तेजस्वि नावधीतमस्तु
ISO 9001:2008 & 14001:2004

FAIRFIELD
Institute of Management & Technology
Managed by 'The Fairfield Foundation'
( Affiliated to GGSIP University, New Delhi )

## Paper Code: BCA-313
## Paper ID: 20313
## Paper: Web Based Programming (PHP)

**Aim :** To highlight the features of different technologies involved in Web Development

**Objectives**

- Students should be able to design and implement a basic website.
- Students should be able to implement different navigation strategies.
- Students should be able to develop simple back-end database to support a website.
- Students should be able to recognize and evaluate website organizational structure and design elements.

### UNIT – I

Introduction to web applications, HTML, Client Side Scripting Vs Server Side Scripting, Web Servers : Local Servers and Remote Servers, Installing Web servers, Internet Information Server (IIS) and Personal Web Server (PWS). Static website vs Dynamic website development.

### UNIT – II

Introduction to PHP, Start and End Tags of PHP, Data types in PHP, Variables, Constants, operators and Expressions, printing data on PHP page, Control statements – if, switch case, for, while, do while.

Arrays: Initialization of an array, Iterating through an array, Sorting arrays, Array Functions, Functions: Defining and Calling Functions, Passing by Value and passing By references, Inbuilt Functions.

### UNIT – III

Working with Forms: Get and Post Methods, Querystrings, HTML form controls and PHP, Maintaining User State: Cookies, Sessions, And Application State.

Working with Files: Opening and Closing Files, Reading and Writing to Files, Getting Information on Files

### UNIT – IV

PHP Database Connectivity: Introduction to MYSQL, Creating database and other operations on database, connecting to a database, Use a particular database, Sending query to database, Parsing of the query results, Checking data errors.

# UNIT-I

## 1. Introduction to web applications

A Web application is a dynamic extension of a Web server. There are two types of Web applications:

- Presentation-oriented. A presentation-oriented Web application generates dynamic Web pages containing various types of markup language (HTML, XML, and so on) in response to requests.
- Service-oriented. A service-oriented Web application implements the endpoint of a fine-grained Web service. Service-oriented Web applications are often invoked by presentation-oriented applications.

## 2. HTML.

HTML (Hypertext Markup Language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element (but many people also refer to it as a tag). Some elements come in pairs that indicate when some display effect is to begin and when it is to end.

## 3. Client Side Scripting Vs Server Side Scripting

| **Client Side Scripting** | **Server Side Scripting** |
|---|---|
| 1. Client side scripting is used when the users' browser already has all the code and the page is altered on the basis of the users input. | 1.Server side scripting is used to create dynamic pages based a number of conditions when the users browser makes a request to the server |
| 2.The Web Server executes the server side scripting that produces the page to be sent to the browser | 2. The Web Browser executes the client side scripting that resides at the user's computer. |
| 3.The browser receives the page sent by the server and executes the client-side scripts | 3.Server executes server-side scripts to send out a page but it does not execute client-side scripts |
| 4.Client side scripting cannot be used to connect to the databases on the web server | 4. Server side scripting is used to connect to the databases that reside on the web server. |
| 5.Client side scripting can't access the file system that resides at the web server | 5. Server side scripting can access the file system residing at the web server. |
| 6. The files and settings that are local at the user's computer can be accessed using Client side scripting. | 6. The settings that belong to Web server can be accessed using Server side scripting. |
| 7. Client side scripting is possible to be blocked by the user. | 7.Server side scripting can't be blocked by the user |
| 8. Response from a client-side script is faster as compared to a server-side script | 8.Response from a server-side script is slower as compared to a client-side script because the |

| | |
|---|---|
| because the scripts are processed on the local computer. | scripts are processed on the remote computer |
| 9.Examples of Client side scripting languages : Javascript, VB script, etc. | 9. Examples of Server side scripting languages : PHP, JSP, ASP, ASP.Net, Ruby, Perl n many more. |

## 4. Web Servers

Web servers are computers that deliver (*serves up*) Web pages. Every Web server has an IP address and possibly a domain name. For example, if you enter the URL *http://www.pcwebopedia.com/index.html* in your browser, this sends a request to the Web server whose domain name is *pcwebopedia.com*. The server then fetches the page named *index.html* and sends it to your browser.

http://whatismyipaddress.com/web-server

## 5. Local Servers

A local server plays a pivotal role for most developers. It allows development and testing on a local machine without the concerns of losing Internet connection, accidentally creating temporary security holes, or constantly uploading files via FTP. Depending on the option you choose, setup will be more or less difficult, but it is a necessary foundation for your development environment.

