Chapter 1

Markup Languages

LEARNING OBJECTIVES

- ☞ *HTML*
- ☞ *Structure of an HTML document*
- ☞ *Tags*
- ☞ *Attributes*
- ☞ *External linking*
- ☞ *Hyperlinks*
- ☞ *Breaks*
- ☞ *Tables*
- ☞ *Images and color*
- ☞ *Frames*
- ☞ *CSS*
- ☞ *XML*

HyperText Markup Language (HTML)

HyperText markup language is a specialized markup language to create a web page. The language consists of ordinary text and special commands called tags.

HTML is not a formatting language. Rather it defines the parts of a document such as titles, headings, body text and block quotations. These parts are called elements. To define an element, tags are used. Tags give the browsers what they want to display on the web page.

Structure of an HTML Document

All HTML documents follow the same structure—a head which contains control information used by the browser and server and a body.

The body contains the content that displays on the screen and tags which control how that content is formatted by the browser.

```
<html>
     <head>
         <title> HTML document </title>
         </head>
         <body>
<h1> Largest heading </h1>
<p> A sample paragraph </p>
<hr>
</body>
</html>
```

- The entire document is surrounded by <html> ……. </html> which tell the software that it is now processing HTML.
- Format of our content should be according to the W3C recommendations.
- <head>……</head> and <body>…..</body> tags are compulsory in all HTML documents.
- Programming languages include a mechanism called the comment that lets developers write plain text inside their code files. Comment tags start <! ……….>. Each comment can contain as many lines of text as you like.
- If the comment runs over a number of lines, each must start and end with **- -** and must not contain **- -** within its body.

Example: <! ……..- -
- - ……. - -
- -……. - - >

Comments can be placed in either the head or body of the document.

Tags

Tags are instructions in HTML that are embedded directly into the text of an HTML document. Tags, their attributes and values are enclosed between angular brackets '<' '>'. Tags that come in pairs have a start tag and an end tag. The slash mark is used to denote the end tag. All the text with in the start tag and end tag is to be considered part of the element that the tag defines.

Tags are of two types—empty tag and container tag.

A formatted text document is composed of a set of elements such as paragraphs, headings and lists.

- A tag is a format name surrounded by angle brackets, end tags which switch a format off also contain a forward slash.
- Tags are delimited by angled brackets <h1>.

- Tags are not case sensitive. The following tags are equivalent:
  <HEAD>, <head> and <hEad>
- Styles must be switched off by an end tag.
- Some characters have to be replaced in the text by escape sequences if '<' was not escaped, the software would attempt to process anything that followed it as part of a tag.
- White space, tabs and new lines are ignored by the browser; they can be used to make the HTML source more readable without affecting the way the page is displayed.
- Multiple white spaces are replaced by a single space while new lines and tabs are treated as spaces.
- If a browser doesn't understand a tag it will ignore it.

Container tags

Tags specified in pairs, delimiting text that will have some type of formatting is called a container tag. A container along with a companion tag, encloses the text to be formatted. The effect of a container tag is applied only to the text they contain. A container tag is also called a paired tag since they always appear as a pair. The general form of a container tag can be represented as:
<tag>
Text to be formatted
</tag>

The '<tag>' is often called the opening tag and the '</tag>' is called the closing tag. The closing tag will always have a slash '/' to indicate the end of a tag. The opening tag activates the effect and the closing tag turns the effect off.

Example: <B> text </B>; the text will appear bold in the browser.

Empty tag

An empty tag is a single tag representing some formatting commands in HTML. It will not have a companion tag and are hence called stand alone or singular tag.

Example: <BR>
The tag will insert a line break at the specified position.

Attributes

HTML tags sometimes require additional information to be supplied to them. The additional information supplied to an HTML tag is known as attributes of a tag. Attributes are written immediately following the tag, separated by a space. Multiple attributes can be associated with a tag, also separated by a space.

Example: <FONT Face = "Arial" size = 12>
Welcome </FONT>
The face and size are attributes of the FONT tag.

- The document body encloses all the page formatting commands. The tags used to indicate the start and end of the main body of textual information are <BODY> and </BODY>. Page defaults like background colour, text colour, font size, etc. can be specified as attributes of the <BODY> tag.

Attributes of BODY tag

Attributes	Descriptions
Bg colour	Changes the default background colour to the colour specified with this tag. The colour can be specified by name or equivalent hexa decimal number. Example: Bg colour = RED
Text	Changes the body text colour from its default value to the colour specified with this attribute. Example: text = green
Background	Specifies the name of the 'GIF' file that will be used as the back ground of the document. This tiles up across the page to give a back ground. Background = "br. gif"

Text formatting tags

Tags	Descriptions
<P>	Paragraph break: The browser, moves onto a new line skipping one line between the previous line and the new line.
<BR>	A line break is required when the text needs to start from a new line and not continues on the same line. It is an empty tag used to simply instruct the browser to start displaying the remaining text in a new line.
<CENTER>	Center tags are used to centre not only text but anything found between them, like texts, lists, rules, tables, etc.

HR element attributes

The HR element has no ending tag.

- **ALIGN:**
  ALIGN = "______", sets the alignment of the line on the page to LEFT, RIGHT or CENTER
  The default is CENTER.
  The alignment has no purpose if the line width is 100%.
- **SIZE:**
  Sets the thickness or size of the line in pixels.
- **WIDTH:**
  Sets the width of the line across the page as a % (or) in pixels.

Linking

Links are elements in a web page which can be selected by clicking on it. Linking is one of the most important features of HTML. Link allows to connect a text or an image to another web page or section of a web page. A link will be displayed in a special way in a browser. Links will be highlighted with colours or underlines to indicate that it is a hyperlink. The target of a hyperlink can be another web page, another location on the current page, an image or any other computer file available in the server. Links can be classified into two:

1. External links
2. Internal links

External linking

External linking refers to linking two documents. When a link in a web page is clicked a new document to which the hyperlink is linked will be opened. An external link points to another HTML document located anywhere in the www.

