

# HTML Overview and Tags

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.

## Why to Learn HTML?

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.

**HTML** is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning HTML:

- **Create Web site** - You can create a website or customize an existing web template if you know HTML well.
- **Become a web designer** - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.
- **Understand web** - If you want to optimize your website, to boost its speed and performance, it is good to know HTML to yield best results.
- **Learn other languages** - Once you understand the basic of HTML then other related technologies like javascript, php, or angular are become easier to understand.

## Applications of HTML

As mentioned before, HTML is one of the most widely used language over the web. I'm going to list few of them here:

- **Web pages development** - HTML is used to create pages which are rendered over the web. Almost every page of web is having html tags in it to render its details in browser.
- **Internet Navigation** - HTML provides tags which are used to navigate from one page to another and is heavily used in internet navigation.
- **Responsive UI** - HTML pages now-a-days works well on all platform, mobile, tabs, desktop or laptops owing to responsive design strategy.
- 
- **Offline support** HTML pages once loaded can be made available offline on the machine without any need of internet.
- **Game development**- HTML5 has native support for rich experience and is now useful in gaming development arena as well.

## Audience

This **HTML 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.

## 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 –

Sr.No	Tag & Description
1	<b>&lt;!DOCTYPE...&gt;</b> This tag defines the document type and HTML version.
2	<b>&lt;html&gt;</b> This tag encloses the complete HTML document and mainly comprises of document header which is represented by <b>&lt;head&gt;...&lt;/head&gt;</b> and document body which is represented by <b>&lt;body&gt;...&lt;/body&gt;</b> tags.
3	<b>&lt;head&gt;</b> This tag represents the document's header which can keep other HTML tags like <b>&lt;title&gt;, &lt;link&gt;</b> etc.
4	<b>&lt;title&gt;</b> The <b>&lt;title&gt;</b> tag is used inside the <b>&lt;head&gt;</b> tag to mention the document title.
5	<b>&lt;body&gt;</b> This tag represents the document's body which keeps other HTML tags like <b>&lt;h1&gt;, &lt;div&gt;, &lt;p&gt;</b> etc.
6	<b>&lt;h1&gt;</b> This tag represents the heading.
7	<b>&lt;p&gt;</b> 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 4.

## HTML Document Structure

A typical HTML document will have the following structure –

```
<html>

    <head>
        Document header related tags
    </head>

    <body>
        Document body related tags
    </body>

</html>
```

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

## 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.

## 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 –

# This is heading 1

## This is heading 2

### This is heading 3

#### This is heading 4

##### This is heading 5

This is heading 6

## Paragraph Tag

The **<p>** tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening **<p>** and a closing **</p>** 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 –

Here is a first paragraph of text.

Here is a second paragraph of text.

Here is a third paragraph of text.

## Line Break Tag

Whenever you use the `<br />` 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 `<br />` 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 `<br>` 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 />
            Mahnaz</p>
    </body>

</html>
```

This will produce the following result –

Hello  
You delivered your assignment  
on time.  
Thanks  
One Byte Labs

## Centering Content

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

### Example

```
<!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 following result –

This text is not in the center.

This text is in the center.

## 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 –

This is paragraph one and should be on top

---

This is paragraph two and should be at bottom

Again **<hr />** tag is an example of the **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

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.

## Example

```
<!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 –

```
function testFunction( strText ){
    alert (strText)
}
```

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

## Nonbreaking Spaces

Suppose you want to use the phrase "12 Angry Men." Here, you would not want a browser to split the "12, Angry" and "Men" across two lines –

An example of this technique appears in the movie "12 Angry Men."

In cases, where you do not want the client browser to break text, you should use a nonbreaking space entity `&nbsp;` instead of a normal space. For example, when coding the "12 Angry Men" in a paragraph, you should use something similar to the following code –

### Example

```
<!DOCTYPE html>
<html>

    <head>
        <title>Nonbreaking Spaces Example</title>
```

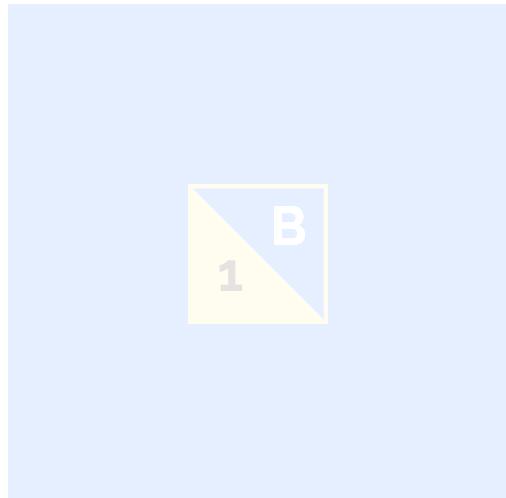
```
</head>

<body>
    <p>An example of this technique appears in the movie
    "12 &nbsp;Angry &nbsp;Men."</p>
</body>

</html>
```

This will produce the following result –

An example of this technique appears in the movie  
"12 Angry Men."



## HTML Elements and Attributes

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
<p>	This is paragraph content.	</p>
<h1>	This is heading content.	</h1>
<div>	This is division content.	</div>

So here <p>....</p> 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 <br /> 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, <p> is starting tag of a paragraph and </p> is closing tag of the same paragraph but <p>This is paragraph</p> is a paragraph element.

## Nested HTML Elements

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

### Example

```
<!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 –

## This is *italic* heading

This is underlined paragraph

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.

### Example

```
<!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 –

This is left aligned

This is center aligned

This is right aligned

## 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.

### Example

```
<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. Their 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.

### Example

```
<!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 –

**Titled Heading Tag Example**

Now try to bring your cursor over "Titled Heading Tag Example" and you will see that whatever title you used in your code is coming out as a tooltip of the cursor.

## 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 –



At this point of time, we are not learning CSS, so just let's proceed without bothering much about CSS. Here, you need to understand what are HTML attributes and how they can be used while formatting content.

## 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 –

This is how IE 5 renders right-to-left directed text.

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.

The values of the *lang* attribute are ISO-639 standard two-character language codes.

### Example

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

    <head>
        <title>English Language Page</title>
    </head>

    <body>
        This page is using English Language
    </body>

</html>
```

This will produce the following result –

This page is using English Language

## The **xml:lang** Attribute

The *xml:lang* attribute is the XHTML replacement for the *lang* attribute. The value of the *xml:lang* attribute should be an ISO-639 country code as mentioned in previous section.

### 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.
width	Numeric Value	Specifies the width of tables, images, or table cells.
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.

## 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 **<b>...</b>** 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 –

The following word uses a **bold** typeface.

### Italic Text

Anything that appears within *<i>...</i>* 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 an <i>italicized</i> typeface.</p>
</body>

</html>
```

This will produce the following result –

The following word uses an *italicized* typeface.

## 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 an <u>underlined</u> typeface.</p>
    </body>

</html>
```

This will produce the following result –

The following word uses an underlined typeface.

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 –

The following word uses a ~~strikethrough~~ typeface.

## 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.

## Example

```
<!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 –

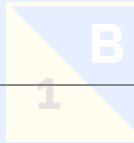
The following word uses a `monospaced` typeface.

## Superscript Text

The content of a **<sup>...</sup>** 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 –

The following word uses a <sup>superscript</sup> typeface.

## Subscript Text

The content of a **<sub>...</sub>** 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 –

The following word uses a <sub>subscript</sub> typeface.

## 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 –

I want to drink ~~cola~~ wine

## Deleted Text

Anything that appears within **<del>...</del>** 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 –

I want to drink ~~cola~~ wine

## 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 –

### Example

```
<!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 –

The following word uses a **big** typeface.

## 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 –

### Example

```
<!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 –

The following word uses a small typeface.

## Grouping Content

The **<div>** and **<span>** 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 **<div>** element to indicate that all of the elements within that **<div>** element relate to the footnotes. You might then attach a style to this **<div>** element so that they appear using a special set of style rules.

## Example

```
<!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 –

[HOME](#) | [CONTACT](#) | [ABOUT](#)

**Content Articles**

Actual content goes here.....

The **<span>** 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 **<span>** 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 –

This is the example of **span tag** and the **div tag** along with CSS

These tags are commonly used with CSS to allow you to attach a style to a section of a page.

## HTML Phrase Tags

The phrase tags have been designed 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 **<em>...</em>** element is displayed as emphasized text.

#### Example

```
<!DOCTYPE html>
<html>

    <head>
        <title>Emphasized Text Example</title>
    </head>

    <body>
        <p>The following word uses an <em>emphasized</em>
        typeface.</p>
    </body>

</html>
```

This will produce the following result –

The following word uses an *emphasized* typeface.

### 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 –

The following word has been **marked** with yellow

## Strong Text

Anything that appears within **<strong>...</strong>** 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 –

The following word uses a **strong** typeface.

