

# HERAT UNIVERSITY COMPUTER SCIENCE FACULTY

Web Engineering II (Semester 6)

(PHP Basics & Variables)

---

LECTURER: HAMED AMIRY

# PHP Introduction

---

PHP is a recursive acronym for “PHP: Hypertext Preprocessor” -- It is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

PHP development began in 1994 when [Rasmus Lerdorf](#)



# PHP Introduction

---

- > PHP is a server-side scripting language
- > PHP scripts are executed on the server
- > PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)
- > PHP is open source software
- > PHP is free to download and use

# PHP Introduction

---

- > 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](http://www.php.net)
- > PHP is easy to learn and runs efficiently on the server side

# PHP Introduction

---

Some info on MySQL which we will cover in the next workshop...

- > MySQL is a database server
- > MySQL is ideal for both small and large applications
- > MySQL supports standard SQL
- > MySQL compiles on a number of platforms
- > MySQL is free to download and use

# PHP Introduction

---

Instead of lots of commands to output HTML (as seen in C or Perl), PHP pages contain HTML with embedded code that does "something" (like in the next slide, it outputs "Hi, I'm a PHP script!").

The PHP code is enclosed in special start and end processing instructions `<?php` and `?>` that allow you to jump into and out of "PHP mode."

# PHP Introduction

---

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Example</title>
  </head>
  <body>

    <?php
        echo "Hi, I'm a PHP script!";
    ?>

  </body>
</html>
```

# PHP Introduction

---

PHP code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was.


A visual, if you please...




# PHP Introduction

**ON SERVER**

```
<html>
<head> <title>Welcome</title> </head>
<body>
<?
    echo "Hello";
    print "<br />";
    echo "<b>I'm here..</b>";
?>
</body>
</html>
```



```
<html>
<head> <title>Welcome</title> </head>
<body>
Hello<br /><b>I'm here..</b></body>
</html>
```



Welcome - Mozilla

Hello  
I'm here..

Done

# PHP Getting Started

---

On windows, you can download and install WAMP. With one installation and you get an Apache webserver, database server and php.

<http://www.wampserver.com>

On mac, you can download and install MAMP.

<http://www.mamp.info/en/index.html>

# PHP Hello World

---

```
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <?php echo '<p>Hello World</p>'; ?>
  </body>
</html>
```

Above is the PHP source code.

# PHP Hello World

---

It renders as HTML that looks like this:

```
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>
```

# PHP Hello World

---

This program is extremely simple and you really did not need to use PHP to create a page like this. All it does is display: Hello World using the PHP **echo()** statement.

Think of this as a normal HTML file which happens to have a set of special tags available to you that do a lot of interesting things.

# PHP Comments

---

In PHP, we use `//` to make a single-line comment or `/*` and `*/` to make a large comment block.

```
<html>
<body>

<?php
//This is a comment

/*
This is
a comment
block
*/
?>

</body>
</html>
```

# PHP Variables

---

- > Variables are used for storing 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 Variables

---

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

- > In PHP, a variable does not need to be declared before adding a value to it.
- > In the example above, you see that you do not have to tell PHP which data type the variable is.
- > PHP automatically converts the variable to the correct data type, depending on its value.



# PHP Variables

---

- > A variable name must start with a letter or an underscore "\_" -- not a number
- > A variable name can only contain alpha-numeric characters, 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 Concatenation

---

- > The concatenation operator (.) is used to put two string values together.
- > To concatenate two string variables together, use the concatenation operator:

```
<?php
$txt1="Hello World!";
$txt2="What a nice day!";
echo $txt1 . " " . $txt2;
?>
```

# PHP Concatenation

---

The output of the code on the last slide will be:

```
Hello World! What a nice day!
```

If we look at the code you see that we used the concatenation operator two times. This is because we had to insert a third string (a space character), to separate the two strings.



# Any Question??

---



**Please write your questions and suggestions on the group!**