Internal linking

Internal linking refers to linking different sections of the same document. When a link is clicked a different section in the same document will be displayed in the browser window.

Text

The text on an HTML page can be altered in many ways. The actual font used can be changed to attempt to force the browser to use a specific font and the look of the text can be changed for emphasis.

- <base font size = "n">
  We can specify the minimum font size for basic text but not for headings. The size argument takes an integer from 1 to 7.
- <font size = " [+/-]"color = " # rrggbb"> The colour of the text is set with the colour argument. This takes a hex value which represents the amounts of red, green and blue in the chosen colour.

Example:

```
<html>
    <head >
        <title> changing font sizes </title >
        </head>
    <body>
<h1> Font sizes </h1>
<base font size = "3">
<p> Here is some text in size 3
<p> Here is some < font size = "7" >
                Larger </font>
<font size = "+4" > t </font >
<font size = "+3" > e </font >
<font size = "+2" > x </font >
<font size = "-1" > t </font >
</base font>
</body>
</html>
```

- Other Alternates are
  <b> ……. </b> → Bold
  <i> …….. </i> → Italic
  <strong > ……. </Strong > used as a form of emphasis
  < tt > …… </tt> mono spaced font
  <sub> …… </sub> subscript
  <sup>…….. </sup > super script

Break

<br>
Forces a line break within a passage of text where a paragraph is not desirable. On complex pages it is sometimes useful to put a <br> before and after tables, lists

- To display Escape sequences we need to use the following replacement sequences which always start with an ampersand '&' and are terminated with a semicolon.
  & amp; → &
  & nbsp; → (white space)
  & It ; → <
  & gt; → >
  & quot ; → "
  & copy ; → ©

Hyperlinks

The benefit of hypertext is that it lets us create links within a document.

- Links should be used within documents where they either add to the understanding of the work or can be used to reduce download times.
- It is better to have many links to medium sized documents containing about a screenful of information rather than forcing readers to download a single massive document.
  $\boxed{\text{<a href = " address"> …… </a>}}$
  The link tag has 3 sections:
  1. The address of the referenced document
  2. A piece of text to display as the link
  3. Closing tag
- The link text can be formatted using any of the text formatting options. Hypertext references, the 'href' part of the tag, can be
  1. links to documents or services at other internet sites
  2. links to documents within the same website
  3. links to a specific part of either the current page or another page.

Example:

<a href = "page three . html" > Next page </a> Links to another page in the same directory. The browser displays 'Next page' on the screen and highlights it so that readers know it is a hyperlink

Example:

<a href = "http :// www. Time4education.com/index. html" > some site </a> links to another website. This time some sight is displayed and highlighted.

Lists

One of the most effective ways of structuring a website or its contents is to use lists. For Example, a commercial website may use pictures of its products instead of text in hyperlinks. These can be built as nested lists to provide an interesting graphical interface to the site.

HTML provides 3 types of lists:

1. Basic bulleted list
2. A numbered list
3. A definition list

Each has a different use but generally the definition list is the most flexible of the three lists.

- **Ordered and unordered lists:**
  <li>…. </li>
  The ordered and unordered lists are each made up of sets of list items. Elements of a list may be formatted with any of the usual text formatting tags and may be images or hyperlinks.
  The closing tag is not part of HTML.
- <ul [type = " disc"/"square"/"circle"] [compact] > ….</ul]
  The basic unordered list has a bullet in front of each list item.
  Everything between the tags must be encapsulated within <li>…</li> tags.
- To minimize the amount of space that a list uses, we have to add the compact attribute
- <ol [type = "1" |"a"|"A"| "I" | "i"]
- [start = "n"][compact] > …. </ol>

An ordered list has a number instead of a bullet in front of each list item. Different numbering schemes can be specified depending upon preference.

A list can number from any value that you desire. The starting value is given by the "start" attribute. All items in an ordered list must be enclosed within <li>….. </li>tags

Tables

Tables have two uses:

1. Structuring pieces of information
2. Structuring the whole web page
   - Alternatively we can structure a page using frames or images.
   - A table is a grid of information such as, we might have seen in a ledger or spreadsheet.
     Unlike a table from a spreadsheet the data items in an HTML table do not need to have any kind of relationship.
   - Most browsers struggle to process complex tables. The browsers are not optimized for tables and where tables are deeply nested on a page the browser may have difficulty displaying the page.
   - Web browsers have a layout engine which arranges the pieces before the web page is displayed.

- It is more difficult if table consists images where the size attribute of the image have not been set.
- <table [align = "center "/ "left" / "right"]
  [border [= "n"]]
  [cell padding = "n"] [width = "nn%"]
  [cell spacing = "n"]> ……. </table >
- Everything between <table>…..</table> tags will be part of table.
  These attributes control the formatting of the table as a whole, not that of the items in each cell.
- Tables can be aligned on the screen.
- A table can have a border, which includes a border between the cells. If the border attribute is not set, the table has no border.
- When the border attribute is set but a valid value is not given, a single pixel wide default border is drawn "cell padding" determines how much space is there between the contents of a cell and its border in pixels. Cell spacing sets the amount of white space between cells.
  - The "Width" attribute sets the amount of the screen that the table will use.
- <th [align = "left"/ "center"/ "right"]
  [valign = "top" / " center" / "bottom"]>…. <|tr>
  Each row of the table has to be delimited by these tags. The row can be aligned horizontally and vertically within the table.
- <th [align = "left " / "center" /"right"]
  [valign = "top" | "center" / "bottom"]
  [nowrap] [colspan = "n"] [rowspan = "n"] > …. <|th>
- These are table cells which are to be used for headings.
- The contents of the cell can be aligned vertically and horizontally within their row.
- If "nowrap" is set, the contents of the cell will not be automatically wrapped as the table is formatted for the screen.
- The "colspan" and "rowspan" attributes allow individual cells to be larger than a one by one grid.
- <td [align = "left"/ "center"/ "right"]
  [valign = " top"| "center"| "bottom"] [nowrap]
  [colspan = n] [rowspan = n] > …. </td>
  These describe the basic data cell.

