

UNIT 2: Hyper Text Markup Language

HTML

HTML stands for **hypertext markup language**. It is not a programming language. A markup language specifies the *layout and style* of a document. A markup language consists of a set of **markup tags**. HTML uses markup tags to describe web pages. An HTML tag is commonly defined as a set of characters constituting a formatted command for a Web page. At the core of HTML, tags provide the directions or recipes for the visual content that one sees on the Web. HTML tags are keywords surrounded by **angle brackets** like `<html>`. Most HTML tags normally come in pairs like `` and ``. The first tag is called the **start tag** (or **opening tag**) and the second tag is called the **end tag** (or **closing tag**). HTML documents describe Web pages. HTML documents contain HTML tags and plain text. HTML documents are also called Web pages. A web browser reads HTML documents and displays them as Web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. A simple HTML document is given below:

```
<html>
  <head>
    <title>This is my first web page</title>
  </head>
  <body>
    <h1>My first heading</h1>
    <p>My first paragraph</p>
  </body>
</html>
```

Save this page with **.html** or **.htm** extension. However, it is good practice to use **.htm** extension.

HTML Elements

An HTML element is an individual component of an HTML document or web page. HTML is composed of a tree of HTML nodes, such as text nodes. Each node can have HTML attributes specified. Nodes can also have content, including other nodes and text. Many HTML nodes represent semantics, or meaning. For example, the `<title>` node represents the title of the document. An HTML element is everything from the start tag to the end tag. For example, `<p>My first paragraph</p>`. An HTML element consists of start tag, end tag, and element content. The element content is everything between the start tag and end tag. Empty elements are closed in the start tag. Most HTML elements can have attributes. For example, `src` attribute of `img` tag.

HTML Attributes

HTML attributes are special words used inside the opening tag to control the element's behavior. HTML attributes are a modifier of an HTML element type attributes provide additional information about HTML elements. Attributes are always specified in the start tag. Attributes come in name/value pair like name = "value". For example, HTML links are defined with <a> tag and the link address is provided as an attribute href like cdcsit.

Note: Always quote attribute values and use lowercase attributes.

HTML Headings

HTML headings are defined with the <h1> to <h6> tags. <h1> displays largest text and <h6> smallest.

For example,

```
<html>
  <h1>This is heading with h1 tag</h1>
  <h2>This is heading with h2 tag</h2>
  <h3>This is heading with h3 tag</h3>
  <h4>This is heading with h4 tag</h4>
  <h5>This is heading with h5 tag</h5>
  <h6>This is heading with h6 tag</h6>
</html>
```

Output:

This is heading with h1 tag

This is heading with h2 tag

This is heading with h3 tag

This is heading with h4 tag

This is heading with h5 tag

This is heading with h6 tag

HTML Paragraphs

HTML paragraphs are defined with <p> tag. Paragraphs are separated by new line.

For example:

```
<html>
<body>
  <p>This is a paragraph.</p>
  <p>This is a paragraph.</p>
  <p>This is a paragraph.</p>
</body>
</html>
```

Output:

This is a paragraph.

This is a paragraph.

This is a paragraph.

HTML Rules (Lines)

We use <hr /> tag to create horizontal line.

HTML Comments

We use comments to make our HTML code more readable and understandable. Comments are ignored by the browser and are not displayed. Comments are written between <!-- and -->.

For example

```
<html>
<body>
  <p>Comments are not displayed in browser</p>
  <!--comments are used to make code understandable and readable-->
</body>
</html>
```

Output:

← → ↻ ⓘ File | C:/Users/dell/Desktop/comments.html

Comments are not displayed in browser

HTML Line Breaks

If you want a new line (line break) without starting a new paragraph, use `
` tag.

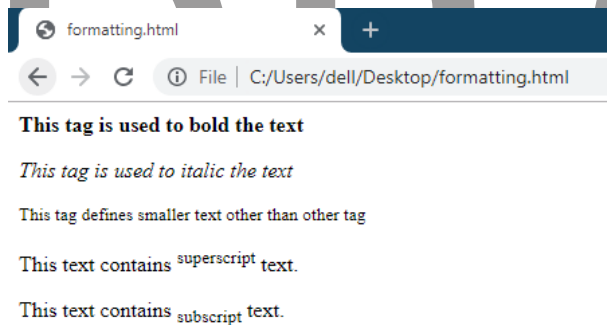
HTML Formatting Tags

We use different tags for formatting output, like ``, `<i>`, `<small>`, `<sup>`, `<sub>` illustrated in example.

