

HTML

HTML stands for **H**ypertext **M**arkup **L**anguage, and it is the most widely used language to write Web Pages.

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

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

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

Basic HTML Document

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

```
<!DOCTYPE html>

<html>

<head>

<title>This is document title</title>

</head>

<body>

<h1>This is a heading</h1>

<p>Document content goes here.....</p>

</body>

</html>
```

HTML Basic Tags

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

Above example of HTML document uses the following tags:

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

<code><h1></code>	This tag represents the heading.
<code><p></code>	This tag represents a paragraph.

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>
```

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 third paragraph of text.</p>

<p>Here is a second paragraph of text.</p>

</body>

</html>
```

Line Break Tag

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

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

Centering Content

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

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.

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 ** ** instead of a normal space. For example, when coding the "12 Angry Men" in a paragraph, you should use something similar to this.

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

HTML ELEMENTS

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 **
** elements. These are known as **void elements**.

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.

HTML ATTRIBUTES

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**.

Example

```
<!DOCTYPE html>

<html>

<head>

<title>Align AttributeExample</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>
```

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.

For now, let's use the id attribute to distinguish between two paragraph elements as shown below.

```
<p id="html">This para explains what is HTML</p>

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

The title Attribute

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

Example

```
<h3 title="Hello HTML!">Titled Heading Tag Example</h3>
```

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.

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:

Some text...

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.

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, hexadecimal, 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.

HTML FORMATING

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

Bold Text

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

<p>The following word uses a bold typeface.</p>

Italic Text

Anything that appears within `<i>...</i>` element is displayed in italicized as shown below:

`<p>The following word uses a <i>italicized</i> typeface.</p>`

Underlined Text

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

`<p>The following word uses a <u>underlined</u> typeface.</p>`

Strike Text

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

`<p>The following word uses a <strike>strikethrough</strike> typeface.</p>`

Superscript Text

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

`<p>The following word uses a ^{superscript} typeface.</p>`

Subscript Text

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

`<p>The following word uses a _{subscript} typeface.</p>`

Inserted Text and Deleted Text

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

`<p>I want to drink cola <ins>wine</ins></p>`

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:

`<p>The following word uses a <big>big</big> typeface.</p>`

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:

`<p>The following word uses a <small>small</small> typeface.</p>`

Grouping Content

The **`<div>`** and **``** elements allow you to group together several elements to create sections or subsections of a page.

For example, you might want to put all of the footnotes on a page within a `<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>

<!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>

```

PHRASE TAGS

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

Emphasized Text

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

Example

```

<!DOCTYPE html>

<html>

<head>

<title>Emphasized Text Example</title>

</head>

<body>

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

</body></html>

```

This will produce the following result:

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>
```

Strong Text

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

Example

```
<!DOCTYPE html>

<html>

<head>

<title>Strong Text Example</title>

</head>

<body>

<p>The following word uses a <strong>strong</strong> typeface.</p>

</body>

</html>
```

This will produce the following result:

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>
```

Computer Code

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

Example

```
<!DOCTYPE html>

<html>

<head>

<title>Computer Code Example</title>

</head>

<body>

<p>Regular text. <code>This is code.</code> Regular text.</p>

</body>

</html>
```

This will produce the following result:

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.

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, Jubilee Hills - Hyderabad

Insert Image

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

```

```

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

Attributes for **** are height, width, border and align.

TABLES

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

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

Example

```
<!DOCTYPE html>

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

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

Table Heading

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

Example

```
<!DOCTYPE html>

<html>

<head>

<title>HTML Table Header</title>

</head>

<body>

<table border="1">

<tr>

<th>Name</th>

<th>Salary</th>

</tr>

<tr>

<td>Ramesh Raman</td>

<td>5000</td>

</tr>

<tr>

<td>Shabbir Hussein</td>

<td>7000</td>

</tr>

</table>

</body></html>
```

Cellpadding and Cellspacing Attributes

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

Example

```
<!DOCTYPE html>

<html>

<head>

<title>HTML Table Cellpadding</title>

</head>

<body>

<table border="1" cellpadding="5" cellspacing="5">

<tr>

<th>Name</th>

<th>Salary</th>

</tr>

<tr>

<td>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>
```

Tables Backgrounds

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

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

You can also set border color also using **bordercolor** attribute.

Example

```
<!DOCTYPE html>

<html>

<head>

<title>HTML Table Background</title>

</head>

<body>

<table border="1" bordercolor="green" bgcolor="yellow"> <tr>

<th>Column 1</th>

<th>Column 2</th>

<th>Column 3</th>

</tr>

<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>

<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>

<tr><td colspan="3">Row 3 Cell 1</td></tr>

</table>

</body>

</html>
```

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

```
<table border="1" bordercolor="green" background="/images/test.png">
```

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.

```
<!DOCTYPE html>

<html>

<head>

<title>HTML Table Width/Height</title></head>

<body>

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

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>
```

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 footer</td>

</tr>

<tbody>

<tr>

<td>Cell 1</td>

<td>Cell 2</td>

<td>Cell 3</td>

<td>Cell 4</td>

</tr>

</tbody>

</table>

</body></html>
```

LISTS

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

- **** - An unordered list. This will list items using plain bullets.
- **** - An ordered list. This will use different schemes of numbers to list your items.

- **<dl>** - A definition list. This arranges your items in the same way as they are arranged in a dictionary.

HTML Unordered Lists

An unordered list is a collection of related items that have no special order or sequence.

This list is created by using HTML **** 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 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

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 **** tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with ****.

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>
```

The type Attribute

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

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

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

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

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

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

Example

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

```
<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

    <ol type="1">

        <li>Beetroot</li>

        <li>Ginger</li>

        <li>Potato</li>

        <li>Radish</li>

    </ol>

</body>

</html>
```

The start Attribute

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

<code><ol type="1" start="4"></code>	- Numerals starts with 4.
<code><ol type="I" start="4"></code>	- Numerals starts with IV.
<code><ol type="i" start="4"></code>	- Numerals starts with iv.
<code><ol type="a" start="4"></code>	Letter starts - s with d.
<code><ol type="A" start="4"></code>	Letter starts - s 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>
```

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

Linking Documents

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

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

Example

```
<!DOCTYPE html>

<html>

<head>

<title>Hyperlink Example</title>

</head>

<body>

<p>Click following link</p>

<a href="http://www.google.com" target="_self">Google </a>

</body></html>
```

This will produce the following result, where you can click on the link generated to reach to the home page of Google(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:

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

</head>

<body>

<p>Click any of the following links</p>

<a href="(path)" target="_blank">Opens in New</a> |

<a href="(path)" target="_self">Opens in Self</a> |

<a href="(path)" target="_parent">Opens in Parent</a> |

<a href="(path)" 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.

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>, , , <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 , <i>, <u>, , , <sup>, <sub>, <big>, <small>, , <ins>, , <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) 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</h4>
    <p>Following is a list of vegetables</p>
    <ul>
        <li>Beetroot</li>
        <li>Ginger</li>
        <li>Potato</li>
        <li>Radish</li>
    </ul>
</div>
<!-- Second group of tags -->
<div style="color:green">
```



```
<h4>This is second group</h4>

<p>Following is a list of fruits</p>

<ul>

<li>Apple</li>

<li>Banana</li>

<li>Mango</li>

<li>Strawberry</li>

</ul>

</div>

</body>

</html>
```

The tag

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

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

Example

```
<!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>
```

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>
```

Attribute	Description
Action	Backend script ready to process your passed data.
Method	Method to be used to upload data. The most frequently used are GET and POST methods.

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.

Attributes

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

Attribute	Description
Type	Indicates the type of input control and for text input control it will be set to text.
Name	Used to give a name to the control which is sent to the server to be recognized and get the value.
Value	This can be used to provide an initial value inside the control.
Size	Allows to specify the width of the text-input control in terms of characters.
maxlength	Allows to specify the maximum number of characters a user can enter into the text box.

Password Input controls

This is also a single-line text input but it masks the character as soon as a user enters it.

They are also created using HTML `<input>` tag but type attribute is set to **password**.

Example

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

```
<form >

User ID : <input type="text" name="user_id" />

<br>

Password: <input type="password" name="password" />

</form>
```

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>
```

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

Checkbox Control

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

Example

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

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

Attribute	Description
type	Indicates the type of input control and for checkbox input control it will be set to checkbox.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	The value that will be used if the checkbox is selected.
checked	Set to <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**.

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

checked	Set to <i>checked</i> if you want to select it by default.
---------	--

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>
```

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.

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

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

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

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

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>
```


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>
```

Button Controls

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

Type	Description
submit	This creates a button that automatically submits a form.
Reset	This creates a button that automatically resets form controls to their initial values.

button	This creates a button that is used to trigger a client-side script when the user clicks that button.
image	This creates a clickable button but we can use an image as background of the button.

Example

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

```
<!DOCTYPE html>

<html>

<head>

<title>File Upload Box</title>

</head>

<body>

<form>

<input type="submit" name="submit" value="Submit" /> <input
type="reset" name="reset" value="Reset" /> <input type="button"
name="ok" value="OK" />

<input type="image" name="imagebutton" src="/html/images/logo.png" />

</form>

</body>

</html>
```

MARQUEE

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 HTML `<marquee>` tag may not be supported by various browsers so it is not recommended to rely on this tag, 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.

Attribute	Description
width	This specifies the width of the marquee. This can be a value like 10 or 20% etc.
height	This specifies the height of the marquee. This can be a value like 10 or 20% etc.
direction	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> .
behavior	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> .
scrollDelay	This specifies how long to delay between each jump. This will have a value like 10 etc.

scrollamount	This specifies the speed of marquee text. This can have a value like 10 etc.
loop	This specifies how many times to loop. The default value is INFINITE, which means that the marquee loops endlessly.
bgcolor	This specifies background color in terms of color name or color hex value.
hspace	This specifies horizontal space around the marquee. This can be a value like 10 or 20% etc.
vspace	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>
```

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>
```

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 HTMLdocument:

```
<!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">

<!-- 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 <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>

<style type="text/css">

.myclass{

    background-color: #aaa;

    padding: 10px;

}

</style>

</head>

<body>

<p class="myclass">Hello, World!</p>

</body>

</html>
```

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>

<script type="text/JavaScript">

function Hello(){

    alert("Hello, World");

}

</script>

</head>

<body>

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

</body>

</html>
```

HTML Button Tag

The HTML <button> tag is used for creating a button within HTML form. You can also use <input> tag to create similar buttons.

Example

```
<!DOCTYPE html>

<html>

<head>

<title>HTML Button Tag</title>

</head>

<body>

<form>

<button name="button" value="OK" type="button">Click Me</button>

</form>

</body>

</html>
```


HTML <canvas> Tag

The HTML <canvas> tag is for drawing graphics, animations etc using scripting.

```
<body>

<canvas id="newCanvas">Your browser does not support canvas tag.</canvas>

<script>

var c=document.getElementById('newCanvas');
var ctx=c.getContext('2d');
ctx.fillStyle='#00FD00';
ctx.fillRect(0,0,200,60);

</script>

</body>
```

EXAMPLE: TO DRAW A CIRCLE:

```
<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid
#d3d3d3;">

Your browser does not support the HTML5 canvas tag.</canvas>

<script>

var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();

</script>

</body>

</html>
```

The HTML <svg> Element

The HTML `<svg>` element is a container for SVG graphics.

SVG has several methods for drawing paths, boxes, circles, text, and graphic images.

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

<!--circle-->

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>

<!--rectangle-->

<svg width="400" height="100">
  <rect width="400" height="100" style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)" />
</svg>

<!--rounded rectangle-->

<svg width="400" height="180">
  <rect x="50" y="20" rx="20" ry="20" width="150" height="150"
    style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />
</svg>

<!--polygon: star-->

<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
    style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />
</svg>

</body>
</html>
```

The HTML <video> Element

To show a video in HTML, use the `<video>` element:

```
<video width="320" height="240" controls>
  <source src="path/filename.extention " type="video/(extention of the file)">
Your browser does not support the video tag.
</video>
```

The HTML <audio> Element

To play an audio file in HTML, use the `<audio>` element:

```
<audio controls>
  <source src="filename.extention/path" type="audio/(extention)">
  < Your browser does not support the audio element.
</audio>
```

HTML <footer> Tag

```
<footer>
  <p>Posted by: aaaaa</p>
  <p>Contact information: <a href="(path for example) ">
    someone@example.com</a>.</p>
</footer>
```

HTML <progress> Tag

Shows you the progress what is done.

```
<label for="file">Downloading progress:</label>
<progress id="file" value="32" max="100"> 32% </progress>
```

HTML <legend> and <fieldset> Tag

The <fieldset> tag is used to group related elements in a form.

The <fieldset> tag draws a box around the related elements.

The <legend> tag defines a caption for the [<fieldset>](#) element.

```
<form>
  <fieldset>
    <legend>Personalia:</legend>
    Name: <input type="text"><br>
    Email: <input type="text"><br>
    Date of birth: <input type="text">
  </fieldset>
</form>
```

HTML <embed> Tag

The <embed> tag defines a container for an external application or interactive content (a plug-in).

```
<embed src="helloworld.swf">
```

HTML <datalist> Tag

The <datalist> tag specifies a list of pre-defined options for an <input> element.

The <datalist> tag is used to provide an "autocomplete" feature on <input> elements. Users will see a drop-down list of pre-defined options as they input data.

Use the <input> element's list attribute to bind it together with a <datalist> element.

```
<input list="browsers">

<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

HTML <article> Tag

The <article> tag specifies independent, self-contained content.

An article should make sense on its own and it should be possible to distribute it independently from the rest of the site.

Potential sources for the <article> element:

- Forum post
- Blog post
- News story
- Comment

```
<article>
  <h1>Google Chrome</h1>
  <p>Google Chrome is a free, open-source web browser developed by
Google, released in 2008.</p>
</article>
```

HTML <blockquote> Tag

```
<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 50 years, WWF has been protecting the future of nature. The world's
leading conservation organization, WWF works in 100 countries and is
supported by 1.2 million members in the United States and close to 5 million
globally.
</blockquote>
```

The <blockquote> tag specifies a section that is quoted from another source.

Browsers usually indent <blockquote> elements.

