HTML - Formatting

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

Bold Text

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

Example

Live Demo

```
<!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 ... element is displayed in italicized as shown below −

Example

Live Demo

```
<!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

Live Demo

```
<!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.

Strike Text

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

Example

Live Demo

```
<!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

Live Demo

```
<!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 ^{...} 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

Live Demo

```
<!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 superscript typeface.

Subscript Text

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

Example

Live Demo

```
<!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 _{subscript} typeface.

Inserted Text

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

Example

Live Demo

```
<!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~~ <u>wine</u>

Deleted Text

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

Example

Live Demo

```
<!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

Live Demo

```
<!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

Live Demo

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

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

Example

Live Demo

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

Example

Live Demo

```
<!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 alongwith CSS

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

HTML - Lists

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

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

HTML Unordered Lists

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

Example

Live Demo

```
<!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">

Live Demo

```
<!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"> −

Live Demo

```
<!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"> −

Live Demo

```
<!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 −

- 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 <ol> tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with <li>.

Example

Live Demo

```
<!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 −

1. Beetroot
2. Ginger
3. Potato
4. 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">

Live Demo

```
<!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">

Live Demo

```
<!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">

Live Demo

```
<!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" >

Live Demo

```
<!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">

Live Demo

```
<!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 −

```
<ol type = "1" start = "4">    - Numerals starts with 4.
<ol type = "I" start = "4">    - Numerals starts with IV.
<ol type = "i" start = "4">    - Numerals starts with iv.
<ol type = "a" start = "4">    - Letters starts with d.
<ol type = "A" start = "4">    - Letters starts with D.
```

Example

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

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Ordered List</title>
   </head>

   <body>
      <ol type = "i" start = "4">
         <li>Beetroot</li>
         <li>Ginger</li>
         <li>Potato</li>
         <li>Radish</li>
      </ol>
   </body>

</html>
```

This will produce the following result −

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

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

Live Demo

```
<!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 - 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

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Tables</title>
   </head>

   <body>
      <table border = "1">
         <tr>
            <td>Row 1, Column 1</td>
            <td>Row 1, Column 2</td>
         </tr>

         <tr>
            <td>Row 2, Column 1</td>
            <td>Row 2, Column 2</td>
         </tr>
      </table>

   </body>
</html>
```

This will produce the following result −

Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2

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

Live Demo

```
<!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>
```

This will produce the following result −

Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

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

Live Demo

```
<!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>
```

This will produce the following result −

Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

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

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Table Colspan/Rowspan</title>
   </head>

   <body>
      <table border = "1">
```

```
         <tr>
            <th>Column 1</th>
            <th>Column 2</th>
            <th>Column 3</th>
         </tr>
         <tr>
            <td rowspan = "2">Row 1 Cell 1</td>
            <td>Row 1 Cell 2</td>
            <td>Row 1 Cell 3</td>
         </tr>
         <tr>
            <td>Row 2 Cell 2</td>
            <td>Row 2 Cell 3</td>
         </tr>
         <tr>
            <td colspan = "3">Row 3 Cell 1</td>
         </tr>
      </table>
   </body>

</html>
```

This will produce the following result −

Column 1	Column 2	Column 3
Row 1 Cell 1	Row 1 Cell 2	Row 1 Cell 3
	Row 2 Cell 2	Row 2 Cell 3
Row 3 Cell 1		

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

Live Demo

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

```
   <head>
      <title>HTML Table Background</title>
   </head>

   <body>
      <table border = "1" bordercolor = "green" bgcolor = "yellow">
         <tr>
            <th>Column 1</th>
            <th>Column 2</th>
            <th>Column 3</th>
         </tr>
         <tr>
            <td rowspan = "2">Row 1 Cell 1</td>
            <td>Row 1 Cell 2</td>
            <td>Row 1 Cell 3</td>
         </tr>
         <tr>
            <td>Row 2 Cell 2</td>
            <td>Row 2 Cell 3</td>
         </tr>
         <tr>
            <td colspan = "3">Row 3 Cell 1</td>
         </tr>
      </table>
   </body>

</html>
```

This will produce the following result −

Column 1	Column 2	Column 3
Row 1 Cell 1	Row 1 Cell 2	Row 1 Cell 3
	Row 2 Cell 2	Row 2 Cell 3
Row 3 Cell 1		

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

Live Demo