## 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 =
"Abhishek">Abhy</abbr>.</p>
    </body>

</html>
```

This will produce the following result –

My best friend's name is Abhy.

## 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 –

This chapter covers marking up text in XHTML.

## 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>
```



## Special Terms

The **<dfn>...</dfn>** element (or HTML Definition Element) allows you to specify that you are introducing a special term. Its 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 –

The following word is a *special* term.

## 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>
```

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

This will produce the following result –

The following description of XHTML is taken from the W3C Web site:

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.

## 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 –

Amit is in Spain, “I think I am wrong.”

## 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 –

This HTML tutorial is derived from *W3 Standard for HTML*.

## 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>

```

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

This will produce the following result –

Regular text. This is code. Regular text.

## 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 –

Regular text. This is inside kbd element Regular text.

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 –

```
document.write("user-name")
```

## 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 –

```
Result produced by the program is Hello World!
```

## 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 –

```
388A, Road No 22, Whitefield - Bangalore
```

## 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 –

Sr.No	Attribute & Description
1	<b>Name</b> Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc.
2	<b>content</b> Specifies the property's value.
3	<b>scheme</b> Specifies a scheme to interpret the property's value (as declared in the content attribute).
4	<b>http-equiv</b> 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 –

Hello HTML5!

## 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 = "Tutorialspoint, 3/7/2014"
/>
    </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 = "Tutorialspoint, 3/7/2014"
/>
        <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

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 = "Tutorialspoint, 3/7/2014"
/>
        <meta http-equiv = "refresh" content = "5; url =
http://www.tutorialspoint.com" />
    </head>
```

```
<body>
    <p>Hello HTML5!</p>
</body>

</html>
```

## 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 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 = "author" content = "Mahnaz Mohtashim" />
</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 = "author" content = "Mahnaz Mohtashim" />
        <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 –

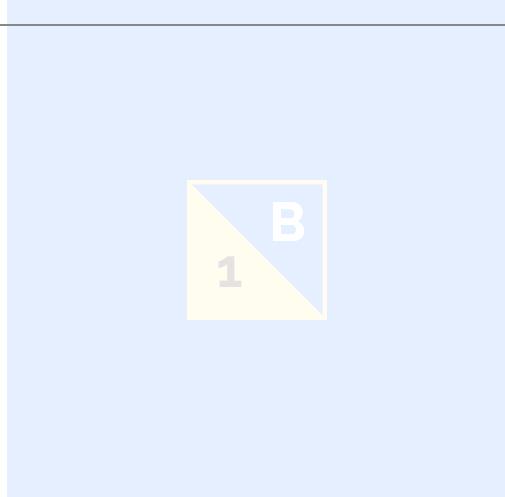
```
<!DOCTYPE html>
```

```
<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 = "author" content = "Mahnaz Mohtashim" />
        <meta http-equiv = "Content-Type" content = "text/html;
charset = Big5" />
    </head>

    <body>
        <p>Hello HTML5!</p>
    </body>

</html>
```



## 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 within `<!-- ... -->` 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>
```

This will produce the following result –

```
Document content goes here.....
```

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 –

```
Document content goes here.....
```

## 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 –

```
<!-- This is not a valid comment -->
```

```
Document content goes here.....
```

## Conditional Comments

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.

## Using Comment Tag

There are few browsers that support <comment> tag to comment a part of HTML code.

**Note** – The <comment> tag deprecated in HTML5. Do not use this element.

### Example

```
<!DOCTYPE html>  
<html>  
  
    <head>  
        <title>Using Comment Tag</title>  
    </head>  
  
    <body>  
        <p>This is <comment>not</comment> Internet Explorer.</p>  
    </body>  
  
</html>
```

If you are using IE, then it will produce following result –

This is Internet Explorer.

But if you are not using IE, then it will produce following result –

This is not Internet Explorer.

## Commenting Script Code

Though you will learn JavaScript with HTML, in a separate tutorial, but here you must make a note that if you are using Java Script or VB Script in your HTML code then it is recommended to put that script code inside proper HTML comments so that old browsers can work properly.

## Example

```
<!DOCTYPE html>
<html>

    <head>
        <title>Commenting Script Code</title>

        <script>
            <!--
                document.write("Hello World!")
            //-->
        </script>
    </head>

    <body>
        <p>Hello , World!</p>
    </body>

</html>
```

This will produce the following result –



## Commenting Style Sheets

Though you will learn using style sheets with HTML in a separate tutorial, but here you must make a note that if you are using Cascading Style Sheet (CSS) in your HTML code then it is recommended to put that style sheet code inside proper HTML comments so that old browsers can work properly.

## Example

```
<!DOCTYPE html>
<html>

    <head>
        <title>Commenting Style Sheets</title>

        <style>
            <!--
                .example {

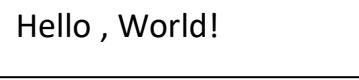
```

```
        border:1px solid #4a7d49;
    }
//-->
</style>
</head>

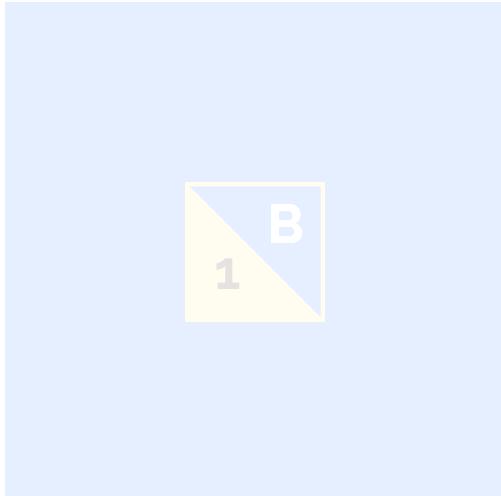
<body>
    <div class = "example">Hello , World!</div>
</body>

</html>
```

This will produce the following result –



```
Hello , World!
```



## HTML Images and Text Links

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 = "/html/images/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 = "/html/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 = "/html/images/test.png" alt = "Test Image" width =
"150" height = "100"/>
    </body>

</html>
```

## 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 = "/html/images/test.png" alt = "Test Image" border = "3"/>
    </body>

</html>
```

## 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 = "/html/images/test.png" alt = "Test Image" border = "3" align = "right"/>
    </body>

</html>
```

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.

## 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 `</a>` 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

Let's try following example which links <http://www.onebytelabs.in> at your page –

```
<!DOCTYPE html>
<html>

    <head>
        <title>Hyperlink Example</title>
    </head>

    <body>
        <p>Click following link</p>
        <a href = "https://www.onebytelabs.in" target =
        "_self">Tutorials Point</a>
    </body>

</html>
```



This will produce the following result, where you can click on the link generated to reach to the home page of Tutorials Point (in this example).

## 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 –

Sr.No	Option & Description
1	<b>_blank</b> Opens the linked document in a new window or tab.
2	<b>_self</b> Opens the linked document in the same frame.
3	<b>_parent</b> Opens the linked document in the parent frame.
4	<b>_top</b> Opens the linked document in the full body of the window.
5	<b>targetframe</b> Opens the linked document in a named <i>targetframe</i> .

## Example

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

```
<!DOCTYPE html>
<html>

    <head>
        <title>Hyperlink Example</title>
        <base href = "https://www.onebytelabs.in/">
    </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>
        <base href = "https://www.onebytelabs.in/">
    </head>

    <body>
        <p>Click following link</p>
        <a href = "/html/index.htm" target = "_blank">HTML
Tutorial</a>
    </body>

</html>
```

This will produce the following result, where you can click on the link generated **HTML Tutorial** to reach to the HTML tutorial.

Now given URL `<a href = "/html/index.htm"` is being considered as `<a href = "http://www.onebytelabs.in/html/index.htm"`

## 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.

**Note** – The *name* attribute deprecated in HTML5. Do not use this attribute. Use *id* and *title* attribute instead.

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 <a name = "top"></a></h1>
```

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

```
<a href = "/html/html_text_links.htm#top">Go to the Top</a>
```

This will produce following link, where you can click on the link generated **Go to the Top** to reach to the top of the HTML Text Link tutorial.

[Go to the Top](#)

## Setting Link Colors

You can set colors of your links, active links and visited links using **link**, **alink** and **vlink** attributes of `<body>` tag.

### Example

Save the following in test.htm and open it in any web browser to see how **link**, **alink** and **vlink** attributes work.

```
<!DOCTYPE html>
<html>

    <head>
        <title>Hyperlink Example</title>
        <base href = "https://www.onebytelabs.in/">
    </head>

    <body alink = "#54A250" link = "#040404" vlink = "#F40633">
        <p>Click following link</p>
        <a href = "/html/index.htm" target = "_blank" >HTML
Tutorial</a>
    </body>

</html>
```

This will produce the following result. Just check color of the link before clicking on it, next check its color when you activate it and when the link has been visited.

## Download Links

You can create text link to make your PDF, or DOC or ZIP files downloadable. This is very simple; you just need to give complete URL of the downloadable file as follows –

```
<!DOCTYPE html>
<html>

    <head>
        <title>Hyperlink Example</title>
    </head>

    <body>
        <a href = "https://www.onebytelabs.in/page.pdf">Download PDF File</a>
    </body>

</html>
```

This will produce following link and will be used to download a file.

