Q.2 a. What are tags? Explain in brief.

Answer:

Any formatted text documents is composed of a set of elements such as paragraphs, headings, and lists. Each element has to be surrounded by control information which tells the presentation or printing software when to switch on a piece of formatting and when to switch it off. In HTML, formatting in specified by using tags. A tag is a format name surrounded by angle brackets. End tags switch a format off also contain a forward slash. For instance, the following example sets the text to the style h1and switches that style off before processing any more of the document:

<h1> text in an h1 style </h1>

- Tags are delimited by angled brackets: <h1>
- They are case sensitive: <HEAD>, <head>, and <hEaD> are not equivalent. In HTML up to version for tags were not case sensitive. XHTML is case sensitive &, furthermore, tags must be in lower case letters;
- Each element is terminated by an end tag. There are a few exceptions to this rule – generally elements which never contain content. Those element which act differently will be identified when all the of the components of XHTML are described letter in this chapter
- White space, tabs, and new lines are ignored by the browser, they can be use to make the HTML source more readable without effecting the way that the page is displayed.
- If a browser does not understand a tag it will usually ignore it.

b. How to insert an image into a Web page? Explain.

Answer:

Images :- Images are the second aspect of a pleasant Web experience. The problems with images are legion if they are not used wisely. First, experienced or inpatients Web surfers often switch image loading off by default, on their browser. If your site relies on an image too get important information across, these people may never see it. Second, loading images is a slow process and if you use too many, or your images are too large, the download times can easily become intolerable.

If you want high quality, good compressions, and lots of color use JPG, for instance when displaying photograph. Generally, though GIF are more common as they tend to be smaller files, lots of software can manipulate them-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 you want to use a single image across the width of a page make it 1281 pixels vide than it can not be tiled horizontally. This is a useful technique if you have an image with a differently colored left edge and want a classy looking page. Background images tend to work best in pale greys and browns, but if they are complex they may hide the text

<img src= “url” | “name” height= “ n” width = “n” [alt= “string”]

[align= “top” | “center” | “bottom”] [use map= “url”] / >

Display 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 browser also use these to shrink / stretch images to fit, generally it is safest to use the correct size for the image.

<area shape=”circle”|”rect”|”poly”|”default”

Href=”URL” coords=”string” alt=”string”>

Creates a click able 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 you do not supply an alt, your 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, spares of integers which gives location on the image in pixels:

- The default location does not require coordinates and is used to indicate what happens if the b
- A circle is defined by its center and its radius. The center is given by a pair of values, the radius by a single value. Therefore requires just 3 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 an example image map with the mapping in the same file as the image link might look like :

1. <image src= “./mappic.gif” usemap= “#main_map”

Height=30width=50/>

5 <map name = “main_map” >

<area shape = “rect” href= “./images/img1.jpg”

Alt = “image one” cords= “0,0,25,25”>

<area shape = “rect” href= “./page 1 . html”

10. alt = “page one “ cords = “26,26,50,50,”>

<area shape = default href = “./page 32.html”

Alt = “page32” >

</map>

It is important that we provide a piece of text to be displayed if the image is not loaded.

c. Explain Iframes with suitable example?

Answer:

Frames do not have to occupy large areas of the screen. A frame can be embedded within a document structuring a large word processed document is to break it into a number of small files which are brought together by including them in a single frame work. Each file can be worked on separately without affecting the whole piece of work. HTML pages can be constructed in exactly the same way. When we look at the programming language Perl and PHP we will see technologies which can build a single HTML page from a set of scripts, each of which is itself a complete document those approaches use dynamic creation of documents by scripts which execute on the web server. The HTML iframe element let's the browser client access document which are addressable through URLs and then display those documents in a single page. URLs can be used to identify many types of resource from images to text pages. These resources may be contents which is static and simply returned by a server or dynamically created on-the-fly before being sent to the browser.

