

PHP:

Introduction

PHP Facts

- PHP stands for **PHP: Hypertext Preprocessor**
- PHP is a server-side scripting language
- 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"
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)

Installations (for Ubuntu)

To make use of our uplb local repository, first it would be a good idea to backup your sources.list through:

```
user@pc:$ sudo cp/etc/apt/sources.list /etc/apt/sources.list.bak
```

Open your sources.list for editing:

```
user@pc:$ sudo gedit /etc/apt/sources.list
```

Delete the entire text and paste this:

```
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.

deb http://ubuntu.intranet.uplb.edu.ph/ hardy main restricted
#deb-src http://ph.archive.ubuntu.com/ubuntu/ hardy main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://ubuntu.intranet.uplb.edu.ph/ hardy-updates main restricted multiverse
#deb-src http://ph.archive.ubuntu.com/ubuntu/ hardy-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## universe WILL NOT receive any review or updates from the Ubuntu security
## team.
deb http://ubuntu.intranet.uplb.edu.ph/ hardy universe
#deb-src http://ph.archive.ubuntu.com/ubuntu/ hardy universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://ubuntu.intranet.uplb.edu.ph/ hardy multiverse
#deb-src http://ph.archive.ubuntu.com/ubuntu/ hardy multiverse

## Uncomment the following two lines to add software from the 'backports'
## repository.
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
```

```
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
# deb http://ph.archive.ubuntu.com/ubuntu/ feisty-backports main restricted universe multiverse
# deb-src http://ph.archive.ubuntu.com/ubuntu/ feisty-backports main restricted universe multiverse
deb http://ubuntu.intranet.uplb.edu.ph/ hardy-security main restricted
#deb-src http://security.ubuntu.com/ubuntu hardy-security main restricted
deb http://ubuntu.intranet.uplb.edu.ph/ hardy-security universe
#deb-src http://security.ubuntu.com/ubuntu hardy-security universe
deb http://ubuntu.intranet.uplb.edu.ph/ hardy-security multiverse
#deb-src http://security.ubuntu.com/ubuntu hardy-security multiverse
```

Type this command in the terminal to see the version of the OS you are using:

```
user@pc:$ lsb_release -a
```

Replace all instances of the word “hardy” in your sources.list into the codename of your OS. If the codename is also hardy, there is no need to change the same instance

Save your sources.list

Now refresh the package list:

```
user@pc:$ sudo apt-get update
```

We can now install PHP and Apache web server

```
user@pc:$ sudo apt-get install apache2 php5
```

To check if we have successfully installed Apache and Php open a web browser and type in the address:

http://localhost then click the directory apache2-default

We can now run our php files by placing them in our /var/www folder and viewing them in the browser:

e.g. http://localhost/index.php

Let us now create our work folder by activating a public_html directory

```
user@pc:$ sudo a2enmod userdir
```

This command requires apache service to restart. To do this we type the following commands.

```
user@pc:$ sudo /etc/init.d/apache2 stop
user@pc:$ sudo /etc/init.d/apache2 start
```

or simply type

```
user@pc:$ sudo /etc/init.d/apache2 restart
```

or

```
user@pc:$ sudo /etc/init.d/apache2 force_reload
```

if u want to force it.

Go to /home/user directory in your terminal, lets create our public_html folder:

```
user@pc:$ mkdir public_html
```

We can now place our php files in our work folder located in home/user/public_html and can be run on a browser using the address:

```
http://localhost/~user/*.php
```

We can now start programming! Phew! ☺

PHP Basic Syntax

A PHP scripting block always starts with <?php and ends with ?>.

```
<?php
?>
```

A PHP scripting block can be placed anywhere in the document, it normally contains HTML tags, just like an HTML file, and some PHP scripting code.

```
<html>
<body>

<?php
echo "Hello World";
?>

</body>
</html>
```

Above is an example of a simple php script that sends the text Hello World to the browser.

