Q.2 a. What's the difference between h1 and the title tag?

(4)

(6)

Answer:

The title applies to the document as a whole, and h1 to the titles of the major subdivisions within the document. For example, consider a resume. An appropriate title would be "John Doe's Resume", and some likely h1's would be "Objective," "Experience," and "References." In terms of how they're used by browsers, graphical browsers tend to display the title as part of the text appearing in the title bar. Both graphical and text browsers often use it when displaying stored favorites or bookmarks.

b. Demonstrate the use of some HTML tags through an example code.

Answer:

HTML documents have a head which contains control information and a large body. The body contains the content that displays on the screen and tags which control how that content is formatted by the browser. The basic document is demonstrated through example code below:

```
<html>
<head> <title>A Basic Web Page</title> </head>
<body>
<h1>Big on Desserts</h1>
Desserts are good tasting and appealing for you ...
There are many varieties, ...
and here is a short list: 

Apple Pie 
Ice cream
Rabri

</body></html>
```

c. What is HTML, SGML and XML?

(6)

Answer:

- What is HTML?
 This is an SGML (Standard Generalized Markup Language) application conforming to International Standard ISO 8879. HTML is widely regarded as the standard publishing language of the World Wide Web.
- What is SGML?
 This is a language for describing markup languages, particularly those used in electronic document exchange, document management, and document publishing. HTML is an example of a language defined in SGML.
- What is XML? XML is the shorthand name for Extensible Markup Language. XML is a markup language much like HTML and was designed to describe data. XML tags are not predefined. You must define your own tags according to your needs.

Q.3 a. What is the use of using an external style sheet?

(6)

Answer:

By using the Link Tag to load a basic external style sheet (CSS), it's possible to control the look n feel of multiply Websites by making changes to One style sheet. An external style sheet (CSS) is perfect - when the same style / format / look n feel is required on numerous pages. With an external style sheet, the webmaster can change the look of the entire site - by changing one file.

b. Write an example code for an external style sheet.

(6)

Answer:

```
body {
background-color: #FFFFF0;
font-family: Arial, Verdana, sans-serif;
font-size: 18px;
color: #00008B;
a { font-family: Arial, Verdana, sans-serif; font-size: 18px; color: #483D8B; text-
decoration: underline}
a:hover { font-family: Arial, Verdana, sans-serif; font-size: 18px; color: #A52A2A;
background-color: #FAEBD7}
h1 { font-family: Arial, Verdana, sans-serif; font-size: 32px; color: #A52A2A }
h2 { font-family: Arial, Verdana, sans-serif; font-size: 24px; color: #A52A2A }
hr{ color:brown; background-color:tan; width:90%; height:2px; }
table {
font-family: Arial, Verdana, sans-serif;
 font-size: 18px;
 color: #00008B;
 margin-top: 0px;
margin-right: 0px;
margin-bottom: 0px;
margin-left: 0px;
 padding-top: 0px;
 padding-right: 0px;
```

```
padding-bottom: Opx;
padding-left: Opx;
}
.note {
font-family: Arial, Verdana, sans-serif;
font-size: 14px;
color: purple;
font-weight: bold;
}
```

c. What is a style rule? Provide an example.

(4)

Answer:

A style rule has two 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 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. Declarations must be separated using colons and terminated using semicolons.

```
selector {property: value; property: value; property: value;...}
```

This form is used for all style declarations in stylesheets. The declaration has three items: the property, a colon, and the value. If we miss the colon or fail to put the semicolon between declarations, the style cannot be processed. Rules do not have to be formatted. Examples of a simple rule:

```
body {
    background-color: #eebd2;
}
h1 {
    color: #eebd2;
    background-color: #d8a29b;
    font-family: "Book Antiua", Times, serif;
    border: thin groove #9baab2;
}
```

Q.4 a. How to use strings as array indexes using JavaScript?

