

SYLLABUS

CA4CRT13 - Web Programming Using PHP

Unit 1

Introduction to web, WWW architecture, Fundamentals of HTML, text formatting tags, marquee, inserting images, links, lists, creating tables, frames, working with form elements.

Unit 2

CSS introduction, <link> and <style> elements, CSS properties, Controlling Fonts, Text formatting, Text- pseudo classes, Selectors, Links, Backgrounds, lists. Introduction to Java Script, Java Script variables, operators, decision control statements, looping, functions, arrays, events, popup boxes-alert, prompt, conform box, built-in objects, writing JavaScript, form validation

Unit 3

Introduction to PHP, server side scripting, role of web server software, php comments, variables, echo and print, PHP operators, data types, branching statements, loops, arrays

Unit 4

PHP functions, PHP form, Passing information between pages, \$_GET, \$_POST, \$_REQUEST. String functions, include and require, session and cookie management, error handling in PHP, Object Oriented Programming using PHP

Unit 5

Introduction to MySQL, datatypes, SQL commands-CREATE, UPDATE, INSERT, DELETE, SELECT, PHP functions for MySQL connectivity and operation-
mysql_connect, mysql_select_db, mysql_query, mysql_fetch_row,
mysql_fetch_array, mysql_result, mysql_list_fields, mysql_num_fields, insertion,
updation and deletion of data using PHP, displaying data from MySQL in webpage.

UNIT 1

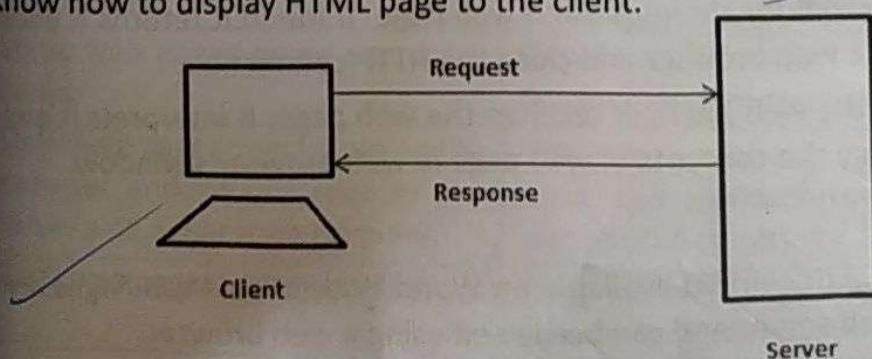
INTRODUCTION TO WEB

INTERNET

The Internet is a global system of interconnected computer networks that use the standard Internet protocol suite (TCP/IP) to serve billions of users worldwide. It is a network of networks that consists of millions of private, public, academic, business, and government networks, of local to global scope, that are linked by a broad array of electronic, wireless and optical networking technologies. The Internet carries a vast range of information resources and services, such as the interlinked hypertext documents of the World Wide Web (WWW) and the infrastructure to support electronic mail.

Introduction to Web

Web consists of billions of clients and server connected through wires and wireless networks. The web clients make requests to web server. The web server receives the request, finds the resources and return the response to the client. When a server answers a request, it usually sends some type of content to the client. The client uses web browser to send request to the server. The server often sends response to the browser with a set of instructions written in HTML(HyperTextMarkup Language). All browsers know how to display HTML page to the client.



WWW Overview

WWW stands for World Wide Web. All the resources and users on the Internet are using the protocol Hypertext Transfer Protocol (HTTP). World Wide Web was created by Tim Berners Lee in 1989 at CERN in Geneva. World Wide Web came into existence as a proposal by him, to allow researchers to work together effectively and efficiently at CERN. Eventually it became World Wide Web.

The World Wide Web is the universe of network-accessible information, an embodiment of human knowledge.

In simple terms, World Wide Web is a way of exchanging information between computers on the Internet, tying them together into a vast collection of interactive multimedia resources. Note that Internet and Web is not the same: Web uses Internet to pass over the information.

WWW Operation

WWW works on client- server approach. Following steps shows how the web works:

1. User enters a URL (<http://www.google.com>) of a web page in the address bar of browser.
2. The browser requests the DNS(Domain Name Server) for the IP address of the corresponding url. DNS stores the url and the corresponding IP address.
3. After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.
4. Then web server receives request using HTTP protocol and checks its search for the requested web page. If found it returns it back to the web browser and close the HTTP connection.
5. Now the web browser receives the web page, It interprets it and display the contents of web page in web browser's window.

Web Page

Web page is a document available on World Wide Web. Web Pages are stored on web server and can be viewed using a web browser.

A web page can contain the different types of information including text, graphics, audio, video and hyperlinks. Hyperlinks are the link to other web pages. That is links points to another webpage. Hyperlink contains the address of webpage.

Collection of linked web pages on a web server is known as website. There is unique Uniform Resource Locator (URL) is associated with each web page. Home page or Index page is the first page of a website.

There are two types of Webpages. They are:

1. Static web page
2. Dynamic web page

Static Web page

Static web pages are also known as stationary web page. They are loaded on the client's browser as exactly they are stored on the web server. These type of web pages contain only static information. User can only read the information but cannot change the information in the webpage.

Static web pages are created using only HTML. Static web pages are only used when the information is no more required to be modified.

Dynamic Web page

Dynamic web page shows different information at different point of time. It is possible to change a portion of a web page without loading the entire web page.

Static web page	Dynamic web page
The content and layout of a web page is fixed.	The content and layout may change during run time.
Static web pages never use databases.	Database is used to generate dynamic content through queries.
Static web pages directly run on the browser and do not require any server side application program.	Dynamic web page runs on the server side application program and displays the results.
Static web pages are easy to develop.	Dynamic web page development requires programming skills.

Scripting Languages

Scripting languages are like programming languages that allow us to write programs in form of script. These scripts are interpreted not compiled and executed line by line. Scripting language is used to create dynamic web pages.