```
<!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>
```

This will produce the following result. Here background image did not apply to table's header.

Column 1	Column 2	Column 3
Row 1 Cell 1	Row 1 Cell 2	Row 1 Cell 3
	Row 2 Cell 2	Row 2 Cell 3
Row 3 Cell 1		

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

Live Demo

```
<!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>
```

This will produce the following result −

Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2

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

Live Demo

```
<!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, columnn 2</td>
         </tr>

         <tr>
            <td>row 2, column 1</td><td>row 2, columnn 2</td>
         </tr>
      </table>
   </body>

</html>
```

This will produce the following result −

This is the caption

row 1, column 1	row 1, column 2
row 2, column 1	row 2, column 2

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

Live Demo

```
<!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>
```

This will produce the following result −

This is the head of the table			
Cell 1	Cell 2	Cell 3	Cell 4
This is the foot of the table			

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.

Live Demo

```
<!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>
```

This will produce the following result −

Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

HTML - Frames

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

Disadvantages of Frames

There are few drawbacks with using frames, so it's never recommended to use frames in your webpages −

- **Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.**
- **Sometimes your page will be displayed differently on different computers due to different screen resolution.**
- **The browser's *back* button might not work as the user hopes.**
- **There are still few browsers that do not support frame technology.**

Creating Frames

To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines, how to divide the window into frames. The rows attribute of <frameset> tag defines horizontal frames and cols attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

> **Note − The <frame> tag deprecated in HTML5. Do not use this element.**

Example

Following is the example to create three horizontal frames −

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Frames</title>
   </head>

   <frameset rows = "10%,80%,10%">
      <frame name = "top" src = "/html/top_frame.htm" />
      <frame name = "main" src = "/html/main_frame.htm" />
      <frame name = "bottom" src = "/html/bottom frame.htm" />
```

```
      <noframes>
         <body>Your browser does not support frames.</body>
      </noframes>

   </frameset>

</html>
```

This will produce the following result −

Example

Let's put the above example as follows, here we replaced rows attribute by cols and changed their width. This will create all the three frames vertically −

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Frames</title>
   </head>

   <frameset cols = "25%,50%,25%">
```

```
      <frame name = "left" src = "/html/top_frame.htm" />
      <frame name = "center" src = "/html/main_frame.htm" />
      <frame name = "right" src = "/html/bottom_frame.htm" />

      <noframes>
         <body>Your browser does not support frames.</body>
      </noframes>
   </frameset>


</html>
```

This will produce the following result −

Top Frame	Main Frame	Bottom Frame

The <frameset> Tag Attributes

Following are important attributes of the <frameset> tag −

Sr.No	Attribute & Description
1	**cols** Specifies how many columns are contained in the frameset and the size of each column. You can specify the width of each column in one of the four ways − Absolute values in pixels. For example, to create three vertical frames, use *cols = "100, 500, 100"*. A percentage of the browser window. For example, to create three vertical frames, use *cols = "10%, 80%, 10%"*. Using a wildcard symbol. For example, to create three vertical frames, use *cols = "10%, *, 10%"*. In this case wildcard takes remainder of the window. As relative widths of the browser window. For example, to create three vertical frames, use *cols = "3*, 2*, 1*"*. This is an alternative to percentages. You can use relative widths of the browser window. Here the window is divided into sixths: the first column takes up half of the window, the second takes one third, and the third takes one sixth.
2	**rows** This attribute works just like the cols attribute and takes the same values, but it is used to specify the rows in the frameset. For example, to create two horizontal frames, use *rows = "10%, 90%"*. You can specify the height of each row in the same way as explained above for columns.
3	**border** This attribute specifies the width of the border of each frame in pixels. For example, border = "5". A value of zero means no border.
4	**frameborder** This attribute specifies whether a three-dimensional border should be displayed between frames. This attribute takes value either 1 (yes) or 0 (no). For example frameborder = "0" specifies no border.
5	**framespacing** This attribute specifies the amount of space between frames in a frameset. This can take any integer value. For example framespacing = "10" means there should be 10 pixels spacing between each frames.

The <frame> Tag Attributes

Following are the important attributes of <frame> tag −

Sr.No	Attribute & Description
1	**src** **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	**name** **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 an another frame, in which case the second frame needs a name to identify itself as the target of the link.**
3	**frameborder** **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	**marginwidth** **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	**marginheight** **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	**noresize** **By default, you can resize any frame by clicking and dragging on the borders of a frame. The noresize attribute prevents a user from being able to resize the frame. For example noresize = "noresize".**
7	**scrolling** **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	**longdesc** **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"**

Browser Support for Frames

If a user is using any old browser or any browser, which does not support frames then <noframes> element should be displayed to the user.

So you must place a <body> element inside the <noframes> element because the <frameset> element is supposed to replace the <body> element, but if a browser does not understand <frameset> element then it should understand what is inside the <body> element which is contained in a <noframes> element.

You can put some nice message for your user having old browsers. For example, *Sorry!! your browser does not support frames.* as shown in the above example.

Frame's name and target attributes

One of the most popular uses of frames is to place navigation bars in one frame and then load main pages into a separate frame.

Let's see following example where a test.htm file has following code −

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Target Frames</title>
   </head>

   <frameset cols = "200, *">
      <frame src = "/html/menu.htm" name = "menu_page" />
      <frame src = "/html/main.htm" name = "main page" />


      <noframes>
         <body>Your browser does not support frames.</body>
      </noframes>
   </frameset>

</html>
```