(4)

Answer:

Javascript does not have a true hashtable object, but you can use the array as a hashtable.

```
<script type="text/javascript">
var days = ["Sunday","Monday","Tuesday","Wednesday",
"Thursday","Friday","Saturday"];
```

```
for(var i=0; i < days.length; i++) {
  days[days[i]] = days[i];
}

document.write("days[\"Monday\"]:"+days["Monday"]);
</script>
This produces
  days["Monday"]:Monday
```

b. JavaScript's anonymous functions and event handling features, along with a couple of little quirks, mean that handling scope in your applications can become frustrating. Explain. (8)

Answer:

When something is global means that it is accessible from anywhere in our code. Take this for example:

```
1.var monkey = "Gorilla";
2.
3.function greetVisitor() {
4.return alert("Hello dear blog reader!");
5.}
```

If that code was being run in a web browser, the function scope would be window, thus making it available to everything running in that web browser window.

As opposed to the global scope, the local scope is when something is just defined and accessible in a certain part of the code, like a function. For instance;

```
1.function talkDirty () {
2.var saying = "Oh, you little VB lover, you";
3.return alert(saying);
4.}
5.alert(saying); // Throws an error
```

If we take a look at the code above, the variable saying is only available within the talkDirty function. Outside of it it isn't defined at all. Note of caution: if we were to declare saying without the var keyword preceding it, it would automatically become a global variable.

What this also means is that if we have nested functions, the inner function will have access to the containing functions variables and functions:

```
1.function saveName (firstName) { 2.function capitalizeName () {
```

```
3.return firstName.toUpperCase();
4.}
5.var capitalized = capitalizeName();
6.return capitalized;
7.}
8.alert(saveName("Robert")); // Returns "ROBERT"
```

As we just saw, the inner function capitalizeName didn't need any parameter sent in, but had complete access to the parameter firstName in the outer saveName function. For clarity, let's take another example:

```
01.function siblings () {
02.var siblings = ["John", "Liza", "Peter"];
03.function siblingCount () {
04.var siblingsLength = siblings.length;
05.return siblingsLength;
06.}
07.function joinSiblingNames () {
08.return "I have " + siblingCount() + " siblings:\n\n" + siblings.join("\n");
09.}
10.return joinSiblingNames();
11.}
12.alert(siblings()); // Outputs "I have 3 siblings: John Liza Peter"
```

Both inner functions have access to the siblings array in the containing function, and each inner function have access to the other inner functions on the same level (in this case, joinSiblingNames can access siblingCount). However, the variable siblingsLength in the siblingCount is only available within that function, i.e. that scope.

c. How to make an array as a stack using JavaScript?

(4)

Answer:

The pop() and push() functions turn a harmless array into a stack

```
<script type="text/javascript">
var numbers = ["one", "two", "three", "four"];
numbers.push("five");
numbers.push("six");
document.write(numbers.pop());
document.write(numbers.pop());
document.write(numbers.pop());
</script>
This produces
sixfivefour
```

Q.5 a. How to open a new window with JavaScript? Describe and provide an example.

(8)

Answer:

To open a new window, we need to use a ready-made JavaScript function. Here is what it looks like:

window.open('url to open', 'window name', 'attribute1, attribute2')

This is the function that allows you to open a new browser window for the viewer to use. Note that all the names and attributes are separated with a comma rather than spaces. Here is what all the stuff inside is:

1.'url to open'

This is the web address of the page you wish to appear in the new window.

2. 'window name'

You can name your window whatever you like, in case you need to make a reference to the window later.

3. 'attribute1, attribute2'

As with a lot of other things, you have a choice of attributes you can adjust.

Below is a list of the attributes you can use:

1. width=300

Use this to define the width of the new window.

2. height=200

Use this to define the height of the new window.

3. resizable=yes or no

Use this to control whether or not you want the user to be able to resize the window.

4. scrollbars=yes or no

This lets you decide whether or not to have scrollbars on the window.

5. toolbar=yes or no

Whether or not the new window should have the browser navigation bar at the top (The back, forward, stop buttons..etc.).

6. location=yes or no

Whether or not you wish to show the location box with the current url (The place to type http://address).

7. directories=yes or no

Whether or not the window should show the extra buttons. (what's cool, personal buttons, etc...).