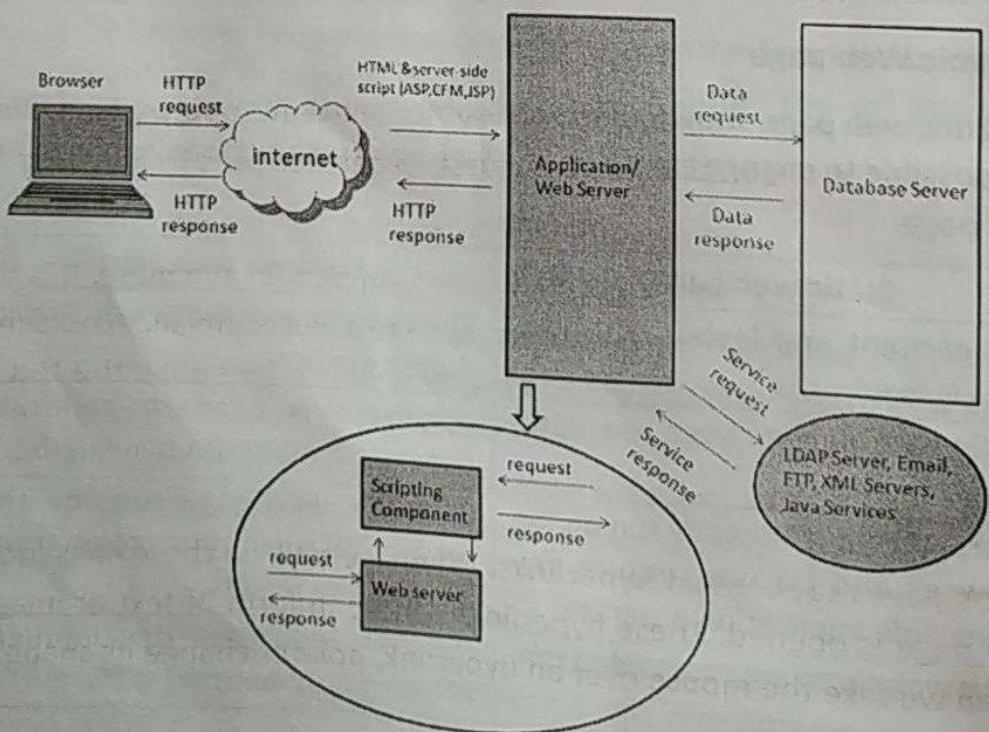
Client-side Scripting

Client-side scripting refers to the programs that are executed on client-side. Client-side scripts contains the instruction for the browser to be executed in response to certain user's action.

Client-side scripting programs can be embedded into HTML files or also can be kept as separate files. JavaScript, VB Script, Action Script, etc. are some of the examples of client side scripting languages.

Server-side Scripting

Server-side scripting acts as an interface for the client and also limit the user access the resources on web server. It can also collects the user's characteristics in order to customize response.



Commonly used Server-Side scripting languages are ASP, ASP.Net, JSP, Python, PHP, etc. Following table shows the comparison between client side scripting and server side scripting.

Client side scripting	Server side scripting
Client side scripting	Script remains in the web server
Server side scripting	Script is executed in the web server and the web page produced is returned to the client browser
Client side scripts are mainly used for validation of data at the client.	Server side scripts are usually used to connect to databases and return data from the web server
Users can block client side scripting	Server side scripting cannot be blocked by a user
The type and version of the web browser affects the working of a client side script	The features of the web browser does not affect the working of server side script

Web Browser

Web Browser is an application software that allows us to view and explore information on the web. User can request for any web page by just entering a URL into address bar.

Web browser can show text, audio, video, animation and more. It is the responsibility of a web browser to interpret text and commands contained in the web page. *Accessing web page* is very simple. Just enter the **URL** in the address bar.

Navigation

A web page may contain **hyperlinks**. When we click on these links other web page is opened. These hyperlinks can be in form of text or image. When we take the mouse over an hyperlink, pointer change its shape to hand.

Web server

Web server is a computer where the web content is stored. Basically web server is used to host the web sites but there exists other web servers also such as gaming, storage, FTP, email etc.

Web site is collection of web pages while web server is a software that respond to the request for web resources.

Web Server Working

- When client sends request for a web page, the web server search for the requested page if requested page is found then it will send it to client with an HTTP response.
- If the requested web page is not found, web server will send an **HTTP response: Error 404 Not found.**
- If client has requested for some other resources then the web server will contact to the application server and data store to construct the HTTP response.

Fundamentals of HTML

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

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

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

Tim Berners-Lee, scientist and academic, was the primary author of HTML.

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

You can write your HTML with any available text editor. All you need to do is type in the code, then save the document, making sure to put an .html extension or an .htm extension to the file (for example "mypage.html").

Structure of HTML Document

A HTML file has two parts: Head part and Body part.

Head Part: <head> </head>

Body Part: <body> </body>

Details of the head section can be viewed within the head tag. The body section encloses the information to be displayed in a web page. Head section should come first and body section must follow. The structure of an html file is given below.

```
<!DOCTYPE html>

<html>
  <head>
    <title>This is document title</title>
  </head>

  <body>
    <h1>This is a heading</h1>
    <p>Document content goes here.....</p>
  -----
  </body>
</html>
```

} Head part

} Body part

Above example of HTML document uses the following tags:

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

Four key concepts of HTML

1. Elements

All HTML pages are made up of elements. An element is just like a container in which a portion of a page is placed. An element consists of two tags namely an 'opening tag and a closing tag'. Eg. <html> ... </html>

2. Tags

The most essential components of HTML are 'tags'. Every webpage is a collection of tags. A tag is a keyword enclosed in angular brackets (<> or </>).

That is tags are commands written in text document. They are not displayed in the web pages. A tag begins with a "less than" sign (<) then the element name, followed by a "greater than" sign (>). Such a tag is called as an opening tag. In the case of a closing tag, it includes a back slash (/) before the element name.

For example, opening tag for the paragraph element will be <p> , closing tag for the paragraph element will be </p> .