Here, we have created two columns to fill with two frames. The first frame is 200 pixels wide and will contain the navigation menu bar implemented by menu.htm file. The second column fills in remaining space and will contain the main part of the page and it is implemented by main.htm file. For all the three links available in menu bar, we have mentioned target frame as main_page, so whenever you click any of the links in menu bar, available link will open in main page.

Following is the content of menu.htm file

Live Demo

```
<!DOCTYPE html>
<html>

   <body bgcolor = "#4a7d49">
      <a href = "http://www google com" target = "main page">Google</a>
```

```
      <br />
      <br />

      <a href = "http://www.microsoft.com" target = "main_page">Microsoft</a>

      <br />
      <br />

      <a href = "http://news.bbc.co.uk" target = "main_page">BBC News</a>
   </body>

</html>
```

Following is the content of main.htm file −

Live Demo

```
<!DOCTYPE html>
<html>

   <body bgcolor = "#b5dcb3">
      <h3>This is main page and content from any link will be displayed here.</h3>
      <p>So now click any link and see the result.</p>
   </body>

</html>
```

When we load test.htm file, it produces following result −

Google

Microsoft

BBC News

This is main page and content from any link will be displayed here.

So now click any link and see the result.

Now you can try to click links available in the left panel and see the result. The *targetattribute* can also take one of the following values −

Sr.No	Option & Description
1	**_self** **Loads the page into the current frame.**
2	**_blank** **Loads a page into a new browser window. Opening a new window.**
3	**_parent** **Loads the page into the parent window, which in the case of a single frameset is the main browser window.**
4	**_top** **Loads the page into the browser window, replacing any current frames.**
5	**targetframe** **Loads the page into a named targetframe.**

HTML - Images

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

Insert Image

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

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

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

Example

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

Live Demo

```
<!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 −

Simple Image Insert

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 −

Live Demo

```
<!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 −

Simple Image Insert

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

Live Demo

```
<!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 = "
   </body>

</html>
```

This will produce the following result −

Setting image width and height

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

Live Demo

```
<!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>
```

This will produce the following result −

Setting image Border

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

Live Demo

```
<!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 = "ri
   </body>

</html>
```

This will produce the following result −

Setting image Alignment

Free Web Graphics

For Free Web Graphics including patterns you can look into Free Web Graphics

HTML - Embed 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 *autostart* attributes deprecated in HTML5. Do not use these attributes.**

Sr.No	Attribute & Description
1	**align** Determines how to align the object. It can be set to either center, *left or right*.
2	**autostart** This boolean attribute indicates if the media should start automatically. You can set it either true or false.
3	**loop** 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	**playcount** Specifies the number of times to play the sound. This is alternate option for *loop* if you are usiong IE.
5	**hidden** Specifies if the multimedia object should be shown on the page. A false value means no and true values means yes.
6	**width** Width of the object in pixels
7	**height** Height of the object in pixels
8	**name** A name used to reference the object.
9	**src** URL of the object to be embedded.
10	**volume** 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.**