Table elements

1. <caption> string </caption>
   This optional element is used to provide a string which describes the contents of the table. If used it must immediately follow the table element.
   - <thead> …… </thead>
     <tfoot> …… </tfoot>
     <tbody> …… </tbody>
     The rows in a table can be grouped into one of the three divisions.

- The idea is that the browsers will be able to scroll the tbody section of the table without moving either the thead or tfoot sections.
- When long tables extend over more than one page the information in thead and tfoot can be automatically replicated on each page.
- <colgroup [span = 'n'] [width = 'n']> ….. </ col group>

Columns within a table can be logically grouped together. Each group of columns can be assigned a default width which will apply to all columns.

- The span indicates the number of columns in the group.
- <col [span = "n"] [width = "n"]> …. </col>

The attributes of individual columns are set using the "col" elements. The 'span' and 'width' attributes work in the same way as the 'colgroup' element.

Example:

```
<html>
<head>
   <title> A table </title>
      </head>
      <body>
<h1> A small table </h1>
<table align = "center" width = "75%"
                border = "1">
< caption> small table </caption >
< colgroup width = "30%" span = "2">
</colgroup>
<colgroup span = "3" > </col group>
<thead>
<tr> <td colspan = "5"> The table header
</td> </tr>
   </thead>
   <tbody>
   <tr>
<td>First  </td>
<td> Second </td>
<td> Third </td>
<td> Fourth </td>
<td> Fifth </td>
   </tr>
   <tr>
<td>First </td>
<td> Second </td>
<td> Third </td>
<td> Fourth </td>
<td> Fifth </td>
   </tr>
   </tbody>
   </t foot>
   <tr> td co|span = "5"> Table Footer </td> </tr>
   </t foot>
   </table>
   </tbody>
   </html>
```

Images and Colour

- Colour can be used in a number of places on a web page; the background can be coloured, individual elements can be altered, and links which are already coloured can have their colours adjusted.
- To change the colours of links or the page background hexadecimal values are placed in the <body> tag.

```
<body bg color = "# nnnnnn" text = "# nnnnnn"
link = "#nnnnnn" vlink = "#nnnnnn"
alink = "#nnnnnn">
```

- The 'Vlink' attribute sets the colour of links visited recently, 'a link' the colour of a currently active link.
- The six figure hexadecimal values must be enclosed in double quotes and preceded by a hash (#).
- The colours of page elements can be altered by using the colour modifier. To change the colour of an individual heading we can use

```
<h2 color = "#a b a b a b" > Heading < /h2> and within a
table the table headers could be coloured by:
<th bgcolor = "#a b a b a b">
```

- Images: If we want high quality, good compression and lots of colours use JPG, GIFS are more common as they tend to be smaller files and can be animated.
- <body background = "URL"> ….. </body>
- Sets the background of your page to use the given image. Images are tiled (repeated) to fill the available space by default.
- If we want to use a single image across the width of a page make it 1281 pixels wide then it cannot be tiled horizontally.

```
<img src = "URL"| "name" height = "n" width = "n" [alt
= "string"] [align = top"| "center"/ "bottom"]
usemap = "URL"]>
```

Displays an inline image, that is an image which appears in the body of the text rather than on a page of its own or in a spawned viewer program.

- The height and width of the image, in pixels tell the browser how much space to allocate to an image when displaying a page.
- Some browsers use these to shrink/stretch images to fit.
- By default any text which follows an image will be aligned alongside its bottom edge. We can alter this so that the first line of text displays alongside the centre or top of the image.
- If we want a block of text shown next to an image we must use a table. To display an image without text, make it into paragraph.

```
<p align = "center"> <img src = "/mygif.gif"
alt =" mine"> </p>
```

The 'usemap' attribute is used in image mapping.

1. <a href = "URL"> text message </a>
2. <a href = "URL"><img src = "filename"> </a>

The first case uses an ordinary hypertext link but the URL should point to the image file, giving its name and type.

- In the second case we are using an image as the link to another image. This can be useful if we want to display a page of thumbnail images and allow the reader to choose which ones to view full-size. This is one way of speeding up the loading times of graphically intensive sites.
- An **image map** is a large picture which has areas that the reader can click with a mouse.
  Each clickable area provides a hypertext link. The image map has 2 parts:
  1. Image
  2. Map

```
<img src = "URL" use map = "URL">
```

It tells the browser to display the source image and to map the second URL, the image map, onto it.

<area shape = "circle" | "rect"| "poly" | "default" href = "URL" Coords = "string" alt = "string"> creates a clickable area on an image map. The 'alt' text in this case is displayed by the browser as an indicator for the reader of where the link goes.

- If we do not supply an 'alt', our image map is invalid and may not be displayed.
- The meaning of href should be clear, it is the destination of the link. The clickable area can have one of four shapes. Each shape is defined by coordinates, pairs of integers which give locations on the image in pixels.
- The default location does not require coordinates and is used to indicate what happens if the user clicks outside of the mapped areas.
  Each image map can have only one default
- A **'rect'** has four coordinates which are paired. The first pair defines the top left corner and the second pair the bottom right corner of the area.
- A **'circle'** is defined by its centre and its radius centre is given by a pair of values and the radius by a single value. This requires just three values in the coordinate string.
- A **'polygon'** is made from a set of coordinates with the last pair listed being joined to the first to complete the shape.
  The following example shows an image map with the mapping in the same file as the image link

```
<img src = "/mappicture.gif" usemap =" # main – map"
height = 30 width = 50>
<a name = "#main – map">
<map name = "main – map">
<area shape = "rect" href = "./images/ img1.jpg"
alt = "Image one" cords =" 0,0,25,25">
<area shape = "rect"  href = "./page1.html/"
alt = "page one" coords =" 26,26,50,50">
<area shape = default href  "./page26.html"
alt = "page 26">
</map>
</a>
```

Frames

If we want to represent a complex page structure and not confident about using a table to create it, then we can use frames. Frames are part of the HTML 4 specification. When we talk about frames what we refer to is a 'frameset' which is a special type of web page.

