

**Q.2 a. What are HTML tags? Discuss about the attributes of the following HTML tags**

**(i) <font>**

**(ii) <img>**

**(iii) <a>**

**(iv) <div>**

**(v) <table>**

**(vi) <tr>**

**(4+6)**

**Answer:**

HTML markup tags are usually called HTML tags

- HTML tags are keywords (tag names) surrounded by angle brackets like <html>
- HTML tags normally come in pairs like <b> and </b>
- The first tag in a pair is the start tag, the second tag is the end tag
- The end tag is written like the start tag, with a forward slash before the tag name

(i) <font>

The HTML font tag is used to specify the font to use.

```
<font size="3" color="red">This is some text!</font>
```

```
<font size="2" color="blue">This is some text!</font>
```

```
<font face="verdana" color="green">This is some text!</font>
```

(ii) <img>

The HTML img tag is used for embedding images into an HTML document.

- To use this tag, the image you specify in the src attribute needs to be available on a web server. This could be your own web server or another web server (such as PhotoBucket).
- For more information on creating graphics for the web, check out the web graphics tutorial.
- To link an image to another web page (or a larger version of the image), you simply nest the <img> tag inside an <a> tag. Here's an example of linking an image.
- You can also link your image to an image map.

(iii) <a>

The HTML `a` tag is used for creating a hyperlink to either another document, or somewhere within the current document.

You can use the `a` tag to link text or images. To link an image, simply nest your `img` tag inside the `a` tag.

(iv) `<div>`

The HTML `div` tag is used for defining a section of your document. With the `div` tag, you can group large sections of HTML elements together and format them with CSS. For example, you could have your navigation section wrapped in one `div` and your main content section in another `div`.

(v) `<table>`

The HTML `<table>` tag is used for defining a table. The `table` tag contains other tags that define the structure of the table.

### Table Elements

Tables consist of the `<table>` element as well as other table-related elements. These other elements are nested inside the `<table>` tags to determine how the table is constructed.

The children of a `<table>` element must be, in order:

1. Zero or one `<caption>` elements
2. Zero or one `<colgroup>` elements
3. Zero or one `<thead>` elements
4. Zero or one `<tfoot>` elements (but only if the last element in the table is not a `<tfoot>` element)
5. Either:
  - Zero or more `<tbody>` elements, or
  - One or more `<tr>` elements (these represent the table rows)
6. Zero or one `<tfoot>` elements (but only if there are no other `<tfoot>` elements in the document)

(vi) `<tr>`

Each `<tr>` element represents a row in the table. A row can have one or more `<td>` or `<th>` elements, which determine the columns in the table. Specifically, `<td>` represents table data and `<th>` represents a table header.

|         |   |
|---------|---|
| Example | <pre>&lt;table border = "1"&gt; &lt;tr&gt; &lt;td&gt;Cell 1&lt;/td&gt; &lt;td&gt;Cell 2&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt;</pre> |
|---------|---|

The HTML <tr> tag is used for specifying a table row within a table. A table row contains one or more td and/or th tags which

**b. What is the role of HTML Forms? (6)**

**Answer:**

HTML form on a [web page](#) allows a user to enter data that is sent to a [server](#) for processing. Forms can resemble [paper](#) or [database](#) forms because web users fill out the forms using checkboxes, [radio buttons](#), or [text fields](#). For example, forms can be used to enter [shipping](#) or [credit card](#) data to order a product, or can be used to retrieve search results from a [search engine](#).

Forms are enclosed in the HTML `form` tag. This tag specifies the [communication endpoint](#) the data entered into the form should be submitted to, and the method of submitting the data, GET or POST.

The most important form element is the <input> element.

The <input> element is used to select user information.

An <input> element can vary in many ways, depending on the type attribute. An <input> element can be of type text field, checkbox, password, radio button, submit button, and more.

**Q.3 a. Write HTML document with CSS, such that when you position the mouse over a link it should display crosshair cursor. (3)**

**Answer:**

CSS can be added to HTML in the following ways:

- Inline - using the style attribute in HTML elements
- Internal - using the <style> element in the <head> section
- External - using an external CSS file

The preferred way to add CSS to HTML, is to put CSS syntax in separate CSS files. Cascading Style Sheets (CSS) is a [style sheet language](#) used for describing the [presentation semantics](#) (the look and formatting) of a document written in a [markup language](#). CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the [layout](#), [colors](#), and [fonts](#). It is much more efficient and convenient to create separate files for your Cascading Style Sheets when the properties apply to all pages. Using CSS in this way, you can have all your h1's red on all your pages and change them to green by just changing the style in one file. To do this, you create a file with the extension .css, include style "rules"

for each HTML element you want to define, and use a link element within the head element in the format of the following example.

- b. What do you understand by ‘style’? What are the benefits of using styles compared with placing formatting directly into the text of webpage? State the different styles which can be applied to HTML document. (10)**

**Answer:**

- c. What shortcomings of cascading style sheets does CSS2 address? (3)**

**Answer:**

- Q.4 a. What are the governing rules for forming name of JavaScript variable? JavaScript has four types of data, discuss about them. (8)**

**Answer:**

Follow these rules when naming variables:

- A variable name must be of the form `$name` or `_${name}`.
- The string name can consist of letters (a-z and A-Z), numbers (0-9), and underscore characters (`_`).
- The first character of name must be an alphabetic character (a-z or A-Z).
- If the `_${name}` form is used, spaces can also appear in name. This allows you to use multiple word names without the underscore character.

The Java programming language defines the following kinds of variables:

**Instance Variables (Non-Static Fields)** Technically speaking, objects store their individual states in "non-static fields", that is, fields declared without the static keyword. Non-static fields are also known as instance variables because their values are unique to each instance of a class (to each object, in other words); the currentSpeed of one bicycle is independent from the current Speed of another.

**Class Variables (Static Fields)** A class variable is any field declared with the static modifier; this tells the compiler that there is exactly one copy of this variable in existence, regardless of how many times the class has been instantiated. A field defining the number of gears for a particular kind of bicycle could be marked as static since conceptually the same number of gears will apply to all instances. The code `static int numGears = 6;` would create such a static field. Additionally, the keyword `final` could be added to indicate that the number of gears will never change.

**Local Variables** Similar to how an object stores its state in fields, a method will often store its temporary state in local variables. The syntax for declaring a local variable is similar to declaring a field (for example, `int count = 0;`). There is no special keyword designating a variable as local; that determination comes entirely from the location in which the variable is declared — which is between the opening and closing braces of a method. As such, local variables are only visible to the methods in which they are declared; they are not accessible from the rest of the class.

Parameters You've already seen examples of parameters, both in the Bicycle class and in the main method of the "Hello World!" application. Recall that the signature for the main method is `public static void main(String[] args)`. Here, the `args` variable is the parameter to this method.

**b. Discuss about String manipulation functions and Math Library in JavaScript. (8)**

**Answer:**

```
strcpy (char dest[], const char source[]);
```

This function copies the string `source` into the string `dest`, including the end of string terminator `'\0'`. The `dest` array must be large enough to hold all of the characters in `source` array.

[Click here for an example](#)

```
strncpy (char dest[], const char source[], int n);
```

This function copies `n` characters from the string `source` into the string `dest`, including the end of string terminator `'\0'`. The `dest` array must be large enough to hold all of the characters copied from the `source` array.

[Click here for an example](#)

```
strcat (char s1[], const char s2[]);
```

This function appends the string `s2` onto the end of string `s1`, including the end of string terminator `'\0'`. The array `s1` must be large enough to hold the extra characters being appended from the `s2` array.

[Click here for an example](#)

```
strncat (char s1[], const char s2[], int n);
```

This function appends `n` characters from the string `s2` onto the end of string `s1`, including the end of string terminator `'\0'`. The array `s1` must be large enough to hold the extra characters being appended from the `s2` array.

```
int strcmp (const char s1[], const char s2[]);
```

This function compares the strings `s1` and `s2`. If `s1` and `s2` are identical then the function will return 0. If `s1` is alphabetically lower than `s2` then a negative value is returned. If `s1` is alphabetically higher than `s2` then a positive value is returned.