<iframe[name= "string"] src= "uri" [scrolling= "yes"| "auto" | "no"]

[frameborder= "one" | "0"] [height= "nn"] [width [= "nn"]

[marginwidth= "nn"] [marginheight= "nn"] ></iframe>

The Iframe element has many of the same attribute as the non-inlined frame . The most important is src which takes the URI of the page being included as its value. The name attribute gives an optional text string which can be used as the target of hyperlinks. The other attributes control the visual presentation of the Iframe. The Iframe may be presented in an area which is smaller than its content, in which case scroll bar can be added by default, scroll bar are provided when needed but others can specify that they are always used or that they are never used.

We have already seen how frames are used. The inline frame is very similar except that it can be displayed anywhere in the page. This example is very simple.

```
1.  <html>
<head><title> Roman farm management</title></head>
<body>
5.   <Iframe src= "Iframelatin.html" width= "40%" >latin solution </Iframe >

     <h2> of the hands </h2>

10.  <p>LVI the following are the customary allowance for food for the hands, four packs of meal for the winters, and four and one –half for the summer. For the overseer, the housekeeper, the waggoner the shepherc, three pecks each. For the slaves, four pounds of bread for the winter, but when they begin to cultivate the winse this is increased to five ponds until the figs are ripe, than return to four pounds. </p>

15 </body>

</html>
```

Q.3 a. What are Classes? How to use classes in style sheet?

Answer:

CLASSES:-

Selecter . classname {property: value; property: value }

<selector class=classname>

These examples show how classes should be used 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 you want to use a name style the tag is extended by including class = and the unique name.

1. H1. Fred {
   Color:#eeebd2;

   Background –color: #d8a29b;

   Font-family: “bookantiqua”, times, serif;

5. Border: thin groove # 9baab2;

   }

   <h1 claas = “fred”>A simpleheading</h1>

The benefit of classes is that they can provide a lot of variety. They are especially good if you want to redefine the paragraph style so that your introduction looks different from your content.

Anonymous Classes:- Sometimes you want to apply a piece of formatting to many different elements within a page but not necessarily to the entire you could redefine every element in a stylsheet to make it use your formatting, and then redefine individual elements back to their defaults as you needed to. This is a rather awkward approach and would inevitably lead to a lot of duplication of effort.

Including style sheets:-

The following, adapted for your local needs, must be included in the <head > of your html page.

[<style type = “text/css” >]

<!-@import url) ; - ></style>

These lines are both needed if you intend to use more than one stylesheet. The first sheet is included as if it were the only one; any further stylesheets have to be imported. Notice that the @ import is enclosed with in a comment so that it can be easily ignored by older browsers.

1. <linkrel= “stylesheet”

Href = http://www.smiggins.co.uk/mainstyles.css

Type= “text/css” media = “screen”>
<styletype= “text/css”>

5. <! -- @import

url (http://www..smiggins. com/ style.css)

-->

</style>

Properties and values in Styles:-

A number of properties of the text can be altered. These can be grouped together. The best way of discovering how styles work is to play around with some of these properties try giving absurd values to elements and see what happens.

b. Compare and contrast the use of and commands used for formatting of elements.

Answer: Page Number 108 of Text Book

c. Define Cascading styles in brief.

Answer:

Cascading Styles: - Conventionally, styles are cascaded. This means you do not have to use just a single set of styles inside a document-you can import as many style sheets as you like. This is useful if you define a set of organizational stylish that can be modified by each department. The only difficulty with importing multiple style sheets is that they cascade. This means that the first is overridden by the second, the second by the third, and so on. Of-course the overriding only happens if a letter stylesheets contains a definition of a style that is already defined. You can also override stylish by defining styles within the body of the page.

Rules: A style rule has two parts: a selector and a set of declaration. The selector is used to create a link between the rule and HTML tag. The declaration has two parts: a property and a value. Selectors can be placed into classes so that a tag can be formatted in a variety of ways. Declaration must be separated using colons and terminated using semi colons (;)

Q.4 a. What are the different types of operators that are used in Java Scripts?

