UNIT I: Introduction to PHP

1.1 BASIC SYNTAX

Introduction to PHP

PHP is a server-side scripting language.

Script is a small programs included in a web page. There are two type of scripting languages. These are:

1. Client Side Scripting Language.

Script on a page runs on client side or client machine such script is called client side script. JavaScript, VBScript are the Languages used for writing client side script.

2. Server Side Scripting Language.

Script on a page runs on server side or server machine, and only sends output to client machine, such script are called server side script.

PHP, ASP.NET (using C#.NET or VB.NET), ASP, JAVA (using JSP-Java Server Pages), are the Languages used for writing server side script.

For understanding PHP you have a basic understanding of:

HTML/XHTML and JavaScript

What is PHP?

PHP stands for PHP: Hypertext Preprocessor
PHP is a server-side scripting language, like ASP, JSP etc.
PHP scripts are executed on the server
PHP is used for creating dynamic websites.

PHP supports many databases (for ex: MySQL, Informix, Oracle, Sybase, Solid, Postgre SQL, Generic ODBC, etc.)

PHP is an open source software technology. Means source code of PHP software is free.

PHP is free to download and use

What is a PHP File?

PHP files can contain text, HTML tags and scripts
PHP files are returned to the browser as plain HTML
PHP files have a file extension of ".php", ".php3", or ".phtml"

Why PHP?

PHP runs on different platforms (Windows, Linux, Unix, etc.)
PHP is compatible with almost all servers used today (Apache, IIS, etc.)
PHP is FREE to download from the official PHP resource: www.php.net.
PHP is easy to learn and runs efficiently on the server side

To begin Programming in PHP for web development, first we require software bundle named XAMPP. There are other similar software bundles that can run PHP, but XAMPP works well in all the major platforms: Windows, Mac and Linux.

Where XAMPP stands for:

X: Stands for Cross platform (Means you can install xampp on any Operating System)

A: Apache server (To run PHP script on local machine we need a server)

M: MySQL (To store data in database and to perform database operation we need MySQL)

P: PHP (Well known Object oriented Scripting language to create dynamic website- PHP Interpreter is required to Compile and Run the PHP Script)

P: Perl programming language.

Apache is the most widely used Web Server software. Developed and maintained by Apache Software Foundation, Apache is open source software available for free. It runs on 67% of all webservers in the world. It is fast, reliable, and secure

Web Server is the software that receives your request to access a web page. It runs a few security checks on your HTTP request and takes you to the web page.

MySQL:

MySQL is an open source relational database management system (RDBMS) based on Structured Query Language (SQL).

MySQL runs on virtually all platforms, including Linux, UNIX, and Windows.

Although it can be used in a wide range of applications, MySQL is most often associated with web-based applications and online publishing.

After you install XAMPP,
1. Open any Text Editor.
2. Write the following PHP Program / CODE in the Text Editor:

```
<?PHP
echo 'Hello From PHP! <br />';
?>
This is test.php File.
```

3. Save the file in XAMPP Installation Directory \ Web Root Directory

Note-1: Default XAMPP Installation Directory in Windows is C:\xampp

Note-2: Default Web Root Directory in XAMPP is htdocs.

All your php files will have to be in this htdocs folder.

That means, for a typical installation of XAMPP in Windows, you will have to save the PHP CODE in C:\xampp\htdocs folder.

4. When you save the file, name it test.php (just as an example, any valid file name with .php in the end will work).

5. Then, go to XAMPP installation folder (typically, C:\xampp) and run xampp-control.exe by double clicking it.

6. In the xampp-control window, click the start button beside Apache.
7. Now, in your web browser's address bar, type the address:
http://localhost/test.php

Hello from PHP! This is test.php File.

BASIC SYNTAX

The PHP script is executed on the server, and the plain HTML result is sent back to the browser.

A PHP script always starts with <?php and ends with ?>. A PHP script can be placed anywhere in the document.

```
<?php
?>
```

A PHP file must have a .php extension.

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

```
<html>
<body>
<?php
print "Hello World";
?>
</body>
</html>
```

Each code line in PHP must end with a semicolon.

The semicolon is a separator and is used to distinguish one set of instructions from another.

There are two basic statements to output text with PHP: print and echo.

What Does PHP Do?

PHP is a simple yet powerful language designed for creating HTML content.

PHP runs on all major operating systems, from UNIX variants including Linux, FreeBSD, and Solaris to such diverse platforms as Windows and Mac OS X.

It can be used with all leading web servers, including Apache, Microsoft IIS, and the Netscape/iPlanet servers.

The language is very flexible. PHP has built-in support for generating PDF files, GIF, JPG, and PNG images, and Flash movies

LEXICAL STRUCTURE OF PHP:

Computer languages, like human languages, have a lexical structure. A source code of a PHP script consists of tokens. Tokens are code elements.

In PHP language, we have comments, variables, literals, operators, delimiters and keywords.

Comments

Comments are used for describing code or giving the information about code.

All comments in PHP follow the #character.

With that we can use C & C++ like comments

```
<?php
# Program or page to display simple message.
Print "This is output of PHP script";
?>
```

Variables

A variable is an identifier, which holds a value. In programming we say, that we assign a value to a variable.

```
<?php
$number = 10;
$Number = 11;
$NUMBER = 12;
echo $number, $Number, $NUMBER; echo "\n";
?>
```

Output:?

A literal

A literal is any notation for representing a value within the PHP source code. Technically, a literal will be assigned a value at compile time, while a variable will be assigned at runtime.

```
$age = 29;
$nationality = "Indian";
```

Here we assign two literals to variables. Number 29 and string Indian are literals.

```
<?php
$name1 = "Jane ";
$age1 = 17;
Print $name1; Print age1;
?>
```

Operators

An operator is a symbol used to perform an action on some value.

Operator	Sign
Arithmetic operators	+,-,*,/,%
Assignment operators	=,+=,-=,*=,/=,%=
Comparison operators	==,===,!=,<>,!==,>,<,>=,<=
Increment/Decrement operators	$a++,++$a,$a--,--$a
Logical operators	and,or,xor,&&,||,!
String operators	.,.=
Array operators	+,==,===,!=,<>,!==
Conditional assignment operators	?:,??

Delimiters

A delimiter is a sequence of one or more characters used to specify the boundary between separate, independent regions in plain text or other data stream.

```
$a = "PHP is lang";
$b = 'Java';
```

The single and double characters are used to mark the beginning and the end of a string.

```
if ( $a > $b)
{
echo "a is bigger than b";
}
```

Parentheses are used to mark the function signature. The signature is the function parameters. Curly brackets are used to mark the beginning and the end of the function body. They are also used in flow control.

Keywords

A keyword is a reserved word in the PHP programming language. Keywords are used to perform a specific task in the computer program.

For example, print a value, do repetitive tasks or perform logical operations. A programmer cannot use a keyword as an ordinary variable.

The following is a list of PHP keywords.

Abstract, and, array, as , break , callable,
Case, catch, class, clone, const, continue,
Declare, default , die , do , echo, else,
elseif , empty, enddeclare, endfor, endforeach, endif,
endswitch, endwhile , extends, final , finally, for,
foreach, function, global , goto, if, implements,
include, include,_once, instanceof, insteadof, interface,
isset, list, namespace, new, or , print,
private, protected, public, require , require_once, return,
static, switch, throw , trait, try, unset,
use, var, while, xor , yield, __halt_compiler.

1.3 Sending Data to the Web Browser

To create dynamic Web sites with PHP, you must know how to send data to the Web browser. PHP has a number of built-in functions for this purpose, the most common being echo () and print ().

echo 'Hello, world!'; echo "What's new?";

You can use print() instead, if you prefer:

print 'Hello, world!'; print "What's new?";

The first quotation mark after the function name indicates the start of the message to be printed. The next matching quotation mark (i.e., the next quotation mark of the same kind as the opening mark) indicates the end of the message to be printed.
You can either use single or double quotation marks, but when you want to print single quote in a output, as in above example you must have to use double quote or escape sequence character \ as bellow

Inside double quote we can use single quote and inside single quote we can use double quote.

print "What's new?";
Or
print 'What\'s new?';

Following example shows how we can use single quote and double quote.

```
print "Hello 'Ram' "; //output : Hello 'Ram'
Or
print 'Hello "Ram" '; //output : Hello "Ram"
```

Alternately we can use one of these as bellow

```
print 'Hello \'Ram\' '; //output : Hello 'Ram'
Or
print "Hello \"Ram\" "; //output : Hello "Ram"
```

To Print the value of Variable we can use the following format

```
$a=20;
$b=30
```

```
print “a=$a,b=$b”;
```
OR
```
$a=20;
$b=30;
print “a=” . $a . “b=”. $b;
```

The echo construct lets you print many values at once, while print( ) prints only one value. The printf( ) function builds a formatted string by inserting values into a template. The print_r( ) function is useful for debugging it prints the contents of arrays, objects, and other things, in a more-or-less human-readable form.

Echo

To put a string into the HTML of a PHP-generated page, use echo. While it looks and for the most part behaves like a function, echo is a language construct. This means that you can omit the parentheses, so the following are equivalent:

```
echo "Printy";
echo (“Printy”); //valid also
```

You can specify multiple items to print by separating them with commas:

```
echo "First", "second", "third";
```

print( )

The print( ) function sends one value (its argument) to the browser. It returns True if the string was successfully displayed and false otherwise (e.g., if the user pressed the Stop button on her browser before this part of the page was rendered):

printf( )

The printf( ) function outputs a string built by substituting values into a template (the format string).

It is derived from the function of the same name in the standard C library.

The first argument to printf( ) is the format string.

The remaining arguments are the values to be substituted.

A % character in the format string indicates a substitution.

Format modifiers

Each substitution marker in the template consists of a percent sign (%), possibly followed by modifiers from the following list, and ends with a type specifier.

(Use '%%' to get a single percent character in the output.) The modifiers must appear in the order in which they are listed here:

Type specifiers

The type specifier tells printf( ) what type of data is being substituted. This determines the interpretation of the previously listed modifiers. There are eight types, as listed below.

A floating-point number to two decimal places:

```
printf('%.2f', 27.452);
27.45
```

Decimal and hexadecimal output:

```
printf('The hex value of %d is %x', 214, 214);
The hex value of 214 is d6
```

Padding an integer to three decimal places:

```
printf('Bond. James Bond. %03d.', 7);
Bond. James Bond. 007.
```

Formatting a date:

```
printf('%02d/%02d/%04d', $month, $day, $year);
02/15/2005
```

A percentage:

```
printf('%.2f%% Complete', 2.1);
2.10% Complete
```

1.4 Understanding PHP, HTML, and White Space

With PHP you send data (like HTML tags and text) to the Web browser, which will, in turn, render that data as the Web page the end user sees.
Thus, what you are doing with PHP is creating the HTML source of a Web page.

With this in mind, there are three areas of notable white space (extra spaces, tabs, and blank lines): in your PHP scripts, in your HTML source, and in the rendered Web page.

PHP is generally white space insensitive, meaning that you can add any number of spaces in your code to make your scripts more readable. But those spaces will not be displayed in the output page.

HTML is also generally white space insensitive. Specifically, the only white space in HTML that affects the rendered page is a single space (multiple spaces still get rendered as one).

Multiple spaces added in the HTML document will be ignored and only single space will be added.

If your HTML document has text on multiple lines, that doesn't mean it'll appear on multiple lines in the output page.

To add the spacing in a Web page, use the HTML tags
 (line break,
 in older HTML standards) and (paragraph).

To Add the Unformatted text we can also use <pre> </pre> tag.

To add the spacing in the HTML contents created with PHP, you can
Use echo () or print () several times. Or

Print the newline character (\n) within double quotation marks.
We can also use $nbsp; tag inside double quote

1.5 Writing Comments

Comments are used for describing code or giving the information about code.
All comments in PHP follow the #character.
With that we can use C & C++ like comments
Everything that follows the # character is ignored by the PHP interpreter.

```
<?php
// Program or page to display simple message.
/*
This PHP Program is written at COCSIT: On Date: 06/12/12
*/
Print "PHP is Server Side Scripting Language ";
?>
```

PHP also recognizes the comments from the C and C++ language.

1.6 What Are Variables?

Variables are the named memory locations which are used to store values. OR Variables are containers used to temporarily store values. These values can be numbers, text, or much more complex data.
A variable is an identifier, which holds a value. In programming we say, that we assign a value to a variable. Technically speaking, a variable is a reference to a computer memory, where the value is stored.

PHP has eight types of variables. These include:

Four scalar (single-valued) types:
1. Boolean (TRUE or FALSE),
2. Integer,
3. Floating point (decimals),
4. Strings (characters);

Two non-scalars (multivalued) types:
5. arrays
6. objects;

And remaining two are:
7. Resources (this is used when interacting with databases) and
8. NULL (this is a special type that has no value).

All variables in PHP follow some syntactical rules:

A variable's name—also called its identifier—must start with a dollar sign ($), for example, $name.
The variable's name can contain a combination of strings, numbers, and the underscore, for example, $my_report1.
The first character after the dollar sign must be either a letter or an underscore (it cannot be a number).

Variable names in PHP are case-sensitive. This is a very important rule. It means that $name and $Name are entirely different variables.

In PHP, variable references are alternative or another name of existing variable or variable alias. To make $black an alias for the variable $white, use:

$black =& $white;

The old value of $black is lost. Instead, $black is now another name for the value that is stored in $white:

```
<?php
$white = "snow";
$black =& $white; print $black;
$black=”Snowfall”; print $white;
?>
```

Output:
snow Snowfall

1.8 About Strings,

Strings are group of characters, or array of characters.
Because strings are so common in web applications, PHP includes core-level support for creating and manipulating strings. A string is a sequence of characters of arbitrary length. String literals are delimited by either single or double quotes:
'big dog' "fat hog"
Variables are expanded within double quotes, while within single quotes they are not:

```
<?php
$name = "Darsh"; echo "Hi, $name\n"; echo 'Hi, $name';
?>
```

Hi, Darsh

Double quotes also support a variety of string escapes, as listed in following table. Escape sequences in double-quoted strings
Escape sequence Character represented

\"	Double quotes
\n	Newline
\r	Carriage return
\t	Tab
\\	Backslash
\$	Dollar sign
\{	Left brace
\}	Right brace
\[	Left bracket
\]	Right bracket
\0	through
\777	ASCII character represented by octal value
\x0	through
\xFF	ASCII character represented by hex value

String Concatenation: String Concatenation is like addition for strings, whereby characters are added to the end of the string. It's performed using the concatenation operator, which is the period (.) example:

```
<?php
$fname="Ram";
$lname="Patil";
$fullname= $fname ." ". $lname; //will concat $lname with $fname and add
single space
//between them
print "Mr: ". $fullname;
?>
```

Output:

Mr.Ram Patil

1.9 About Numbers

There are three main numeric types in PHP:

Integer
Float
Number Strings

Variables of numeric types are created when you assign a value to them:

```
$a = 5;
$b = 5.34;
$c = "25";
```

PHP Integers

2, 256, -256, 10358, -179567 are all integers.

An integer is a number without any decimal part.

An integer data type is a non-decimal number between -2147483648 and 2147483647 in 32 bit systems, and between -9223372036854775808 and 9223372036854775807 in 64 bit systems.

Some rules for integers:

An integer must have at least one digit
An integer must NOT have a decimal point
An integer can be either positive or negative
Integers can be specified in three formats: decimal (10-based), hexadecimal (16-based - prefixed with 0x) or octal (8-based - prefixed with 0)

PHP Floats

A float is a number with a decimal point or a number in exponential form.

2.0, 256.4, 10.358, 7.64E+5, 5.56E-5 are all floats.

The float data type can commonly store a value up to 1.7976931348623E+308 (platform dependent), and have a maximum precision of 14 digits.

PHP Numerical Strings

The PHP is_numeric() function can be used to find whether a variable is

numeric. The function returns true if the variable is a number or a numeric string, false otherwise.

```
<!DOCTYPE html>
<html>
<body>

<?php
// Check if the variable is numeric
$x = 5985;
var_dump(is_numeric($x));

echo "<br>";

$x = "5985";
var_dump(is_numeric($x));

echo "<br>";

$x = "59.85" + 100;
var_dump(is_numeric($x));

echo "<br>";

$x = "Hello";
var_dump(is_numeric($x));
?>

</body>
</html>
```

About Constants

PHP constants are name or identifier that can't be changed during the execution of the script except for magic constants, which are not really constants.

PHP constants can be defined by 2 ways:

Using define() function
Using const keyword

Constants are similar to the variable except once they defined, they can never be undefined or changed. They remain constant across the entire program.

PHP constants follow the same PHP variable rules. For example, it can be started with a letter or underscore only.

Conventionally, PHP constants should be defined in uppercase letters.

PHP constant: define()

Use the define() function to create a constant. It defines constant at run time. Let's see the syntax of define() function in PHP.

Syntax

define(name, value, case-insensitive)

name: It specifies the constant name.
value: It specifies the constant value.
case-insensitive: Specifies whether a constant is case-insensitive. Default value is false. It means it is case sensitive by default.

Ex.

```
<?php
define("GRETTING","Hello Gopal Good Morning",true);//not case sensitive
echo GRETTING, "</br>";
echo gretting;
?>
```

PHP constant: const keyword

PHP introduced a keyword const to create a constant. The const keyword defines constants at compile time. It is a language construct, not a function.

The constant defined using const keyword are case-sensitive.

```
<?php
const MESSAGE="Hello Gopal";
echo MESSAGE;
?>
```

Constant() function

There is another way to print the value of constants using constant() function instead of using the echo statement.

Syntax

constant (name)

```
<?php
   define("MSG", "Gopal");
   echo MSG, "</br>";
   echo constant("MSG");
   //both are similar
?>
```

The End