Each script is ended with a semi-colon, this separates the statement from succeeding statements

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

Comments in PHP uses C-Style */*comment*/* and C++ style comment *//comment* . Below is an example.

```
<html>
<body>

<?php
//This is a comment

/*
This is
a comment
block
*/
?>

</body>
</html>
```

PHP Variables

All variables in PHP start with a \$ sign symbol.

```
$var_name = some_value;
```

php variables declared without the \$ sign will not work.

PHP variables are loosely typed.

```
<?php
$str="Hello World!";
$num=16;
?>
```

Like javascript, you do not need to declare a data type for the variables. In PHP, the variable is declared automatically when you use it.

Naming rules for PHP Variables go as follows.

- A variable name must start with a letter or an underscore "_".
- A variable name can only contain alpha-numeric characters and underscores (a-z, A-Z, 0-9, and _).
- A variable name should not contain spaces. If a variable name is more than one word, it should be separated with an underscore (\$my_string), or with capitalization (\$myString)

PHP String Variable Examples

```
<?php
$txt="Hello World";
echo $txt;
```

```
echo "<br>" // break line
```

```
// Concatenation using (.)
//$txt1="Hello World!";
//$txt2="What a nice day!";
//echo $txt1 . " " . $txt2;
```

```
//strlen() function
//echo strlen("Hello world!");
```

```
// strpos() function
//echo strpos("Hello world!","world");
```

```
?>
```

PHP Operators

Arithmetic Operators

Operator	Description	Example	Result
+	Addition	x=2 x+2	4
-	Subtraction	x=2 5-x	3
*	Multiplication	x=4 x*5	20
/	Division	15/5 5/2	3 2.5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

Assignment Operators

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
=	x=y	x=x*y
/=	x/=y	x=x/y
.=	x.=y	x=x.y
%=	x%=y	x=x%y

Comparison Operators

Operator	Description	Example
==	is equal to	5==8 returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

Logical Operators

Operator	Description	Example
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
	or	x=6 y=3 (x==5 y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

Conditional Statements

if Statement	if...else Statement	if...else if...else Statement
<pre><html> <body> <?php \$d=date("D"); if (\$d=="Fri") echo "Have a nice weekend!"; ?> </body> </html></pre>	<pre><html> <body> <?php \$d=date("D"); if (\$d=="Fri") echo "Have a nice weekend!"; else echo "Have a nice day!"; ?> </body> </html></pre>	<pre><html> <body> <?php \$d=date("D"); if (\$d=="Fri") echo "Have a nice weekend!"; elseif (\$d=="Sun") echo "Have a nice Sunday!"; else echo "Have a nice day!"; ?> </body> </html></pre>

Switch Statement

<pre><html> <body> <?php switch (\$x) { case 1: echo "Number 1"; break; case 2: echo "Number 2"; break; case 3: echo "Number 3"; break; default:</pre>
--

```

    echo "No number between 1 and 3";
}
?>

</body>
</html>

```

Loops

For Loops	While Loops	Do...While Loops
<pre> <html> <body> <?php for (\$i=1; \$i<=5; \$i++) { echo "The number is " . \$i . "
"; } ?> </body> </html> </pre>	<pre> <html> <body> <?php \$i=1; while(\$i<=5) { echo "The number is " . \$i . "
"; \$i++; } ?> </body> </html> </pre>	<pre> <html> <body> <?php \$i=1; do { \$i++; echo "The number is " . \$i . "
"; } while (\$i<=5); ?> </body> </html> </pre>

Functions

<pre> <html> <body> <?php function writeName() { echo "Mr Bean"; } echo "My name is "; writeName(); ?> </body> </html> </pre>	<pre> <html> <body> <?php function add(\$x,\$y) { \$total=\$x+\$y; return \$total; } echo "1 + 16 = " . add(1,16); ?> </body> </html> </pre>
--	---

We shall discuss more next time :D