## File Download Dialog Box

Sometimes it is desired that you want to give an option where a user will click a link and it will pop up a "File Download" box to the user instead of displaying actual content. This is very easy and can be achieved using an HTTP header in your HTTP response.

For example, if you want make a **Filename** file downloadable from a given link then its syntax will be as follows.

```
#!/usr/bin/perl

# Additional HTTP Header
print "Content-Type:application/octet-stream; name =
\"FileName\"\r\n";
print "Content-Disposition:attachment; filename =
\"FileName\"\r\n\r\n";

# Open the target file and list down its content as follows
open( FILE, "<FileName" );

while(read(FILE, $buffer, 100)){
    print("$buffer");
}
```

## 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. The elements under **<td>** are regular and left aligned by default

### 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>
```

1	B
2	C

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. Headings, which are defined in **<th>** tag are centered and bold by default.

### 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>Ramesh Raman</td>
                <td>5000</td>
            </tr>

            <tr>
                <td>Shabbir Hussein</td>
                <td>7000</td>
            </tr>
        </table>
    </body>
</html>
```

1

B

## 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 space between table cells, 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>Ramesh Raman</td>
    <td>5000</td>
</tr>
<tr>
    <td>Shabbir Hussein</td>
    <td>7000</td>
</tr>
</table>
</body>

</html>

```

## 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>

```



```
</table>
</body>

</html>
```

## 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.

**Note** – The *bgcolor*, *background*, and *bordercolor* attributes deprecated in HTML5. Do not use these attributes.

### 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>
```

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

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 =  
"/images/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>
```

## 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 Width/Height</title>
</head>

<body>
    <table border = "1" width = "400" height = "150">
        <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>

```

## 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 Caption</title>
    </head>

    <body>
        <table border = "1" width = "100%">
            <caption>This is the caption</caption>

            <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>

```

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

## 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

```
<!DOCTYPE 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>

```

## 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.

```

<!DOCTYPE html>
<html>

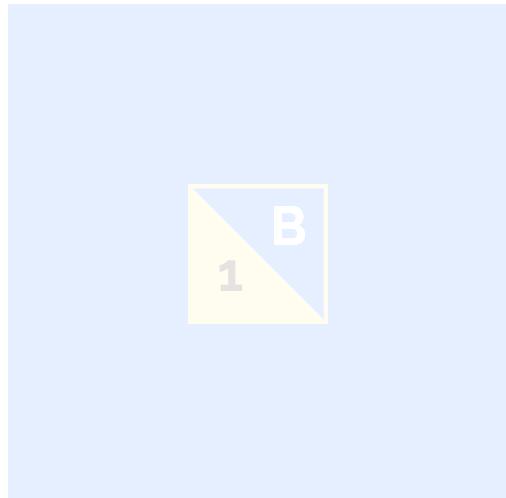
    <head>
        <title>HTML Table</title>
    </head>

    <body>
        <table border = "1" width = "100%">

            <tr>
                <td>
                    <table border = "1" width = "100%">
                        <tr>
                            <th>Name</th>
                            <th>Salary</th>
                        </tr>
                        <tr>
                            <td>Ramesh Raman</td>
                            <td>5000</td>
                        </tr>
                        <tr>
                            <td>Shabbir Hussein</td>
                            <td>7000</td>
                        </tr>
                    </table>
                </td>
            </tr>

        </table>
    
```

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



## 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 –

- **<ul>** – An unordered list. This will list items using plain bullets.
- **<ol>** – 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 –

- Beetroot
- Ginger
- Potato
- Radish

## The type Attribute

You can use **type** attribute for `<ul>` tag to specify the type of bullet you like. By default, it is a disc. Following are the possible options –

```
<ul type = "square">  
<ul type = "disc">  
<ul type = "circle">
```

### Example

Following is an example where we used `<ul type = "square">`

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



This will produce the following result –

- Beetroot
- Ginger
- Potato
- Radish

### Example

Following is an example where we used `<ul type = "disc">` –

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

```
<title>HTML Unordered List</title>
</head>

<body>
  <ul type = "disc">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ul>
</body>

</html>
```

This will produce the following result –

- Beetroot
- Ginger
- Potato
- Radish

## Example

Following is an example where we used `<ul type = "circle">` –

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Unordered List</title>
  </head>

  <body>
    <ul type = "circle">
      <li>Beetroot</li>
      <li>Ginger</li>
      <li>Potato</li>
      <li>Radish</li>
    </ul>
  </body>

</html>
```

This will produce the following result –

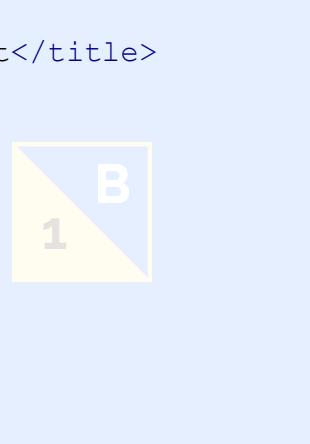
- o Beetroot
- o Ginger
- o Potato
- o Radish

## 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 –

5. Beetroot
6. Ginger
7. Potato
8. Radish

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

1. Beetroot
2. Ginger
3. Potato
4. Radish

## 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 –

- I. Beetroot
- II. Ginger
- III. Potato
- IV. Radish

### 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 –

- i. Beetroot
- ii. Ginger
- iii. Potato
- iv. Radish

## 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 –

- A. Beetroot
- B. Ginger
- C. Potato
- D. Radish

## 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 –

- a. Beetroot
- b. Ginger
- c. Potato
- d. Radish

## 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 –

<code>&lt;ol type = "1" start = "4"&gt;</code>	- Numerals starts with 4.
<code>&lt;ol type = "I" start = "4"&gt;</code>	- Numerals starts with IV.
<code>&lt;ol type = "i" start = "4"&gt;</code>	- Numerals starts with iv.
<code>&lt;ol type = "a" start = "4"&gt;</code>	- Letters starts with d.
<code>&lt;ol type = "A" start = "4"&gt;</code>	- 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>
```

- iv. Beetroot
- v. Ginger
- vi. Potato
- vii. Radish

It –

## 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 –

### HTML

This stands for Hyper Text Markup Language

### HTTP

This stands for Hyper Text Transfer Protocol

## 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>`, `<em>`, `<strong>`, `<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) `<div>` tag and (ii) `<span>` tag

#### The `<div>` 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 `<div>` tag can be used to create webpage layout where we define different parts (Left, Right, Top etc.) of the page using `<div>` 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 `<div>` tag. We will learn Cascading Style Sheet (CSS) in a separate chapter but we used it here to show the usage of `<div>` 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
        Following is a list of vegetables</h4>

        <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
        Following is a list of fruits</h4>

        <ul>
            <li>Apple</li>
            <li>Banana</li>
            <li>Mango</li>
            <li>Strawberry</li>
        </ul>
    </div>
</body>

</html>

```

This will produce the following result –

**This is first group *Following is a list of vegetables***

- Beetroot
- Ginger
- Potato
- Radish

**This is second group Following is a list of fruits**

- Apple
- Banana
- Mango
- Strawberry

## The <span> tag

The HTML <span> 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 <span> tag and the <div> tag is that the <span> tag is used with inline elements whereas the <div> tag is used with block-level elements.

### Example

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

This is red and this is green

## HTML Iframes

You can define an inline frame with HTML tag **<iframe>**. The **<iframe>** tag is not somehow related to **<frameset>** tag, instead, it can appear anywhere in your document. The **<iframe>** tag defines a rectangular region within the document in which the browser can display a separate document, including scrollbars and borders. An inline frame is used to embed another document within the current HTML document.

The **src** attribute is used to specify the URL of the document that occupies the inline frame.

### Example

Following is the example to show how to use the **<iframe>** –

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML Iframes</title>
    </head>

    <body>
        <p>Document content goes here...</p>
        
        <iframe src = "/html/menu.htm" width = "555" height = "200">
            Sorry your browser does not support inline frames.
        </iframe>
        <p>Document content also go here...</p>
    </body>

</html>
```

### The **<Iframe>** Tag Attributes

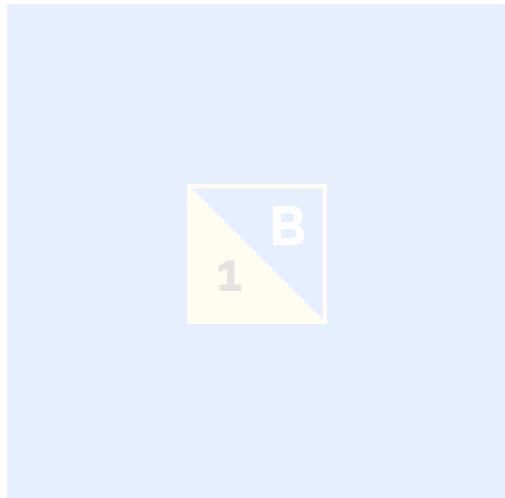
Most of the attributes of the **<iframe>** tag, including *name*, *class*, *frameborder*, *id*, *longdesc*, *marginheight*, *marginwidth*, *name*, *scrolling*, *style*, and *title* behave exactly like the corresponding attributes for the **<frame>** tag.