Eg. <html> </html>
<head> </head>

The tags can be classified as

i. **Empty tags**

Empty tag is a tag which contains only an opening tag and there is no need of closing tag. For example <p>,

ii. **Container tags.**

Container tag is a tag which requires an opening tag as well as a closing tag. It has the following syntax <keyword> </keyword>

Examples are

<html> </html>
<body> </body>
<title> </title>

The **container tags** are also known as '**Paired Tags**'.

3. Attributes and values

Attributes are another important part of HTML. An attribute is used to define the characteristics of an element and is placed inside the element's opening tag. For example, if the size of the image is 300 X 200 pixels, the HTML statement will be

4. Nesting

Nesting means arrangement of the elements in such a way that there is no overlapping of elements.

For example

```
<html>
<head>
    <title> My first web page </title>
</head>
<body>Hai</body>
</html>
```

<BODY> element

The <body> element contains the entire content of a webpage. It must be the second element inside of the parent <html> element, following only the <head> element.

The <body> tag defines the document's body. The **HTML body tag** acts as a container for all the renderable content of the document. The <body> element contains all the contents of an HTML document, such as text, hyperlinks, images, tables, lists, etc. There can be only one <body> element in a document.

The <body> element may be the most important HTML element there is. The contents of the <body> element are what actually get displayed to the user visiting your web page or viewing your document. The following are the important attributes of body tag:

Attribute	Value	Description
Alink	rgb(x,x,x) #xxxxxx colorname	Specifies the color of the active links in the document.
background	URL	Specifies the background image file path.
bgcolor	rgb(x,x,x) #xxxxxx colorname	Specifies the background color.
Link	rgb(x,x,x) #xxxxxx colorname	Specifies the color of all the links in the document.
Text	rgb(x,x,x) #xxxxxx colorname	Specifies the color of the text in the document.
Vlink	rgb(x,x,x) #xxxxxx colorname	Specifies the color of the visited links in the document.

Shown below is an example of a complete body tag:

```
<body bgcolor="#c0c0c0" background="bubbles.gif" text="#000000"  
link="#3399ff" vlink="#9966ff" alink="#000000" bgproperties="fixed">
```

If you want your background to be set in place and not scroll, add the attribute `bgproperties="fixed"` into the body. This will leave the image as a **watermark**.

`Leftmargin`, `rightmargin`, `topmargin` and `bottommargin` are the attributes of `<body>` tag to specify the different margin width.

HTML Comments

Comments can be inserted into the HTML code to make it more readable and understandable. Comments are ignored by the browser and are not displayed. Comments are written like this:

Example

```
<!-- This is a comment -->
```

Text Formatting Tags

HTML Formatting is a process of formatting text for better look and feel. There are many formatting tags in HTML. These tags are used to make text bold, italicized, or underlined. There are almost 12 options available that how text appears in HTML and XHTML.

1) Bold Text

If you write anything within `.....` element, is shown in bold letters.

Example:

```
<b>Write Your First Paragraph in bold text.</b>
```

Output:

Write Your First Paragraph in bold text.

2) Italic Text

If you write anything within `<i>.....</i>` element, is shown in italic letters.

Example:

<i>Write Your First Paragraph in italic text.</i>

Output:

Write Your First Paragraph in italic text.

3) HTML Marked formatting

If you want to mark or highlight a text, you should write the content within <mark>.....</mark>.

Example:

<h2> I want to put a <mark> Mark</mark> on your face</h2>

Output:

I want to put a Mark on your face

4) Underlined Text

If you write anything within <u>.....</u> element, is shown in underlined text.

Example:

<u>Write Your First Paragraph in underlined text.</u>

Output:

Write Your First Paragraph in underlined text.

5) Strike Text

Anything written within <strike>.....</strike> element is displayed with strikethrough. It is a thin line which cross the statement.

Example:

<strike>Write Your First Paragraph with strikethrough</strike>

Output:

~~Write Your First Paragraph with strikethrough.~~

6) Monospaced Font

If you want that each letter has the same width then you should write the content within `<tt>.....</tt>` element.

Note: We know that most of the fonts are known as variable-width fonts because different letters have different width. (for example: 'w' is wider than 'i'). Monospaced Font provides similar space among every letter.

Example:

Hello `<tt>Write Your First Paragraph in monospaced font.</tt>`

Output:

Hello Write Your First Paragraph in monospaced font.

7) Superscript Text

If you put the content within `^{.....}` element, is shown in superscript ; means it is displayed half a character's height above the other characters.

Example:

Hello `^{Write Your First Paragraph in superscript.}`

Output:

Hello Write Your First Paragraph in superscript.

8) Subscript Text

If you put the content within `_{.....}` element, is shown in subscript ; means it is displayed half a character's height below the other characters.

Example:

Hello `_{Write Your First Paragraph in subscript.}`

Output:

Hello Write Your First Paragraph in subscript.

9) Deleted Text

Anything that puts within `.....` is displayed as deleted text.

Example:

```
<p>Hello <del>Delete your first paragraph.</del>
```

Output:

Hello ~~Delete your first paragraph.~~

10) Inserted Text

Anything that puts within `<ins>.....</ins>` is displayed as inserted text.

Example:

```
<del>Delete your first paragraph.</del><ins>
```

Write another paragraph.</ins>

Output:

~~Delete your first paragraph.~~ Write another paragraph.

11) Larger Text

If you want to put your font size larger than the rest of the text then put the content within `<big>.....</big>`. It increase one font size larger than the previous one.

Example:

```
Hello <big>Write the paragraph in larger font.</big>
```

Output:

Hello Write the paragraph in larger font.

12) Smaller Text

If you want to put your font size smaller than the rest of the text then put the content within `<small>.....</small>` tag. It reduces one font size than the previous one.

Example:

Hello <small>Write the paragraph in smaller font.</small>

Output:

Hello Write the paragraph in smaller font.

13) Preserve Formatting

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

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

```
<pre>
function testFunction(strText ){
    alert (strText)
}
</pre>
```

Output

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

14) Centering Content

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

Example:

```
<body>
    <p>This text is not in the center.</p>
    <center>
        <p>This text is in the center.</p>
    </center>
</body>
```

Output

This text is not in the center.