8. status=yes or no

Whether or not to show the window status bar at the bottom of the window.

9. menubar=yes or no

Whether or not to show the menus at the top of the window (File, Edit, etc...).

10. copyhistory=yes or no

Whether or not to copy the old browser window's history list to the new window.

Example code for opening a new window:

<FORM>

<INPUT type="button" value="New Window!"</pre>

onClick="window.open('http://www.somewebsite.com/jscript/jex5.htm','mywindow','wid th=400,height=200')">

</FORM>

b. Write short code example of an image moving across screen in JavaScript. (8)

Answer:

```
<html>
<head>
<title>Demo of Image moving across screen in JavaScript</title>
<script language='JavaScript' type='text/JavaScript'>
<!--
function reset1(){
clearTimeout(my time);
document.getElementById('i1').style.left= "500px";
document.getElementById('i1').style.top= "100px";
document.getElementById("msg").innerHTML="";
}
function move img(str) {
var x=document.getElementById('i1').offsetTop;
x = x + 100;
document.getElementById('i1').style.top= x + "px";
}
function disp(){
var step=1; // Change this step value
//alert("Hello");
var y=document.getElementById('i1').offsetTop;
var x=document.getElementById('i1').offsetLeft;
if(y < 600) {y= y +step;
document.getElementById('i1').style.top= y + "px"; // vertical movment
}else{
if(x < 800) \{x = x + step;
document.getElementById('i1').style.left= x + "px"; // horizontal movment
function timer(){
disp();
var y=document.getElementById('i1').offsetTop;
var x=document.getElementById('i1').offsetLeft;
document.getElementById("msg").innerHTML="X: " + x + " Y: " + y
my time=setTimeout('timer()',10);
```

```
//-->
</script>
</head>
<body>
<img src=images/help.jpg id='i1' style="position:absolute; left: 500; top: 100;">
<br>
<br>
<br>
<br>
<input type=button onClick=timer() value='Start'>
<input type=button onClick=reset1() value='Reset'>
<div id='msg'></div>
</body>
</html>
```

Q.6 a. What is hash in perl? Explain some of the features of hash with skeleton code.

Answer:

(8)

A hash is a kind of lookup table, it consists of a collection of key-value pairs, where both the key and value are scalars. You can retrieve a value from the hash by providing the key used to enter it.

Although you can have duplicate values in a hash the keys must be unique. If you try to insert the same key into a hash twice the second value will overwrite the first. Hashes do not preserve the order in which data was added to them. They store your data in an efficient manner which does not guarantee ordering. If you need things to be ordered use an array. If you need efficient retrieval use a hash!

Hash names all start with the % symbol. Hash keys are simple scalars. Hash values can be accessed by putting the hash key in curly brackets {} after the hash name (which would now start with a \$ as we're talking about a single scalar value rather than the whole hash. For example to retrieve the value for "simon" from %ages we would use \$ages{simon}.

When you create a hash you can populate it with data from a list. This list must contain an even number of elements which come as consecutive key value pairs.

```
my %hair_colour = ("Simon", "brown", "Iain", "brown", "Conor", "blonde");
print $hair colour{Simon}; # prints brown
```

In the above example note that you do not have to quote the key data (Simon) when retrieving data from a hash, even though it is a bare string.

Because the syntax used above for populating the hash doesn't make it clear which scalars are keys and which are values Perl allows an alternative syntax for writing lists.

In this version you use the => operator in place of a comma (it's also known as a fat comma). This has the same effect as a comma, and in addition it also automatically quotes the value to its left so you don't need to put quotes around the key names. The code below does exactly the same thing as the one above.

```
my %hair_colour = (Simon => "brown",
Iain => "brown",
Conor => "blonde",);
```

print \$hair_colour{Simon}; # prints brown

This version makes it much clearer which are the keys and which are the values. One useful tip is that when you create a list this way you can leave a trailing comma at the end of the list. This means that if you later come back and add an extra key-value pair you are unlikely to forget to put in the extra comma which would be required.

b. What is the meaning of "Automatic Data Context" in perl? Illustrate through example. (8)

Answer:

Perl makes programming easier by detecting the context within which a variable is used and automatically converts its value appropriately.

```
For example you can use strings as numbers and vice versa. $str1 = "12.5"; $str2 = "2.5"; $sum = $str1 + $str2; ## adding as numbers (A) print "$sum\n"; ## displaying (B)
```

We used the strings as numbers on line A and the \$sum, which is a number, as a string on line B.