For example:

```
<html>
<body>
  <p><b>This tag is used to bold the text</b></p>
  <p><i>This tag is used to italic the text</i></p>
  <p><small>This tag defines smaller text other than other tag</small></p>
  <p>This text contains <sup>superscript</sup> text.</p>
  <p>This text contains <sub>subscript</sub> text.</p>
</body>
</html>
```

Output:



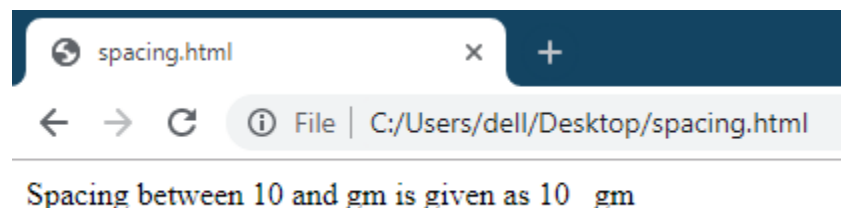
HTML Spacing

A common character entity used in HTML is the non-breaking space. i.e: ** **; Two words separated by a non-breaking space will stick together (not break into a new line)

For example:

```
<html>
<body>
  Spacing between 10 and gm is given as 10 &nbsp; gm
</body>
</html>
```

Output:



DRAFT

HTML Formatting text

The following are the tags used in formatting the text like: <pre>,
, tags , , <tt>, <a> and . The <pre> tag is used to defines preformatted text. Text in a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks. The
 tag inserts a single line break as illustrated in example 1

The tag is used to group inline-elements in a document. The tag provides no visual change by itself. The tag provides a way to add a hook to a part of a text or a part of a document. The tag gives text a strong emphasis which traditionally means that the text is displayed as bold by the browser. The <tt> tag defines teletype text which is not supported in HTML 5 and onwards. The <a> tag defines hyperlink, which is used to link from one page to another. The tag is the image element tag defines an image in an HTML page, which requires two attributes such as src and alt. The src attributes specifies the URL of an image and the alt attribute specifies an alternate text for an image as illustrated in the example 2

For example 1:

```
<html>
<body>
  <pre>
    Text in a pre element
    is displayed in a fixed-width
    font, and it preserves
    both   spaces and
    line breaks
  </pre>
  This line has br <br> tag.
</body>
</html>
```

Output:

```
← → ↺ ⓘ File | C:/Users/dell/Desktop/spacing.html

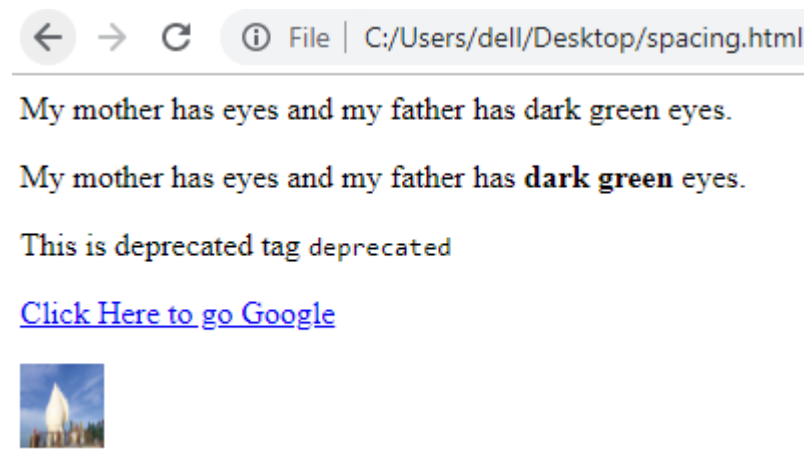
Text in a pre element
is displayed in a fixed-width
font, and it preserves
both   spaces and
line breaks

This line has br
tag.
```

Example 2:

```
<html>
<body>
  <p>My mother has eyes and my father has <span>dark green</span> eyes.</p>
  <p>My mother has eyes and my father has <strong>dark green</strong> eyes.</p>
  <p>This is deprecated tag <tt>deprecated</tt></p>
  <p><a href="#">Click Here to go Google</a></p>
  
</body>
</html>
```

Output:



HTML List

List is the collection of items or elements. In HTML list are categorized as:

- a. Unordered List
- b. Ordered List
- c. Description List

Unordered List

The list which are not ordered by numbers are known as unordered list. An unordered list starts with the `` tag. Each list item starts with the `` tag. The list items will be marked with bullets by defaults however it can be changed. To change the bullets an attribute type is added. The value of type are disc, circle, square, none. Like `<ul type="circle">`. The example following shows the unordered list

```
<html>
<body>

<h2>An unordered HTML list</h2>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

</body>
</html>
```

Output:

An unordered HTML list

- Coffee
- Tea
- Milk

Ordered List

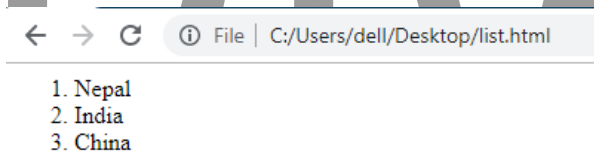
The list which are ordered by numbers are known as ordered list. An ordered list starts with the `` tag. Each list item starts with the `` tag. The list items will be marked with numbers by default. The ordered list has the type attribute of the `` tag, defines the type of the list item marker. The following table shows the type attribute with the descriptions

Type	Description
<code>type="1"</code>	The list items will be numbered with numbers (default)
<code>type="A"</code>	The list items will be numbered with uppercase letters
<code>type="a"</code>	The list items will be numbered with lowercase letters
<code>type="i"</code>	The list items will be numbered with lowercase roman numbers
<code>type="I"</code>	The list items will be numbered with uppercase roman numbers

Example: List starting with numbers

```
<html>
<body>
  <ol type="1">
    <li>Nepal</li>
    <li>India</li>
    <li>China</li>
  </ol>
</body>
</html>
```

Output:



Description List

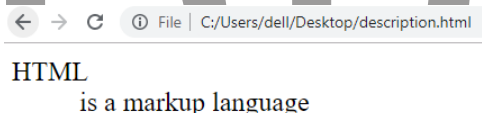
Definition List displays elements in definition form like in dictionary. The <dl>, <dt> and <dd> tags are used to define description list. The 3 HTML description list tags are given below:

1. **<dl> tag** defines the description list.
2. **<dt> tag** defines data term.
3. **<dd> tag** defines data definition (description).