This text is in the center.

15) -tag

The tag allows us to change the size, style and colour of the text closed within and tags. It is generally used for changing the appearance of a short segment of the document. The attributes of tag are:

- Color : This attribute sets the text colour using either a ColorName or a colour in the Hexadecimal format.
- Face : This attribute specifies the font face. If no face attribute is mentioned, the document text in the default style is used in the first font face that the browser supports.
- Size : This attribute specifies the font size whose value ranges from 1 to 7, with default value 3.

Example:

```
<font Size="6" Face="Courier New" Color="#B22222">  
Successful people don't do great things, they only do small things in  
a great way.  
</font>
```

HTML Heading

A HTML heading or HTML h tag can be defined as a title or a subtitle which you want to display on the webpage. When you place the text within the heading tags <h1>.....</h1>, it is displayed on the browser in the bold format. Size of the text depends on the number of heading.

There are six different HTML headings which are defined with the <h1> to <h6> tags. h1 is the largest heading tag and h6 is the smallest one. So h1 is used for most important heading and h6 is used for least important. Attribute of heading tag in html is *align*. It specifies the alignment of heading. Possible values of align attribute are left, right and center. The default alignment is left.

For example:

```
<h1>Heading no. 1</h1>
<h2>Heading no. 2</h2>
<h3>Heading no. 3</h3>
<h4>Heading no. 4</h4>
<h5>Heading no. 5</h5>
<h6>Heading no. 6</h6>
```

Output

Heading no. 1

Heading no. 2

Heading no. 3

Heading no. 4

Heading no. 5

Heading no. 6

HTML Paragraph

HTML paragraph or HTML `<p>` tag is used to define a paragraph in a webpage. It is a notable point that a browser itself add an empty line before and after a paragraph. Attribute of paragraph tag in html is `align`. It specifies the alignment of text within paragraph. Possible values of align attribute are `left`, `right`, `center` and `justify`. The default alignment is left.

For example:

```
<p>This is first paragraph.</p>
<p>This is second paragraph.</p>
```

Output:

This is first paragraph.

This is second paragraph.

HTML - line breaks

A line break is used in HTML text elements, and it is the equivalent of pressing Enter or Return on your keyboard. In short, a line break ends the line you are currently on and resumes on the next line. `
` tag is used for line break.

Example

This is first line
`
`this is second line

Output

This is first line
this is second line

Horizontal rule

Use the `<hr />` tag to display lines across the screen. Note: the horizontal rule tag has no ending tag like the line break tag. Attributes of `<hr>` tag are:

- Width: to specify the width of the line. Width can be specified in pixel or percentage (`<hr width="60%">` or `<hr width="600">`).
- Size: specifies the thickness of the line
- Align: specifies the alignment of the line and center is the default alignment.

Example : This `<hr/>` is a line

Output

This

is a line

Inserting images

HTML provides a tag `` to insert images in HTML pages. The `` tag is an empty tag, which means that, it can contain only list of attributes and it has no closing tag.

``

Following are the important attributes of -tag.

Attribute	Value	Description
src	URL	the url of an image
align	Top, bottom, middle, left, right	Deprecated – Specifies the alignment for the image.
alt	Text	Specifies alternate text
border	Pixels	Deprecated " Specifies the width of the image border.
height	pixels or %	Specifies the height of the image.
hspace	Pixels	Deprecated " Amount of white space to be inserted to the left and right of the object.
vspace	Pixels	Deprecated " Amount of white space to be inserted to the top and bottom of the object.
width	pixels or %	Sets the width of an image in pixels or in %.

Grouping Content

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

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

Example

```

<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

```

<!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> along with CSS</p>
</body>
</html>

```

This will produce the following result:

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

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

HTML Hyperlinks (Linking)

The anchor tag can be used to create a link to another document. A hyperlink (or link) is a word, group of words, or image that you can click on to jump to a new document or a new section within the current document. When you move the cursor over a link in a Web page, the arrow will turn into a little hand. This type of text that provides linking is called "Hyper Text". HTML provides two types of linking

1. External linking
2. Internal linking

Links are specified in HTML using the `<a>` tag.

```
<a href="url">Link text</a>
```

Here `a` = creates an anchor

`href` = Target URL(Uniform Resource Locator)

`Link text` = Text to be displayed as substitute for the URL

Example

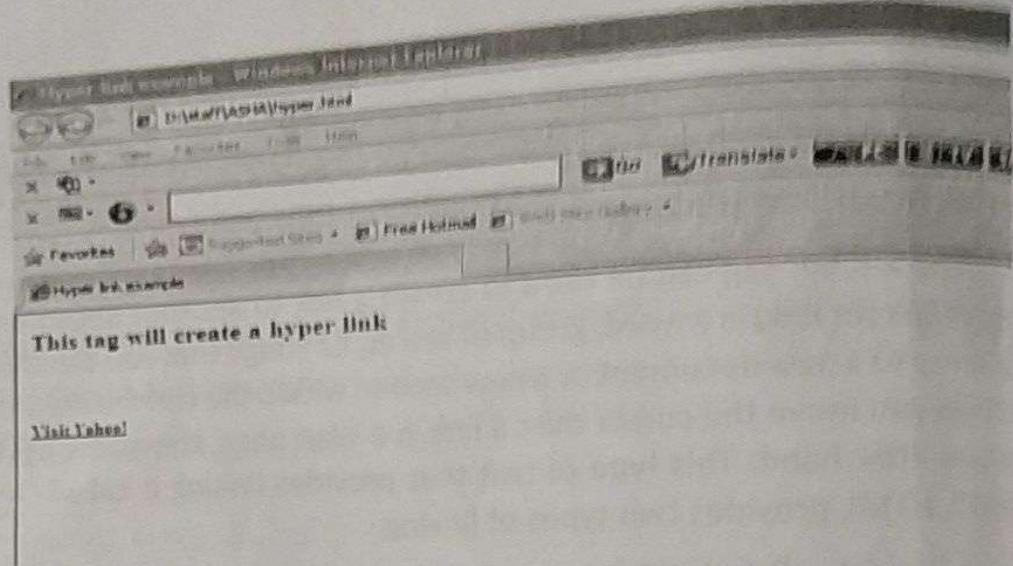
```
<a href="http://www.w3schools.com/">Visit W3Schools</a>
<a href="http://www.yahoo.com/">Visit Yahoo!</a>
```

External linking

External linking is provided with the anchor tag `<a> ..`. Important attribute related to external linking is `href` (hypertext reference). This attribute is used to indicate the source file of a document link.

Example