The assignment \$len = @arr uses @arr in a scalar context and turns its value to the length of the array. The automatic conversion of a hash to an array on line 5 is another example.

In a CGI program, the need to convert strings to numbers arises often. For example:

```
$total = param('payment'); ## a string representing a number if ( $total > 50 ) ## number context
{ ... }
```

not represent a number to a number results in the number zero. Thus if \$str != 0 is true then \$str is a valid nonzero number.

Q.7 a. Why is the Common Gateway Interface (CGI) vulnerable?

(8)

Answer:

The Common Gateway Interface (CGI) suffers for a lot of vulnerabilities but one of the most exploited vulnerability is the user input. The user input allows users to input data into Common Gateway Interface (CGI) scripts, this makes it do other things that where not originally intended. File execution and ability to read files are among these. Special user input can force these programs to perform functions that there not suppose to do. Functions such as if the user input were the parameter to a program like "echo". Permissions of the echo have many catastrophic consequences below is an example of one of them.

Normal Echo Use. This is simple a script that executes a string to the window. Echo "Text Goes Here" >> testing

Modifying Echo Parameters To Execute Commands. This script will echo the string to the terminal, and then cat the passwd file. The passwd file is where the passwords are stored on the target system.

Echo "Text Goes Here"; cat /etc/passwd

Remember never trust user input. Reason for this is because many Common Gateway Interfaces (CGI) scripts do not check user input for unrecognized characters. Some do it partially; this still leaves potential holes to exploit. The sample I have listed above is a very easy but effect sample of what you can do with a Common Gateway User Interface (CGI) script.

b. Discuss how to perl handles a relational database with example. (8)

Answer:

In accessing relational databases Perl has a de-facto standard library called DBI or Database independent interface for Perl.

Architecture

The Perl scripts use DBI, which in turn uses the appropriate Database Driver (e.g. DBD::Oracle for Oracle, DBD::Pg for PostgreSQL and DBD::SQLite to access SQLite). Those drivers are compiled together with the C client libraries of the respective database engines. In case of SQLite, of course all the database engine gets embedded in the perl application.

Simple example

#!/usr/bin/perl

```
use strict;
use warnings;
use DBI;
my $dbfile = "sample.db";
           = "dbi:SQLite:dbname=$dbfile";
my $dsn
           = "";
my $user
my $password = "";
my $dbh = DBI->connect($dsn, $user, $password, {
 PrintError \Rightarrow 0,
               =>1,
 RaiseError
  AutoCommit \Rightarrow 1,
 FetchHashKeyName => 'NAME_lc',
# ...
$dbh->disconnect;
```

Q.8 a. Write a PHP script to create a form that would be used to collect a person's username and password. The form would use the POST method to pass these details to this php script which would be called login.php (8)

Answer:

```
<?php
include 'config.php';
// establish connection to MySQL and select database
                                 $db username, $db password)
                                                                       die
                                                                             ('Error
$db=@mysql connect($db host,
connecting to MySQL server!');
@mysql_select_db($db_username, $db) or die ('Error selecting database.');
function testAuthentication($username, $password)
       $auth result = mysql query("SELECT
                                                      FROM
                                                               members
                                                                           WHERE
SO='$username' AND password='$password''') or die('Error.');
       return mysql fetch row($auth result) != false;
// ensure username and password were entered
```

```
if (!isset($_PO$T['username']) || !isset($_PO$T['password']))
{
          die('Please go back and enter both your username and password.');
}

// user details

$username = ($_PO$T['username']);
$password = ($_PO$T['password']);

// test authentication

if (!testAuthentication($username, $password))

{
          mysql_close($db);
          die('Invalid username or password.');
}

?>
```

b. Please write a PHP script to check whether the user has entered information in all the required fields in a form. Also demonstrate the use of array_search() function. (8)

