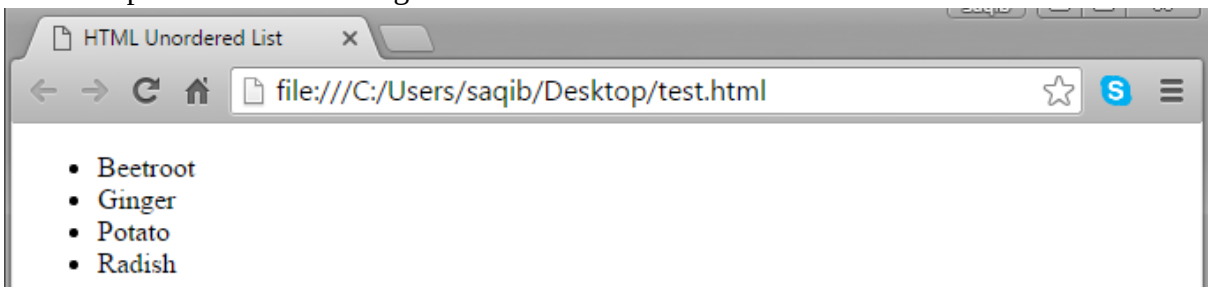
HTML Unordered Lists

An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML <ul> tag. Each item in the list is marked with a bullet.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Unordered List</title>
</head>
<body>
    <ul>
        <li>Beetroot</li>
        <li>Ginger</li>
        <li>Potato</li>
        <li>Radish</li>
    </ul>
</body>
</html>
```

This will produce the following result:



HTML Ordered Lists

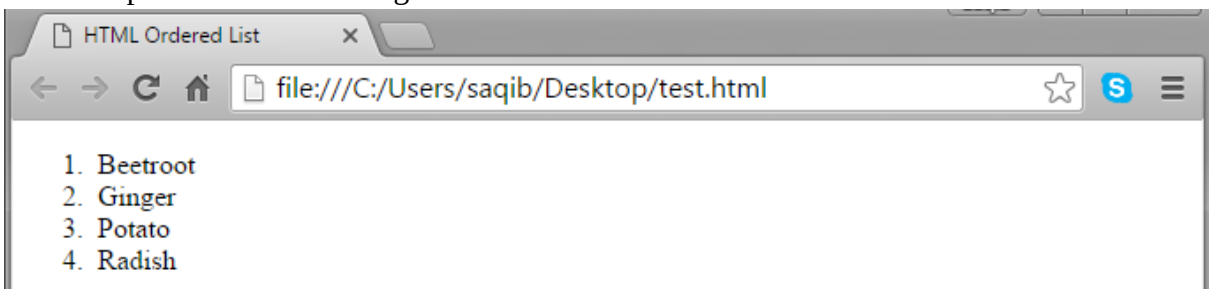
If you are required to put your items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using <ol> tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with <li>.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Ordered List</title>
</head>
<body>
    <ol>
        <li>Beetroot</li>
```

```
        <li>Ginger</li>
        <li>Potato</li>
        <li>Radish</li>
    </ol>
</body>
</html>
```

This will produce the following result:



The type Attribute

You can use type attribute for <ol> tag to specify the type of numbering you like. By default, it is a number. Following are the possible options:

<ol type="1"> - Default-Case Numerals.

<ol type="I"> - Upper-Case Numerals.

<ol type="i"> - Lower-Case Numerals.

<ol type="a"> - Lower-Case Letters.

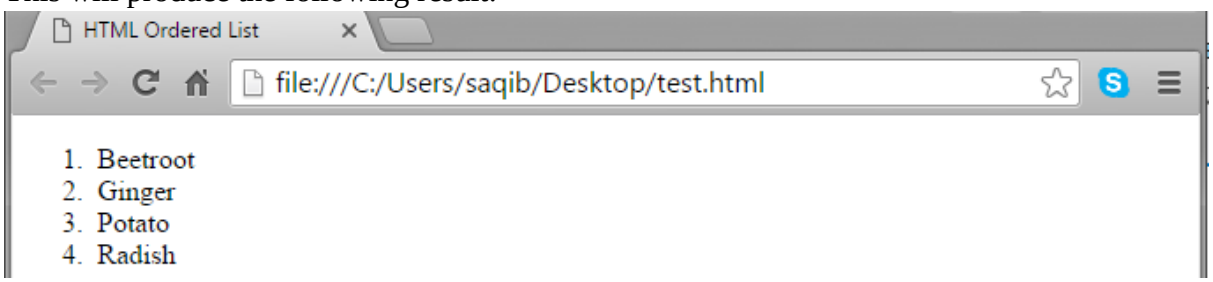
<ol type="A"> - Upper-Case Letters.

Example

Following is an example where we used <ol type="1">

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Ordered List</title>
</head>
<body>
    <ol type="1">
        <li>Beetroot</li>
        <li>Ginger</li>
        <li>Potato</li>
        <li>Radish</li>
    </ol>
</body>
</html>
```

This will produce the following result:

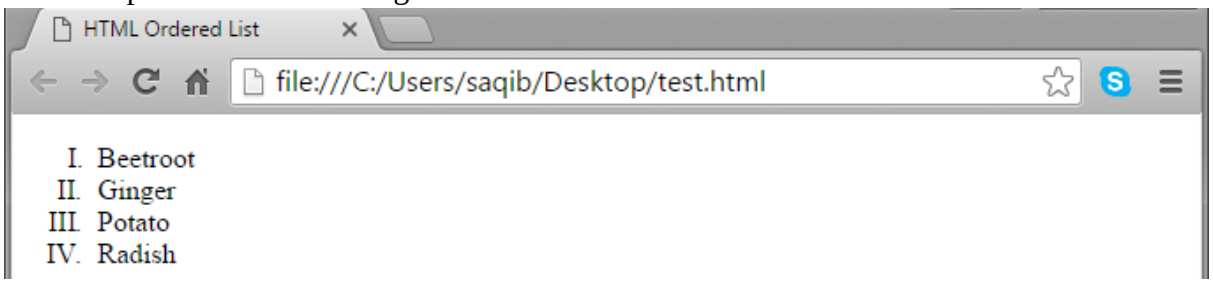


Example

Following is an example where we used <ol type="I">

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Ordered List</title>
</head>
<body>
    <ol type="I">
        <li>Beetroot</li>
        <li>Ginger</li>
        <li>Potato</li>
        <li>Radish</li>
    </ol>
</body>
</html>
```

This will produce the following result:

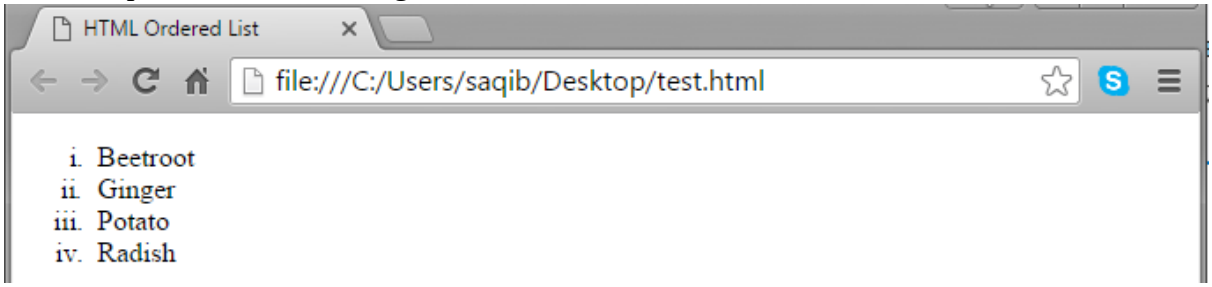


Example

Following is an example where we used <ol type="i">

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Ordered List</title>
</head>
<body>
    <ol type="i">
        <li>Beetroot</li>
        <li>Ginger</li>
        <li>Potato</li>
        <li>Radish</li>
    </ol>
</body>
</html>
```

This will produce the following result:

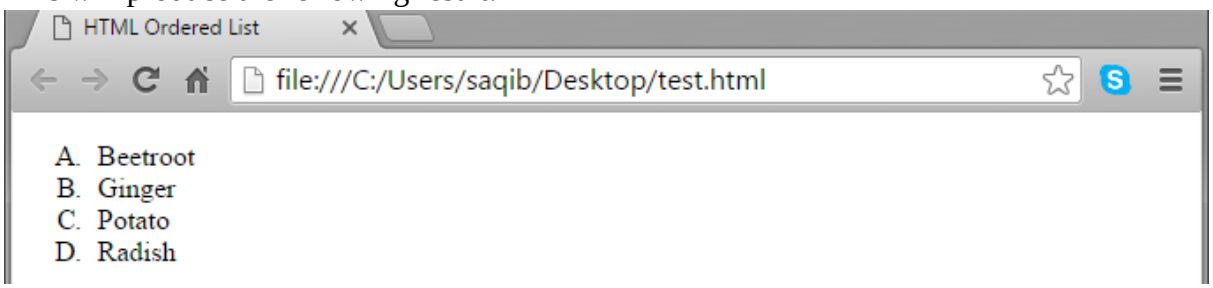


Example

Following is an example where we used <ol type="A">

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Ordered List</title>
</head>
<body>
    <ol type="A">
        <li>Beetroot</li>
        <li>Ginger</li>
        <li>Potato</li>
        <li>Radish</li>
    </ol>
</body>
</html>
```