- The frameset page contains a set of references to HTML files, each of which is displayed inside a separate frame.
- All of the pages within a frameset are displayed inside the same browser window and can actually be made to appear to be a single page.
- Frame-based sites display more than one page at the same time, they can be complex to set up.
- Frame-based page is actually made from a set of documents, each displayed in its own frame. Each sub-documents can have its own scrollbars and can be loaded, reloaded, and printed.
- The tags that are needed are

```
<frameset [cols="%, %"] [rows = "%,%"> … </frame
set>
```

  - This tag determines how the screen will be divided between the various frames we can have as many frames either vertically or horizontally as we want.
  - Each frame has to be allocated a percentage of the screen.
  - We can also nest framesets so that individual rows or columns can themselves be broken up into frames.

```
<frame [name = "name"] src = "filename"
[scrolling = "yes"| "auto"|"no"]
[frame border = "zero" | "1"]>
```

  The src attribute works like an image source or a hyperlink address. It should point to a valid HTML file or image which can be displayed within the frame

- by setting the frame border attribute to "zero" stops it being displayed.
  <a href = " source" target = "n">
- to ensure that pages display in the correct frame we need to extend the basic address tag.

Example:

```
<html>
  <head>
    <title> TIME pvt. Limited  </title>
<frameset rows = "25% , 50%">
<frame name = "A" src = " ./ company. html">
< frame name = "B" src = " ./ orders html" scrolling = "no">
</frameset>
</html>
```

Forms

Forms are used to add an element of interactivity to a website. They are used to let the reader send information back to the server but can also be used to simplify navigation on complex websites.

```
<form action = "URL" method = "post" | "get" > … </
form>
```

- A form can contain virtually all other markup tags but cannot be rested within another form..
- The action attribute specifies the name and location of a CGI script that will be used to process the data.
  Data can be sent in 2 ways:
  1. post
  2. get
- We should use 'get' to retrieve information from a server and 'post' to send information to a server. The choice of approach is made by the 'method' attribute.
- 'post' is secured than 'get'.
- 'post' is capable of sending a wide variety of character sets but 'get' can only return ASCII data.
- 'post' is used to get data written in non-English languages.

```
   <input type = "text"| "password"| "checkbox" "radio"|
"password"| "submit"| "reset" | "button"| "image"
   name = "string" [value = "string"] [checked] [size =
"n"] [maxlength = "n"] [src = "URL"]
   [align = "top"| "bottom" | "middle" | "left"| "right"] >
```

Following are several types of input widgets:

- **Text** creates an input device up to size characters long and is able to accept up to max length characters as input.
- If value is set, that string will be used as the default text. These fields support only a single line of text. If we want to enter larger amount of text then use a "text area".
- **Password** works exactly like text but the input is not displayed to the screen. Each character is replaced by '*'. The password is not encoded but is sent to the server as plain text.
- **Radio** creates a radio button. These are always grouped; buttons within a group should have the same name but different values.
  The CGI script differentiates them by name + value.
- **Checkbox** produces a simple checkbox. It will be returned to the server as name = on if checked at submission.
- **Submit** creates a button which displays the value attribute as its text. It is used to send the data to the server.
- **Reset** also creates a button but this one is used to clear the form.
- **Image** can be used to place a picture on the page instead of a button.
- <select name = "string"> … </select>
  It often very useful to have a list of items from which the user can choose.
  The tag encloses a set of options and, when sent to the server, the name of the particular select tag and the name of the chosen option are returned.
- <option value = "string" [selected]> … </option> the 'select' statement will have several options from which the user can choose. The values will be displayed as the user moves through the list and the chosen one retuned to the server.
- <text area name = "string" rows = "n" cols = "n"> … </text area>
  creates a free format plain text area into which the user can enter anything. The area will be sized at 'rows' by cols but will support automatic scrolling.

Example:

```
<html>
  <head>
    <title> my company < /title>
    </head>
    <body>
<h2 align = "center"> Feedback form </h2>
<hr width = "60%">
<form action = "http:// www.My company.Com / cgi – bin/
feed back.cgi"
Method = "post">
<p align = "left" > name : <input type = "text"
Max length = "32" size = "16">
<p align = "left"> Email Address:
<input type = "text" max length = "32" size = "16">
<p align = "left"> Location:
<select name = "city" size = "1">
<option value = "Hyderabad" selected>
Hyderabad
<option value = "Chennai"> chennai
<option value = "Banglore"> Banglore
<option value = "Pune"> Pune
</select>
<p> comments:
<br> <textarea name = "comments" rows = "6" cols = "40">
</text area>
<P align = "center"> <input type = "submit" name = "feed-
back" value = "submit details">
</form>
<hr width = "60%">
</body>
</html>
```

Cascading Style Sheets

One of the most important aspects of HTML is the capability to separate presentation and content. HTML does not have the facilities that are needed to cope with this diversity, but style sheets provide them.

- A style is simply a set of formatting instructions that can be applied to a piece of text.
- There are 3 mechanisms by which we can apply styles to our HTML documents:
  1. The style can be defined within the basic HTML tag.
  2. Styles can be defined in the <head> section and applied to the whole document.
  3. Styles can be defined in external files called style sheets which can then be used in any document by including the style sheet via a URL.