## 5 Remote Servers

Computer that is not attached to a user's keyboard but over which he or she has some degree of control, whether it is in the same room, another part of a building, or another part of the world. Also called remote computer or remote host.

## 6. Internet Information Server (IIS)

IIS (Internet Information Server) is a group of Internet servers (including a Web or Hypertext Transfer Protocol server and a File Transfer Protocol server) with additional capabilities for Microsoft's Windows

NT and Windows 2000 Server operating systems. IIS is Microsoft's entry to compete in the Internet server market that is also addressed by Apache, Sun Microsystems, O'Reilly, and others. With IIS, Microsoft includes a set of programs for building and administering Web sites, a search engine, and support for writing Web-based applications that access databases. Microsoft points out that IIS is tightly integrated with the Windows NT and 2000 Servers in a number of ways, resulting in faster Web page serving.

http://searchwindowsserver.techtarget.com/definition/IIS

## 7. personal web server

A **personal web server** is an appliance that allows users to store and access private files remotely from any device connected to the web. A personal web server might just be an occasional user's computer loaded with the requisite software for remote management or a dedicated high-end computer with dedicated network hardware and software. A personal web server is a primary example of e-commerce application.

## 8.Static website vs Dynamic website development.

### Static Websites

The closest approximation of a static website is an online brochure. It acts and feels like a printed brochure we can view through your computer screen and managing a website is not easy.

### Dynamic Websites

Dynamic websites have far more functionality and flexibility. Their initial set up is more complicated and uses more complex coding types such as PHP or ASP. managing a dynamic website is much easier and, with the right Content Management System (CMS), updating content, or creating new pages, can be done with no programming skills.

## 1. Introduction to PHP

- PHP stands for **PHP**: **H**ypertext **P**reprocessor
- PHP code is executed on the server.
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use
- PHP files can contain text, HTML, JavaScript code, and PHP code

- PHP code are executed on the server, and the result is returned to the browser as plain HTML
- PHP files have a default file extension of ".php"

## 2. Start and End Tags of PHP

A PHP script starts with **<?php** and ends with **?>**:

Syntax:

```
<?php
// PHP code goes here

?>
```

The default file extension for PHP files is ".php".

A PHP file normally contains HTML tags, and some PHP scripting code.

**Example**

```
<!DOCTYPE html>

<html>
<body>
<h1>My first PHP page</h1>

<?php
echo "Hello World!";

?>

</body>
</html>
```

## 3. Data types in PHP

A Data type defines the type of data. In PHP we do not need to strictly define data type for data.PHP data types are divided into two broad categories.

- Scalar Data Type: for example integer, float, string and boolean
- Compound Data Type: they are container of other data types. examples of compound data types are arrays, resources and objects

PHP has eight data types:

Scalar types

- boolean
- integer
- float
- string

Compound types

- array
- object

Special types

- resources

- NULL

## 4. Variables

Variables are used for storing a values, like text strings, numbers or arrays. When a variable is declared, it can be used over and over again in your script. All variables in PHP start with a $ sign symbol. The correct way of declaring a variable in PHP

```
$var_name = value;
```

PHP programmers often forget the $ sign at the beginning of the variable. In that case it will not work.

```
<?php $txt="Hello World!"; $x=16; ?>
```

## 5. Constants

A constant is a name or an identifier for a simple value. A constant value cannot change during the execution of the script. By default a constant is case-sensitiv. By convention, constant identifiers are always uppercase. A constant name starts with a letter or underscore, followed by any number of letters, numbers, or underscores. If you have defined a constant, it can never be changed or undefined.