Example of description list

```
<html>
<body>
  <dl>
    <dt>HTML</dt>
    <dd>is a markup language</dd>
  </dl>
</body>
</html>
```

Output:



HTML
is a markup language

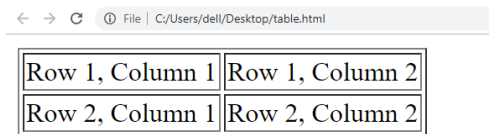
HTML Tables

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:

```
<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>
```

Output:



A screenshot of a web browser window. The address bar shows the file path C:/Users/dell/Desktop/table.html. The browser displays a simple table with two rows and two columns. Each cell contains text describing its position: 'Row 1, Column 1', 'Row 1, Column 2', 'Row 2, Column 1', and 'Row 2, Column 2'. The table has a visible border.

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".

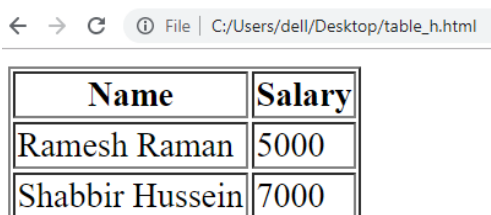
Table Heading

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 top row is considered as table heading as shown below, otherwise <th> element is used in any row. Headings, which are defined in <th> tag are centered and bold by default.

Example:

```
<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>
```

Output:



A screenshot of a web browser window. The address bar shows the file path C:/Users/dell/Desktop/table_h.html. The browser displays a table with two columns: 'Name' and 'Salary'. The first row contains the data 'Ramesh Raman' and '5000'. The second row contains the data 'Shabbir Hussein' and '7000'. The headings 'Name' and 'Salary' are bold and centered. The table has a visible border.

Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

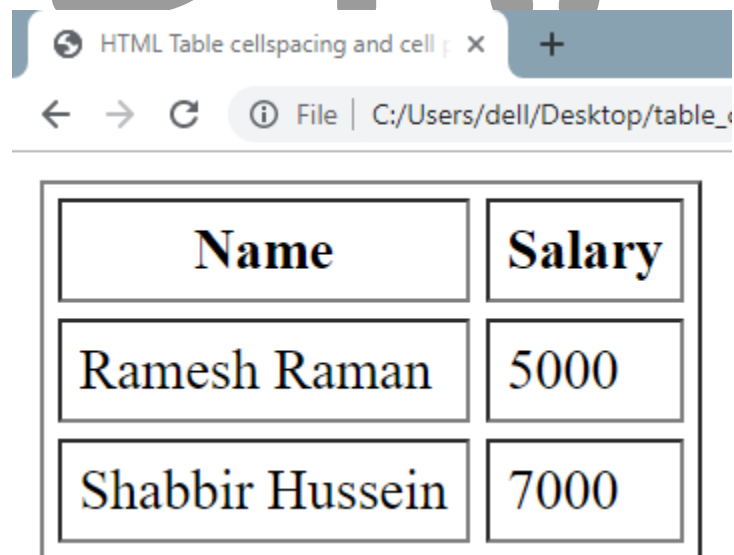
Cellpadding and Cellspacing Attributes

There are two attributes called cellpadding and cellspacing which is used to adjust the white space in the 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:

```
<html>
<head>
  <title>HTML Table cellspacing and cell padding</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>
```

Output:



Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

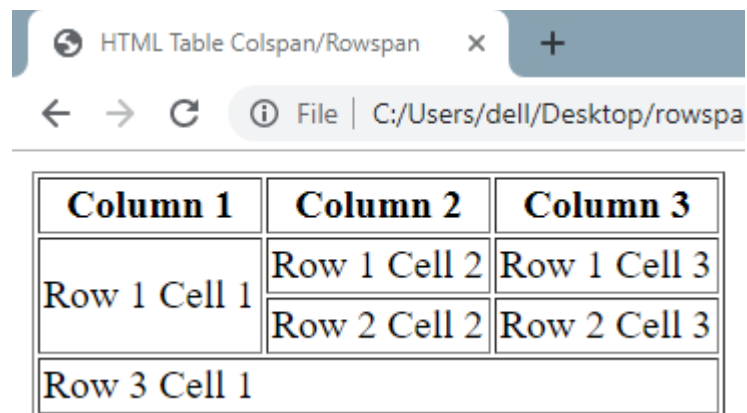
Colspan and Rowspan Attributes

The colspan attribute is used to merge two or more columns into a single column. Similar way the rowspan is used to merge two or more rows.

Example:

```
<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>
```

Output:



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

Table background can be set using one of the following two ways –

- bgcolor attribute – Used to set background color for whole table or just for one cell.
- background attribute – Used to set background image for whole table or just for one cell.

To set border color also using bordercolor attribute. Like <table border = "1" bordercolor = "green" bgcolor = "yellow">

Table Height and Width

Table width and height can be adjusted using width and height attributes. The table width or height in terms of pixels or in terms of percentage of available screen area can be specified by height and width attributes.

Like: <table border = "1" width = "400" height = "150">

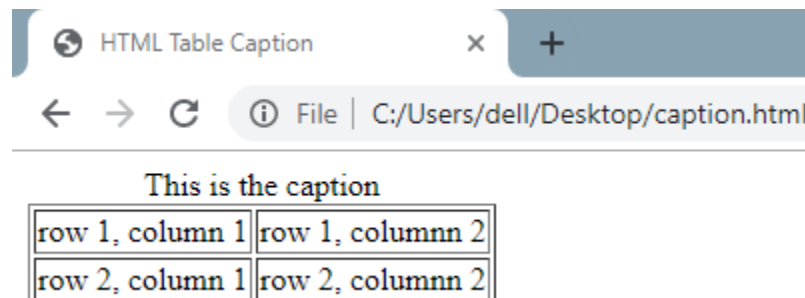
Table Caption

The caption tag will serve as a title or explanation for the table and it shows up at the top of the table.

Example:

```
<html>
<head>
  <title>HTML Table Caption</title>
</head>
<body>
  <table border = "1">
    <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>
```

Output:



row 1, column 1	row 1, columnn 2
row 2, column 1	row 2, columnn 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:

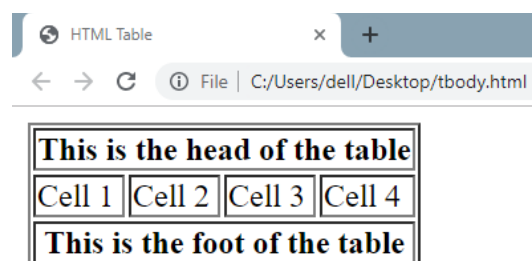
- `<thead>` – to create a separate table header.
- `<tbody>` – to indicate the main body of the table.
- `<tfoot>` – to create a separate table footer.

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:

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

Output:



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

HTML Frames

HTML frames are used to divide the 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. The disadvantages of using frames are:

- 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 `<frameset>` tag instead of `<body>` tag is used. 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.

Important: You cannot use the `<body></body>` tags together with the `<frameset></frameset>` tags. However, if you add a `<noframes>` tag containing some text for browsers that do not support frames, you will have to enclose the text in `<body></body>` tags.

Example:

```
<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>
```

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>
  //input elements
</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.

Syntax:

```
<form action="index.html" method="post">
</form>
```


Difference between GET and POST

GET	POST
Parameters remain in browser history because they are part of the URL	Parameters are not saved in browser history.
Can be bookmarked.	Can not be bookmarked.
GET requests are re-executed but may not be re-submitted to server if the HTML is stored in the browser cache.	The browser usually alerts the user that data will need to be re-submitted.
application/x-www-form-urlencoded	multipart/form-data or application/x-www-form-urlencoded Use multipart encoding for binary data.
Can send but the parameter data is limited to what we can stuff into the request line (URL). Safest to use less than 2K of parameters, some servers handle up to 64K	Can send parameters, including uploading files, to the server.
Easier to hack for script kiddies	More difficult to hack
Yes, only ASCII characters allowed.	No restrictions. Binary data is also allowed.
GET is less secure compared to POST because data sent is part of the URL. So it's saved in browser history and server logs in plaintext.	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs.
Yes, since form data is in the URL and URL length is restricted. A safe URL length limit is often 2048 characters but varies by browser and web server.	No restrictions
GET method should not be used when sending passwords or other sensitive information.	POST method used when sending passwords or other sensitive information.
GET method is visible to everyone (it will be displayed in the browser's address bar) and has limits on the amount of information to send.	POST method variables are not displayed in the URL.
Data can be cached	Data cannot be cached

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.

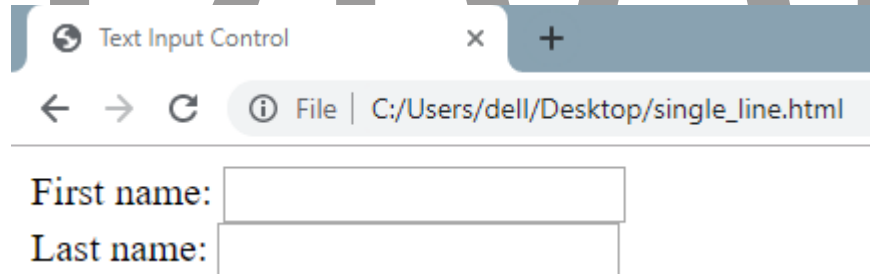
Single-line text input controls

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:

```
<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>
```

Output:



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.

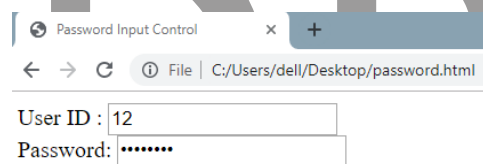
Password input controls

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:

```
<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>
```

Output:



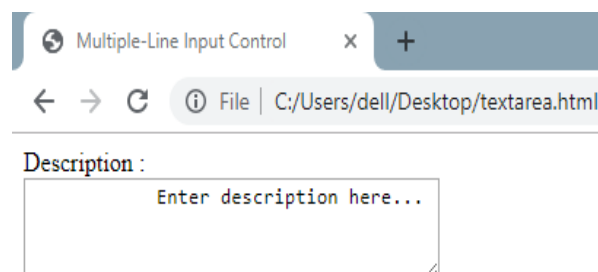
Multiple-Line Text Input Controls

The most commonly used form tag is <input> tag. The type of input is specified with the type attribute within the <input> tag.