This will produce the following result:

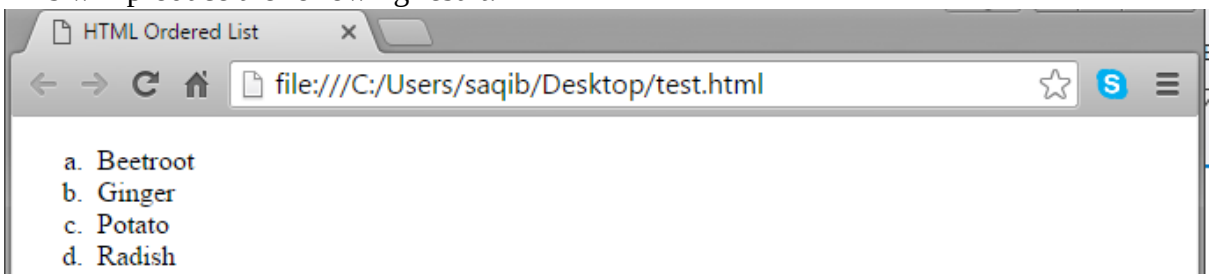


Example

Following is an example where we used <ol type="a">

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Ordered List</title>
</head>
<body>
    <ol type="a">
        <li>Beetroot</li>
        <li>Ginger</li>
        <li>Potato</li>
        <li>Radish</li>
    </ol>
</body>
</html>
```

This will produce the following result:



The start Attribute

You can use start attribute for <ol> tag to specify the starting point of numbering you need. Following are the possible options:

<ol type="1" start="4"> - Numerals starts with 4.

<ol type="I" start="4"> - Numerals starts with IV.

<ol type="i" start="4"> - Numerals starts with iv.

<ol type="a" start="4"> - Letters starts with d.

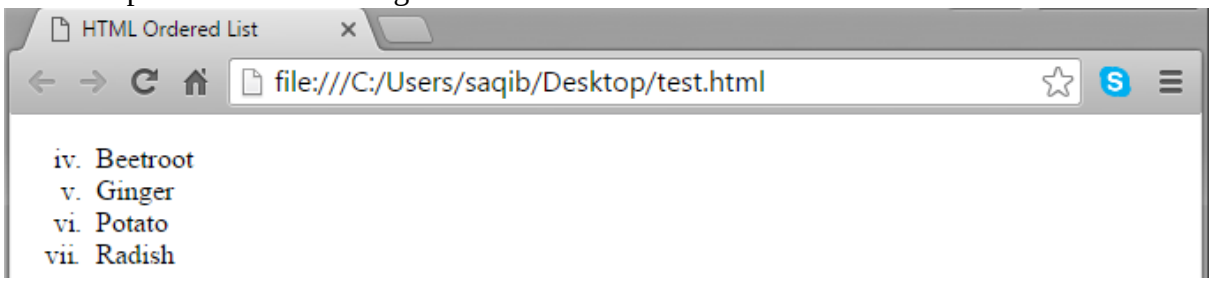
<ol type="A" start="4"> - Letters starts with D.

Example

Following is an example where we used <ol type="i" start="4" >

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Ordered List</title>
</head>
<body>
    <ol type="i" start="4">
        <li>Beetroot</li>
        <li>Ginger</li>
        <li>Potato</li>
        <li>Radish</li>
    </ol>
</body>
</html>
```

This will produce the following result:



HTML Definition Lists

HTML and XHTML supports a list style which is called definition lists where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.

Definition List makes use of following three tags.

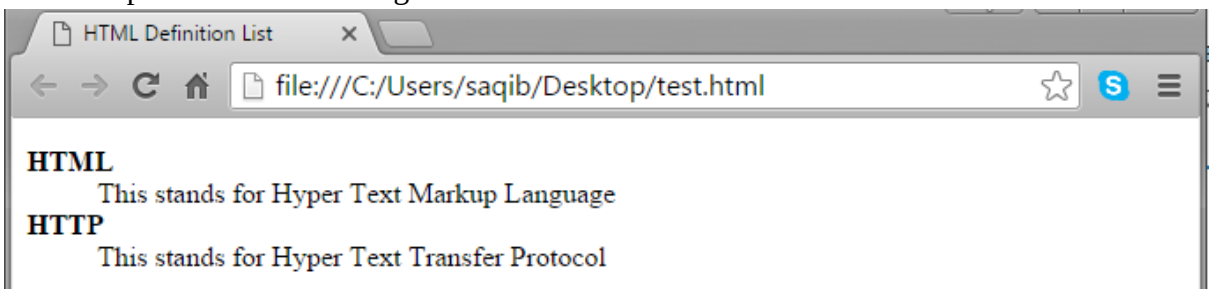
- <dl> - Defines the start of the list
- <dt> - A term
- <dd> - Term definition
- </dl> - Defines the end of the list

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Definition List</title>
</head>
<body>
    <dl>
        <dt><b>HTML</b></dt>
```

```
        <dd>This stands for Hyper Text Markup Language</dd>
        <dt><b>HTTP</b></dt>
        <dd>This stands for Hyper Text Transfer Protocol</dd>
    </dl>
</body>
</html>
```

This will produce the following result:



12. HTML – TEXT LINKS

A webpage can contain various links that take you directly to other pages and even specific parts of a given page. These links are known as hyperlinks.

Hyperlinks allow visitors to navigate between Web sites by clicking on words, phrases, and images. Thus you can create hyperlinks using text or images available on a webpage.

Note: I recommend you to go through a short tutorial on Understanding URL

Linking Documents

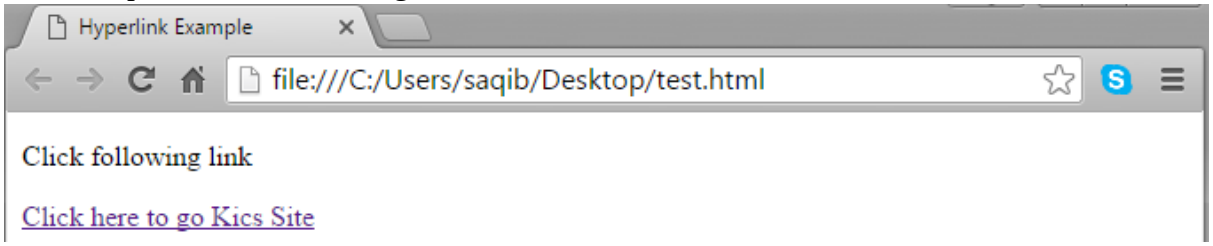
A link is specified using HTML tag <a>. This tag is called anchor tag and anything between the opening <a> tag and the closing tag becomes part of the link and a user can click that part to reach to the linked document. Following is the simple syntax to use <a> tag.