```
<body>
<h3>This tag will create a hyper link </h3>
<br><br>
<h5><a href = http://www.yahoo.com/> Visit Yahoo! </a></h5>
<body>
```



2. Internal Linking

The process in which link one document to another part of the same document is called internal linking. The internal linking is also provided with the use of `<a> `. This type of tag is commonly called 'named anchor'.

To create internal link, we have to do 2 steps

1. Naming the target portion: The attribute `name` is used to give name to a particular portion.

Syntax: ``

2. Create a link to the particular portion: The attribute `href` is used for this purpose

Example of internal linking

```
<html>
<body>
.....
<a name = "portion1"> When you click the following link, you will
reach to me </a>
.....
.....
<a href = "#portion1"> Click Me! </a>
</body>
</html>
```

Other attributes of `<a>` are `alt`, `name` and `target`.

Creating Banner

The tag `<marquee> </marquee>` is used for creating marquee or moving text. Its attributes are:

1. align:- Sets the alignment of the text. Values are top, middle or bottom
2. behavior:- Sets how the text in the marquee should move. Values are Scroll, Slide(text enters from one side and stops at the other side), alternate – text seems to bounce from one side to other.
3. bgcolor:- Sets the background color for marquee box.
4. direction:- Sets the direction the text should move (values:- left, right, up or down).
5. height : - Specifies the height of the marquee box.
6. width:- Specifies the width of the marquee box.
7. loop:- Sets how many times we want the marquee to cycle. Its values may be a positive integer (eg: 10,5,100) or infinite for continuous cycling.

Example of marquee

```
<body>
<marquee align ="bottom" loop ="infinite" direction="right"
behavior="scroll"bgcolor="red">
<h2> Here is Marquee1</h2>
</marquee>
<marqueeart="middle" loop =3 direction="left" behavior="slide"
bgcolor="red">
<h2> Here is Marquee2</h2>
</marquee>
</body>
```

A banner is like a title, but it always stays at that place and while scrolling through a web page it does not move on the screen. Generally, the Banner is on the top of a web page. Generally, the Banner contains the organization logo, or its name and motto.

Lists

The item that appear in 'One by One' format is called list. There are three types of lists:

- i. Ordered list (Numbered list)
- ii. Unordered list (Unnumbered list)
- iii. Definition list.

1. Ordered List

The list of items that appears with a use of numbers or alphabets is called 'Ordered List'. The tag ` ` is used to specify the starting and end of the ordered list and list items presented in the list by using the tag ``. The attributes that can be specified with the tag `` are:

- TYPE: its values may be:
 - TYPE = "1" will give counting numbers (1,,2,....)
 - TYPE = "A" will give Uppercase letters (A, B,)
 - TYPE = "a" will give Lowercase letters (a,b,)
 - TYPE = "I" will give Uppercase Roman Numbers (I,II,)
 - TYPE = "i" will give Lowercase letters (i,ii,)
- START : Alters the numbering sequence. Can be set to any value.

For example:

```
<body>
  <b>My Ordered List</b>
  <ol>
    <li>Item 1
    <li>Item 2
    <li>Item 3
  </ol>
</body>
```

Output

My Ordered List:

1. Item 1
2. Item 2
3. Item 3

2. Unordered lists

Unordered Lists, or ` ..` tags, are ones that appear as a list of items with “bullets” or markers in the front. The bullet marks will depend on the particular version of your web browser. Each list items are represented within the tag ``. The closing tag is optional. Most important attribute of an unordered list is ‘type’. Its values may be

- Disc
- Circle
- Square

```
<body>
<b>My Unordered List:</b>
<ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ul>
</body>
```

Output**My Unordered List:**

- Item 1
- Item 2
- Item 3

3. Definition List

Definition list is the special arrangement of the list items without bullets or numbers. In other words, definition list is the list of items and definitions. A definition list starts with `<DL>` end with `</DL>`. It contains two types of tags.

32

- i. <dt> </dt> Definition term or definition title
- ii. <dd> </dd> Definition description

For example

```
<body>
  <dl>
    <dt> keyboard
        <dd> an input device

    <dt> printer
        <dd> an output device
  </dl>
</body>
```

Output

KEYBOARD

An input device

PRINTER

An output device

Example program for Ordered and Unordered list

