

# INTRODUCTION TO PHP

Ibrahim Al-Fifi

## OBJECTIVE

- What is PHP ?
- PHP Installation
- Basic PHP Syntax
- PHP Variables
- PHP Arrays
- Resources

## WHAT IS PHP ?

- PHP stands for **PHP: Hypertext Preprocessor**
- PHP is a server-side scripting language, like ASP
- PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)
  
- PHP is an **open source** software
- PHP is **free** to download and use

## PHP INSTALLATION

- Here is a link to a good tutorial from PHP.net on how to install PHP5:  
<http://www.php.net/manual/en/install.php>
  
- Download PHP for free here:  
<http://www.php.net/downloads.php>
  
- Download Apache for free here:  
<http://httpd.apache.org/download.cgi>

## BASIC PHP SYNTAX

- A PHP scripting block always starts with `<?php` and ends with `?>` and it can be placed anywhere in the document.
- On servers with shorthand support enabled you can start a scripting block with `<?` and end with `?>`.
- For maximum compatibility, we recommend that you use the standard form (`<?php`) rather than the shorthand form.

5

## BASIC PHP SYNTAX

- Below, we have an example of a simple PHP script which sends the text "Hello World" to the browser:

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

- Each code line in PHP must end with a semicolon.
- There are two basic statements to output text with PHP: `echo` and `print`.

6

## PHP VARIABLES

- All variables in PHP start with a `$` sign symbol.
- In PHP a variable does not need to be declared before being set.

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

- Variable Naming Rules:

- ✓ 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 underscore (`$my_string`), or with capitalization (`$myString`)

7

## CREATING VARIABLE VARIABLES

- A variable variable is one that holds the name of a variable.
- Example:  
Create a new variable, `$fruitname`, that holds the name of the `$apples` variable:

```
<?php
$apples = 4;
$fruitname = "apples";
echo "Number of apples: ", $$fruitname;
?>
```

This script displays: Number of apples: 4

8

## PHP ARRAYS

- There are three different kind of arrays:
- ✓ **Numeric array** - An array with a numeric ID key
- ✓ **Associative array** - An array where each ID key is associated with a value
- ✓ **Multidimensional array** - An array containing one or more arrays

9

## PHP ARRAYS – NUMERIC ARRAYS

- **Example 1:** The ID key is automatically assigned:  

```
$names = array("Peter","Quagmire","Joe");
```
- **Example 2:** The ID key manually assigned:  

```
$names[0] = "Peter";  
$names[1] = "Quagmire";  
$names[2] = "Joe";
```

10

## PHP ARRAYS – ASSOCIATIVE ARRAYS

- **Example 1:** use an array to assign ages to the different persons:  

```
$ages = array("Peter"=>32, "Quagmire"=>30, "Joe"=>34);
```
- **Example 2:** The Same as example 1, but shows a different way of creating the array:  

```
$ages['Peter'] = "32";  
$ages['Quagmire'] = "30";  
$ages['Joe'] = "34";
```

11

## PHP ARRAYS – MULTIDIMENSIONAL ARRAYS

```
$families = array  
(  
    "Griffin"=>array  
    (  
        "Peter",  
        "Lois",  
        "Megan"  
    ),  
    "Quagmire"=>array  
    (  
        "Glenn"  
    ),  
    "Brown"=>array  
    (  
        "Cleveland",  
        "Loretta",  
        "Junior"  
    )  
);
```

```
Array  
(  
    [Griffin] => Array  
    (  
        [0] => Peter  
        [1] => Lois  
        [2] => Megan  
    )  
    [Quagmire] => Array  
    (  
        [0] => Glenn  
    )  
    [Brown] => Array  
    (  
        [0] => Cleveland  
        [1] => Loretta  
        [2] => Junior  
    )  
)
```

12

## RESOURCES

- <http://www.php.net/>
- <http://www.w3schools.com/php/>