```
<a href="Document URL" ... attributes-list>Link Text</a>
```

Example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Hyperlink Example</title>
</head>
<body>
    <p>Click following link</p>
    <a href="http://www.kics.edu.pk" target="_self">Click here to go Kics Site</a>
</body>
</html>
```

This will produce the following result:



The target Attribute

We have used target attribute in our previous example. This attribute is used to specify the location where linked document is opened. Following are the possible options:

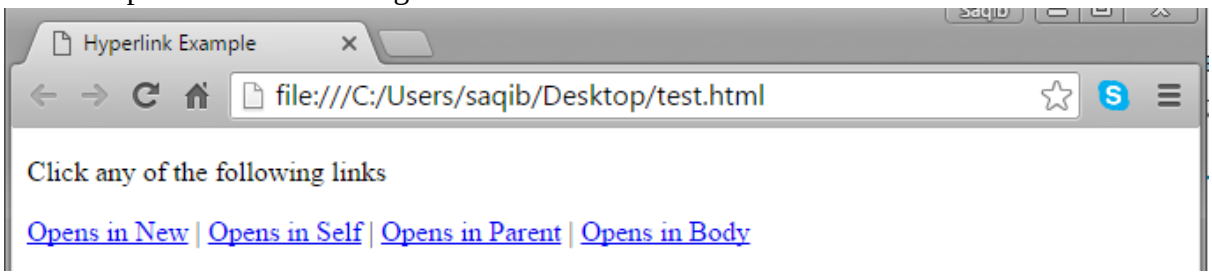
Option	Description
_blank	Opens the linked document in a new window or tab.
_self	Opens the linked document in the same frame.
_parent	Opens the linked document in the parent frame.
_top	Opens the linked document in the full body of the window.
targetframe	Opens the linked document in a named targetframe.

Example: Try following example to understand basic difference in few options given for target attribute.

```
<!DOCTYPE html>
<html>
<head>
    <title>Hyperlink Example</title>
```

```
</head>
<body>
    <p>Click any of the following links</p>
    <a href="/html/index.htm" target="_blank">Opens in New</a> |
    <a href="/html/index.htm" target="_self">Opens in Self</a> |
    <a href="/html/index.htm" target="_parent">Opens in Parent</a> |
    <a href="/html/index.htm" target="_top">Opens in Body</a>
</body>
</html>
```

This will produce the following result:



where you can click on different links to understand the difference between various options given for target attribute.

Use of Base Path

When you link HTML documents related to the same website, it is not required to give a complete URL for every link. You can get rid of it if you use <base> tag in your HTML

document header. This tag is used to give a base path for all the links. So your browser will concatenate given relative path to this base path and will make a complete URL.

Example: Following example makes use of <base> tag to specify base URL and later we can use relative path to all the links instead of giving complete URL for every link.

```
<!DOCTYPE html>
<html>
<head>
    <title>Hyperlink Example</title>
</head>
<body>
    <p>Click following link</p>
    <a href="/html/index.htm" target="_blank">HTML Tutorial</a>
</body>
</html>
```

Linking to a Page Section

You can create a link to a particular section of a given webpage by using name attribute. This is a two-step process.

First create a link to the place where you want to reach with-in a webpage and name it using <a...> tag as follows:

<h1>HTML Text Links </h1>

Second step is to create a hyperlink to link the document and place where you want to reach:

Go to the Top

13. HTML – IMAGE LINKS

We have seen how to create hypertext link using text and we also learnt how to use images in our webpages. Now, we will learn how to use images to create hyperlinks.

Example: It's simple to use an image as hyperlink. We just need to use an image inside hyperlink at the place of text as shown below:

```
<!DOCTYPE html>
<html>
<head>
    <title>Image Hyperlink Example</title>
</head>
<body>
    <p>Click following link</p>
    <a href="http://www.kics.edu.pk" target="_self">
        <img src="logo.png" alt="Open Image" border="0" />
    </a>
</body>
</html>
```

This will produce the following result,

This was the simplest way of creating hyperlinks using images. Next we will see how we can create Mouse-Sensitive Image Links.

14. HTML – EMAIL LINKS

It is not difficult to put an HTML email link on your webpage but it can cause unnecessary spamming problem for your email account. There are people, who can run programs to harvest these types of emails and later use them for spamming in various ways.

You can have another option to facilitate people to send you emails. One option could be to use HTML forms to collect user data and then use PHP or CGI script to send an email.

A simple example, check our Contact Us Form. We take user feedback using this form and then we are using one CGI program which is collecting this information and sending us email to the one given email ID.

Note: You will learn about HTML Forms in HTML Forms and you will learn about CGI in our another tutorial Perl CGI Programming.

HTML Email Tag

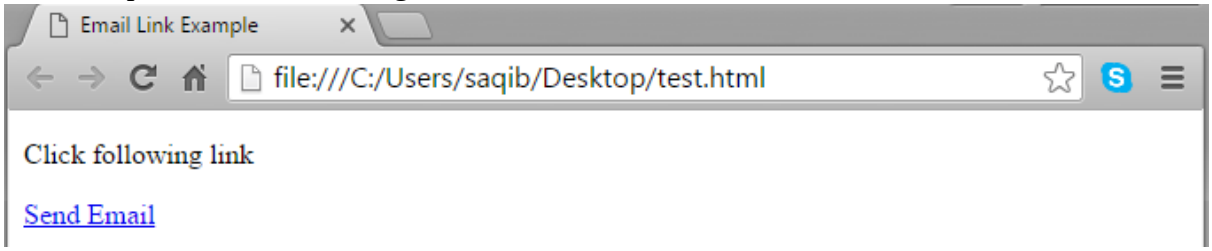
HTML <a> tag provides you option to specify an email address to send an email. While using <a> tag as an email tag, you will use mailto: email address along with href attribute. Following is the syntax of using mailto instead of using http.

Send Email

Example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Email Link Example</title>
</head>
<body>
    <p>Click following link</p>
    <a href="mailto: abc@example.com">Send Email</a>
</body>
</html>
```

This will produce the following result:



Now, if a user clicks this link, it launches one Email Client (like Lotus Notes, Outlook Express etc.) installed on your user's computer. There is another risk to use this option to send email because if user do not have email client installed on their computer then it would not be possible to send email.

Default Settings

You can specify a default email subject and email body along with your email address. Following is the example to use default subject and body.

 Send Feedback

Example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Email Link Example</title>
</head>
<body>
    <p>Click following link</p>
    <a href="mailto:abc@example.com?subject=Feedback&body=Message">Send Feedback
    </a>
</body>
</html>
```

This code will generate the following link which you can use to send email.

15. HTML – BLOCKS

All the HTML elements can be categorized into two categories (a) Block Level Elements (b)Inline Elements.

Block Elements

Block elements appear on the screen as if they have a line break before and after them. For example, the <p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <ul>, <ol>, <dl>,

<pre>, <hr />, <blockquote>, and <address> elements are all block level elements. They all start on their own new line, and anything that follows them appears on its own new line.

Inline Elements

Inline elements, on the other hand, can appear within sentences and do not have to appear on a new line of their own. The <b>, <i>, <u>, , , <sup>, <sub>, <big>,

<small>, <li>, <ins>, <del>, <code>, <cite>, <dfn>, <kbd>, and <var> elements are all inline elements.

Grouping HTML Elements

There are two important tags which we use very frequently to group various other HTML

tags (i) tag and (ii) tag

The tag

This is the very important block level tag which plays a big role in grouping various other HTML tags and applying CSS on group of elements. Even now tag can be used to create webpage layout where we define different parts (Left, Right, Top etc.) of the page using tag. This tag does not provide any visual change on the block but this has more meaning when it is used with CSS.