Live Demo

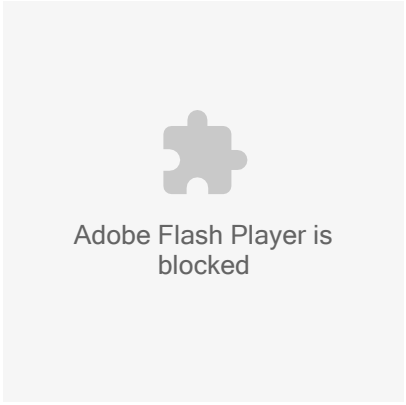
```
<!DOCTYPE html>
<html>

   <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>
```

This will produce the following result −



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

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 −

Live Demo

```
<!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	**type** **Indicates the type of input control and for text input control it will be set to text.**
2	**name** **Used to give a name to the control which is sent to the server to be recognized and get the value.**
3	**value** **This can be used to provide an initial value inside the control.**
4	**size** **Allows to specify the width of the text-input control in terms of characters.**
5	**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 −

Live Demo

```
<!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 :
Password:

Attributes

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

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

Live Demo

```
<!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:

Enter description here...

Attributes

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

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

Live Demo

```
<!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 −

☐ Maths ☐ Physics

Attributes

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

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for checkbox input control it will be set to checkbox..
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** The value that will be used if the checkbox is selected.
4	**checked** Set to *checked* if you want to select it by default.

Radio Button Control

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

Example

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

Live Demo

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

Select Box Control

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

Example

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

Live Demo

```
<!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	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
2	**size** This can be used to present a scrolling list box.
3	**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 −

Sr.No	Attribute & Description
1	**value** The value that will be used if an option in the select box box is selected.
2	**selected** Specifies that this option should be the initially selected value when the page loads.
3	**label** An alternative way of labeling options

File Upload Box

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

Example

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

Live Demo

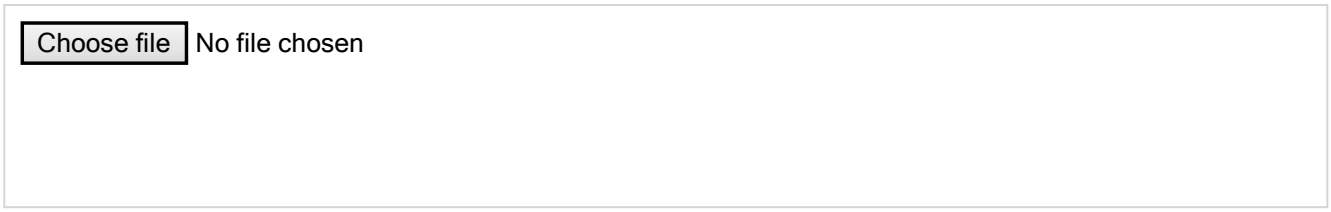
```
<!DOCTYPE html>
<html>

   <head>
      <title>File Upload Box</title>
   </head>

   <body>
      <form>
         <input type = "file" name = "fileupload" accept = "image/*" />
      </form>
   </body>

</html>
```

This will produce the following result −



Attributes

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

Sr.No	Attribute & Description
1	**name** **Used to give a name to the control which is sent to the server to be recognized and get the value.**
2	**accept** **Specifies the types of files that the server accepts.**

Button Controls

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

Sr.No	Type & Description
1	**submit** This creates a button that automatically submits a form.
2	**reset** This creates a button that automatically resets form controls to their initial values.
3	**button** This creates a button that is used to trigger a client-side script when the user clicks that button.
4	**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 −

Live Demo

```
<!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>
```

This will produce the following result −

Submit Reset OK

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 −

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>File Upload Box</title>
   </head>

   <body>
      <form>

         <p>This is page 10</p>
         <input type = "hidden" name = "pagename" value = "10" />
         <input type = "submit" name = "submit" value = "Submit" />
         <input type = "reset" name = "reset"  value = "Reset" />
      </form>
   </body>

</html>
```

This will produce the following result −

This is page 10

Submit Reset