## 6. operators and Expressions

Operators are catalysts of operations. In other words operators are signs that perform a special operation. For example + operator is used for adding values. PHP has many operators to perform different operations. Some of the operators are:

| | |
|---|---|
| + | addition |
| - | subtraction |
| * | multiplication |
| / | division |
| % | modulus or remainder |
| = | assignment |
| == | comparison |
| != | not equals |

| | |
|---|---|
| . | concatenation/Combination |
| **&&** | logical AND |
| \|\| | logical OR |
| **XOR** | logical xor |
| < | less than |
| > | greater than |
| <= | less than, equal to |
| >= | greater than, equal to |

## 7. printing data on PHP page

Echo is not actually a function (it is a language construct), so you are not required to use parentheses with it. echo (unlike some other language constructs) does not behave like a function, so it cannot always be used in the context of a function. Additionally, if you want to pass more than one parameter to echo, the parameters must not be enclosed within parentheses.Echo statement is uses to print data in PHP pages. The PHP command echo is a means of outputting text to the web browser. Printing data on PHP page by table tag, grid tag etc.

## 8. Control statements

- **Conditional statements** are used when we want to perform an action based on a condition. For instance, I can check if it's raining to decide whether to stay home or go out.
- Conditional statements have a body structure (wrapped in between curly braces {} ). PHP keywords for condional statements are **if**, **else**, **else if**.

**if/else:**

```
<?php #if-statements.php

  if (condition) {
    //Code to be executed if condition is true
  }
  else
  {
    //Alternative code
  }
?>
```

- The else part of the above is optional. **A condition is an expression that can be evaluated to true or false.** PHP treats 0 and the empty string as false, and everything else as true. Here is some example code:

```
<?php #if-statements.php

  $name = "Bill";
  if($name == "Bill")
  {
    echo "Hi Bill!";
  }
  else
  {
    echo "Hello person-who-is-not-Bill";
  }
?>
```

- IF statements can be chained together using ELSEIF :

```
<?php #if-statements.php

  $width = 100;

  $height = 20;


  if($width > $height)
  {
    echo "The object is wider than it is high";
  }
  elseif ($width == $height)
  {
    echo "The object has similar width and height";
  }
  else
  {
    echo "The object is higher than it is wide";
  }
?>
```

### switch case

If you want to select one of many blocks of code to be executed, use the Switch statement. The switch statement is used to avoid long blocks of if..elseif..else code.

Syntax:

```
 switch (expression)
{
case label1:
  code to be executed if expression = label1;
  break;
case label2:
  code to be executed if expression = label2;
  break;
default:
  code to be executed
  if expression is different
  from both label1 and label2;
}
```

## 8. Loop

Loops in PHP are used to execute the same block of code a specified number of times. PHP supports following four loop types.

- **for -** loops through a block of code a specified number of times.
- **while -** loops through a block of code if and as long as a specified condition is true.
- **do...while -** loops through a block of code once, and then repeats the loop as long as a special condition is true.
- **foreach -** loops through a block of code for each element in an array.

A **FOR** loop requires three parameters:

- start: the starting point, a value for the counter variable
- condition: the condition that has to be met to run the loop again
- increment: the counter has to be incremented (or decremented) before the loop is run again

**Syntax**

```
for (initialization; condition; increment)
{
  code to be executed;
}
```

## while loop statement

The while statement will execute a block of code if and as long as a test expression is true. If the test expression is true then the code block will be executed. After the code has executed the test expression will again be evaluated and the loop will continue until the test expression is found to be false.

**Syntax**

```
while (condition)
{
   code to be executed;
}
```

## do...while loop statement

The do...while statement will execute a block of code at least once - it then will repeat the loop as long as a condition is true.

**Syntax**

```
do
{
  code to be executed;
}while (condition);
```

## Foreach loop statement

The foreach statement is used to loop through arrays. For each pass the value of the current array element is assigned to $value and the array pointer is moved by one and in the next pass next element will be processed.

**Syntax**

```
foreach (array as value)
{
   code to be executed;

}
```

### The continue statement

The PHP **continue** keyword is used to halt the current iteration of a loop but it does not terminate the loop.Just like the **break** statement the **continue** statement is situated inside the statement block containing the code that the loop executes, preceded by a conditional test. For the pass encountering **continue** statement, rest of the loop code is skipped and next pass starts

### The break statement

The PHP **break** keyword is used to terminate the execution of a loop prematurely.