Styles can be cascaded. This means that formats override any, which were defined or included earlier in the document. We may include an external style sheet which redefines the h_1 tag, then write an h_1 style in the head of your page before finally redefining h_1 in the body of page. The browser will use the last of these definitions when showing the content in the following example the <h_1> tag is redefined.

```
<html>
  <head>
    <title> simple style sheet </title>
    <style>
    <! - -
    h1{
      color : black;
      border : thin groove;
      text – align : center ;
      }
    - - >
  </style>
</head>
<body>
<h1> simple style sheet </h1>
</body>
</html>
```

Rules

A style rule has 2 parts, a selector and a set of declarations. The selector is used to create a link between the rule and the HTML tag.

The declaration has 2 parts, a property and a value.

- Selectors can be placed into classes so that a tag can be formatted in a variety of ways.
- Declarations must be separated using colons and terminated using semicolons.

  Selector {property: value; property: value …}

Classes

If we want to apply a style to some paragraphs we have to use classes.

Selector: classname {property : value ; property : value }
<selector class = classname>

- In the style sheet itself the rule is slightly modified by giving the style a unique name which is appended to the selector using a dot.
- In the HTML document when we want to use a named style the tag is extended by including

  Class = and the unique name

```
h1. f {
color :# a b a b a b ;
background – color : # d 9a b 29;
font – family : “Book Antiqua”, Times, Serif;
border : thin groove # 9 a b a b a;
}
<h1 class = “f” > simple heading </h1>
```

Anonymous classes

Some times we want to apply a piece of formatting to many different elements within a page but not necessarily to the entire page.

- Cascading style sheets provides a way of defining styles within reusable classes.

Example:

```
<html>
  <head>
    <title> Anonymous classes </title>
    </style>
    <! - -
      f {
        color : # a b a b a b ;
        background – color :# d9a b29;
  font – family : “Book Anitqua”, Times, serif;
  border : thin groove #9ab aba;
  - ->
</style>
</head>
<body>
<h1 class = “f”> A simple Heading </h1>
<P class = “f”> Appling the style f to a paragraph of text
</P>
</body>
</html>
```

Including style sheets:

$\boxed{\text{< link rel = “stylesheet” href = “url”}}$

Type = “text / css” media = “screen”>

- The href is a hyperlink to your style sheet, ‘rel’ tells the browser what type of link you are using.
- We have to tell the browser what type of document we are including, the type statement gives the relevant MIME type.

Extensible Markup Language (XML)

Extensible markup language (XML) is a way to apply structure to a web page. It provides a standard open format and mechanisms for structuring a document so that it can be exchanged and manipulated. Like HTML, XML uses tags to ‘markup’ data content. Unlike HTML, in XML you define your own tags that meet the exact needs of your document. The custom tags make data easier to organize and search. XML will not change the way your web page look, but it will change the way the documents are read and the way documents are filed and stored.

XML is a markup language. The term ‘markup’ is used to identify anything put within a document which either adds or provides special meaning. A mark up language is the set of rules. It declares what constitutes markup in a document and defines exactly what the markup means. It also provides a description of document layout and logical structure.

There are three types of markup:

1. Stylistic – How a document is presented.
2. Structural – How the document is to be structured.
3. Semantic – Tells about the content of the data.

In XML, the only type of markup that we are concerned with is structural.

An XML document must begin with the XML declaration statement. This statement alerts the browser or other processing tools that the document contains XML tags.

The declaration is-

<? XML version = "1.0 "?>

This is the first line of an XML document.

Tags

Tags carry the smallest unit of meaning signifying structure, format or style of the data. They are always enclosed with angular brackets, '<' and '>'. Tags are case-sensitive. This means that the tags <fruit>, <Fruit>, <FRUIT> carry different meanings and cannot be used interchangeably. All the tags must be paired so that they have a start tag and an end tag. For example, <fruit> and </fruit>.

Elements

Tags combined with data form elements. Elements are the building block of a document. An element consists of a start tag, an end tag and the content between them:

<fruit> orange is a fruit </fruit>

Attributes

An attribute gives information about an element. Attributes are embedded in the element start tag. An attribute consists of an attribute name and attribute value. The attribute name precedes its value and they are separated by an equal sign. Also the attribute value is enclosed in quotes to delimit multiple attributes in the same element.

Example: <fruit number = "6"> orange </fruit>

Document Type Definition

- Well formed XML documents are those which have tags, elements and attributes in a correct nesting structure without really providing further definitions.
- Valid XML documents are documents which follow a more formal structure.
- The main difference between well-formed XML and valid XML is the document type definition.

The document type definition (DTD) is a set of rules that define the elements that may be used and where they may be applied in relation to each other.

XML Parser

The process of taking a file and breaking it into components is called parsing.

The components are defined by a grammar, the rules of the language. Although this may be implied by the file structure rather than formally specified.

XML is not a simple data structure and cannot be handled with regular expressions for parsing. There are 4 parameters which can be used to categorize parsers:

1. Validating
2. Non-validating
3. Stream-based
4. Tree-based

Validating parser A validating parser uses both an XML file and a DTD to check that the XML adheres to the rules of the application.

If the XML breaks the rules by straying from the DTD then the parser will create an error and stop processing the files.

Non-validating parser These parsers will only use the XML document and are quite content if it is well-formed.

Stream-based parser A stream-based parser must read the entire document each time that an operation is requested and send a message to the controlling application when specific events occur.

Example: SAX

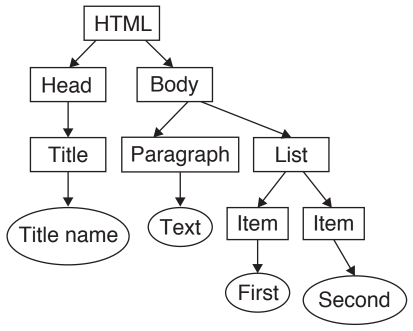
Tree-based parser This type of parser builds a static representation of the document which corresponds to the structure of the original XML.