```
<body>
  <ol type = 1>
    <li> orange
    <ul type = circle>
      <li> large
    <li> sweet
      </ul>
      <li> grape
      <ul type = circle>
    <li> small
    <li> tart
      </ul>
  <ol>
</body>
```

Output

1. ORANGE

- Large
- Sweet

2. GRAPE

- Small
- Tart

TABLES

Data items represented with the use of rows and columns are called table. `<Table>` and `</Table>` is used for this purpose. A table has different parts: such as rows, columns, headings, captions etc. Most important tags related to table creation are

1. `<TABLE></TABLE>`
 2. `<CAPTION></CAPTION>`
 3. `<TR></TR>`
 4. `<TD></TD>`
 5. `<TH></TH>`
- 1. `<TABLE> <TABLE>`**

Most important attributes of table tag are

- (i) Align:- Takes LEFT, RIGHT OR CENTER as its values and aligns the table with respect to surrounding information.
- (ii) Background: - The value of this attribute has to be an image URL. This image is displayed as the background of the table.
- (iii) BgColor: - Specifies the background color.
- (iv) Border: - It makes a border around the table. The thickness of this border depends on the number of pixels supplied as its value.
- (v) BorderColor:- Table borders can be coloured using his attribute.
- (vi) CellPadding:- Its value supplied as number of pixels, denotes the amount of space between the edge of the table and its contents.

vii) CellSpacing: - This attribute sets the width in pixels between the individual table cells.

viii) Height:- Specifies the table height , value has o be in pixels.

ix) Width:- Defines the width of the table. It value has to be pixels.

2. <CAPTION>.....<CAPTION>

Often tables need to be given a heading, which gives the reader a content for the information in the tables. Table headings are called captions. Captions can be provided to the table using the <caption> tag and will appear within the <Table><Table> tags. It has an attribute ALIGN.

ALIGN = BOTTOM will place the caption immediately below the table
TOP will place the caption immediately above the table.

3. <TR></TR>

The data row is defined using <TR> </TR> tags. Text matter displayed in a row is left justified by default. Any special formatting like boldface or italics is done by including appropriate formatting tags inside the <TR> </TR> tags. An image can also displayed in a data cell.

4. <TD></TD>

It is used to insert data into the table. Its attributes are

(i) Align :- Its value can be left, right, center

(ii) Valign:- Aligns the contents to the top, middle or bottom of the cell.

(iii) BGCOLOR:- Specifies background color

(iv) BorderColor:- Determines the border colors

(v) Rowspan:- To extend data cell across more than one row. Value has to be an integer number.

(vi) COLSPAN: - To extend data cells across more than one columns. Its value has to be an integer number.

(vii) Height:- Specifies the height of the cell.

(viii) Width:- Specifies the width of the cell.

5. <TH> </TH>

To create table headings.

Example

```
<body>
<h2>Basic HTML Table</h2>
<table align = "center" BgColor = "green" border = "6" >
    <tr>
        <th>Roll Number</th>
        <th> Name </th>
    </tr>
    <tr>
        <td>1</td>
        <td>Smith</td>
    </tr>
    <tr>
        <td>3</td>
        <td>Willy</td>
    </tr>
</table>
</body>
```

Basic HTML Table

Roll Number	Name
1	Smith
3	Willy

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. HTML 5 is not supported the frames.

1. Each frame is given a name
2. It can load its own URL independent of other frames. Other wise each frame will be targeted by HTML document.
3. Each frame resizes itself in response to the changes in the size of visible client area.

Frames are generated by (i) Frame document (ii) Frameset tags (iii) Frame tags.

(i) Frame document

In the basic structure of frame document, the tag <body> is replace by <frameset> tag, that describes the sub html documents or frames that make up the page. Syntax of frame document is

```
<html>
<head> .... </head>
<frameset>
.....
.....
</frameset>
</html>
```

(ii) The <frameset> </frameset> tag

A set of frames are defined using <frameset> and ends with </frameset>. The main attributes of frameset tag are

1. Rows- It specifies the row height value list

Eg:-<frameset Rows = "25%,50%,25%">
</frameset>

2. Cols- It specifies column width

Eg:-<frameset Cols = "50%,40%,*> * implies remaining part
</frameset>

3. Border:- It specifies the width of the border

Syntax:- Border = n

4. BorderColor:- It specifies the color of the frame border.

Syntax :BorderColor = color

The size of the frames are mentioned by any of the following forms:

(i) Pixel Unit

The pixel unit simply represents number of pixels in each frame.
Commas must separate the numbers.

Eg: <frameset rows = "150,70,80"> </frameset>

This definition creates 3 row wise frames with the first frame of 150 pixels height, second of 70 pixels height and third of 80 pixels height.

(ii) Percentage unit

The percentage unit divides the window according to the specified percentages.

Eg: <frameset rows = "50%,30%,20%">

If the total of percentages is greater than 100 all %s are scaled down or reduced. If the sum of the %s is less than 100, the extra spaces are left out.

The <frame> tag

This defines a single frame in a frameset. This tag is an empty tag. It has following attributes.

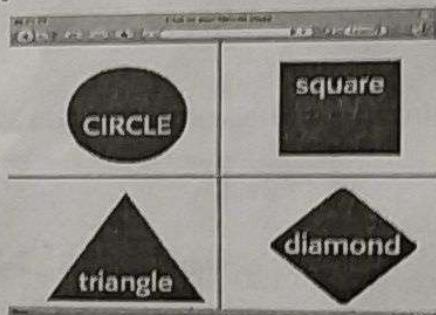
- i) SRC : "URL" Specifies the document to be displayed in the frame.
- ii) Name: Name of each frame
- iii) MarginWidth: Syntax MarginWidth=n
- iv) MarginHeight: Syntax:MarginHeight = n
- v) Scrolling - Syntax : Scrolling = "yes/no/auto"

It specifies, if the frame should have a scroll bar or not. If the value is "yes" the scrollbar will display on the frame. The no value will result the Scrollbar never being visible. 'Auto' value instructs the browser to decide whether scrollbars are needed and placed them wherever they are needed. By default the value is 'Auto'.

- vi) **Noresize:** This attribute has no value. It informs the user the frame is not resizable by the user.

```
<html>
<frameset rows="50,*" cols="50,*">
  <frame src="circle.html" />
  <frame src="square.html" />
  <frame src="triangle.html" />
  <frame src="diamond.html" />
</frameset>
</html>
```

Output



iframe

iframe stands for **Inline Frame**. An inline frame is used to embed another document or web page within the current HTML document including scrollbars and borders.

Syntax `<iframe>...</iframe>`

Attribute Specifications to Adjust Appearance and Behavior

- o `src="(URL of initial iframe content)"`
- o `name="(name of frame, required for targeting)"`
- o `longdesc="(link to long description)"`
- o `width=(frame width, % or pixels)`
- o `height=(frame height, % or pixels)`

- o align=[top | middle | bottom | left | right | center] (frame alignment, pick two, use comma)
 - o frameborder=[1 | 0] (frame border, default is 1)
 - o marginwidth=(margin width, in pixels)
 - o marginheight=(margin height, in pixels)
- scrolling=[yes | no | auto] (ability to scroll)

Example