The **break** statement is situated inside the statement block. If gives you full control and whenever you want to exit from the loop you can come out. After coming out of a loop immediate statement to the loop will be executed.

## 9. Array

An array is a data structure that stores one or more similar type of values in a single value. For example if you want to store 100 numbers then instead of defining 100 variables its easy to define an array of 100 length.

There are three different kind of arrays and each array value is accessed using an ID c which is called array index.

- **Numeric array** - An array with a numeric index. Values are stored and accessed in linear fashion
- **Associative array** - An array with strings as index. This stores element values in association with key values rather than in a strict linear index order.
- **Multidimensional array** - An array containing one or more arrays and values are accessed using multiple indices

## 10. Initialization of an array

array( *key* => *value*, *key2* => *value2*, *key3* => *value3*,... )

Example: $array = array( "foo" => "bar", "bar" => "foo",);

## 12. Sorting Arrays

PHP has several functions that deal with sorting arrays, and this document exists to help sort it all out.

The main differences are:

- Some sort based on the array keys, whereas others by the values: *$array['key'] = 'value';*

- Whether or not the correlation between the keys and values are maintained after the sort, which may mean the keys are reset numerically (0,1,2 ...)
- The order of the sort: alphabetical, low to high (ascending), high to low (descending), numerical, natural, random, or user defined
- Note: All of these sort functions act directly on the array variable itself, as opposed to returning a new sorted array
- If any of these sort functions evaluates two members as equal then the order is undefined (the sorting is not stable).

## 13. Array Functions

we will go through the following PHP array sort functions:

- sort() - sort arrays in ascending order
- rsort() - sort arrays in descending order
- asort() - sort associative arrays in ascending order, according to the value
- ksort() - sort associative arrays in ascending order, according to the key
- arsort() - sort associative arrays in descending order, according to the value
- krsort() - sort associative arrays in descending order, according to the key

## 10 Function

A **Function** is a reusable piece of code. To use a function we have two steps:

- function declaration
- function invocation (or call)

**To declare a function:**

We use the keyword ***function*** followed by the name of the function, two parenthesis and the block of code in between two curly braces:

```
function fooName(parameters)

{
        // Function body goes here
        return $returnValue;
}
```

Functions do not have to return anything, so the return statement can be left out.

Variables used within the function are **local** to that function and do not effect the global scope.

## User defined PHP Functions(//function call)

**zero parameter funtion**

Using **return** keyword is optional, here two samples:

**returning a value..**

```
<?php //functions.php
  //function declaration
  function returnName()
 {
    $returnValue = "<p>my name is Shahid</p>";

    return $returnValue;

 }

  //function call
  echo returnName();
?>
```

**without returning a value..**

```
<?php //functions.php
  //function declaration
  function displayName()
 {
    echo "<p>my name is Shahid</p>";

 }

  //function call
  displayName();
?>
```

**function with parameters**

### passing by Reference and passing by value

```
<?php //functions.php
  //function declaration (the function asks for 2 parameters)
  function displayFirstAndLastName($first, $last)
 {
    echo "<p>my first name is <strong>$first</strong>"
 .
          " and my last name is <strong>$last</strong></p>";
 }

 //call the function parsing two strings
 Display FirstAndLastName("John", "Doe"); //my first name is John and my last name is Doe
?>
```

## UNIT – III

**Working with Forms:** Get and Post Methods, Querystrings, HTML form controls and PHP, Maintaining User State: Cookies, Sessions, Application State.
Working With Files: Opening and Closing Files, Reading and Writing to Files, Getting Information on Files

## 1. Get and Post Methods

### The GET Method

The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the **?** character.

```
http://www.test.com/index.htm?name1=value1&name2=value2
```

- The GET method produces a long string that appears in your server logs, in the browser's Location: box.
- The GET method is restricted to send upto 1024 characters only.
- Never use GET method if you have password or other sensitive information to be sent to the server.
- GET can't be used to send binary data, like images or word documents, to the server.
- The data sent by GET method can be accessed using QUERY_STRING environment variable.
- The PHP provides **$_GET** associative array to access all the sent information using GET method.

## 2. The POST Method

The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY_STRING.