Example:

```
<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>
```

Output:



Attributes used in multiple line

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

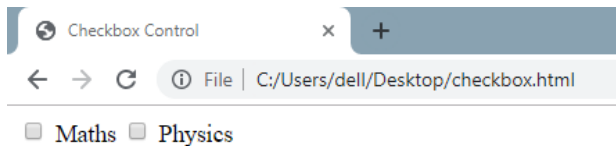
Checkbox Control

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:

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

Output:



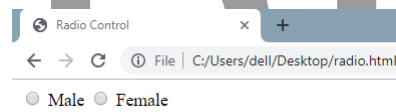
Radio Button Control

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. The name attribute in radio button control must be same in every input tag which is shown in example below

Example:

```
<html>
<head>
  <title>Radio Control</title>
</head>
<body>
  <form>
    <input type = "radio" name = "gender" value = "male"> Male
    <input type = "radio" name = "gender" value = "female"> Female
  </form>
</body>
</html>
```

Output:



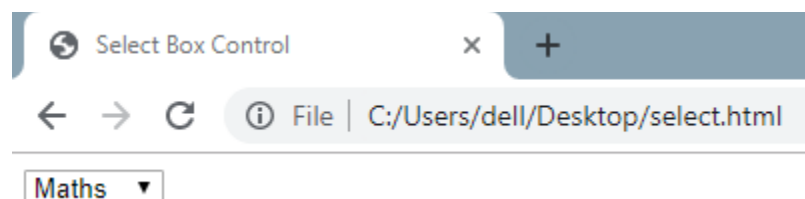
Select Box Control

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:

```
<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>
```

Output:



DRAFT

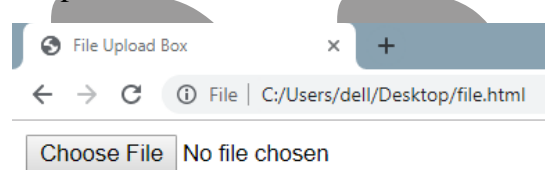
File Upload Box

To upload a file to in the web site file upload box is needed, also known as a file select box. This is also created using the `<input>` element but type attribute is set to file.

Example:

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

Output:



Button Controls

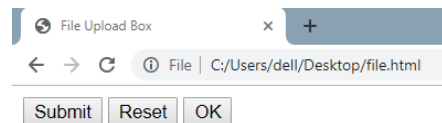
There are various ways in HTML to create clickable buttons. You can also create a clickable button using `<input>` tag by setting its type attribute to button. The type attribute can take the following values:

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

```
<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" />
  </form>
</body>
</html>
```

Output:



When the user clicks on the "Submit or Ok" button, the content of the form is sent to the server. The form's **action attribute** defines the name of the file to send the content to. The file defined in the action attribute usually does something with the received input. For example,

```
<form name="input" action="submit.php" method="get">
```

Username:

```
<input type="text" name="user" />
```

```
<input type="submit" value="Submit" />
```

```
</form>
```

If you type some characters in the text field above, and click the "Submit" button, the browser will send your input to a page called "submit.php". The page will show you the received input.

The **Reset** button resets the input field.

The **method** attribute of `<form>` tag specifies how to send form-data (the form-data is sent to the page specified in the action attribute). We can use **“get”** and **“post”** as values of method attribute. When we use **get**, form-data can be sent as URL variables and when we use **post**, form-data are sent as HTTP post.

Notes on the "get" method:

- This method appends the form-data to the URL in name/value pairs
- There is a limit to how much data you can place in a URL (varies between browsers), therefore, you cannot be sure that all of the form-data will be correctly transferred
- Never use the "get" method to pass sensitive information! (password or other sensitive information will be visible in the browser's address bar)

Notes on the "post" method:

- This method sends the form-data as an HTTP post transaction□
- The "post" method is more robust and secure than "get", and "post" does not have size limitations

ID attributes of HTML elements

the ID attribute in HTML is a unique identifier for the element. It provides a way to identify an area of a web page for CSS styles, anchor links, and targets for scripts.

What is the ID attribute used for?

The ID attribute can perform several actions for web pages:

- **A style sheet selector:** This is the function most people use the ID attribute for. Because they are unique, you can be sure you'll be styling just the one item on your web page when you style using an ID property. The downside to using an ID for styling purposes is that it has a very high level of specificity, which can make it very challenging if you need to override a style for some reason later in a stylesheet. Because of this, current web practices lean toward using classes and class selectors in place of IDs and ID selectors for general styling purposes.
- **Named anchors for linking to:** web browsers allow you to target precise locations in your web documents by pointing to the ID at the end of the URL. You simply add the id to the end of the page URL, preceded by a pound-sign (#). You can also link to these anchors with the page itself by adding the pound-sign (#) and the ID name in the href attribute for the element. For instance, if you have a division with an ID of contact, you could link to it on that page with #contact.
- **A reference for scripts:** If you write any Javascript functions, you will want to use the ID attribute so that you can make changes to the precise element on the page with your scripts.
- **Other processing:** The id allows you to process your web documents in whatever way you need to. For example, you might extract the HTML into a database, and the ID attribute identifies fields.