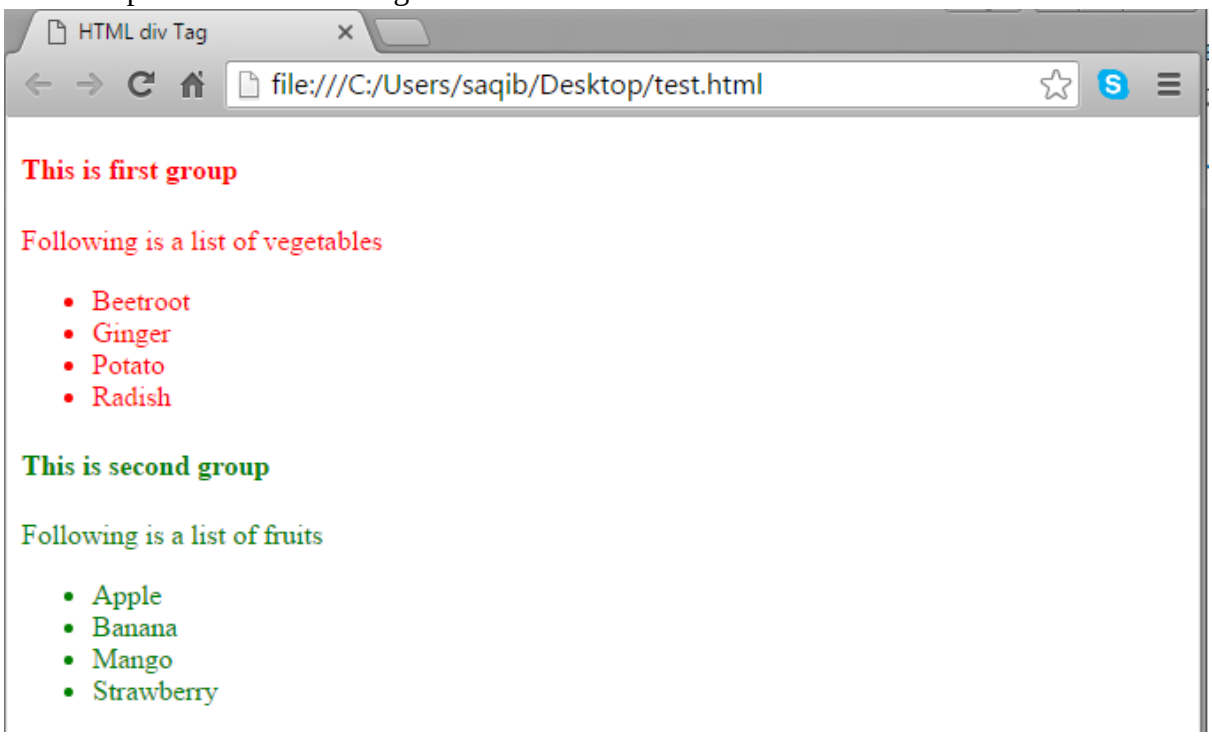
Example

Following is a simple example of tag. We will learn Cascading Style Sheet (CSS) in a separate chapter but we used it here to show the usage of tag:

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML div Tag</title>
</head>
<body>
    <!-- First group of tags -->
    <div style="color: red">
        <h4>This is first group</h4>
        <p>Following is a list of vegetables</p>
        <ul>
            <li>Beetroot</li>
            <li>Ginger</li>
            <li>Potato</li>
            <li>Radish</li>
        </ul>
    </div>
```

```
    <!-- Second group of tags -->
    <div style="color: green">
        <h4>This is second group</h4>
        <p>Following is a list of fruits</p>
        <ul>
            <li>Apple</li>
            <li>Banana</li>
            <li>Mango</li>
            <li>Strawberry</li>
        </ul>
    </div>
</body>
</html>
```

This will produce the following result:



The tag

The HTML is an inline element and it can be used to group inline-elements in an HTML document. This tag also does not provide any visual change on the block but has more meaning when it is used with CSS.

The difference between the tag and the tag is that the tag is used with inline elements whereas the tag is used with block-level elements.

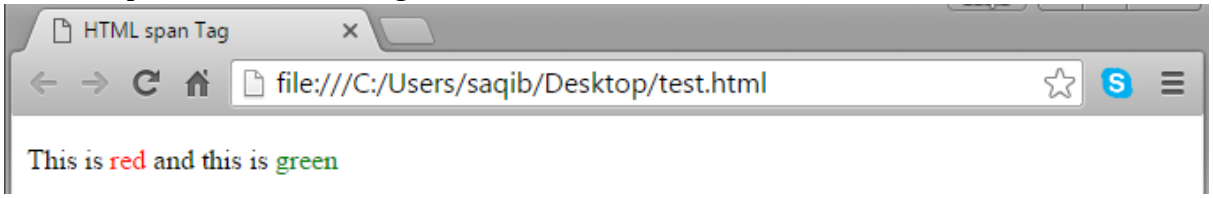
Example

Following is a simple example of tag. We will learn Cascading Style Sheet (CSS) in a separate chapter but we used it here to show the usage of tag:

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML span Tag</title>
</head>
<body>
```

```
    <p>This is <span style="color: red">red</span> and this is <span style="color:
green">green</span></p>
</body>
</html>
```

This will produce the following result:



16. HTML – BACKGROUNDS

By default, your webpage background is white in color. You may not like it, but no worries. HTML provides you following two good ways to decorate your webpage background.

- Html Background with Colors
- Html Background with Images

Now let's see both the approaches one by one using appropriate examples.

Html Background with Colors

The bgcolor attribute is used to control the background of an HTML element, specifically page body and table backgrounds. Following is the syntax to use bgcolor attribute with any HTML tag.

<tagname bgcolor="color_value"...>

This color_value can be given in any of the following formats:

<!-- Format 1 - Use color name -->

<table bgcolor="lime" >

<!-- Format 2 - Use hex value -->

<table bgcolor="#f1f1f1" >

<!-- Format 3 - Use color value in RGB terms -->

<table bgcolor="rgb(0,0,120)" >

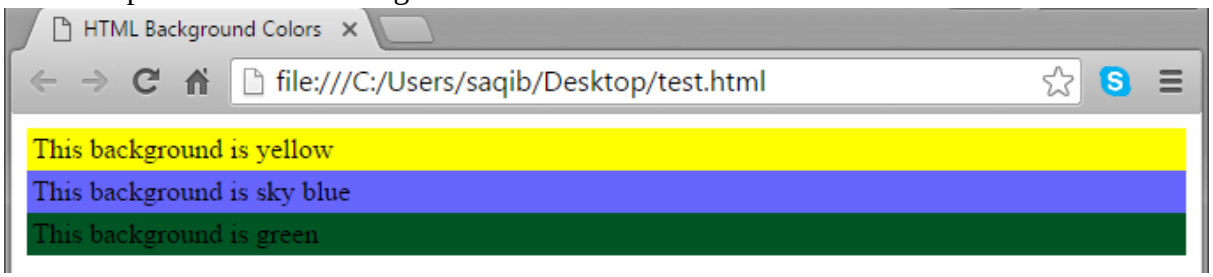
Example

Here are the examples to set background of an HTML tag:

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Background Colors</title>
</head>
<body>
    <!-- Format 1 - Use color name -->
    <table bgcolor="yellow" width="100%">
        <tr>
            <td>This background is yellow
            </td>
        </tr>
    </table>
    <!-- Format 2 - Use hex value -->
    <table bgcolor="#6666FF" width="100%">
        <tr>
            <td>This background is sky blue
            </td>
        </tr>
    </table>
    <!-- Format 3 - Use color value in RGB terms -->
    <table bgcolor="rgb(255,0,255)" width="100%">
```

```
        <tr>
            <td>This background is green
            </td>
        </tr>
    </table>
</body>
</html>
```

This will produce the following result:



Html Background with Images

The background attribute can also be used to control the background of an HTML element, specifically page body and table backgrounds. You can specify an image to set background of your HTML page or table. Following is the syntax to use background attribute with any HTML tag.

Note: The background attribute is deprecated and it is recommended to use Style Sheet for background setting.

<tagname background="Image URL"...>

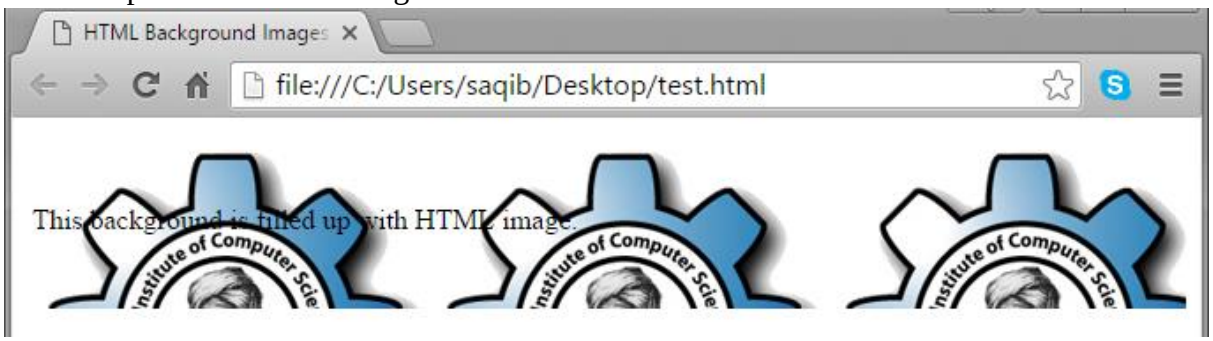
The most frequently used image formats are JPEG, GIF and PNG images.

Example

Here are the examples to set background images of a table.

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Background Images</title>
</head>
<body>
    <!-- Set table background -->
    <table background="/images/html.gif" width="100%" height="100">
        <tr>
            <td>This background is filled up with HTML image.
            </td>
        </tr>
    </table>
</body>
</html>
```

This will produce the following result:



17. HTML – COLORS

Colors are very important to give a good look and feel to your website. You can specify colors on page level using <body> tag or you can set colors for individual tags using bgcolor attribute.