Example: DOM

Document object model (DOM)

- The DOM is an application program interface (API) for XML documents.
- API is a set of data items and operations which can be used by developers of application programs.
- DOM API specifies the logical structure of XML documents and the ways in which they can be accessed and manipulated.
- DOM API is just a specification.
- DOM-Compliant applications include all the functionality needed to handle XML documents.
- They can build static documents, navigate and search through them, add new elements, delete elements and modify the content of existing elements
- The DOM views XML documents as trees, but this is very much a logical view of the document.
- Each node of the tree, each XML element, is modeled as an object.
- Each node encompasses both data and behavior and that the whole document can be seen as a single complex object.

- Sample DOM is shown below:



- DOM exposes the whole document to applications.

Namespaces

- A namespace is a way of keeping the names used by applications separate from each other.
- Within a particular namespace no duplicate names can exist.
- Applications may use many different namespaces at the same time.
- The implementation of namespaces is system dependent.
- XML developers can specify their own namespaces which can be used in many applications.
- A namespace can be included in the same way as a DTD.

Example:
<?xml version = "1.0"?>
<!DOCTYPE items SYSTEM "items.dtd ">
<!xml : namespace ns = "http : //URL/namespaces/jam" prefix = "jam">
<?!xml : namespace ns = "http : //URL/namespaces/bread prefix = "bread">
<items>
<item1>
<jam : name> kissan </jam : name>
</item1>
<item>
<bread : name> Roasted </bread : name>
</item>
</items>

Each item1 of items has a name element, But a namespace have been declared, so there is no chance of an application to get confused with the two names.

Attributes

- Attributes are important and useful when we are handling complexity.
- Some XML elements need to hold more than one piece of information.
- Some of these pieces are used to control the behaviour of the application. These are included as attributes.

Example:
<Quantity amount = "800" unit = "MC"> water </Quantity>
Here amount and unit are attributes of Quantity.

- The attributes of XML elements needs to be included in the DTD.
- Associated with the element declaration is an ATTLIST which may contain:
  - The name of the element
  - The name of each attribute
  - The data type of the attribute
  - Any value which will be used as a default if the attribute is omitted from the XML source.
  - Control information about the use of the element.

ATTLIST declaration

Example: <!ATTLIST Quantity amount CDATA # REQUIRED unit CDATA "g">

The declaration shows an element with two attributes. The first one is 'amount', which is of CDATA type, means that it holds plain text which will not be passed through XML parser.

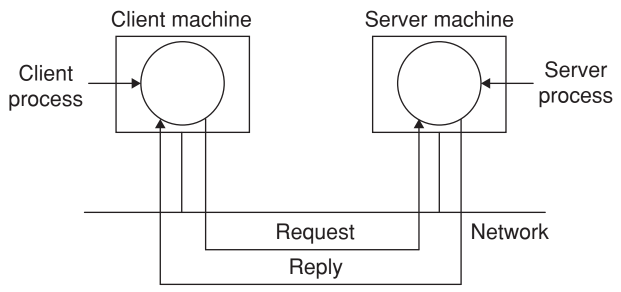
The attribute is REQUIRED which means that it must be included when the element is used. Failure of this will result in an error raised by the parser.

The second element 'unit' is optional which has a default value 'g'. If the value of this attribute is omitted then default value will be used.

Client–Server Computing

Client–server networking grew when personal computers (PC's) became the common alternative to older mainframe computers.

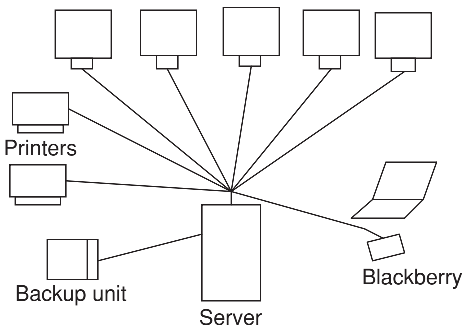
In client-server network, communication generally takes the form of a request message from the client to the server asking for some work to be done. The server does the work and sends back the reply, as shown below:



Usually, there are many clients using a small number of servers. Client devices are typically PC's with network software applications installed, that request and receive information over the network. Mobile devices as well as desktop computers can both function as clients.

A server device typically stores files and databases including more complex applications like websites. Server devices often feature higher powered central processors, more memory and larger disk drives than clients. One server generally supports numerous clients and multiple servers can be networked together in a pool to handle the increased processing load as the number of clients grows.

Client server network



Some of the most popular applications on the internet including email, FTP and web services follow client-server model. Each of these clients features a user interface (either graphic or text-based) and a client application that allows the user to connect to servers. In the case of email and FTP, users enter a computer name (or an IP Address) into the interface to setup connections to the server.

The client-server model was developed to allow more users to share access to database applications. Compared to the mainframe approach, client-server offers improved scalability because connections can be made as needed rather than being fixed. The client-server model also supports modular applications (software applications are divided into modules) that can make creation of software easier.

Example: Users accessing banking services from their computer uses a web browser client to send a request to a web server at a bank. That program may in turn forward the request to its own database client program that sends a request to a database server at another bank computer at bank to retrieve the account information. The balance is returned to the bank database client, which in turn serves it back to the web browser client displaying the results to the user.

Exercises

Practice Problems I

Directions for questions 1 to 15: Select the correct alternative from the given choices.