- The POST method does not have any restriction on data size to be sent.
- The POST method can be used to send ASCII as well as binary data.
- The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.
- The PHP provides **$_POST** associative array to access all the sent information using GET method.

### 3. Query strings

Query string plays important role in building web applications, especially if you want to make nice urls without question mark and many key, value pairs, actually it is not so easy to design application with nice urls however it is always worth doing so, because it not only looks very proffessional, such URLs are search engine friendly, which means that they will get indexed faster but also it happens that search engines have problems with indexing pages with more than 3 key=value pairs in query string.Query-string variables are automatically put into an array called $_GET to be extracted at your leisure. It's so simple it's scary. Maybe that's what's confusing you.

### 4. HTML form controls and PHP

The standard HTML form controls and link elements to provide keyboard operation and assistive technology interoperability of interactive user interface elements.

1) <a>: 'title' attribute, text within <a> element or 'alt' attribute if image link. Concatenated if both text and image 'alt' attribute are provided
2) <button>: text inside <button> element or 'title' attribute
3) <input type = "button", "submit", or "reset">:
4) <input type = "image">
5) <input type = "text">
6) <input type = "password">
7) <input type="checkbox">
8) <input type="radio">
9) <select>
10) <textarea>

### 5. Cookies

Cookies are text files stored on the client computer and they are kept of use tracking purpose. PHP transparently supports HTTP cookies.

There are three steps involved in identifying returning users:

- Server script sends a set of cookies to the browser. For example name, age, or identification number etc.
- Browser stores this information on local machine for future use.
- When next time browser sends any request to web server then it sends those cookies information to the server and server uses that information to identify the user.

### 6. Session

A session creates a file in a temporary directory on the server where registered session variables and their values are stored. This data will be available to all pages on the site during that visit.

The location of the temporary file is determined by a setting in the **php.ini** file called **session.save_path**. Bore using any session variable make sure you have setup this path.

When a session is started following things happen:

- PHP first creates a unique identifier for that particular session which is a random string of 32 hexadecimal numbers such as 3c7foj34c3jj973hjkop2fc937e3443.
- A cookie called **PHPSESSID** is automatically sent to the user's computer to store unique session identification string.
- A file is automatically created on the server in the designated temporary directory and bears the name of the unique identifier prefixed by sess_ ie sess_3c7foj34c3jj973hjkop2fc937e3443

## 7. Web Application State

Web application development with PHP" category:

- Sending Email using PHP
- HTTP Headers and PHP
- Defining Your Own Functions in PHP
- Working with Dates and Times in PHP
- Improving Script and Site Performance

## 8. Working With Files

The fopen() function is used to open files in PHP.

**Opening a File**

The fopen() function is used to open files in PHP.

The first parameter of this function contains the name of the file to be opened and the second parameter specifies in which mode the file should be opened:

```
<html>
<body>

<?php
$file=fopen("welcome.txt","r");
?>

</body>
</html>
```

The file may be opened in one of the following modes:

| **Modes** | **Description** |
|---|---|
| r | Read only. Starts at the beginning of the file |
| r+ | Read/Write. Starts at the beginning of the file |
| w | Write only. Opens and clears the contents of file; or creates a new file if it doesn't exist |
| w+ | Read/Write. Opens and clears the contents of file; or creates a new file if it doesn't exist |
| a | Append. Opens and writes to the end of the file or creates a new file if it doesn't exist |
| a+ | Read/Append. Preserves file content by writing to the end of the file |
| x | **Write only. Creates a new file. Returns FALSE and an error if file already exists** |
| x+ | **Read/Write. Creates a new file. Returns FALSE and an error if file already exists** |

**Closing a File**

The fclose() function is used to close an open file:

```
<?php
$file = fopen("test.txt","r");


//some code to be executed


fclose($file);
?>
```

**End-of-file**

The feof() function checks if the "end-of-file" (EOF) has been reached.The feof() function is useful for looping through data of unknown length.

```
if (feof($file)) echo "End of file";
```

**Reading a File Character by Character**

The fgetc() function is used to read a single character from a file.