Rules for using the ID attribute

There are a few rules you must follow to have a valid document that uses the id attribute anywhere in the document:

- The ID must start with a letter (a-z or A-Z)
- All subsequent characters can be letters, numbers (0-9), hyphens (-), underscores (_), colons (:), and periods (.).
- Each ID must be unique within the document.

Syntax:

```
<element id="id_name">
```

The element may be of any types of HTML elements discussed previous

Class Attributes of HTML Elements

The class attribute specifies one or more class names for an element. The class attribute is mostly used to point to a class in a style sheet. However, it can also be used by a JavaScript (via the HTML DOM) to make changes to HTML elements with a specified class.

Syntax:

```
<element class="class_name">
```

The element may be of any types of HTML elements discussed previous

HTML Color

HTML colors are displayed using RED, GREEN, and BLUE light. Colors are defined using hexadecimal (hex) notation for combination of red, green, and blue color values (RGB). The lowest value that can be given to one of the light sources is 0 (hex 00) and the highest values is 255 (hex FF). We can use HEX (e.g. **#2000FF**) as well as RGB (e.g. **rgb(32, 0, 255)**) values to define different colors.

The combination of Red, Green and Blue values from 0 to 255 gives a total of more than 16 million different colors to play with (256 x 256 x 256).

We can also use color names instead of hex and rgb values. The World Wide Web Consortium (W3C) has listed 16 valid color names for HTML and CSS: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow. Some examples are given below:

```
<body style = "background:rgb(12, 32, 255)">
```

```
<body style = "background:#0008FF">
```

```
<body style = "background:red">
```

HTML Fonts

The **** tag in HTML is deprecated. It is supposed to be removed in a future version of HTML. For example,

```
<p>
```

```
<font size="2" face="Verdana" color = "red">
```

This is a paragraph.

```
</font>
```

```
</p>
```

HTML Character Entities

Character entities are replaced with reserved characters. A character entity looks *&entity_name* OR *&#entity_number*. Some commonly used character entities are:

Result	Description	Entity Name	Entity Number
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	Ampersand	&	&
¢	Cent	¢	¢
£	Pound	£	£
¥	Yen	¥	¥
€	Euro	€	€
©	Copyright	©	©
®	registered trademark	®	®

HTML Head

The head element contains general information, also called meta-information, about a document. The elements inside the head element should not be displayed by a browser. According to the HTML standard, only a few tags are legal inside the head section. These are: <base>, <link>, <meta>, <title>, <style>, and <script>. You must use this element and it should be used just once. It must start immediately after the opening <html> tag and end directly before the opening <body> tag.

HTML Meta

HTML includes a meta element that goes inside the head element. The purpose of the meta element is to provide meta-information about the document. Meta elements are purely used for search engine's use and to provide some additional information about your pages. We use three attributes (**name**, **content**, and **http-equiv**) with <meta> tag.

We use **name** = “**keywords**” to provide information for a search engine. If the keywords you have chosen are the same as the ones they have put in, you come up in the search engine's result pages. For example,

```
<meta name="keywords" content="HTML, DHTML, CSS, XML, XHTML, JavaScript" />
```

We use **name** = “**description**” to define a description of your page. It is sort summary of the content of the page. Depending on the search engine, this will be displayed along with the title of your page in an index. For example,

```
<meta name="description" content="Free Web tutorials on HTML, CSS, XML, and XHTML" />
```

We use **name** = “**generator**” to define a description for the program you used to write your pages. For example,

```
<meta name="generator" content="Homesite 4.5" />
```

We use **name** = “**author**” and **name** = “**copyright**” for author and copyright details. For example,

```
<meta name="author" content="W3schools" />
```

```
<meta name="copyright" content="W3schools 2005" />
```

We use **name** = “**expires**” to give the browsers a data, after which the page is deleted from the browsers cache, and must be downloaded again. This is useful if you want to make sure your visitors are reading the most current version of a page. For example,

```
<meta name="expires" content="13 July 2008" />
```

We use **http-equiv** = “**expires**” to refresh itself to the most current version or change to another location (page) entirely after some time. This is useful if you've moved a page to a new url and want any visitors to the old address to be quietly sent to the new location. For example,

```
<meta http-equiv = "refresh" content="5;url=http://www.tu.edu.np" />
```

Here, the number is the number of seconds to wait before changing to the new page.

Setting it to 0 results in an instant redirect.

HTML Audio

Since the release of HTML5, audios can be added to webpages using the “audio” tag. Previously audios could be only played on webpages using web plugins like Flash. The “audio” tag is an inline element which is used to embed sound files into a web page. It is a very useful tag if you want to add audio such as songs, interviews, etc on your webpage.

Syntax:

```
<audio>
  <source src="sample.mp3" type="audio/mpeg">
</audio>
```

Attributes

The various attributes that can be used with the “audio” tag are listed below :

- Controls : Designates what controls to display with the audio player.
- Autoplay : Designates that the audio file will play immediately after it loads controls.
- Loop : Designates that the audio file should continuously repeat.
- src : Designates the URL of the audio file.
- muted : Designates that the audio file should be muted.