The <body> tag has following attributes which can be used to set different colors:

- bgcolor - sets a color for the background of the page.
- text - sets a color for the body text.
- alink - sets a color for active links or selected links.
- link - sets a color for linked text.
- vlink - sets a color for visited links - that is, for linked text that you have already clicked on.

HTML Color Coding Methods

There are following three different methods to set colors in your web page:

- Color names - You can specify color names directly like green, blue or red.
- Hex codes - A six-digit code representing the amount of red, green, and blue that makes up the color.
- Color decimal or percentage values - This value is specified using the rgb() property.

Now we will see these coloring schemes one by one.

HTML Colors - Color Names

You can specify direct a color name to set text or background color. W3C has listed 16 basic color names that will validate with an HTML validator but there are over 200 different color names supported by major browsers.

Note: Check a complete list of HTML Color Name.

W3C Standard 16 Colors

Here is the list of W3C Standard 16 Colors names and it is recommended to use them.

	Black		Gray		Silver		White
	Yellow		Lime		Aqua		Fuchsia
	Red		Green		Blue		Purple
	Maroon		Olive		Navy		Teal

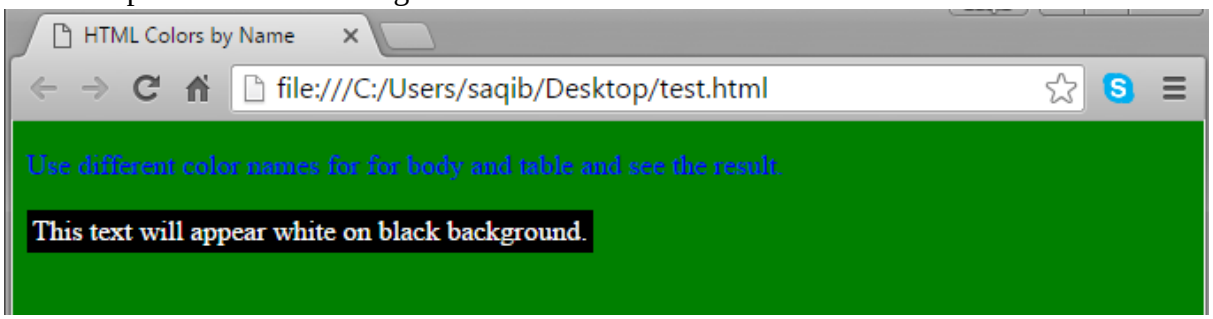
Example

Here are the examples to set background of an HTML tag by color name:

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Colors by Name</title>
```

```
</head>
<body text="blue" bgcolor="green">
    <p>Use different color names for for body and table and see the result.</p>
    <table bgcolor="black">
        <tr>
            <td>
                <font color="white">This text will appear white on black
background.</font>
            </td>
        </tr>
    </table>
</body>
</html>
```

This will produce the following result:



HTML Colors - Hex Codes

A hexadecimal is a 6 digit representation of a color. The first two digits(RR) represent a red value, the next two are a green value(GG), and the last are the blue value(BB).

A hexadecimal value can be taken from any graphics software like Adobe Photoshop, Paintshop Pro or MS Paint.

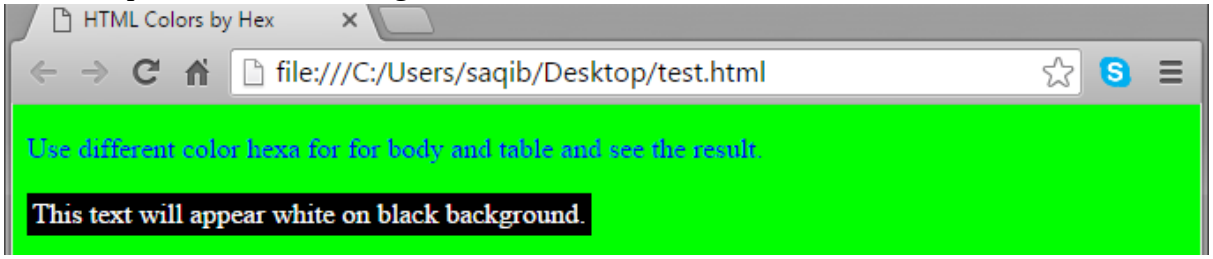
Each hexadecimal code will be preceded by a pound or hash sign #. Following is a list of few colors using hexadecimal notation.

Example

Here are the examples to set background of an HTML tag by color code in hexadecimal:

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Colors by Hex</title>
</head>
<body text="#0000FF" bgcolor="#00FF00">
    <p>Use different color hexa for for body and table and see the result.</p>
    <table bgcolor="#000000">
        <tr>
            <td>
                <font color="#FFFFFF">This text will appear white on black
background.</font>
            </td>
        </tr>
    </table>
</body>
</html>
```

This will produce the following result:



HTML Colors - RGB Values

This color value is specified using the rgb( ) property. This property takes three values, one each for red, green, and blue. The value can be an integer between 0 and 255 or a percentage.

Note: All the browsers does not support rgb() property of color so it is recommended not to use it.

Following is a list to show few colors using RGB values.

Color	Color RGB
	rgb(0,0,0)
	rgb(255,0,0)
	rgb(0,255,0)
	rgb(0,0,255)
	rgb(255,255,0)
	rgb(0,255,255)
	rgb(255,0,255)
	rgb(192,192,192)
	rgb(255,255,255)

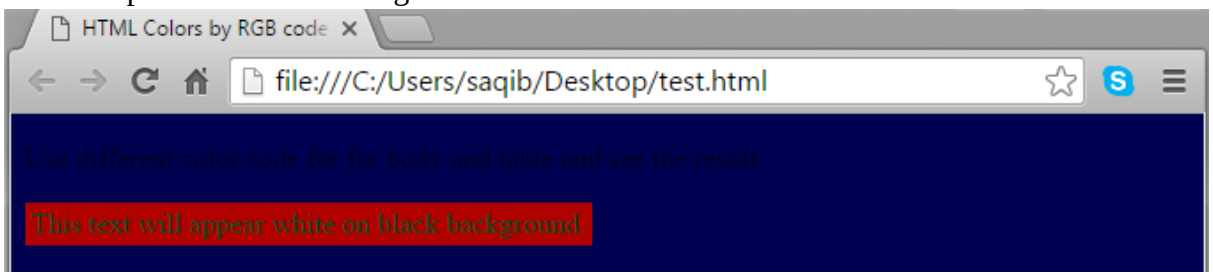
Example

Here are the examples to set background of an HTML tag by color code using rgb() values:

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Colors by RGB code</title>
</head>
<body text="rgb(0,0,255)" bgcolor="rgb(0,255,0)">
    <p>Use different color code for for body and table and see the result.</p>
    <table bgcolor="rgb(0,0,0)">
        <tr>
            <td>
```

```
                <font color="rgb(255,255,255)">This text will appear white on black
background.</font>
            </td>
        </tr>
    </table>
</body>
</html>
```

This will produce the following result:



18. HTML – FORMS

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML <form> tag is used to create an HTML form and it has following syntax:

```
<form action="Script URL" method="GET|POST">
form elements like input, textarea etc.
</form>
```

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes:

Attribute	Description
action	Backend script ready to process your passed data.
method	Method to be used to upload data. The most frequently used are GET and POST methods.
target	Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
enctype	You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are:
application/x-www-form-urlencoded - This is the standard method most forms use in simple scenarios.	mutlipart/form-data - This is used when you want to upload binary data in the form of files like image, word file etc.

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form:

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

Text Input Controls

There are three types of text input used on forms:

- Single-line text input controls - This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.
- Password input controls - This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.
- Multi-line text input controls - This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Single-line text input controls

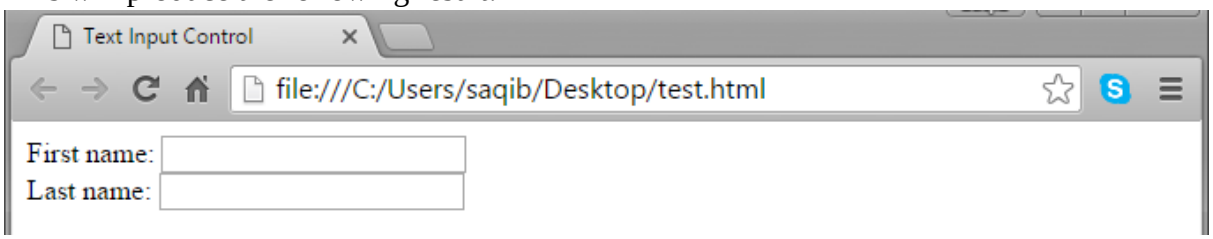
This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

Example

Here is a basic example of a single-line text input used to take first name and last name:

```
<!DOCTYPE html>
<html>
<head>
    <title>Text Input Control</title>
</head>
<body>
    <form>
        First name:
        <input type="text" name="first_name" />
        <br>
        Last name:
        <input type="text" name="last_name" />
    </form>
</body>
</html>
```

This will produce the following result:



Attributes

Following is the list of attributes for <input> tag for creating text field.

Attribute	Description
type	Indicates the type of input control and for text input control it will be set totext.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	This can be used to provide an initial value inside the control.
size	Allows to specify the width of the text-input control in terms of characters.

maxlength	Allows to specify the maximum number of characters a user can enter into the text box.

Password Input controls

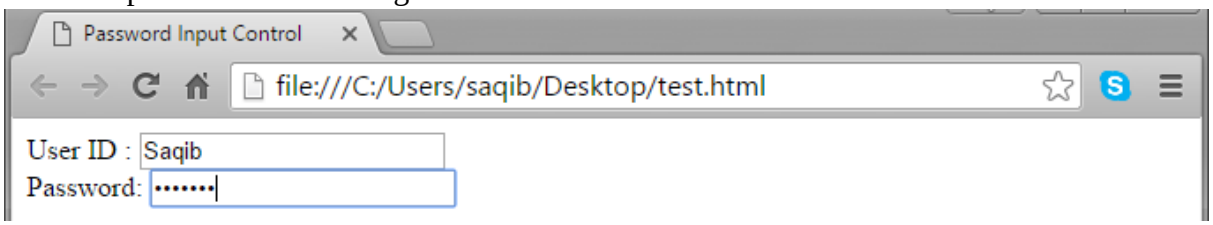
This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input> tag but type attribute is set to password.

Example

Here is a basic example of a single-line password input used to take user password:

```
<!DOCTYPE html>
<html>
<head>
    <title>Password Input Control</title>
</head>
<body>
    <form>
        User ID :
        <input type="text" name="user_id" />
        <br/>
        Password:
        <input type="password" name="password" />
    </form>
</body>
</html>
```

This will produce the following result:



Multiple-Line Text Input Controls

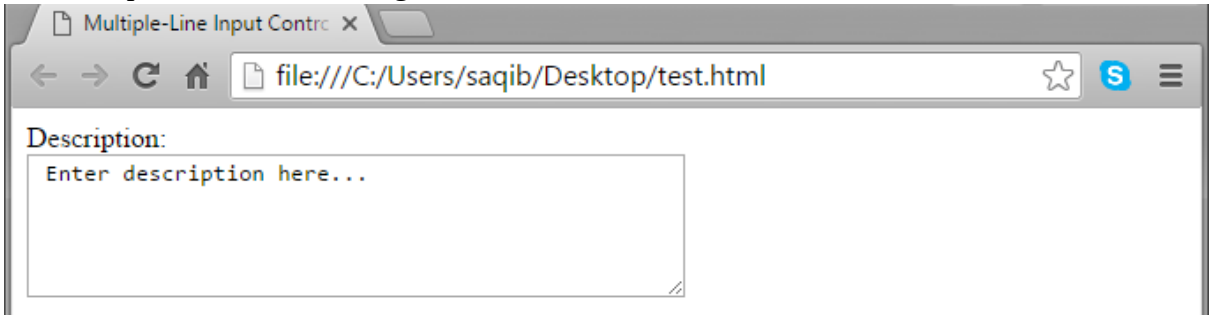
This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Example

Here is a basic example of a multi-line text input used to take item description:

```
<!DOCTYPE html>
<html>
<head>
    <title>Multiple-Line Input Control</title>
</head>
<body>
    <form>
        Description:
        <br />
        <textarea rows="5" cols="50" name="description"> Enter description here...
</textarea>
    </form>
</body>
</html>
```

This will produce the following result:



Attributes

Following is the list of attributes for <textarea> tag.

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
rows	Indicates the number of rows of text area box.
cols	Indicates the number of columns of text area box

Checkbox Control

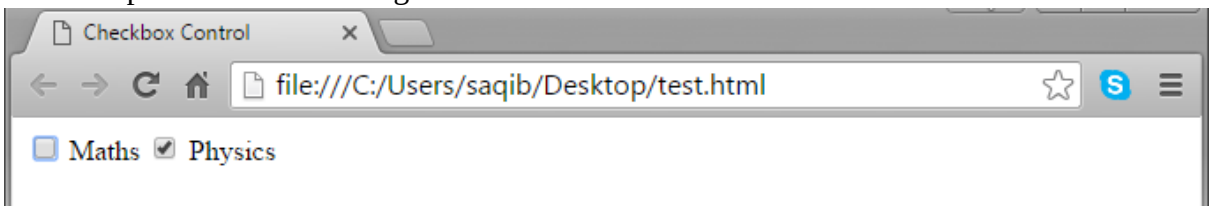
Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to checkbox.

Example

Here is an example HTML code for a form with two checkboxes:

```
<!DOCTYPE html>
<html>
<head>
    <title>Checkbox Control</title>
</head>
<body>
    <form>
        <input type="checkbox" name="maths" value="on">
        Maths
        <input type="checkbox" name="physics" value="on">
        Physics
    </form>
</body>
</html>
```

This will produce the following result:



Attributes	Following is the list of attributes for <checkbox> tag.
Attribute	Description

type	Indicates the type of input control and for checkbox input control it will be set to checkbox.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	The value that will be used if the checkbox is selected.
checked	Set to checked if you want to select it by default.

Radio Button Control

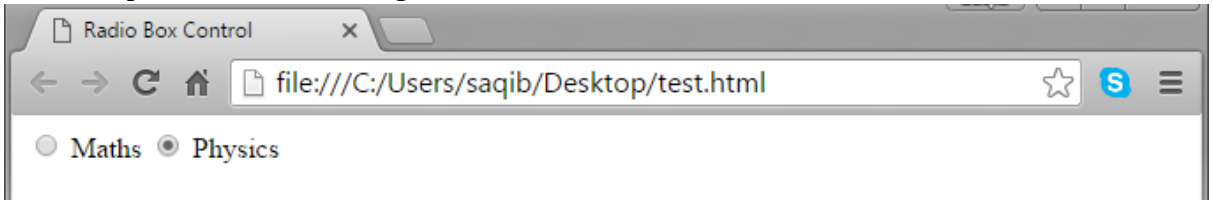
Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to radio.

Example

Here is example HTML code for a form with two radio buttons:

```
<!DOCTYPE html>
<html>
<head>
    <title>Radio Box Control</title>
</head>
<body>
    <form>
        <input type="radio" name="subject" value="maths">
        Maths
        <input type="radio" name="subject" value="physics">
        Physics
    </form>
</body>
</html>
```

This will produce the following result:



Attributes

Following is the list of attributes for radio button.

Attribute	Description
type	Indicates the type of input control and for checkbox input control it will be set to radio.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	The value that will be used if the radio box is selected.
checked	Set to checked if you want to select it by default.

Select Box Control

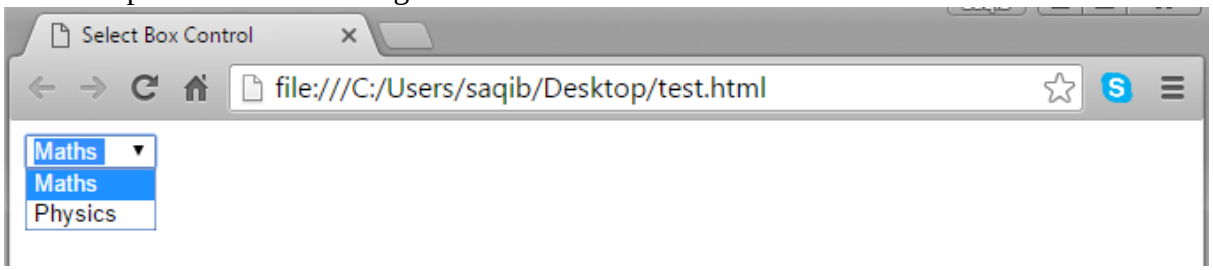
A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

Example

Here is example HTML code for a form with one drop down box

```
<!DOCTYPE html>
<html>
<head>
    <title>Select Box Control</title>
</head>
<body>
    <form>
        <select name="dropdown">
            <option value="Maths" selected="selected">Maths</option>
            <option value="Physics">Physics</option>
        </select>
    </form>
</body>
</html>
```

This will produce the following result:



Attributes

Following is the list of important attributes of <select> tag:

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
size	This can be used to present a scrolling list box.
multiple	If set to "multiple" then allows a user to select multiple items from the menu.

Following is the list of important attributes of <option> tag:

Attribute	Description
value	The value that will be used if an option in the select box box is selected.
selected	Specifies that this option should be the initially selected value when the page loads.
label	An alternative way of labeling options

File Upload Box

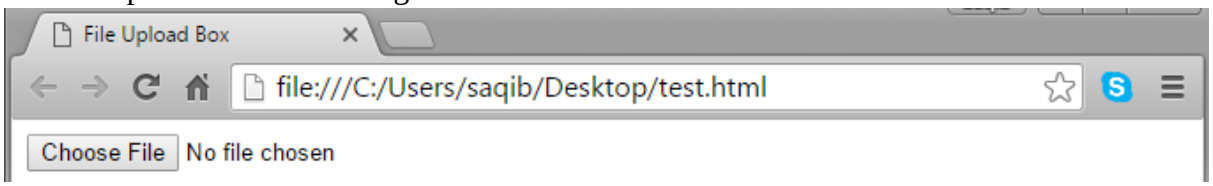
If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to file.

Example

Here is example HTML code for a form with one file upload box:

```
<!DOCTYPE html>
<html>
<head>
    <title>File Upload Box</title>
</head>
<body>
    <form>
        <input type="file" name="fileupload" accept="image/*" />
    </form>
</body>
</html>
```

This will produce the following result:



Attributes

Following is the list of important attributes of file upload box:

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
accept	Specifies the types of files that the server accepts.

Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input> tag by setting its type attribute to button. The type attribute can take the following values:

Type	Description
submit	This creates a button that automatically submits a form.
reset	This creates a button that automatically resets form controls to their initial values.
button	This creates a button that is used to trigger a client-side script when the user clicks that button.
image	This creates a clickable button but we can use an image as background of the button.

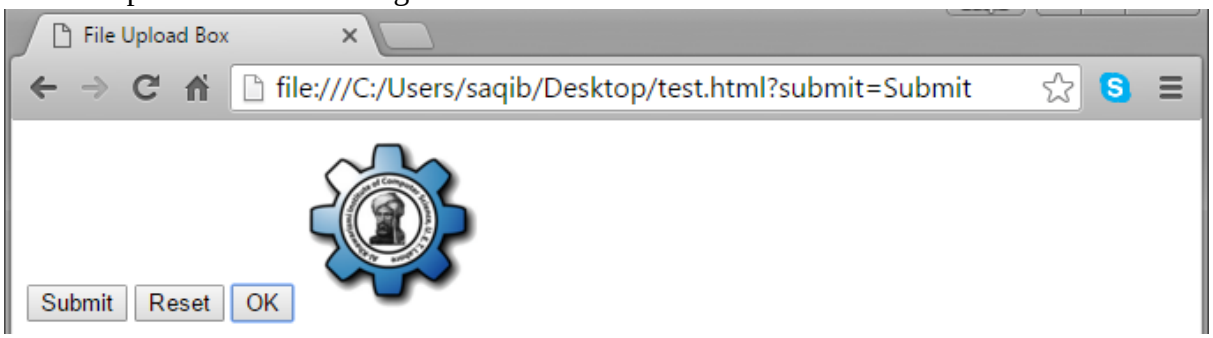
Example

Here is example HTML code for a form with three types of buttons:

```
<!DOCTYPE html>
<html>
<head>
```

```
    <title>File Upload Box</title>
</head>
<body>
    <form>
        <input type="submit" name="submit" value="Submit" />
        <input type="reset" name="reset" value="Reset" />
        <input type="button" name="ok" value="OK" />
        <input type="image" name="imagebutton" src="logo.png" width="100"
height="100"/>
    </form>
</body>
</html>
```

This will produce the following result:



Hidden Form Controls

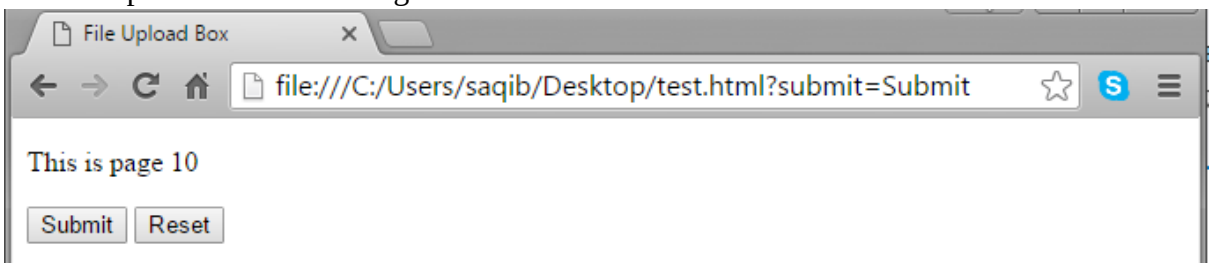
Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page will be displayed next based on the passed current page.

Example

Here is example HTML code to show the usage of hidden control:

```
<!DOCTYPE html>
<html>
<head>
    <title>File Upload Box</title>
</head>
<body>
    <form>
        <p>This is page 10</p>
        <input type="hidden" name="pagename" value="10" />
        <input type="submit" name="submit" value="Submit" />
        <input type="reset" name="reset" value="Reset" />
    </form>
</body>
</html>
```

This will produce the following result:



19. HTML – MULTIMEDIA

Multimedia comes in many different formats. It can be almost anything you can hear or see.

Examples: Pictures, music, sound, videos, records, films, animations, and more.

Web pages often contain multimedia elements of different types and formats.

In this chapter you will learn about the different multimedia formats.

Multimedia Formats

Multimedia elements (like sounds or videos) are stored in media files.

The most common way to discover the type of a file, is to look at the file extension. When a browser sees the file extension .htm or .html, it will treat the file as an HTML file. The .xml extension indicates an XML file, and the .css extension indicates a style sheet file. Pictures are recognized by extensions like .gif, .png and .jpg.

Multimedia files also have their own formats and different extensions like: .swf, .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

Format	File	Description
MPEG	.mpg .mpeg	MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Used to be supported by all browsers, but it is not supported in HTML5 (See MP4).
AVI	.avi	AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
WMV	.wmv	WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
QuickTime	.mov	QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers. (See MP4)

RealVideo	.rm .ram	RealVideo. Developed by Real Media to allow video streaming with low bandwidths. It is still used for online video and Internet TV, but does not play in web browsers.
Flash	.swf .flv	Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers.
Ogg	.ogg	Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5.
WebM	.webm	WebM. Developed by the web giants, Mozilla, Opera, Adobe, and Google. Supported by HTML5.
MPEG-4 or MP4	.mp4	MP4. Developed by the Moving Pictures Expert Group. Based on QuickTime. Commonly used in newer video cameras and TV hardware. Supported by all HTML5 browsers. Recommended by YouTube.

Sound Formats

MP3 is the newest format for compressed recorded music. The term MP3 has become synonymous with digital music.

If your website is about recorded music, MP3 is the choice.

Format	File	Description
MIDI	.mid .midi	MIDI (Musical Instrument Digital Interface). Main format for all electronic music devices like synthesizers and PC sound cards. MIDI files do not contain sound, but digital notes that can be played by electronics. Plays well on all computers and music hardware, but not in web browsers.
RealAudio	.rm .ram	RealAudio. Developed by Real Media to allow streaming of audio with low bandwidths. Does not play in web browsers.
WMA	.wma	WMA (Windows Media Audio). Developed by Microsoft. Commonly used in music players. Plays well on Windows computers, but not in web browsers.
AAC	.aac	AAC (Advanced Audio Coding). Developed by Apple as the default format for iTunes. Plays well on Apple computers, but not in web browsers.
WAV	.wav	WAV. Developed by IBM and Microsoft. Plays well on Windows, Macintosh, and Linux operating systems. Supported by HTML5.

Ogg	.ogg	Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5.
MP3	.mp3	MP3 files are actually the sound part of MPEG files. MP3 is the most popular format for music players. Combines good compression (small files) with high quality. Supported by all browsers.
MP4	.mp4	MP4 is a video format, but can also be used for audio. MP4 video is the upcoming video format on the internet. This leads to automatic support for MP4 audio by all browsers.

Audio on the Web

- Before HTML5, there was no standard for playing audio files on a web page.
- Before HTML5, audio files could only be played with a plug-in (like flash).
- The HTML5 <audio> element specifies a standard way to embed audio in a web page.
- Example:

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
</audio>
```

Playing Videos in HTML

- Before HTML5, there was no standard for showing videos on a web page.
- Before HTML5, videos could only be played with a plug-in (like flash).
- The HTML5 <video> element specifies a standard way to embed a video in a web page.

Example:

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
</video>
```

20. URL - Uniform Resource

- A URL is another word for a web address.
- A URL can be composed of words (kics.com), or an Internet Protocol (IP) address (192.68.20.45).
- Most people enter the name when surfing, because names are easier to remember than numbers.

Syntax Rules:

scheme://host.domain:port/path/filename

Example:

https://www.kics.com/itdepartment/cs.html

21. HTML – TAGS

Following tags have been introduced in HTML

Tag	Description
<!--...-->	Specifies a comment
<!DOCTYPE>	Specifies the document type
<a>	Specifies an anchor
<abbr>	Specifies an abbreviation
<acronym>	Specifies an acronym
<address>	Specifies an address element
<applet>	Deprecated. Specifies an applet
<area>	Specifies an area inside an image map
<article>	Specifies an article
<aside>	Specifies some content loosely related to the page content. If it is removed, the remaining content still makes sense
<audio>	Specifies a sound content
<b>	Specifies bold text
<base>	Specifies a base URL for all the links in a page
<basefont>	Deprecated. Specifies a base font
<bdo>	Specifies the direction of text display
<bdi>	Represents text that must be isolated from its surrounding for bidirectional text formatting. It allows embedding a span of text with a different, or unknown, directionality
<bgsound>	Specifies background music
<big>	Specifies big text
<blink>	Specifies a text which blinks
<blockquote>	Specifies a long quotation
<body>	Specifies the body element
** **	Inserts a single line break
<button>	Specifies a push button
<canvas>	For making graphics with a script
<caption>	Specifies a table caption
** **	Deprecated. Specifies centered text
<cite>	Specifies a citation
<code>	Specifies computer code text
<col>	Specifies attributes for table columns
<colgroup>	Specifies groups of table columns
<comment>	Puts a comment in the document
<datalist>	A list of options for input values
<dd>	Specifies a definition description
<del>	Specifies deleted text
<dfn>	Specifies a definition term
<dialog>	Specifies a dialog box or window
<dir>	Deprecated. Specifies a directory list
** **	Specifies a section in a document
<dl>	Specifies a definition list
<dt>	Specifies a definition term
****	Specifies emphasized text

<embed>	Specifies a container for an external (non- HTML) application
<fieldset>	Specifies a fieldset
<figcaption>	Specifies a caption for a <figure> element
<figure>	Specifies self-contained content
<font>	Deprecated. Specifies text font, size, and color
<footer>	Specifies a footer for a document or section
<form>	Specifies a form
<frame>	Specifies a sub window (a frame)
<frameset>	Specifies a set of frames
<h1> to <h6>	Specifies header 1 to header 6
<head>	Specifies information about the document
<header>	Specifies a header for a document or section
<hr>	Specifies a horizontal rule
<html>	Specifies an html document
<i>	Specifies italic text
<iframe>	Specifies an inline sub window (frame)
<ilayer>	Specifies an inline layer
<img>	Specifies an image
<input>	Specifies an input field
<ins>	Specifies inserted text
<isindex>	Deprecated. Specifies a single-line input field
<kbd>	Specifies keyboard text
<keygen>	Generate key information in a form
<label>	Specifies a label for a form control
<layer>	Specifies a layer
<legend>	Specifies a title in a fieldset
** **	Specifies a list item
<link>	Specifies a resource reference
<main>	Specifies the main or important content in the document. There is only oneelement in the document
<map>	Specifies an image map
<mark>	Specifies a text highlighted for reference purposes, that is for its relevance in another context
<marquee>	Creates a scrolling-text marquee
<menu>	Deprecated. Specifies a menu list
<menuitem>	Specifies a command/menu item that the user can invoke from a popup menu
<meta>	Specifies meta data of an html document which is not displayed on the page
<meter>	Specifies a scalar measurement within a known range (a gauge)
<multicol>	Specifies a multicolumn text flow
<nav>	Specifies a section that contains only navigation links
<nobr>	No breaks allowed in the enclosed text
<noembed>	Specifies content to be presented by browsers that do not support the
<embed>	tag
<noframes>	Specifies a noframe section

<noscript>	Specifies a noscript section
<object>	Specifies an embedded object
** **	Specifies an ordered list
<optgroup>	Specifies an option group
<option>	Specifies an option in a drop-down list
<output>	Specifies the result of a calculation
<p>	Specifies a paragraph
<param>	Specifies a parameter for an object
<plaintext>	Deprecated. Render the remainder of the document as preformatted plain text
<pre>	Specifies preformatted text
<progress>	Specifies a completion progress of a task
<q>	Specifies a short quotation
<rp>	Specifies to show browsers that do not support the ruby element
<rt>	Specifies an text ruby annotation
<ruby>	Specifies an ruby annotation
<s>	Deprecated. Specifies strikethrough text
<samp>	Specifies sample computer code
<script>	Specifies a script
<section>	Specifies a section in a document
<select>	Specifies a selectable list
<spacer>	Specifies a white space
<small>	Specifies small text
<source>	Specifies a media resources for media elements, defined inside video or audio elements
****	Specifies a section in a document
<strike>	Deprecated. Specifies strikethrough text
****	Specifies strong text
<style>	Specifies a style definition
<sub>	Specifies subscripted text
<summary>	Specifies a summary, caption, or legend for a given <details>
<sup>	Specifies superscripted text
<table>	Specifies a table
<tbody>	Specifies a table body
<td>	Specifies a table cell
<textarea>	Specifies a text area
<tfoot>	Specifies a table footer
<th>	Specifies a table heading
<thead>	Specifies a table header
<time>	Specifies a date and time <details>
<title>	Specifies the document title
<tr>	Specifies a table row
<track>	Specifies a text tracks used in mediaplayers
<tt>	Specifies teletype text
<u>	Deprecated. Specifies underlined text
** **	Specifies an unordered list
<var>	Specifies a variable

<video>	Specifies a text tracks used in media players
<wbr>	Indicates a potential word break point within a <nobr> section
<xmp>	Deprecated. Specifies preformatted text