Answer: Page Number 174 of Text Book

b. What are events? What events can Java Script handle?

Answer: Page Number 231 of Text Book

Q.5 a. Why are rollover buttons so popular among web developers?

Answer: Page Number 252 of Text Book

b. What are difficulties that you might experience if you use Java Script to perform data validation.

Answer: Page Number 238 of Text Book

Q.6 a. What is Scalar in Perl? Also explain the functions which operate on Scalars in brief.

Answer:

The basic data type is called the Scalar. Scalar item are identified by having a $ at the front of their name. So what is a singular piece of data? Well, single data item might be numbers or characters, strings or individual data item inside a structure such as an array.

Because the types of a scalar is not predetermined they can be used rather creatively. Try saving the following script in file called scalar.pl.

```
1. #!/usr/bin/perl –w
$item = 0;
$item = 34* 54.364762;
5. $item .= “fred”;

   Print $item . “\n”;

   Print “$item \n”;
```

- It is initialized as a number with the value 0.
- It then takes the results of the multiplication operation.
- A string is than joint on to the numerical value. Notice the .= construct which is used for certain string concatenation operations.
- The scalar is than displayed, using the print function, with a new line character joined to its end.
- Finally the scalar is displayed form within a string.

Function which operate on scalar:-

A number of Perl functions operate on scalar data. Some of the more useful ones are:-

Chop [(variable)]

This works exactly like chomp but returns the character which was removed

Lc[(parameter)]

This return its parameter with all characters converted to lower case. If the parameter is omitted lc will operate on standard input.

Length [(parameter)]

This return the length in characters of its parameter. Its operate on standard input if no parameter is specified.

q/string/

Places single quotes around the string.

qq/string/

Places double quotes around the string.

Substr (string, offset [, length])

Extracts and returns a substring from the string, starting at the supplied off set. The first character of the expression is at position 0. An optional length may be supplied to indicate how many characters are returned. If this is omitted the substring continues to the end of the string.

Uc[(parameter)]

Returns its parameter with all characters converted to upper case. If the parameter is omitted uc will operate on standard input.

b. How arrays can be used in Perl? Explain with a suitable example.

Answer:

Arrays:- The first of the plural data types is the array. These have an @ before their name and are, broadly, like those which you may have met in java script. Although you have already used arrays, it is probably worth giving a quick refresher on arrays in general before I write about how they are used in perl. If you know your array from your linked list, your stack from your queue, feel free to skip the next bit and leap to the discussion arrays in perl. A couple of warnings: in perl an array can be used as a stack so you need to keep your wits about you, and perl arrays are not types sensitive-we can mix and match numbers, strings, arrays, and hashes as item in an array.

The array is a common, popular, and useful data structure which is found in most programming languages. An array is an ordered list of scalar variables. To access items in an array you use its position in the list. This is called its index. If we take some simple items: “dog”, “cat”, 234, “uncle bill”, we can put them in to an array. The following code crates an array and display its contents:

1. #!/usr/bil/perl –w

```
@firsttarr = (“dog”, “ cat” ,234, “uncle bill”);
Print “@firstarr\in”
Exit (0)
```

Having put our strange list in to an array, it is now ordered and we can access items base upon that order. The first item in the array is at index 0 (Zero), not index (1), many programming languages count from 0 and Perl is no exception. This is not simply a convention design to trap the unwary novice, but is very useful in counting through loops. In Perl, as in java script, you need to get used to counting the first instance of anything as instance 0. The index of the last items in our arrays not the same as the number of items in the array. In the example I have 6^{th} things in the array, yet if the first is at index 0 the last must be at index 5. That is , the last item is at index “number of items in the array” -1.