```
<html>
<body>
<iframe src="https://www.google.com" width="500" height="250">
</iframe>
</body>
</html>
```

HTML Forms and Controls

HTML Forms are required to collect different kinds of user inputs, such as contact details like name, email address, phone numbers, or details like credit card information, etc.)

Forms contain special elements called controls like inputbox, checkboxes, radio-buttons, submit buttons, etc. Users generally complete a form by modifying its controls e.g. entering text, selecting items, etc. and submitting this form to a web server for processing. The <form> tag is used to create an HTML form.

Web forms also called (Fill-out Forms) let a user return information to a web server for some action. The processing of incoming data is handled by a script or program written in language that manipulates text files and information. Form element is used to delimit a data input form. There can be several forms in a document, but form elements cannot be nested. The forms are not hard to code. These follow the same constructs as other HTML elements.

Creating HTML Forms

As with most things in HTML, a structure starts with an opening tag, and ends with a closing tag. With forms, these tags are **<FORM>** and **</FORM>**.

```
<FORM METHOD=method ACTION=url NAME=frmnameENCTYPE=
enctype TARGET=target>
.....
</FORM>
```

The url points to the remote file or application used for digesting the information. The method is either GET or POST. Inside the **<FORM>**, a Web page can contain any standard HTML formatting information, graphics, links to other pages and a bunch of new tags specific to forms.

<FORM> Element has the following attributes:

- **METHOD**- The method can be set to either POST or GET. The difference between the two has to do with the way the information is sent to the asp program. Always use the POST method.
- **ACTION**- The action attribute is URL specifying the location where the contents of the form are to be submitted to elicit a response. If the **ACTION** attribute is missing, the URL of the document itself is assumed. This tells the form, what program should be executed by the server when the form's data is submitted. The path to your asp file will vary from system to system.
- **NAME** attribute is used to refer to the form when working with controls in it from JavaScript. Despite their names, Forms are not visible objects on the screen; it is purely a logical concept to hold the controls together.
- Target specifies the name of the frame to put the results in.
- **ENCTYPE** This attribute specifies how the data is to be encoded. It sets the MIME type used to encode the name/value pair when sent to the action URL. This attribute is applied only if POST method is used. The default value is "application/X-www-form-urlencoded". For the File control it has the value "multipart/form-data"

These tags start and end a form (all input fields of the form are placed between these two tags). METHOD specifies which technical protocol the web server will use to pass the form data to the program which processes it (always set it to POST), and ACTION tells the server exactly which program that is. Note: POST must be capitalized; otherwise the method defaults to "GET".

GET: This method sends the information entered in the form to the server at the end of the URL. Get is the default method, a question mark (?) and the form data is appended to the target URL of the code on the server and this complete variable is called query-string, which is environment variable of the server.

If there are two text fields named *name* and *id* on a form holding data RAM KUMAR and 100 respectively, then the data from these text fields would be appended to the URL like this

Filename.asp?name=RAM+KUMAR&id=100

But GET method has disadvantage from security point of view as it shows all the values entered by the user.

POST: This method is preferred over the GET method as it sends the contents of the form as a data block through the standard input stream using the http header.

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.

1. 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

```
<form>
    First name: <input type="text" name="first_name"/>
    <br>
    Last name: <input type="text" name="last_name"/>
</form>
```

This will produce following result:

Top of Form

First name:

Last name:

Attributes

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

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

2. 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:

```
<body>
<form>
User ID : <input type="text" name="user_id"/>
<br>
Password: <input type="password" name="password"/>
</form>
</body>
```

This will produce following result:

User ID :

Password:

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

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

4. Multiple-Line Text Input Controls

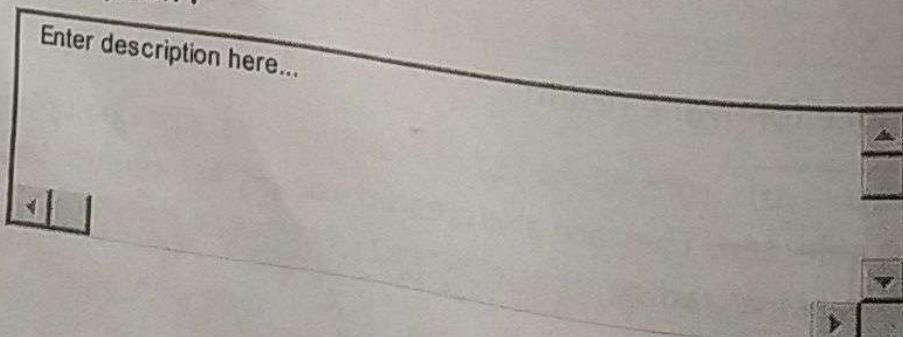
This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Example

```
<form>
Description :<br/>
<textarearows="5" cols="50" name="description">
Enter description here...
</textarea>
</form>
```

This will produce following result:

Description :



Enter description here...

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

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

5. 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

```
<form>
<input type="checkbox" name="maths" value="on">Maths
<input type="checkbox" name="physics" value="on"> Physics
</form>
```

This will produce following result:

Maths Physics

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

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

6. 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

```
<form>
<input type="radio" name="subject" value="maths">Maths
<input type="radio" name="subject" value="physics"> Physics
</form>
```

This will produce following result:

Maths Physics

Following is the list of attributes for radio button.

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

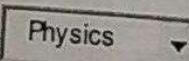
7. Select Box Control

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

```
<form>
<select name="dropdown">
<option value="Maths" selected>Maths</option>
<option value="Physics">Physics</option>
</select>
</form>
```

This will produce following result:



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

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

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

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

8. 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

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

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

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

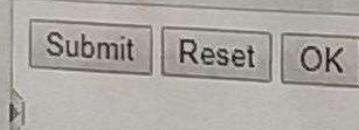
9. Button Controls

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

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

Example

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



10. Creating HIDDEN data

This is Hidden Control for "Hidden" input items. The 'hidden' input is not shown to the user. Instead Web designer will provide the value of the field. Usually, the hidden fields will contain configuration information for the server-side action script which handles the form data.

- NAME : Gives the element a name. Sets to alphanumeric characters.
- STYLE : Inline style indicating how to render the element.

- TITLE : Holds additional information, which might be displayed in tool tips for the element
- TYPE : Specifies the type of the element
- VALUE: sets the caption of the element. Set the alphanumeric characters.

Example:

```
<INPUT TYPE="HIDDEN" NAME="backup" VALUE="Hello! From HTML!">
```

Labelling Form Controls

When introducing form controls, the code was kept simple by indicating the purpose of each one in text next to it. However, each form control should have its own `<label>` element as this makes the form accessible to vision-impaired users. The `<label>` element can be used in two ways. It can:

1. Wrap around both the text description and the form input
2. Be kept separate from the form control and use the for attribute to indicate which form control it is a label for (as shown with the radio buttons).

Example

```
<label>Age:<input type="text" name="age"></label>
Gender:<input type="radio" name="gender" id="female" value="f">
<label for="female">Female</label>
<input type="radio" name="gender" id="male" value="m">
<label for="male">Male</label>
```

Age:

Gender: Female Male

Some Useful Form Elements In HTML5

Grouping Form Elements

You can group related form controls together inside the `<fieldset>` element. This is particularly helpful for longer forms. Most browsers will show the `<fieldset>` with a line around the edge to show how they are related.

The element can come directly after the opening tag and contains a caption which helps identify the purpose of that group of form controls.

Example

```
<fieldset>
<legend>contact details</legend>
<label>Email:<br>
<input type="text" name="email"></label>
<label>Mobile:<br>
<input type="text" name="mobile"></label>
<label>Telephone:<br>
<input type="text" name="telephone"></label>
</fieldset>
```

Output

Contact details

Email:	<input type="text"/>
Mobile:	<input type="text"/>
Telephone:	<input type="text"/>

Date field

HTML5 introduces new form controls to standardize the way that some information is gathered. Older browsers that do not recognize these inputs will just treat them as a single line text box.

For date field you can use <input type="date">. This will create a date input in browsers that support the new HTML5 input types.

Example

```
<label>date of birth:<input type="date" name="dob"></label>
```

Output

date of birth:

Number field

```
<input type="number" id="tentacles" name="tentacles" min="10"  
max="100">
```

<input> elements of type **number** are used to let the user enter a number. They include built-in validation to reject non-numerical entries. The browser may opt to provide stepper arrows to let the user increase and decrease the value using their mouse or by simply tapping with a fingertip.

Month field

```
<input type="month" id="start" name="start" min="2018-03"  
value="2018-05">
```

<input> elements of type **month** create input fields that let the user enter a month and year allowing a month and year to be easily entered. The value is a string whose value is in the format "YYYY-MM", where YYYY is the four-digit year and MM is the month number.

New HTML5 Elements

The interesting new HTML5 elements are:

- <header>
- <footer>
- <article>
- <section>
- <nav>

<section> Element

The <section> element defines a section in a document. A section is a thematic grouping of content, typically with a heading. A home page could normally be split into sections for introduction, content, and contact information.

Example

```
<section>  
  <h1>Introduction</h1>  
  <p>one of the new HTML 5 element is section element.....</p>  
</section>
```

<article> Element

The <article> element specifies independent, self-contained content. An article should make sense on its own, and it should be possible to read it independently from the rest of the web site.

Example

```
<article>
  <h1>article element</h1>
  <p>The <article> element specifies independent, self-contained
  content.</p>
</article>
```

Nesting <article> in <section> or Vice Versa?

The <article> element specifies independent, self-contained content. The <section> element defines section in a document. We can use <section> and <article> elements in a nested manner.

You can use pages with <section> elements containing <article> elements, and <article> elements containing <section> elements. And also pages with <section> elements containing <section> elements, and <article> elements containing <article> elements.

<header> Element

The <header> element specifies a header for a document or section. The <header> element should be used as a container for introductory content. You can have several <header> elements in one document. The following example defines a header for an article:

Example

```
<article>
  <header>
    <h1>Header element</h1>
    <p>New element in HTML 5</p>
  </header>
  <p>The <header> element specifies a header for a document or
  section.</p>
</article>
```

<footer> Element

The <footer> element specifies a footer for a document or section. A footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc. You may have several <footer> elements in one document.

Example

```
<footer>
  <p>created by: Amala</p>
  <p>Contact information: <a href="mailto:amala@gmail.com">
    amala@gmail.com</a>.</p>
</footer>
```

<nav> Element

The <nav> element defines a set of links used for navigation in a webpage. The <nav> element is proposed only for main navigational links; not for all links of a document. For example

```
<nav>
  <a href="/home.html">Home</a> |
  <a href="/contact.html">Contact US</a> |
  <a href="/abuts.html">About us</a> |
</nav>
```

<aside> Element - portion of a document whose content is only indirectly related to the document's main content.

The <aside> element defines some content aside from the content it is placed in (like a sidebar). The <aside> content should be related to the surrounding content. . . presented as Sidebars

Example

```
<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>
```

REVIEW QUESTIONS

Part A

1. What is internet?
2. What is WWW?
3. What is a website?
4. What is a webpage?
5. What is web server?
6. What is HTML?
7. How to insert images in a web page?
8. What is hypertext?
9. What are frames?
10. What are list? List the different types of lists in HTML.
11. What are hyperlinks?
12. What are attributes?
13. What are tags?
14. Define the different types of webpages.

Part B

1. Explain WWW architecture.
2. What is the difference between client side scripting and server side scripting?
3. Explain the structure of an HTML document.
4. Explain the different text formatting tags.
5. Explain marquee.
6. Explain the attributes of <body> tag with example.
7. Explain hyperlinks in HTML with example.
8. Create a webpage to display your class time table.
9. Explain lists.

Part C

1. Explain the different tags and their attributes used for creating tables in HTML with suitable examples.
2. Explain the form controls in HTML.
3. Create a webpage to display bio data form.