JavaScript supports many [arithmetic operations](#) as well as the following math functions

```
Math.abs(a) // the absolute value of a
Math.acos(a) // arc cosine of a
Math.asin(a) // arc sine of a
Math.atan(a) // arc tangent of a
Math.atan2(a,b) // arc tangent of a/b
Math.ceil(a) // integer closest to a and not less than a
Math.cos(a) // cosine of a
Math.exp(a) // exponent of a (Math.E to the power a)
Math.floor(a) // integer closest to a, not greater than a
Math.log(a) // log of a base e
```

Math.max(a,b) // the maximum of a and b  
 Math.min(a,b) // the minimum of a and b  
 Math.pow(a,b) // a to the power b

**Q.5 a. Write a JavaScript to handle mouse rollover event on a button in a page. (8)**

**Answer:**

Events of types mouseover (and mouseout) occur when a mouse comes over(goes out) an element. For these events, the mouse goes from one element to another, so two elements are actually engaged. Both elements can be retrieved from in event properties.

mouseover

The element under the pointer is event.target(IE: srcElement).

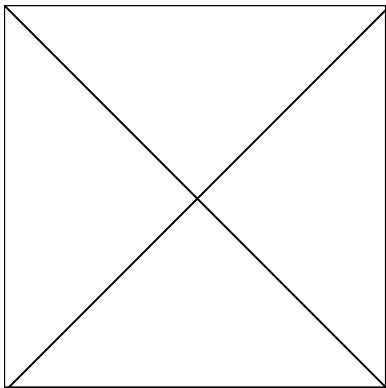
The element the mouse came from is event.relatedTarget(IE: fromElement)

mouseout

The element the mouse came from is event.target(IE: srcElement).

The element under the pointer is event .related Target(IE: event.toElement).

As you can see, the W3C specification joins both from Element and to Element into the single property relatedTarget which serves as from Element for mouse over and to Element for mouseout.



```
<img src      = "polar-green.png"
  alt      = "house icon"
  on Mouse Over = "this.src = 'polar-red.png';"
  on Mouse Out  = "this.src = 'polar-green.png';">
```

**b. Write a function to demonstrate moving of images on the screen. (8)**

**Answer:**

```
<html>
```

```
<head>
```

```
<title>Page Under Construction</title>
```

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-8859-1">
```

```
<style type="text/css">
body {
background-color:#d7df29;
width: 100%;
height: 100%;
}
.img {
text-align: center;
margin: 0 auto
vertical-align: middle;
top:500;
}
#html5 {
position:absolute;
top:0;
right:0;
}
</style>
</head>
<body>
<div class="img"></div>
<div id="html5"> </div>

</body>
```

</html>

**Q.6 a. What is scalar in perl? Discuss about the functions which operate on scalars.**

(i) **substr ( )**

(ii) **uc ( )**

(iii) **chomp ( )**

(iv) **chop ( )**

(v) **lc ( )**

(vi) **length ( )**

(8)

**Answer:**

The most basic kind of variable in Perl is the scalar variable. Scalar variables hold both strings and numbers, and are remarkable in that strings and numbers are completely interchangeable.

Perl's string functions:

**chomp(STRING) OR chomp(ARRAY) --** Uses the value of the `$/` special variable to remove endings from STRING or each element of ARRAY. The line ending is only removed if it matches the current value of `$/`.

**chop(STRING) OR chop(ARRAY)--** Removes the last character from a string or the last character from every element in an array. The last character chopped is returned.

**lc(STRING) --** Returns a string with every letter of STRING in lowercase. For instance, `lc("ABCD")` returns "abcd".

**length(STRING) --** Returns the length of STRING.

**substr(STRING, OFFSET, LENGTH) --** Returns a portion of STRING as determined by the OFFSET and LENGTH parameters. If LENGTH is not specified, then everything from OFFSET to the end of STRING is returned. A negative OFFSET can be used to start from the right side of STRING.

**uc(STRING) --** Returns a string with every letter of STRING in uppercase. For instance, `uc("abcd")` returns "ABCD".

**b. Write a perl program for displaying all elements of array. Discuss about functions which operate on Array.** (8)

**Answer:**

Each element of the array can be indexed using a scalar version of the same array. When an array is defined, PERL automatically numbers each element in the array beginning with zero. This phenomenon is termed array indexing.