Answer:

```
function have_required($array , $required fields) {
  foreach($required fields as $field) {
    if (empty ($array [$field])) return false;
  return true;
if ($submitted)
  echo 'You ';
  echo have_required($_POST, array('name', 'email_address')) ? 'did' :
'did not';
  echo ' have all the required fields.';
2>
<form action="<?= $PHP SELF; ?>" method="POST">
   Name: <input type="text" name="name" /><br />
   Email address: <input type="text" name="email address" /><br />
   Age (optional): <input type="text" name="age" />
 <input type="submit" value="submit" name="submitted" />
  </form>
```

A_variation on in_array() is the array_search() function. While in_array() returns true if the value is found, array_search() returns the key of the found element:

```
$person = array('name' => 'Farid', 'age' => 35, 'wife' => 'Uma');
$k = array_search($person, 'Uma');
echo("Farid's $k is Uma\n");
Farid's wife is Uma
```

The array_search() function also takes the optional third *strict* argument, which requires the types of the value being searched for and the value in the array to match.

Q.9 a. What are DTD – Entities? Explain with examples.

(8)

Answer:

Entities are variables used to define shortcuts to standard text or special characters. Entity references are references to entities. Entities can be declared internal or external.

```
An Internal Entity Declaration
Syntax
```

```
<!ENTITY entity-name "entity-value">
Example
```

DTD Example:

```
<!ENTITY writer "Donald Duck.">
<!ENTITY copyright "Copyright W3Schools.">
```

XML example:

<author>&writer;©right;</author>

Note: An entity has three parts: an ampersand (&), an entity name, and a semicolon (;).

An External Entity Declaration

Syntax

```
<!ENTITY entity-name SYSTEM "URI/URL"> Example
```

DTD Example:

```
<!ENTITY writer SYSTEM "http://www.w3schools.com/entities.dtd"> <!ENTITY copyright SYSTEM "http://www.w3schools.com/entities.dtd">
```

XML example:

<author>&writer;©right;</author>

b. What is an XML namespace?

(4)

Answer:

An XML namespace is a collection of names, identified by a URI reference, which are used in XML documents as element types and attribute names. XML namespaces differ from the "namespaces" conventionally used in computing disciplines in that the XML version has internal structure and is not, mathematically speaking, a set. These issues are discussed in "A. The Internal Structure of XML Namespaces".

What this means, basically, is that the validating rules for some elements are defined in one place, and some others in another.

The RDF (Resource Description Framework), on the other hand, is specifically designed to be a framework for various parties to share data using a common set of XML elements. In the Bibliographic world, there is another framework (called the Dublin Core) which is often used in conjunction with RDF. This is far more complex, with multiple markup vocabularies, so requires namespaces - which requires schema.

c. Write a short note on "Using CSS with XML"

(4)

Answer:

Let's assume that you directly want to render XML contents in a browser. This content may be text-centric XML that you created yourself, contents using a document standard, contents that are pulled out of database, or that are obtained through a web service, e.g. a simple RSS news feed. In most cases, you would use XSLT for styling, i.e. translate XML to HTML or another markup language. However, using CSS is easier and can do a perfect job for some text-centric contents.

CSS for XML is in no way different from CSS for HTML, but you need a browser that implements most of CSS 2 (all modern browsers do).

The only practical difference is that HTML includes default style for each element, e.g. a "h1" title would show in a big font. An XML elements like "header1" would not show at all. XML elements don't have any default styling. In other words, you will have to define properties for each of your elements. Since CSS implements cascading, you may define defaults for the root elements, or else use the "*" selector.

CSS sometimes is used to make XML editing easier. I.e. the Schema author would produce a CSS that is then used to create a sort of WYSYWIG editing interface. With respect to this functionality, XSLT cannot replace CSS as a rendering solution.

TEXT BOOK

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