Supported

Formats

Three formats mp3, ogg, wav are supported by HTML5

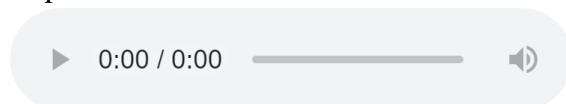
Example

```
<html>
<body>

<audio controls>
  <source src="national.ogg" type="audio/ogg">
  <source src="song.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>

</body>
</html>
```

Output:



HTML Video

The HTML5 <video> element specifies a standard way to embed a video in a web page.

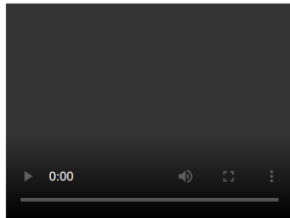
Example:

```
<html>
<body>

<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>

</body>
</html>
```

Output:



HTML Canvas

The HTML `<canvas>` element is used to draw graphics on a web page. The graphic to the left is created with `<canvas>`. It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.

What is HTML Canvas?

The HTML `<canvas>` element is used to draw graphics, on the fly, via JavaScript. The `<canvas>` element is only a container for graphics. You must use JavaScript to actually draw the graphics. Canvas has several methods for drawing paths, boxes, circles, text, and adding images. A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

Syntax:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

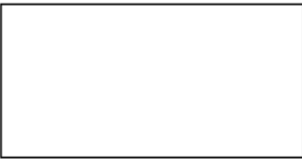
Example:

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

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">
Your browser does not support the HTML5 canvas tag.
</canvas>

</body>
</html>
```

Output:



Example 2: Drawing a circle using canvas

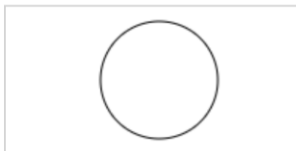
```
<html>
<body>

<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>
```

Output:



HTML Main

The <main> tag specifies the main content of a document. The content inside the <main> element should be unique to the document. It should not contain any content that is repeated across documents such as sidebars, navigation links, copyright information, site logos, and search forms.

Syntax:

```
<main>
//statements
</main>
```

Example:

```
<html>
<body>

<main>
  <h1>Web Browsers</h1>
  <p>Google Chrome, Firefox, and Internet Explorer are the most used browsers today.</p>
  <article>
    <h1>Google Chrome</h1>
    <p>Google Chrome is a free, open-source web browser developed by Google, released in 2008.</p>
  </article>
  <article>
    <h1>Internet Explorer</h1>
    <p>Internet Explorer is a free web browser from Microsoft, released in 1995.</p>
  </article>
  <article>
    <h1>Mozilla Firefox</h1>
    <p>Firefox is a free, open-source web browser from Mozilla, released in 2004.</p>
  </article>
</main>
<p><strong>Note:</strong> The main tag is not supported in Internet Explorer 11 and earlier versions.</p>
</body>
</html>
```

Output:

Web Browsers

Google Chrome, Firefox, and Internet Explorer are the most used browsers today.

Google Chrome

Google Chrome is a free, open-source web browser developed by Google, released in 2008.

Internet Explorer

Internet Explorer is a free web browser from Microsoft, released in 1995.

Mozilla Firefox

Firefox is a free, open-source web browser from Mozilla, released in 2004.

Note: The main tag is not supported in Internet Explorer 11 and earlier versions.

HTML section

The <section> tag defines sections in a document, such as chapters, headers, footers, or any other sections of the document.

Syntax:

```
<section>
//statements
</section>
```

Example:

```
<html>
<body>
<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was founded in 1961.</p>
</section>
<p><strong>Note:</strong> The section tag is not supported in Internet Explorer 8 and earlier versions.</p>
</body>
</html>
```

Output:

WWF

The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was founded in 1961.

Note: The section tag is not supported in Internet Explorer 8 and earlier versions.

HTML Article

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.

Syntax:

```
<article>
//statements
</article>
```

Example:

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

```

<article>
  <h1>Google Chrome</h1>
  <p>Google Chrome is a free, open-
source web browser developed by Google, released in 2008.</p>
</article>
<p><strong>Note:</strong> The article tag is not supported in Internet Explorer 8
and earlier versions.</p>
</body>
</html>

```

Output:

Google Chrome

Google Chrome is a free, open-source web browser developed by Google, released in 2008.

Note: The article tag is not supported in Internet Explorer 8 and earlier versions.

HTML Header

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

- one or more heading elements (<h1> - <h6>)
- logo or icon
- authorship information

You can have several <header> elements in one document.

Example:

```

<html>
<body>
<article>
  <header>
    <h1>Most important heading here</h1>
    <h3>Less important heading here</h3>
    <p>Some additional information here.</p>
  </header>
  <p>This is the heading paragraph</p>
</article>
</body>
</html>

```

Output:

Most important heading here

Less important heading here

Some additional information here.

This is the heading paragraph

HTML Footer

The <footer> tag defines a footer for a document or section. A <footer> element should contain information about its containing element.

A <footer> element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

You can have several <footer> elements in one document.

Example:

```
<html>
<body>
<footer>
  <p>Posted by: Web Tech</p>
  <p>Contact information: <a href="mailto:someone@example.com">web@web.com</a>.</p>
</footer>
<p><strong>Note:</strong> The footer tag is not supported in Internet Explorer 8
and earlier versions.</p>
</body>
</html>
```

Output:

Posted by: Web Tech

Contact information: web@web.com.

Note: The footer tag is not supported in Internet Explorer 8 and earlier versions.

HTML Aside

The <aside> tag defines some content aside from the content it is placed in. The aside content should be related to the surrounding content.

Example:

```

<html>
<body>
<p>My family and I visited The Epcot center this summer.</p>
<aside>
  <h4>Epcot Center</h4>
  <p>The Epcot Center is a theme park in Disney World, Florida.</p>
</aside>
<p><strong>Note:</strong> The aside tag is not supported in Internet
Explorer 8 and earlier versions.</p>
</body>
</html>

```

Output:

My family and I visited The Epcot center this summer.

Epcot Center

The Epcot Center is a theme park in Disney World, Florida.

Note: The aside tag is not supported in Internet Explorer 8 and earlier versions.

HTML Nav

The <nav> tag defines a set of navigation links.

Example:

```

<html>
<body>
<nav>
<a href="/web/">Web</a> |
<a href="/C++/">C++</a> |
<a href="/Microprocesspor/">Microprocesspor</a>
</nav>
<p><strong>Note:</strong> The nav tag is not supported in Internet Explorer 8 and
earlier versions.</p>

</body>
</html>

```

Output:

[Web](#) | [C++](#) | [Microprocesspor](#)

Note: The nav tag is not supported in Internet Explorer 8 and earlier versions.

HTML Figure Tags

The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc. While the content of the <figure> element is related to the main flow, its position is independent of the main flow, and if removed it should not affect the flow of the document.

Example:

```
<html>
<body>
<p>Nepal has a history of nine decades of science teaching/learning at the higher
  level. Several schemes, plans, policies and implementation strategies have been
  experimented in science education. This institute has come to the present form af
  ter facing many challenges and preparing scientifically oriented technical manpow
  er of different levels to meet the requirements of the nation. It is capable to c
  ompete with universities at regional and global levels in terms of curriculum and
  pedagogic standards.
</p>
<figure>
  
</figure>
<p><strong>Note:</strong> The figure tag is not supported in Internet Explorer 8
and earlier versions.</p>

</body>
</html>
```

Output:

Nepal has a history of nine decades of science teaching/learning at the higher level. Several schemes, plans, policies and implementation strategies have been experimented in science education. This institute has come to the present form after facing many challenges and preparing scientifically oriented technical manpower of different levels to meet the requirements of the nation. It is capable to compete with universities at regional and global levels in terms of curriculum and pedagogic standards.



Note: The figure tag is not supported in Internet Explorer 8 and earlier versions.

HTML Div

The <div> element defines logical divisions within the document. When you use a <div> element, you are indicating that the enclosed content is specific section of the page and you can format the section with CSS (Cascading Style Sheet). For example,

```
<div style="background-color:orange;text-align:center">  
<p>Navigation section</p>
```

```
</div>
```

```
<div style="border:1px solid black">
```

```
<p>Content section</p>
```

```
</div>
```

DRAFT

HTML Events

Events trigger actions in the browser, like starting a JavaScript when a user clicks on an HTML element. Below is a list of attributes that can be inserted to HTML tags to define event actions. These HTML events are given below:

Window Events (**Only valid in body and frameset elements**)

Attribute	Value	Description
Onload	<i>Script</i>	Script to be run when a document loads
Onunload	<i>Script</i>	Script to be run when a document unloads

Form Element Events (**Only valid in form elements**)

Attribute	Value	Description
Onchange	<i>Script</i>	Script to be run when the element changes
Onsubmit	<i>Script</i>	Script to be run when the form is submitted
Onreset	<i>Script</i>	Script to be run when the form is reset
Onselect	<i>script</i>	Script to be run when the element is selected
Onblur	<i>script</i>	Script to be run when the element loses focus
Onfocus	<i>script</i>	Script to be run when the element gets focus

Keyboard Events (**Not valid in *base, bdo, br, frame, frameset, head, html, iframe, meta, param, script, style, and title* elements**)

Attribute	Value	Description
Onkeydown	<i>script</i>	What to do when key is pressed
Onkeypress	<i>script</i>	What to do when key is pressed and released
Onkeyup	<i>script</i>	What to do when key is released

Mouse Events (**Not valid in *base, bdo, br, frame, frameset, head, html, iframe, meta, param, script, style, title* elements**)

Attribute	Value	Description
OnClick	<i>script</i>	What to do on a mouse click
Ondblclick	<i>script</i>	What to do on a mouse double-click
Onmousedown	<i>script</i>	What to do when mouse button is pressed
Onmousemove	<i>script</i>	What to do when mouse pointer moves
Onmouseout	<i>Script</i>	What to do when mouse pointer moves out of an element
Onmouseover	<i>Script</i>	What to do when mouse pointer moves over an element
Onmouseup	<i>script</i>	What to do when mouse button is released

Exercise: WAP to create a form containing text field for name, email, password, roll number, faculty, gender, hobbies, description and a submit button and clear button

-----***-----

DRAFT