1. Which of the following statement is false?
(A) HTML is a markup language for hypertext.
(B) VML is used for freehand drawing in web page.
(C) WML is wireless markup language used for micro procure of mobiles and palmtops.
(D) None of these

2. The web browser request goes to the server in
(A) Hex form (B) ASCII form
(C) Binary form (D) Text form

3. The tag that contains information about the document including its title, scripts used, style definitions and documentation description is
(A) <HTML>, <\HTML>
(B) <HEAD>, <\HEAD>
(C) <BODY>, <\BODY>
(D) <TITLE>, <\TITLE>

4. Tag that is considered to be illegal in XML is
(A) <.document> (B) <document>
(C) <\document> (D) None of these

5. _____ specifies that a click in this area will not link anywhere.
(A) NOHREF (B) HREF = 0
(C) NULLHREF (D) HREF

6. A _______ specifies the layout for frames, including the locations and characteristics of the frame.
(A) frameset (B) border layout
(C) table (D) frame border

7. What is the correct syntax of the declaration, which defines the XML version?
(A) <xml version = “1.0”/>
(B) <?xml version = “1.0”?>
(C) <?xml version = “1.0”/>
(D) None of the above

8. Which of the following are predefined attributes?
(A) xml : lang (B) xml : space
(C) both (A) and (B) (D) None of these

9. Which of the following XML documents are well formed?

(A)
```
<firstElement>
  TIME Hyderabad
<secondElement>
  Head Office
</secondElement>
</firstElement>
```
(B)
```
<firstElement>
  TIME Hyderabad
</firstElement>
  Head Office
</secondElement>
<secondElement>
```
(C)
```
<firstElement>
  TIME Hyderabad
<secondElement>
  Head Office
</firstElement>
</second Element
```
(D)
```
</firstElement>
  TIME Hyderabad
</secondElement>
  Head Office
<secondElement>
<firstElement>
```

10. Which of the following XML fragments are well formed?
(A) <myExam mycity = "Hyderabad"/>
(B) <myExam mycity = 'Hyderabad'/>
(C) <myExam mycity = "Hyderabad/">
(D) <myExam mycity = 'Hyderabad/>

11. In the following anchor tag <A HREF = "http://www.time4education.com"> time </A> which one is attribute?
(A) A (B) HREF
(C) time (D) http

12. A link to the document is like this:
– <A HREF =
http://www.time4education/document.html> document </A>
Then the link to proposal section will look like (from within same document)
(A) <A HREF = "#proposal">
(B) <A HREF = "documents.html #proposal">
(C) <A HREF = "→ proposal">
(D) <A HREF = "~ proposal">

13. DTD includes the specifications about the markup that can be used within the document, the specifications consists of all EXCEPT
(A) the browser name
(B) the size of element name
(C) entity declarations
(D) element declarations

14. Every website has a server process listening to TCP port 80 for incoming connections from clients (normally browsers). After a connection has been established the client sends one request and server sends one reply. Then the connection is released. The protocol that defines the legal requests and replies is called
(A) TFTP (B) FTP
(C) gopher (D) HTTP

15. Given below are several usages of the anchor tag in HTML:
I. < A HREF = "http://www.ebay.com/Html/Basic/Pageno.html"> Hello </A>
II. <A HREF = "/Basic /Pageno.html"> Hello </A>
III. <A HREF = "Pageno.html">Hello</A>
IV. <A HREF = "Pageno.html # Hello">Hello</A>
Which of the above are valid?
(A) I and II only (B) I, II, III and IV
(C) I and III only (D) I, II and III only

Practice Problems 2

Directions for questions 1 to 15: Select the correct alternative from the given choices.

1. Which of the following is not case sensitive?
(A) VML (B) XHTML
(C) XML (D) HTML

2. Which of the following statement is false?
(A) W3C stands for World Wide Web consortium.
(B) W3C implements the <layer> tag.
(C) W3C sets the HTML standards.
(D) None of these

3. Which of the following requires a closing tag?
(A) <H_1> (B) <ABBR>
(C) <B> (D) All of these

4. Which of the following does not require a closing tag?
(A) <B> (B) <FONT>
(C) <BR> (D) <ABR>

5. The smallest heading tag in HTML is
(A) <H_0> (B) <H_1>
(C) <H_6> (D) <H_8>

6. The largest size among the heading tags is
(A) H_6 (B) H_5
(C) H_7 (D) H_1

7. The ____ tag is effective for formatting program code or similar information, usually in a fixed font with ample space between words and lines.
(A) <Pre> (B) <Address>
(C) <Blockquote> (D) <Strong>

8. ____ sets the text characteristics for the document.
(A) <font>
(B) <size>
(C) <color>
(D) <basefont>

9. The tag used for creating a row in a HTML table is
(A) <TR> (B) <TD>
(C) <Table row> (D) <TH>

10. The SRC attribute is used to point to a ____ of the image.
(A) folder (B) file
(C) URL (D) pixel

11. How the position of files will be displayed in browser for the following code?
```
<frameset col's = "50%, 50%">
  <frameset rows = "50%, 50%">
    <frame src = "file1.html">
    <frame src = "file3.html">
  </frame set>
    <frame set rows = "50%, 50%">
    <frame src = "file2.html">
    <frame src = "file4.html">
  </frameset?
</frameset>
```

(A)

50	50	
File 1	File 2	30
File 3	File 4	70

(B)

50	50	
File 1	File 3	30
File 2	File 4	70

(C)

70	30	
File 1	File 4	50
File 2	File 3	50

(D)

30	70	
File 1	File 4	50
File 3	File 2	50

12. ‘We may have standalone attributes in XML’. This statement is
(A) True
(B) False
(C) True if it is well-formed
(D) True if it defined in DTD

13. Standalone is one of the possible attributes in the XML declaration. We can set this to ___ if the document does not refer to any external entity.
(A) Yes (B) No
(C) Not required to set it (D) None of these

14. Which of the following is not the difference between HTML and Java script?
(A) HTML is used to create web pages, java script is used to customize the web pages.
(B) HTML provides security where as java script doesn’t provide security.
(C) HTML is more preferable by the clients or users where as java script is not more preferable by the users.
(D) HTML is less efficient than java script.