Array in perl are nice and straight forward with a couple of useful enhancement over the traditional array. You create an array by assigning a list of values:

@myarray = (“dog”, “cat”, “mouse”, 234, “my uncle bill”);

You can also set the value of an individual array item

$myarray [5] = “hourse”;

Perl supplies to operations which allow you to manipulate the last item in the array. These are called push and pop. The following code shows these functions in action:

```
1. #!/usr/bin/perl –w
@myarray = (“dog”, “cat”, 234, “my uncle bill”);
$string = “foobar was my uncle”;
5. Print “@myarray\n”;

   Push (@myarray, $string);

   Print “@myarray\n”;
```

Exit (0): you will not find these operations in a traditional array implementation; rather they are usually reserved for use with stack. However, they are very useful if you many to swap things. If you push into one array and pop them off into another you have quickly reversed their order.

Q.7 a. Write a CGI program that displays a count of how many times a browser on each computer has contacted the server.

Answer: Page Number 252 of Text Book

b. Explain the Use of relational Data Base with Perl.

Answer:

Using Relational Data Base:-

For the developers of large commercial Web Site, two issues override all others: security and data storage. The benefits of using the Web to conduct business disappear totally if you can’t efficiently store, retrieve, and manipulate the data customers. Web Sites which do not handle data are little more than glorified advertising opportunities although they have a rather larger potential audience. However, if you are handling complex data, complex processing or need to keep your data secure than using a relational data base is probably the best solution.

Introduction DBI:-

There are many types of relational database available-some of them are even freely available on the internet. The Multiplicity of such platform could present the end- developers with a problem of potential insurmountable proportions. Imagine developing an application which could talk to an Oracle data base, only for your ISP or company to decide that they were moving from Oracle to SQL server. If your application contained a lot of platform – specific code that you would have to rewrite the whole thing. That is not only time- consuming and expensive but potentially dangerous. All of the debugging and in incremental improvements that had been made to the original oracle application would have to be repeated for the SQL server version. Fortunately, there is a data base - neutral solution available for perl.

RDBMS Neutral Data Base Applications:- The Perl DBI module provides a neutral interface to many relational data bases. It is an application programmer interface (API) which provides a

library of functions, variables and conventions. According to the documentation with the modul these

Provide a consistent data base interface independent of the actual data base being used

The DBI routines do not actually perform much of the processing of the application-database connection. That functionality comes from a driver module which must be specifically developed for the data base being used. Data base drivers are available most of the commonly used relational data bases. The drivers do the actual work while the DBI API provides a frame work with in which those drivers can operate.

Q.8 a. What is the difference between user defined functions and built-in function in PHP?

Answer: Page Number 450, 452 of Text Book

b. Explain with the help of example how arrays are used in PHP.

Answer: Page Number 441 of Text Book

Q.9 Write in detail:
(i) XML documents

Answer:

XML document are composed of just three things: elements, control information, and entities. Let us look at each of those in turn. Most of the markup in an XML document is element markup. Elements are surrounded by tags much as they are in HTML. The content of the document has a single root element which contains all of the other markup. You have already met this idea in HTML where all documents are enclosed inside <html></html> tags. The document is then composed of number of sections, each of which is enclosed between tags. The sections themselves are also elements, of course.

Nesting Tags:-

Even the simplest XML document has nested tags (tags inside tags) unlike HTML these must be nested properly and closed in the reverse of the order in which they were opened. The following code is invalid XML because the order of the tags has become confused, with tags overlapping:

1. <category type =”loaf”>
<name> basic farmhouse</name>
<ingredient></cooking>
5. <serves>

<instructions></serves>

</instructions>

</loaf>

Each XML tag has to have to have a closing tag, again unlike HTML) see section 3.6 for a description of the way that HTML parsing is moving. There is no way that a parser can extract

control information from the structure of the document. In HTML a parser is working within t context of HTML. It is not a general Principal to be elsewhere.

(ii) Document Object Model

Answer: Page Number 533 of Text Book

TEXT BOOK

Web Programming – Building Internet Applications, Chris Bates, Third Edition, Wiley Student Edition, 2006