```
#!/usr/bin/perl
```

```
print "content-type: text/html \n\n"; #HTTP HEADER
```



```
# DEFINE AN ARRAY
@coins = ("Quarter","Dime","Nickel");

# PRINT THE WHOLE ARRAY
print "@coins";

# PRINT EACH SCALAR ELEMENT
print "<br />";
print $coins[0]; #Prints the first element
print "<br />";
print $coins[1]; #Prints the 2nd element
print "<br />";
print $coins[2]; #Prints the 3rd element
```

PERL-*Sorting Arrays*

PERL-*Replacing Array Elements*

PERL-*Slicing Array Elements*

PERL-*Adding and Removing Elements*

PERL-*Finding the length of an Array*

**Q.7 a. What is CGI? Explain its working with a neat diagram. (8)**

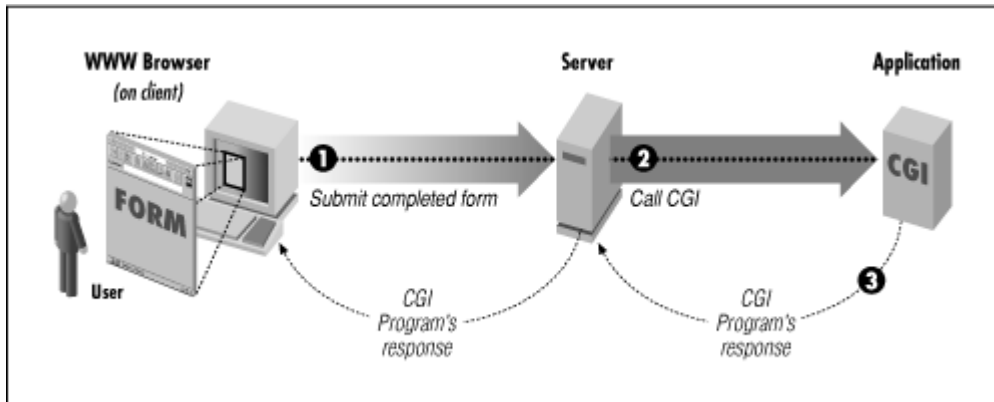
**Answer:**

Common Gateway Interface (CGI) is a standard method for web server software to delegate the generation of web content to executable files. Such files are known as CGI scripts or simply CGIs; they are usually written in a scripting language

CGI is the part of the Web server that can communicate with other programs running on the server. With CGI, the Web server can call up a program, while passing user-specific data to the program (such as what host the user is connecting from, or input the user has supplied using HTML form syntax). The program then processes that data and the server passes the program's response back to the Web browser.

CGI isn't magic; it's just programming with some special types of input and a few strict rules on program output. Everything in between is just programming. Of course, there are special techniques that are particular to CGI, and that's what this book is mostly about. But underlying it all is the simple model shown in Figure 1.1.

Figure 1.1: Simple diagram of CGI



**b. Explain the following applications of CGI: forms, gateways and virtual documents. (8)**

**Answer:**

**Forms:** One of the most prominent uses of CGI is in processing forms. Forms are a subset of HTML that allow the user to supply information. The forms interface makes Web browsing an interactive process for the user and the provider. Generally, forms are used for two main purposes. At their simplest, forms can be used to collect information from the user. But they can also be used in a more complex manner to provide back-and-forth interaction. For example, the user can be presented with a form listing the various documents available on the server, as well as an option to search for particular information within these documents. A CGI program can process this information and return document(s) that match the user's selection criteria.

**Gateways:** Web gateways are programs or scripts used to access information that is not directly readable by the client. For example, say you have an Oracle database that contains baseball statistics for all the players on your company team and you would like to provide this information on the Web. How would you do it? You certainly cannot point your client to the database file (i.e., open the URL associated with the file) and expect to see any meaningful data

**Virtual Documents:** Virtual, or dynamic, document creation is at the heart of CGI. Virtual documents are created on the fly in response to a user's information request. You can create virtual HTML, plain text, image, and even audio documents.

**Q.8 a. What are the differences between Perl and PHP when handling with Web pages? (8)**

**Answer:**

PHP is HTML-embedded: Although it is possible to use PHP for arbitrary tasks by running it from the command line, it is more typically connected to a Web server and used for producing Web pages. If you are used to writing CGI scripts in Perl, the main difference in PHP is that you no longer need to explicitly print large blocks of static

HTML using print or heredoc statements and instead can simply write the HTML itself outside of the PHP code block.

No @ or % variables: PHP has one only kind of variable, which starts with a dollar sign (\$). Any of the datatypes in the language can be stored in such variables, whether scalar or compound.

Arrays versus hashes: PHP has a single datatype called an array that plays the role of both hashes and arrays/lists in Perl.

Specifying arguments to functions: Function calls in PHP look pretty much like subroutine calls in Perl. Function definitions in PHP, on the other hand, typically require some kind of list of formal arguments as in C or Java which is not the case in PERL.

Variable scoping in functions: In Perl, the default scope for variables is global. This means that top-level variables are visible inside subroutines. Often, this leads to promiscuous use of globals across functions. In PHP, the scope of variables within function definitions is local by default.

No module system as such: In PHP there is no real distinction between normal code files and code files used as imported libraries.

Break and continue rather than next and last: PHP is more like C language and uses break and continue instead of next and last statement.

No elsif: A minor spelling difference: Perl's elsif is PHP's elseif.

More kinds of comments: In addition to Perl-style (#) single-line comments, PHP offers C-style multiline comments (/\* comment \*/) and Java-style single-line comments (// comment).

Regular expressions: PHP does not have a built-in syntax specific to regular expressions, but has most of the same functionality in its "Perl-compatible" regular expression functions.

**b. What's the Difference Between Storing Your Data in Cookies and in Session Variables? When to use Sessions rather than Cookies? (8)**

**Answer:**

The main difference between cookies and sessions is that cookies are stored in the user's browser, and sessions are not. This difference determines what each is best used for.

A cookie can keep information in the user's browser until deleted. If a person has a login and password, this can be set as a cookie in their browser so they do not have to re-login to your website every time they visit. You can store almost anything in a browser cookie. The trouble is that a user can block cookies or delete them at any time. If, for example,

your website's shopping cart utilized cookies, and a person had their browser set to block them, then they could not shop at your website.

Sessions are not reliant on the user allowing a cookie. They work instead like a token allowing access and passing information while the user has their browser open. The problem with sessions is that when you close your browser you also lose the session. So, if you had a site requiring a login, this couldn't be saved as a session like it could as a cookie, and the user would be forced to re-login every time they visit.

1. When you need the data stored on the server and not your user's browser
2. When the data is transient, and only relevant for the current browsing session
3. When the data does not contain any information that needs to be securely kept

**Q.9 a. What are the differences between general markup schemes such as XML and proprietary systems such as Rich Text Format? (5)**

**Answer:**

XML (Extensible Markup Language) is a meta markup language that is now widely used. XML was developed by the World Wide Web Consortium, in a committee created and chaired by Jon Bosak. The main purpose of XML was to simplify SGML by focusing on a particular problem — documents on the Internet.<sup>[10]</sup> XML remains a meta-language like SGML, allowing users to create any tags needed (hence "extensible") and then describing those tags and their permitted uses.

XML adoption was helped because every XML document can be written in such a way that it is also an SGML document, and existing SGML users and software could switch to XML fairly easily. However, XML eliminated many of the more complex and human-oriented features of SGML to simplify implementation environments such as documents and publications. However, it appeared to strike a happy medium between simplicity and flexibility, and was rapidly adopted for many other uses. XML is now widely used for communicating data between applications.

The Rich Text Format specification provides a format for text and graphics interchange that can be used with different output devices, operating environments, and operating systems.” Like “.doc,” RTF is a proprietary file format, but it has been designed to be more widely compatible and transferable. In fact, RTF is often the best way to preserve a document’s layout and presentation when Word is not available. Its major drawback is its large file size, especially with documents containing many graphics. Other formats, such as PDF and HTML, produce smaller files, but may require more work to preserve formatting and maintain accessibility.

When posting documents to the Web, it is always important to supply alternative file formats. In addition to “.doc,” you should consider formats such as HTML, PDF, and RTF. Of the three, RTF is perhaps the simplest to create.

b. List three advantages of using Perl to manipulate XML.

(6)

Answer:

```
<event>
<date>03-05-02</date>
<start-time>09:00</start-time>
<end-time>10:00</end-time>
<type>Meeting</type>
<title>Staff Meeting</title>
<description>Weekly staff meeting.</description>
<location>Conference Room</location>
<reminder>
<status>yes</status>
<interval>1-day</interval>
<method>e-mail</method>
</reminder>
</event>
```

c. Why does XML use both streaming and tree-based parsing?

(5)

Answer:

### TEXT BOOK

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