**Note –** The *frameborder*, *marginwidth*, *longdesc*, *scrolling*, *marginheight* attributes deprecated in HTML5. Do not use these attributes.

Sr.No	Attribute & Description

1	<b>src</b>
	This attribute is used to give the file name that should be loaded in the frame. Its value can be any URL. For example, src = "/html/top_frame.htm" will load an HTML file available in html directory.
2	<b>name</b>
	This attribute allows you to give a name to a frame. It is used to indicate which frame a document should be loaded into. This is especially important when you want to create links in one frame that load pages into another frame, in which case the second frame needs a name to identify itself as the target of the link.
3	<b>frameborder</b>
	This attribute specifies whether or not the borders of that frame are shown; it overrides the value given in the frameborder attribute on the <frameset> tag if one is given, and this can take values either 1 (yes) or 0 (no).
4	<b>marginwidth</b>
	This attribute allows you to specify the width of the space between the left and right of the frame's borders and the frame's content. The value is given in pixels. For example marginwidth = "10".
5	<b>marginheight</b>
	This attribute allows you to specify the height of the space between the top and bottom of the frame's borders and its contents. The value is given in pixels. For example marginheight = "10".
6	<b>height</b>
	This attribute specifies the height of <iframe>.
7	<b>scrolling</b>
	This attribute controls the appearance of the scrollbars that appear on the frame. This takes values either "yes", "no" or "auto". For example scrolling = "no" means it should not have scroll bars.
8	<b>longdesc</b>

	This attribute allows you to provide a link to another page containing a long description of the contents of the frame. For example longdesc = "framedescription.htm"
9	<b>width</b> This attribute specifies the width of <iframe>.



## HTML Background and Fonts

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.

**Note** – The *bgcolor* attribute deprecated in HTML5. Do not use this attribute.

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 --> B  
<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>  
    </body>  
</html>
```

```

        </td>
    </tr>
</table>

<!-- Format 2 - Use hex value --&gt;
&lt;table bgcolor = "#6666FF" width = "100%"&gt;
    &lt;tr&gt;
        &lt;td&gt;
            This background is sky blue
        &lt;/td&gt;
    &lt;/tr&gt;
&lt;/table&gt;

&lt;/body&gt;

&lt;/html&gt;
</pre>

```

This will produce the following result –



This background is yellow  
This background is sky blue

## 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.

**Note** – The *background* attribute deprecated in HTML5. Do not use this attribute.

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>

```

## Patterned & Transparent Backgrounds

You might have seen many pattern or transparent backgrounds on various websites. This simply can be achieved by using patterned image or transparent image in the background.

It is suggested that while creating patterns or transparent GIF or PNG images, use the smallest dimensions possible even as small as 1x1 to avoid slow loading.

### Example

Here are the examples to set background pattern of a table –

```

<!DOCTYPE html>
<html>

    <head>
        <title>HTML Background Images</title>
    </head>

    <body>
        <!-- Set a table background using pattern -->
        <table background = "/images/pattern1.gif" width = "100%"
height = "100">
            <tr>
                <td>
                    This background is filled up with a pattern image.
                </td>
            </tr>
        </table>
    </body>

```

```

<!-- Another example on table background using pattern -->
<table background = "/images/pattern2.gif" width = "100%" height = "100">
    <tr>
        <td>
            This background is filled up with a pattern image.
        </td>
    </tr>
</table>
</body>

</html>

```

**Fonts** play a very important role in making a website more user friendly and increasing content readability. Font face and color depends entirely on the computer and browser that is being used to view your page but you can use HTML **<font>** tag to add style, size, and color to the text on your website. You can use a **<basefont>** tag to set all of your text to the same size, face, and color.

The font tag is having three attributes called **size**, **color**, and **face** to customize your fonts. To change any of the font attributes at any time within your webpage, simply use the **<font>** tag. The text that follows will remain changed until you close with the **</font>** tag. You can change one or all of the font attributes within one **<font>** tag.

**Note** –The *font* and *basefont* tags are deprecated and it is supposed to be removed in a future version of HTML. So they should not be used rather, it's suggested to use CSS styles to manipulate your fonts. But still for learning purpose, this chapter will explain font and basefont tags in detail.

## Set Font Size

You can set content font size using **size** attribute. The range of accepted values is from 1(smallest) to 7(largest). The default size of a font is 3.

### Example

```

<!DOCTYPE html>
<html>

    <head>
        <title>Setting Font Size</title>
    </head>

    <body>

```

```

<font size = "1">Font size = "1"</font><br />
<font size = "2">Font size = "2"</font><br />
<font size = "3">Font size = "3"</font><br />
<font size = "4">Font size = "4"</font><br />
<font size = "5">Font size = "5"</font><br />
<font size = "6">Font size = "6"</font><br />
<font size = "7">Font size = "7"</font>
</body>

</html>

```

This will produce the following result –

```

Font size = "1"
Font size = "2"
Font size = "3"
Font size = "4"
Font size = "5"
Font size = "6"
Font size = "7"

```

## Relative Font Size

You can specify how many sizes larger or how many sizes smaller than the preset font size should be. You can specify it like **<font size = "+n">** or **<font size = "-n">**

### Example

```

<!DOCTYPE html>
<html>

    <head>
        <title>Relative Font Size</title>
    </head>

    <body>
        <font size = "-1">Font size = "-1"</font><br />
        <font size = "+1">Font size = "+1"</font><br />
        <font size = "+2">Font size = "+2"</font><br />
    
```

```
<font size = "+3">Font size = "+3"</font><br />
<font size = "+4">Font size = "+4"</font>
</body>

</html>
```

This will produce the following result –

```
Font size = "-1"
Font size = "+1"
Font size = "+2"
Font size = "+3"
Font size = "+4"
```

## Setting Font Face

You can set font face using *face* attribute but be aware that if the user viewing the page doesn't have the font installed, they will not be able to see it. Instead user will see the default font face applicable to the user's computer.

### Example

```
<!DOCTYPE html>
<html>

    <head>
        <title>Font Face</title>
    </head>

    <body>
        <font face = "Times New Roman" size = "5">Times New
Roman</font><br />
        <font face = "Verdana" size = "5">Verdana</font><br />
        <font face = "Comic sans MS" size = "5">Comic Sans
MS</font><br />
        <font face = "WildWest" size = "5">WildWest</font><br />
        <font face = "Bedrock" size = "5">Bedrock</font><br />
    </body>

</html>
```

This will produce the following result –

```
Times New Roman  
Verdana  
Comic Sans MS  
WildWest  
Bedrock
```

## Specify alternate font faces

A visitor will only be able to see your font if they have that font installed on their computer. So, it is possible to specify two or more font face alternatives by listing the font face names, separated by a comma.

```
<font face = "arial,helvetica">  
<font face = "Lucida Calligraphy,Comic Sans MS,Lucida Console">
```

When your page is loaded, their browser will display the first font face available. If none of the given fonts are installed, then it will display the default font face *Times New Roman*.

**Note** – Check a complete list of [HTML Standard Fonts](#).

## Setting Font Color

You can set any font color you like using *color* attribute. You can specify the color that you want by either the color name or hexadecimal code for that color.

**Note** – You can check a complete list of [HTML Color Name with Codes](#).

### Example

```
<!DOCTYPE html>  
<html>  
  
    <head>  
        <title>Setting Font Color</title>  
    </head>  
  
    <body>  
        <font color = "#FF00FF">This text is in pink</font><br />  
        <font color = "red">This text is red</font>
```

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

This will produce the following result –

This text is in pink  
This text is red

## The <basefont> Element

The <basefont> element is supposed to set a default font size, color, and typeface for any parts of the document that are not otherwise contained within a <font> tag. You can use the <font> elements to override the <basefont> settings.

The <basefont> tag also takes color, size and face attributes and it will support relative font setting by giving size a value of +1 for a size larger or -2 for two sizes smaller.

### Example

```
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Setting Basefont Color</title>  
    </head>  
  
    <body>  
        <basefont face = "arial, verdana, sans-serif" size = "2"  
color = "#ff0000">  
            <p>This is the page's default font.</p>  
            <h2>Example of the &lt;basefont&gt; Element</h2>  
  
            <p><font size = "+2" color = "darkgray">  
                This is darkgray text with two sizes larger  
            </font>  
            </p>  
  
            <p><font face = "courier" size = "-1" color = "#000000">  
                It is a courier font, a size smaller and black in  
color.  
            </font>  
            </p>  
        </body>  
  
</html>
```



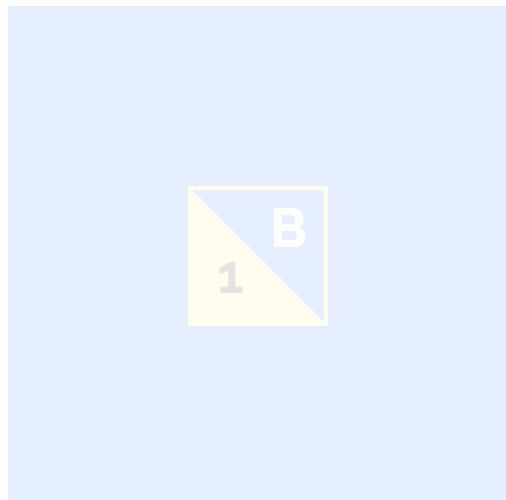
This will produce the following result –

This is the page's default font.