**Example**

```
<?php
$file=fopen("welcome.txt","r") or exit("Unable to open file!");

while (!feof($file))

  {
  echo fgetc($file);

  }
fclose($file);
?>
```

---

## UNIT-IV

### 1. Connecting to MySQL

• To connect to a database, need to create a connection
– At lowest level, this is a network connection
– Involves a login sequence (username/password)
• Since this is a relatively expensive step, web application environments:
– Share connections
– Have multiple connections
• Whether, and how many, are typical configuration items. In MySQL:
– Allow_persistent: whether to allow persistent connections
– Max_persistent: the maximum number of persistent connections
– Max_links: max number of connections, persistent and not
– Connection_timeout: how long the persistent connection is left Open

### 2. Creating Database Connection

• Use either mysql_connect or mysql_pconnect to create database connection
– mysql_connect: connection is closed at end of script (end of page)
– mysql_pconnect: creates persistent connection
• connection remains even after end of the page
• Parameters
– Server – hostname of server
– Username – username on the database
– Password – password on the database
– New Link (mysql_connect only) – reuse database connection created by previous call to mysql_connect
– Client Flags
• MYSQL_CLIENT_SSL :: Use SSL
• MYSQL_CLIENT_COMPRESS :: Compress data sent to MySQL

## Other operations

### 3. Security Note

• Username and password fields imply that database password is sitting there in the source code

– If someone gains access to source code, can compromise the database

– Servers are sometimes configured to view PHP source code when a resource is requested with ".phps" instead of ".php"

– One approach to avoid this: put this information in Web server config. File

• Then ensure the Web server config. file is not externally Accessible

### 4. Selecting a Database

• mysql_select_db()

– Pass it the database name

• Related:

– mysql_list_dbs()

• List databases available

– Mysql_list_tables()

### 5. Sending SQL Query to database

• Create query string

– $query = '*SQL formatted string*'

– $query = 'SELECT * FROM *table*'

• Submit query to database for processing

– $result = mysql_query($query);

– For UPDATE, DELETE, DROP, etc, returns TRUE or FALSE

– For SELECT, SHOW, DESCRIBE or EXPLAIN, $result is an identifier for the results, and does not contain the results themselves

• $result is called a "resource" in this case

• A result of FALSE indicates an error

• If there is an error

– mysql_error() returns error string from last MySQL call

### 6. Process Results

• Many functions exist to work with database results

• mysql_num_rows()

– Number of rows in the result set

– Useful for iterating over result set

• mysql_fetch_array()

– Returns a result row as an array

– Can be associative or numeric or both (default)

– $row = mysql_fetch_array($result);

– $row['*column name*'] :: value comes from database row with specified column name

– $row[0] :: value comes from first field in result set

### 7. Process Results Loop

• Easy loop for processing results: $result = mysql_query($qstring);

```
$num_rows = mysql_num_rows($result);
for ($i=0; $i<$num_rows; $i++) {
$row = mysql_fetch_array($result);
// take action on database results here
}
```

### 8. Closing Database Connection

• mysql_close()
– Closes database connection
– Only works for connections opened with
mysql_connect()
– Connections opened with mysql_pconnect() ignore this call
– Often not necessary to call this, as connections created by mysql_connect are closed at the end of the script anyway

## Web References

- **http://www.tutorialspoint.com/vb.net/vb.net_classes_objects.htm**
- http://docs.oracle.com/javaee/1.4/tutorial/doc/WebApp.html
- http://freefeast.info/general-it-articles/difference-
- http://whatismyipaddress.com/web-server
- http://en.wikipedia.org/wiki/Personal_web_server
- https://drupal.org/node/157602\
- http://www.hscripts.com/tutorials/vbnet/declaring-arrays.html
- http://www.seminarsonly.com/Engineering-Projects/PHP/PHP_Projects.php
- http://www.php.net/manual/en/language.basic-syntax.phptags.php
- http://www.hscripts.com/tutorials/vbnet/textbox-control.html\
- http://www.cramerz.com/php/php_data_types
- http://www.w3schools.com/php/php_intro.asp\
- http://socialpresence.me/integrating-digital-into-the-mix/
- http://www.tutorialspoint.com/vb.net/vb.net_classes_objects.htm database
- http://www.tutorialspoint.com/vb.net/vb.net_classes_objects.htm

: