

PHP

- PHP == PHP Hypertext Preprocessor.
- It is an Open-Source, server-side scripting language.
- It is used to generate dynamic webpages.
- The Scripts reside between reserved PHP tags.
- These allows the programmer to embed PHP scripts within HTML Pages.
- Source Code not visible by client.
- Various built-in functions allows for fast development.

PHP Code :-

- Structurally looks like as similar to C/C++.
- Supports procedural and Object-Oriented paradigm.
- All PHP statements end with a Semi-Colon.
- Each PHP script must be enclosed in the reserved PHP tag.

<?php

?>

Comments in PHP :-

→ Standard C, C++, and shell Comment Symbols

// C++ and Java-style Comment

shell-style Comments

/* C-style Comments

These can span multiple lines */

Variables in PHP :-

→ PHP variables must begin with a \$ sign.

→ Case Sensitive. (\$foo != \$foo! = \$f00)

→ Globally and locally scoped variables.

→ Certain variable names are reserved by PHP.

* Form Variables. (\$_POST, \$_GET)

* Server Variables. (\$_SERVER)

Variable Usage

```
<?php
```

```
$foo = 25; // Numerical variable
```

~~var~~

```
$bar = "Hello"; // string variable
```

```
$foo = ($foo * 7); // multiplies foo by 7.
```

```
$bar = ($bar * 7); // Invalid expression
```

```
?>
```

Echo :-

The PHP command = echo' is used to output the parameters passed to it.

→ The typical usage for this is to send data to the client's web-browser.

Syntax:- void echo(string arg1 [, string argn...])

Example:-

```
<?php
```

```
$foo = 25;
```

```
// Numerical Variable
```

```
$bar = "Hello";
```

```
// String Variable
```

```
echo $bar;
```

```
// Outputs Hello
```

```
echo $foo, $bar;
```

```
// Outputs 25Hello
```

```
echo "5x5 = ", $foo;
```

```
// Outputs 5x5 = 25
```

```
echo "5x5 = $foo";
```

```
// Outputs 5x5 = 25
```

```
echo "5x5 = $foo";
```

```
// Outputs 5x5 = $foo
```

```
?>
```

Arithmetic Operations :-

→ \$a - \$b

// Subtraction

→ \$a * \$b

// multiplication

→ \$a / \$b

// division

→ \$a += 5

// \$a = \$a + 5 Also works for

* = and /=

Ex:- <? php

\$a = 15;

\$b = 30;

\$total = \$a + \$b;

print \$total;

print - <p><h1>\$total</h1>||;

?>

Concatenation:-

→ Use a period to join strings into one.

Ex:- <? php

\$string1 = "Hello";

\$string2 = "PHP";

\$string3 = \$string1 . " " . \$string2;

print \$string3;

?>

Hello PHP

Escaping the character:-

→ If the string has set of double quotation marks that must remain visible, use the \ (backslash) before the quotation marks to ignore and display them.

<?php

\$heading = "\" Computer Science \"";

Print \$heading;

?>

PHP Control Structures :-

- The structures within a language that allows us to control the flow of execution through a program (or) script.
- Grouped into Conditional (if/else) and repetition structures (eg:- while loops)

Ex:- if(\$foo==0)

```
{
    echo 'The variable foo is equal to 0';
}
else if (($foo > 0) && ($foo <= 5)) {
    echo 'The variable foo is between 1
    and 5';
}
else {
    echo 'The variable foo is equal to
    '$foo';
}
```

If... Else...

```
If (condition)
{
    statements;
}
Else
{
    Statement;
}
```

```
<?php
If ($user == "John")
{
    print "Hello John.";
}
Else
{
    print "You are not John.";
}
?>
```

while loops :-

while (condition)

{

Statements;

}

<?php

\$count = 0;

while (\$count < 3)

{

print "hello PHP.";

\$count += 1;

}

?>

Date Display

\$date display = date (-yyyy/m/d||);

Print \$date display;

2009/4/1

\$date display = date (-l, F, m, Y||);

Print \$date display;

Wednesday, April 1, 2009

Month, Day & Date Format Symbols

M	Jan
F	January
m	01
n	1

Day of month	d	01
Day of Month	J	1
Day of week	l	Monday
Day of week	D	Mon

Functions :-

- functions must be defined before they called.
- Function headers format.
function functionName(\$arg-1, \$arg2, ..., \$arg-n)
- No Return type is specified and function names are not Case-Sensitive.

Ex:- <?php

```
function foo($arg-1, $arg-2)
{
    $arg-2 = $arg-1 * $arg-2;
    return $arg-2;
}
```

```
$result_1 = foo(12, 3);
```

```
echo $result_1;
```

```
echo foo(12, 3)
```

```
?>
```

Include files :-

- Include - Opendb.php||; , Include - closedb.php||;
- This insert files; the code in files will be inserted into current code.
- Include (- footer.php||);

PHP - Forms

- Access to the HTTP POST and GET data is simple in PHP.

FIRST PHP SCRIPT :-

Save as Sample.php

```
<!--sample.php-->
```

```
<html><body>
```

```
<strong>Hello World!</strong><br/>
```

```
<?php
```

```
echo - <h2>Hello, world <h2>||; ? >
```

```
<?php
```

```
$myvar = "Hello World";
```

```
echo $myvar;
```

```
?>
```

```
</body></html>
```

Data in the tables :-

→ Function: list all tables in your database. User can select one of tables and show all contents in this table.

Functions Covered :-

- mysql_connect()
- mysql_select_db()
- mysql_num_rows
- mysql_close()
- mysql_query()
- include()
- mysql_fetch_array()

History of PHP :-

→ PHP began in 1995 when Rasmus Lerdorf developed a Perl/CGI script toolset he called the Personal Home Page (or) PHP.

→ And different versions are came for better performance of maintenance & fixing bugs.