15. <HMTL> and </HMTL> tags indicates the beginning and ending of the document which are compulsory because these indicate that the software is _____.
(A) processing HTML (B) processing XML
(C) processing URL (D) deprocessing HTML

Previous Years’ Questions

1. Match the following: **[2015]**

(P) Condition coverage	(i) Black-box testing
(Q) Equivalence class partitioning	(ii) System testing
(R) Volume testing	(iii) White-box testing
(S) Alpha testing	(iv) Performance testing

(A) P–ii, Q–iii, R–i, S–iv
(B) P–iii, Q–iv, R–ii, S–i
(C) P–iii, Q–i, R–iv, S–ii
(D) P–iii, Q–i, R–ii, S–iv

2. Which of the following statements is/are FALSE? **[2015]**
I. XML overcomes the limitations in HTML to support a structured way of organizing content.
II. XML specification is not case sensitive while HTML specification is case sensitive.
III. XML supports user defined tags while HTML uses pre-defined tags.
IV. XML tags need not be closed while HTML tags must be closed.
(A) II only (B) I only
(C) II and IV only (D) III and IV only

3. Consider the following C program segment.

```
while(first <= last)
{
   if (array[middle] < search)
              first = middle + 1;
   else if (array[middle] == search)
                found = TRUE;
              else last = middle - 1;
   middle = (first + last)/2;
}
if (first > last) notPresent = TRUE;
```

The cyclomatic complexity of the program segment is ______.

4. A software requirements specification (SRS) document should avoid discussing which one of the following? **[2015]**
(A) User interface issues
(B) Non-functional requirements
(C) Design specification
(D) Interfaces with third party software

5. Consider the basic COCOMO model where E is the effort applied in person-months, D is the development time in chronological months, KLOC is the estimated number of delivered lines of code (in thousands) and a_b, b_b, c_b, d_b have their usual meanings. The basic COCOMO equations are of the form **[2015]**
(A) $E = a_b(\text{KLOC})\exp(b_b)$, $D = c_b(E)\exp(d_b)$
(B) $D = a_b(\text{KLOC})\exp(b_b)$, $E = c_b(D)\exp(d_b)$
(C) $E = a_b\exp(b_b)$, $D = c_b(\text{KLOC})\exp(d_b)$
(D) $E = a_b\exp(d_b)$, $D = c_b(\text{KLOC})\exp(b_b)$

6. Which one of the following statements is NOT correct about HTTP cookies? **[2015]**
(A) A cookie is a piece of code that has the potential to compromise the security of an Internet user.
(B) A cookie gains entry to the user’s work area through an HTTP header.
(C) A cookie has an expiry date and time.
(D) Cookies can be used to track the browsing pattern of a user at a particular site.

7. Which one of the following assertions concerning code inspection and code walkthrough is true? **[2015]**

(A) Code inspection is carried out once the code has been unit tested.
(B) Code inspection and code walkthrough are synonyms
(C) Adherence to coding standards is checked during code inspection
(D) Code walkthrough is usually carried out by independent test team

8. Consider a software project with the following information domain characteristics for calculation of function point metric.

Number of external inputs (I) = 30

Number of external outputs (O) = 60

Number of external inquiries (E) = 23

Number of files (F) = 08

Number of external interfaces (N) = 02

It is given that the complexity weighting factors for I, O, E, F and N are 4, 5, 4, 10 and 7, respectively. It is also given that, out of fourteen value adjustment factors that influence the development effort, four factors are not applicable, each of the other four factors have value 3, and each of the remaining factors have value 4. The computed value of function point metric is ______ **[2015]**

9. In a web server, ten WebPages are stored with the URLs of the form http://www.yourname.com/var.html; where, var is a different number from 1 to 10 for each Webpage. Suppose, the client stores the Webpage with var = 1 (say W_1) in local machine, edits and then tests. Rest of the WebPages remains on the web server. W_1 contains several relative URLs of the form 'var.html" referring to the other WebPages. Which one of the following statements needs to be added in W_1, so that all the relative URLs in W_1 refer to the appropriate WebPages on the web server? **[2015]**

(A) <a href: http://www.yourname.com/", href: "var.html">
(B) <base href: http://www.yourname.com/">
(C) <a href: http://www.yourname.com/">
(D) <base href: http://www.yourname.com/", range: "var.html">

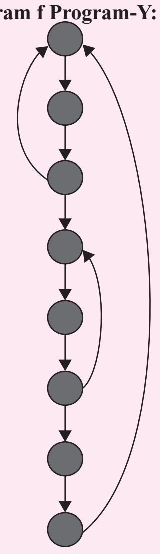
10. Consider a software program that is artificially seeded with 100 faults. While testing this program, 159 faults are detected, out of which 75 faults are from those artificially seeded faults. Assuming that both real and seeded faults are of same nature and have same distribution, the estimated number of undetected real faults is _________. **[2015]**

11. Consider three software items: Program-X, Control Flow Diagram of Program-Y and Control Flow Diagram of Program-Z as shown below **[2015]**

Program-X:

```
Sumcal (int maxint, int value)
{
int result=0, i=0;
if (value <0)
{
Value = -value;
}
While ( (i<value) AND (result <=
    maxint
{
i=i+1;
result = result + 1;
}
if (result <= maxint)
{
printf (result);
}
else
{
printf ("large")
}
printf ("end of program");
}
```

Control Flow Diagram f Program-Y:



Control Flow Diagram of Program-Z:

$$\begin{tikzcd}
\boxed{\text{Control Flow Diagram of Program-}X} \arrow[d] \\
\boxed{\text{Control Flow Diagram of Program-}Y}
\end{tikzcd}$$

The value of McCabe's Cyclomatic complexity of program-X, Program-Y, and Program-Z respectively are

(A) 4, 4, 7 (B) 3, 4, 7
(C) 4, 4, 8 (D) 4, 3, 8

Answer Keys

Exercises

Practice Problems 1

1. D **2.** B **3.** B **4.** A **5.** A **6.** A **7.** B **8.** C **9.** A **10.** A
11. B **12.** A **13.** A **14.** D **15.** D

Practice Problems 2

1. D **2.** B **3.** D **4.** C **5.** C **6.** D **7.** A **8.** D **9.** A **10.** C
11. A **12.** B **13.** A **14.** C **15.** A

Previous Years' Questions

1. C **2.** C **3.** 5 **4.** C **5.** A **6.** A **7.** C **8.** 612 to 613 **9.** B
10. 28 **11.** A