## **Example of the <basefont> Element**

This is darkgray text with two sizes larger

It is a courier font, a size smaller and black in color.



## 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 –

Sr.No	Attribute & Description
1	<b>action</b> Backend script ready to process your passed data.
2	<b>method</b> Method to be used to upload data. The most frequently used are GET and POST methods.
3	<b>target</b> Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
4	<b>enctype</b> 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.

**multipart/form-data** – This is used when you want to upload binary data in the form of files like image, word file etc.

**Note** – You can refer to [Perl & CGI](#) for a detail on how form data upload works.

## 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 –

First name:

Last name:

## Attributes

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

Sr.No	Attribute & Description
1	<b>type</b> Indicates the type of input control and for text input control it will be set to <b>text</b> .
2	<b>name</b> Used to give a name to the control which is sent to the server to be recognized and get the value.
3	<b>value</b>

	This can be used to provide an initial value inside the control.
4	<b>size</b> Allows to specify the width of the text-input control in terms of characters.
5	<b>maxlength</b> 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 –

User ID :	<input type="text"/>
Password:	<input type="password"/>

## Attributes

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

Sr.No	Attribute & Description
1	<b>type</b> Indicates the type of input control and for password input control it will be set to <b>password</b> .
2	<b>name</b> Used to give a name to the control which is sent to the server to be recognized and get the value.
3	<b>value</b> This can be used to provide an initial value inside the control.
4	<b>size</b> Allows to specify the width of the text-input control in terms of characters.
5	<b>maxlength</b> Allows to specify the maximum number of characters a user can enter into the text box.

## 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 –

Description:



## Attributes

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

Sr.No	Attribute & Description
1	<b>name</b> Used to give a name to the control which is sent to the server to be recognized and get the value.
2	<b>rows</b> Indicates the number of rows of text area box.
3	<b>cols</b> 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

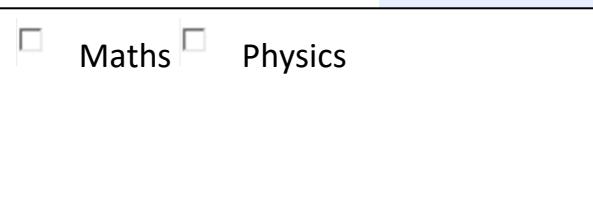
Here is an example HTML code for a form with two checkboxes –

```
<!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 –



The image shows a simple HTML form within a browser window. It contains two checkboxes. The first checkbox, labeled 'Maths', is unchecked. The second checkbox, labeled 'Physics', is checked. The checked state is indicated by a small checkmark icon to the left of the label.

## Attributes

Following is the list of attributes for `<checkbox>` tag.

Sr.No	Attribute & Description
1	<b>type</b> Indicates the type of input control and for checkbox input control it will be set

	<b>to checkbox..</b>
2	<b>name</b> Used to give a name to the control which is sent to the server to be recognized and get the value.
3	<b>value</b> The value that will be used if the checkbox is selected.
4	<b>checked</b> Set to <i>checked</i> 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 –

Maths  Physics

## Attributes

Following is the list of attributes for radio button.

Sr.No	Attribute & Description
1	<b>type</b> Indicates the type of input control and for checkbox input control it will be set to radio.
2	<b>name</b> Used to give a name to the control which is sent to the server to be recognized and get the value.
3	<b>value</b> The value that will be used if the radio box is selected.
4	<b>checked</b> Set to <i>checked</i> 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>
```

```

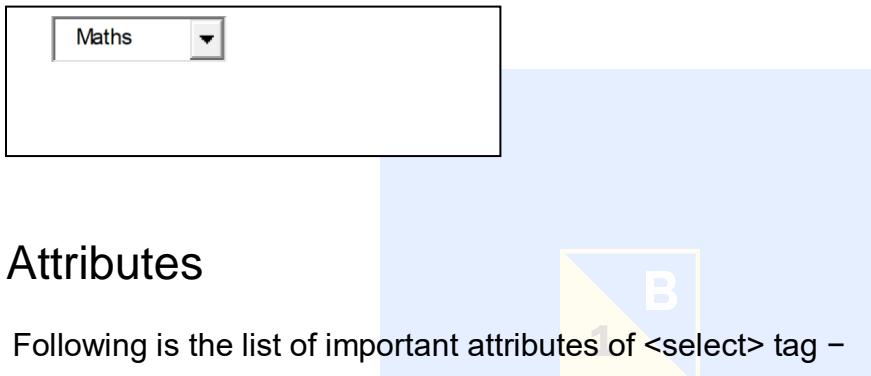
<title>Select Box Control</title>
</head>

<body>
    <form>
        <select name = "dropdown">
            <option value = "Maths" selected>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 –

Sr.No	Attribute & Description
1	<b>name</b> Used to give a name to the control which is sent to the server to be recognized and get the value.
2	<b>size</b> This can be used to present a scrolling list box.
3	<b>multiple</b> 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 –

Sr.No	Attribute & Description
1	<b>value</b> The value that will be used if an option in the select box is selected.
2	<b>selected</b> Specifies that this option should be the initially selected value when the page loads.
3	<b>label</b> 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>

```

## Attributes

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

Sr.No	Attribute & Description
1	<b>name</b> Used to give a name to the control which is sent to the server to be recognized and get the value.
2	<b>accept</b> 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 –

Sr.No	Type & Description
1	<b>submit</b> This creates a button that automatically submits a form.
2	<b>reset</b> This creates a button that automatically resets form controls to their initial values.
3	<b>button</b> This creates a button that is used to trigger a client-side script when the user clicks that button.
4	<b>image</b> 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 =
"/html/images/logo.png" />
        </form>
    </body>

</html>
```

## 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>
        </form>
    </body>
```

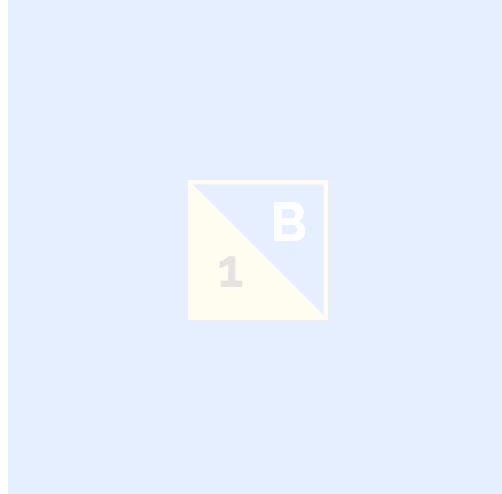
```
<input type = "hidden" name = "pagename" value = "10" />
<input type = "submit" name = "submit" value = "Submit" />
<input type = "reset" name = "reset" value = "Reset" />
</form>
</body>

</html>
```

This will produce the following result –

This is page 10

Submit Reset



## HTML Multimedia

Sometimes you need to add music or video into your web page. The easiest way to add video or sound to your web site is to include the special HTML tag called **<embed>**. This tag causes the browser itself to include controls for the multimedia automatically provided browser supports **<embed>** tag and given media type.

You can also include a **<noembed>** tag for the browsers which don't recognize the **<embed>** tag. You could, for example, use **<embed>** to display a movie of your choice, and **<noembed>** to display a single JPG image if browser does not support **<embed>** tag.

### Example

Here is a simple example to play an embedded midi file –

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML embed Tag</title>
    </head>

    <body>
        <embed src = "/html/yourfile.mid" width = "100%" height =
"60" >
        <noembed><img src = "yourimage.gif" alt = "Alternative
Media" ></noembed>
    </embed>
    </body>

</html>
```

### The **<embed>** Tag Attributes

Following is the list of important attributes which can be used with **<embed>** tag.

**Note** –The *align* and *autoplay* attributes deprecated in HTML5. Do not use these attributes.

Sr.No	Attribute & Description
1	<b>align</b>

	Determines how to align the object. It can be set to either center, <i>left</i> or <i>right</i> .
2	<b>autoplay</b> This boolean attribute indicates if the media should start automatically. You can set it either true or false.
3	<b>loop</b> Specifies if the sound should be played continuously (set loop to true), a certain number of times (a positive value) or not at all (false)
4	<b>playcount</b> Specifies the number of times to play the sound. This is alternate option for <i>loop</i> if you are usiong IE.
5	<b>hidden</b> Specifies if the multimedia object should be shown on the page. A false value means no and true values means yes.
6	<b>width</b> Width of the object in pixels
7	<b>height</b> Height of the object in pixels
8	<b>name</b> A name used to reference the object.
9	<b>src</b> URL of the object to be embedded.
10	<b>volume</b> Controls volume of the sound. Can be from 0 (off) to 100 (full volume).

# Supported Video Types

You can use various media types like Flash movies (.swf), AVI's (.avi), and MOV's (.mov) file types inside embed tag.

- **.swf files** – are the file types created by Macromedia's Flash program.
- **.wmv files** – are Microsoft's Window's Media Video file types.
- **.mov files** – are Apple's Quick Time Movie format.
- **.mpeg files** – are movie files created by the Moving Pictures Expert Group.

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML embed Tag</title>
    </head>

    <body>
        <embed src = "/html/yourfile.swf" width = "200" height =
"200" >
            <noembed><img src = "yourimage.gif" alt = "Alternative
Media" ></noembed>
        </embed>
    </body>

</html>
```

## Background Audio

You can use HTML **<bgsound>** tag to play a soundtrack in the background of your webpage. This tag is supported by Internet Explorer only and most of the other browsers ignore this tag. It downloads and plays an audio file when the host document is first downloaded by the user and displayed. The background sound file also will replay whenever the user refreshes the browser.

**Note** – The bgsound tag is deprecated and it is supposed to be removed in a future version of HTML. So they should not be used rather, it's suggested to use HTML5 tag audio for adding sound. But still for learning purpose, this chapter will explain bgsound tag in detail.

This tag is having only two attributes *loop* and *src*. Both these attributes have same meaning as explained above.

Here is a simple example to play a small midi file –

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

```

<head>
    <title>HTML embed Tag</title>
</head>

<body>
    <bgsound src = "/html/yourfile.mid">
        <noembed><img src = "yourimage.gif" ></noembed>
    </bgsound>
</body>

</html>

```

This will produce the blank screen. This tag does not display any component and remains hidden.

Internet Explorer can also handle only three different sound format files – wav, the native format for PCs; au, the native format for most Unix workstations; and MIDI, a universal music-encoding scheme.

## HTML Object tag

HTML 4 introduces the **<object>** element, which offers an all-purpose solution to generic object inclusion. The **<object>** element allows HTML authors to specify everything required by an object for its presentation by a user agent.

Here are a few examples –

### Example - 1

You can embed an HTML document in an HTML document itself as follows –

```

<object data = "data/test.htm" type = "text/html" width = "300"
height = "200">
    alt : <a href = "data/test.htm">test.htm</a>
</object>

```

Here *alt* attribute will come into picture if browser does not support *object* tag.

### Example - 2

You can embed a PDF document in an HTML document as follows –

```

<object data = "data/test.pdf" type = "application/pdf" width =
"300" height = "200">
    alt : <a href = "data/test.pdf">test.htm</a>
</object>

```

### Example - 3

You can specify some parameters related to the document with the **<param>** tag. Here is an example to embed a wav file –

```
<object data = "data/test.wav" type = "audio/x-wav" width = "200"  
height = "20">  
    <param name = "src" value = "data/test.wav">  
    <param name = "autoplay" value = "false">  
    <param name = "autoStart" value = "0">  
    alt : <a href = "data/test.wav">test.wav</a>  
</object>
```

#### Example - 4

You can add a flash document as follows –

```
<object classid = "clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" id =  
"penguin"  
    codebase = "someplace/swflash.cab" width = "200" height = "300">  
  
    <param name = "movie" value = "flash/penguin.swf" />  
    <param name = "quality" value = "high" />  
    <img src = "penguin.jpg" width = "200" height = "300" alt =  
"Penguin" />  
</object>
```

#### Example - 5

You can add a java applet into HTML document as follows –

```
<object classid = "clsid:8ad9c840-044e-11d1-b3e9-00805f499d93"  
    width = "200" height = "200">  
    <param name = "code" value = "applet.class">  
</object>
```

The **classid** attribute identifies which version of Java Plug-in to use. You can use the optional **codebase** attribute to specify if and how to download the JRE.

## HTML Header

We have learnt that a typical HTML document will have following structure –

```
Document declaration tag
<html>

    <head>
        Document header related tags
    </head>

    <body>
        Document body related tags
    </body>

</html>
```

This chapter will give a little more detail about header part which is represented by HTML `<head>` tag. The `<head>` tag is a container of various important tags like `<title>`, `<meta>`, `<link>`, `<base>`, `<style>`, `<script>`, and `<noscript>` tags.

### The HTML `<title>` Tag

The HTML `<title>` tag is used for specifying the title of the HTML document. Following is an example to give a title to an HTML document –

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML Title Tag Example</title>
    </head>

    <body>
        <p>Hello, World!</p>
    </body>

</html>
```

This will produce the following result –

```
Hello, World!
```

### The HTML `<meta>` Tag

The HTML <meta> tag is used to provide metadata about the HTML document which includes information about page expiry, page author, list of keywords, page description etc.

Following are few of the important usages of <meta> tag inside an HTML document –

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML Meta Tag Example</title>

        <!-- Provide list of keywords -->
        <meta name = "keywords" content = "C, C++, Java, PHP, Perl,
Python">

        <!-- Provide description of the page -->
        <meta name = "description" content = "Simply Easy Learning by
Tutorials Point">

        <!-- Author information -->
        <meta name = "author" content = "Tutorials Point">

        <!-- Page content type -->
        <meta http-equiv = "content-type" content = "text/html;
charset = UTF-8"> 1

        <!-- Page refreshing delay -->
        <meta http-equiv = "refresh" content = "30">

        <!-- Page expiry -->
        <meta http-equiv = "expires" content = "Wed, 21 June 2006
14:25:27 GMT">

        <!-- Tag to tell robots not to index the content of a page -->
        <meta name = "robots" content = "noindex,nofollow">

    </head>

    <body>
        <p>Hello, World!</p>
    </body>

</html>
```

## The HTML <base> Tag

The HTML <base> tag is used for specifying the base URL for all relative URLs in a page, which means all the other URLs will be concatenated into base URL while locating for the given item.

For example, all the given pages and images will be searched after prefixing the given URLs with base URL <http://www.onebytelabs.in/> directory –

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML Base Tag Example</title>
        <base href = "https://www.onebytelabs.in/" />
    </head>

    <body>
        <img src = "/images/logo.png" alt = "Logo Image"/>
        <a href = "/html/index.htm" title = "HTML Tutorial">HTML
Tutorial</a>
    </body>

</html>
```

1

B

But if you change base URL to something else, for example, if base URL is <http://www.onebytelabs.in/> then image and other given links will become like <http://www.onebytelabs.in/images/logo.png> and <http://www.onebytelabs.in/index.htm>

## The HTML <link> Tag

The HTML <link> tag is used to specify relationships between the current document and external resource. Following is an example to link an external style sheet file available in **css** sub-directory within web root –

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML link Tag Example</title>
        <base href = "https://www.onebytelabs.in/" />
        <link rel = "stylesheet" type = "text/css" href =
"/css/style.css">
    </head>

    <body>
        <p>Hello, World!</p>
    </body>
```

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

## The HTML <style> Tag

The HTML <style> tag is used to specify style sheet for the current HTML document. Following is an example to define few style sheet rules inside <style> tag –

```
<!DOCTYPE html>  
<html>  
  
    <head>  
        <title>HTML style Tag Example</title>  
        <base href = "https://www.onebytelabs.in/" />  
  
        <style type = "text/css">  
            .myclass {  
                background-color: #aaa;  
                padding: 10px;  
            }  
        </style>  
    </head>  
  
    <body>  
        <p class = "myclass">Hello, World!</p>  
    </body>  
  
</html>
```



**Note** – To learn about how Cascading Style Sheet works, kindly check a separate tutorial available at css

## The HTML <script> Tag

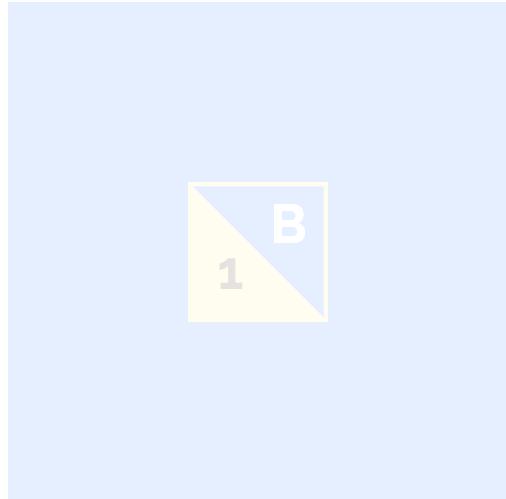
The HTML <script> tag is used to include either external script file or to define internal script for the HTML document. Following is an example where we are using JavaScript to define a simple JavaScript function –

```
<!DOCTYPE html>  
<html>  
  
    <head>  
        <title>HTML script Tag Example</title>  
        <base href = "http://www.onebytelabs.in/" />
```

```
<script type = "text/JavaScript">
    function Hello() {
        alert("Hello, World");
    }
</script>
</head>

<body>
    <input type = "button" onclick = "Hello();" name = "ok" value
= "OK"  />
</body>

</html>
```



## HTML Marquees

An HTML marquee is a scrolling piece of text displayed either horizontally across or vertically down your webpage depending on the settings. This is created by using HTML <marquees> tag.

**Note** – The <marquee> tag deprecated in HTML5. Do not use this element, instead you can use JavaScript and CSS to create such effects.

### Syntax

A simple syntax to use HTML <marquee> tag is as follows –

```
<marquee attribute_name = "attribute_value"....more attributes>
    One or more lines or text message or image
</marquee>
```

### The <marquee> Tag Attributes

Following is the list of important attributes which can be used with <marquee> tag.

Sr.No	Attribute & Description
1	<b>width</b> This specifies the width of the marquee. This can be a value like 10 or 20% etc.
2	<b>height</b> This specifies the height of the marquee. This can be a value like 10 or 20% etc.
3	<b>direction</b> This specifies the direction in which marquee should scroll. This can be a value like <i>up</i> , <i>down</i> , <i>left</i> or <i>right</i> .
4	<b>behavior</b> This specifies the type of scrolling of the marquee. This can have a value like <i>scroll</i> , <i>slide</i> and <i>alternate</i> .
5	<b>scrolldelay</b>

	This specifies how long to delay between each jump. This will have a value like 10 etc.
6	<b>scrollamount</b> This specifies the speed of marquee text. This can have a value like 10 etc.
7	<b>loop</b> This specifies how many times to loop. The default value is INFINITE, which means that the marquee loops endlessly.
8	<b>bgcolor</b> This specifies background color in terms of color name or color hex value.
9	<b>hspace</b> This specifies horizontal space around the marquee. This can be a value like 10 or 20% etc.
10	<b>vspace</b> This specifies vertical space around the marquee. This can be a value like 10 or 20% etc.

Below are few examples to demonstrate the usage of marquee tag.

### Examples - 1

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML marquee Tag</title>
    </head>

    <body>
        <marquee>This is basic example of marquee</marquee>
    </body>

</html>
```

## Examples - 2

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML marquee Tag</title>
    </head>

    <body>
        <marquee width = "50%">This example will take only 50%
width</marquee>
    </body>

</html>
```

## Examples - 3

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML marquee Tag</title>
    </head>

    <body>
        <marquee direction = "right">This text will scroll from left
to right</marquee>
    </body>

</html>
```

## Examples - 4

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML marquee Tag</title>
    </head>

    <body>
        <marquee direction = "up">This text will scroll from bottom
to up</marquee>
    </body>
</html>
```

## HTML JavaScript

A **script** is a small piece of program that can add interactivity to your website. For example, a script could generate a pop-up alert box message, or provide a dropdown menu. This script could be written using JavaScript or VBScript.

You can write various small functions, called event handlers using any of the scripting language and then you can trigger those functions using HTML attributes.

Now-a-days, only **JavaScript** and associated frameworks are being used by most of the web developers, VBScript is not even supported by various major browsers.

You can keep JavaScript code in a separate file and then include it wherever it's needed, or you can define functionality inside HTML document itself. Let's see both the cases one by one with suitable examples.

### External JavaScript

If you are going to define a functionality which will be used in various HTML documents then it's better to keep that functionality in a separate JavaScript file and then include that file in your HTML documents. A JavaScript file will have extension as **.js** and it will be included in HTML files using **<script>** tag.

#### Example

Consider we define a small function using JavaScript in **script.js** which has following code –

```
function Hello() {  
    alert("Hello, World");  
}
```

Now let's make use of the above external JavaScript file in our following HTML document –

```
<!DOCTYPE html>  
<html>  
  
    <head>  
        <title>Javascript External Script</title>  
        <script src = "/html/script.js" type =  
"text/javascript"/></script>  
    </head>  
  
    <body>  
        <input type = "button" onclick = "Hello();" name = "ok" value  
= "Click Me" />  
    </body>
```

```
</html>
```

## Internal Script

You can write your script code directly into your HTML document. Usually we keep script code in header of the document using `<script>` tag, otherwise there is no restriction and you can put your source code anywhere in the document but inside `<script>` tag.

### Example

```
<!DOCTYPE html>
<html>

    <head>
        <title>JavaScript Internal Script</title>
        <base href = "https://www.tutorialspoint.com/" />

        <script type = "text/JavaScript">
            function Hello() {
                alert("Hello, World");
            }
        </script>
    </head>

    <body>
        <input type = "button" onclick = "Hello();" name = "ok" value
= "Click Me" />
    </body>

</html>
```

The diagram illustrates the flow of execution. A yellow triangle labeled '1' points to the `function Hello()` definition. A blue triangle labeled 'B' points to the `onclick = "Hello();"` attribute in the `<input>` tag, indicating that the function is called when the button is clicked.

## Event Handlers

Event handlers are nothing but simply defined functions which can be called against any mouse or keyboard event. You can define your business logic inside your event handler which can vary from a single to 1000s of line code.

Following example explains how to write an event handler. Let's write one simple function `EventHandler()` in the header of the document. We will call this function when any user brings mouse over a paragraph.

```
<!DOCTYPE html>
<html>

    <head>
        <title>Event Handlers Example</title>
```

```

<base href = "https://www.tutorialspoint.com/" />

<script type = "text/JavaScript">
    function EventHandler() {
        alert("I'm event handler!!");
    }
</script>
</head>

<body>
    <p onmouseover = "EventHandler();">Bring your mouse here to
see an alert</p>
</body>

</html>

```

## Hide Scripts from Older Browsers

Although most (if not all) browsers these days support JavaScript, but still some older browsers don't. If a browser doesn't support JavaScript, instead of running your script, it would display the code to the user. To prevent this, you can simply place HTML comments around the script as shown below.

### **JavaScript Example:**

```

<script type = "text/JavaScript">
    <!--
        document.write("Hello JavaScript!");
    //-->
</script>

```

### **VBScript Example:**

```

<script type = "text/vbscript">
    <!--
        document.write("Hello VBScript!")
    '-->
</script>

```

## The **<noscript>** Element

You can also provide alternative info to the users whose browsers don't support scripts and for those users who have disabled script option their browsers. You can do this using the **<noscript>** tag.

### **JavaScript Example:**

```

<script type = "text/JavaScript">
    <!--
        document.write("Hello JavaScript!");
    //-->

```

```
//-->
</script>

<noscript>Your browser does not support JavaScript!</noscript>

VBScript Example:
<script type = "text/vbscript">
    <!--
        document.write("Hello VBScript!")
    '-->
</script>

<noscript>Your browser does not support VBScript!</noscript>
```

## Default Scripting Language

There may be a situation when you will include multiple script files and ultimately use multiple `<script>` tags. You can specify a default scripting language for all your `script` tags. This saves you from specifying the language every time you use a `script` tag within the page. Below is the example –

```
<meta http-equiv = "Content-Script-Type" content =
"text/JavaScript" />
```

Note that you can still override the default by specifying a language within the `script` tag.

## HTML CSS

Cascading Style Sheets (CSS) describe how documents are presented on screens, in print, or perhaps how they are pronounced. W3C has actively promoted the use of style sheets on the Web since the consortium was founded in 1994.

Cascading Style Sheets (CSS) provide easy and effective alternatives to specify various attributes for the HTML tags. Using CSS, you can specify a number of style properties for a given HTML element. Each property has a name and a value, separated by a colon (:). Each property declaration is separated by a semi-colon (;).

### Example

First let's consider an example of HTML document which makes use of `<font>` tag and associated attributes to specify text color and font size –

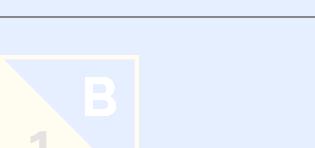
**Note** – The `font` tag deprecated and it is supposed to be removed in a future version of HTML. So they should not be used rather, it's suggested to use CSS styles to manipulate your fonts. But still for learning purpose, this chapter will work with an example using the `font` tag.

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML CSS</title>
    </head>

    <body>
        <p><font color = "green" size = "5">Hello, World!</font></p>
    </body>

</html>
```



We can re-write above example with the help of Style Sheet as follows –

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML CSS</title>
    </head>

    <body>
        <p style = "color:green; font-size:24px;">Hello, World!</p>
    </body>
```

```
</html>
```

This will produce the following result –

```
Hello, World!
```

You can use CSS in three ways in your HTML document –

- **External Style Sheet** – Define style sheet rules in a separate .css file and then include that file in your HTML document using HTML `<link>` tag.
- **Internal Style Sheet** – Define style sheet rules in header section of the HTML document using `<style>` tag.
- **Inline Style Sheet** – Define style sheet rules directly along-with the HTML elements using `style` attribute.

Let's see all the three cases one by one with the help of suitable examples.

## External Style Sheet

If you need to use your style sheet to various pages, then it's always recommended to define a common style sheet in a separate file. A cascading style sheet file will have extension as `.css` and it will be included in HTML files using `<link>` tag.

### Example

Consider we define a style sheet file `style.css` which has following rules –

```
.red {  
    color: red;  
}  
.thick {  
    font-size: 20px;  
}  
.green {  
    color: green;  
}
```

Here we defined three CSS rules which will be applicable to three different classes defined for the HTML tags. I suggest you should not bother about how these rules are being defined because you will learn them while studying CSS. Now let's make use of the above external CSS file in our following HTML document –

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

```

<head>
    <title>HTML External CSS</title>
    <link rel = "stylesheet" type = "text/css" href =
"/html/style.css">
</head>

<body>
    <p class = "red">This is red</p>
    <p class = "thick">This is thick</p>
    <p class = "green">This is green</p>
    <p class = "thick green">This is thick and green</p>
</body>

</html>

```

This will produce the following result –

```


This is red  

    This is thick  

    This is green  

    This is thick and green


```

## Internal Style Sheet

If you want to apply Style Sheet rules to a single document only, then you can include those rules in header section of the HTML document using `<style>` tag.

Rules defined in internal style sheet overrides the rules defined in an external CSS file.

### Example

Let's re-write above example once again, but here we will write style sheet rules in the same HTML document using `<style>` tag –

```

<!DOCTYPE html>
<html>

    <head>
        <title>HTML Internal CSS</title>

        <style type = "text/css">
            .red {
                color: red;

```

```

        }
        .thick{
            font-size:20px;
        }
        .green {
            color:green;
        }
    </style>
</head>

<body>
    <p class = "red">This is red</p>
    <p class = "thick">This is thick</p>
    <p class = "green">This is green</p>
    <p class = "thick green">This is thick and green</p>
</body>

</html>

```

This will produce the following result –

This is red

This is thick

This is green

This is thick and green

## Inline Style Sheet

You can apply style sheet rules directly to any HTML element using **style** attribute of the relevant tag. This should be done only when you are interested to make a particular change in any HTML element only.

Rules defined inline with the element overrides the rules defined in an external CSS file as well as the rules defined in `<style>` element.

### Example

Let's re-write above example once again, but here we will write style sheet rules along with the HTML elements using **style** attribute of those elements.

```

<!DOCTYPE html>
<html>

```

```
<head>
    <title>HTML Inline CSS</title>
</head>

<body>
    <p style = "color:red;">This is red</p>
    <p style = "font-size:20px;">This is thick</p>
    <p style = "color:green;">This is green</p>
    <p style = "color:green;font-size:20px;">This is thick and
green</p>
    </body>

</html>
```

This will produce the following result –

This is red

This is thick

This is green

This is thick and green

## HTML Layouts

A webpage layout is very important to give better look to your website. It takes considerable time to design a website's layout with great look and feel.

Now-a-days, all modern websites are using CSS and JavaScript based framework to come up with responsive and dynamic websites but you can create a good layout using simple HTML tables or division tags in combination with other formatting tags. This chapter will give you few examples on how to create a simple but working layout for your webpage using pure HTML and its attributes.

### HTML Layout - Using Tables

The simplest and most popular way of creating layouts is using HTML <table> tag. These tables are arranged in columns and rows, so you can utilize these rows and columns in whatever way you like.

#### Example

***Execute the below code in browser for better understanding***

For example, the following HTML layout example is achieved using a table with 3 rows and 2 columns but the header and footer column spans both columns using the colspan attribute –

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML Layout using Tables</title>
    </head>

    <body>
        <table width = "100%" border = "0">

            <tr>
                <td colspan = "2" bgcolor = "#b5dcb3">
                    <h1>This is Web Page Main title</h1>
                </td>
            </tr>
            <tr valign = "top">
                <td bgcolor = "#aaa" width = "50">
                    <b>Main Menu</b><br />
                    HTML<br />
                    PHP<br />
                    PERL...
                </td>
            </tr>
        </table>
    </body>
</html>
```

```

        <td bgcolor = "#eee" width = "100" height = "200">
            Technical and Managerial Tutorials
        </td>
    </tr>
    <tr>
        <td colspan = "2" bgcolor = "#b5dcb3">
            <center>
                Copyright © 2020 onebytelabs.in
            </center>
        </td>
    </tr>

</table>
</body>

</html>

```

## Multiple Columns Layout - Using Tables

You can design your webpage to put your web content in multiple pages. You can keep your content in middle column and you can use left column to use menu and right column can be used to put advertisement or some other stuff. This layout will be very similar to what we have at our website [tutorialspoint.com](http://tutorialspoint.com).

### Example

Here is an example to create three column layout –

```

<!DOCTYPE html>
<html>

    <head>
        <title>Three Column HTML Layout</title>
    </head>

    <body>
        <table width = "100%" border = "0">

            <tr valign = "top">
                <td bgcolor = "#aaa" width = "20%">
                    <b>Main Menu</b><br />
                    HTML<br />
                    PHP<br />
                    PERL...
                </td>

```

```

<td bgcolor = "#b5dcb3" height = "200" width = "60%">
    Technical and Managerial Tutorials
</td>

<td bgcolor = "#aaa" width = "20%">
    <b>Right Menu</b><br />
    HTML<br />
    PHP<br />
    PERL...
</td>
</tr>

<table>
</body>
</html>

```

## HTML Layouts - Using DIV, SPAN

The `<div>` element is a block level element used for grouping HTML elements. While the `<div>` tag is a block-level element, the HTML `<span>` element is used for grouping elements at an inline level.

Although we can achieve pretty nice layouts with HTML tables, but tables weren't really designed as a layout tool. Tables are more suited to presenting tabular data.

**Note** – This example makes use of Cascading Style Sheet (CSS), so before understanding this example you need to have a better understanding on how CSS works.

### Example

Here we will try to achieve same result using `<div>` tag along with CSS, whatever you have achieved using `<table>` tag in previous example.

```

<!DOCTYPE html>
<html>

    <head>
        <title>HTML Layouts using DIV, SPAN</title>
    </head>

    <body>
        <div style = "width:100%">

            <div style = "background-color:#b5dcb3; width:100%">
                <h1>This is Web Page Main title</h1>
            </div>
        </div>
    </body>
</html>

```

```
<div style = "background-color:#aaa; height:200px;  
width:100px; float:left;">  
    <div><b>Main Menu</b></div>  
    HTML<br />  
    PHP<br />  
    PERL...  
</div>  
  
<div style = "background-color:#eee; height:200px;  
width:350px; float:left;">  
    <p>Technical and Managerial Tutorials</p>  
</div>  
  
<div style = "background-color:#aaa; height:200px;  
width:100px; float:right;">  
    <div><b>Right Menu</b></div>  
    HTML<br />  
    PHP<br />  
    PERL...  
</div>  
  
<div style = "background-color:#b5dcb3; clear:both">  
    <center>  
        Copyright © 2020 onebytelabs.in  
    </center>  
</div>  
    </div>  
</body>  
</html>
```

You can create better layout using DIV, SPAN along with CSS. For more information on CSS, please refer to CSS Tutorial.

## HTML Tag Reference

Following tags have been introduced in older versions of HTML but all the tags marked with  are part of HTML-5.

Tag	Description	Version
<!--...-->	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	5
<bgsound>	Specifies background music	
<big>	Specifies big text	
<blink>	Specifies a text which blinks	
<blockquote>	Specifies a long quotation	1
<body>	Specifies the body element	
 	Inserts a single line break	
<button>	Specifies a push button	
<canvas>	For making graphics with a script	5
<caption>	Specifies a table caption	
<center>	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	5
<dd>	Specifies a definition description	
<del>	Specifies deleted text	
<dfn>	Specifies a definition term	
<dialog>	Specifies a dialog box or window	5
<dir>	Deprecated. Specifies a directory list	
<div>	Specifies a section in a document	
<dl>	Specifies a definition list	
<dt>	Specifies a definition term	
<em>	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	5
<label>	Specifies a label for a form control	
<layer>	Specifies a layer	
<legend>	Specifies a title in a fieldset	
<li>	Specifies a list item	
<link>	Specifies a resource reference	
<main>	Specifies the main or important content in the document. There is only one element in the document	5
<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	
<ol>	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	5
<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	5
<q>	Specifies a short quotation	
<rp>	Specifies to show browsers that do not support the ruby element	5
<rt>	Specifies an text ruby annotation	5
<ruby>	Specifies an ruby annotation	5
<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	
<span>	Specifies a section in a document	
<strike>	Deprecated. Specifies strikethrough text	
<strong>	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>	5
<title>	Specifies the document title	
<tr>	Specifies a table row	
<track>	Specifies a text tracks used in mediaplayers	5
<tt>	Specifies teletype text	
<u>	Deprecated. Specifies underlined text	
<ul>	Specifies an unordered list	
<var>	Specifies a variable	
<video>	Specifies a text tracks used in media players	5
<wbr>	Indicates a potential word break point within a <nobr> section	
<xmp>	Deprecated. Specifies preformatted text	

