



VIT[®]
BHOPAL
www.vitbhopal.ac.in

VIT BHOPAL

UNIVERSITY

A PLACE TO LEARN; A CHANCE TO GROW

WELCOME TO

**CSE 4001 - INTERNET AND WEB
PROGRAMMING**

Unit 5

PHP and MYSQL PHP Basics – Arrays - Functions– Form Handling - File handling – Date and Time – MYSQL Basics - PHP and MYSQL Database Connectivity- Session - Cookies, E-mail – PHP and XML –PHP and AJAX.

Text Books:

- 1. Thomas Powell, HTML and CSS, Complete Reference, Fifth Edition, Mc Graw Hill, 2010
- 2. Thomas Powell, Fritz Schneider , JavaScript The complete reference, Mc Graw Hill, 2013
- 3. Tom Christiansen, Nathan Torkington, Perl Cookbook, O'Reilly, 2012
- 4. David Powers, PHP Solutions, Dynamic web page design made easy, Apress, 2010
- 5. Joe Fawcett, Danny Ayers, Liam R. E. Quin, Beginning XML, 5th Edition, Wrox, 2012

Reference Books:

- 1. Paul Dietel, Harvey Dietel and Abbey Dietel, Internet and World Wide Web How to program, 5th International Edition, Pearson, 2012

PHP

What is PHP?

- PHP is an acronym for "PHP: Hypertext Preprocessor"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use

PHP is an amazing and popular language!

- It is powerful enough to be at the core of the biggest blogging system on the web (WordPress)!

It is deep enough to run large social networks!

It is also easy enough to be a beginner's first server side language!

What is a PHP File?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code is executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies

- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data
- With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

Why PHP?

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)

- PHP supports a wide range of databases
- PHP is easy to learn and runs efficiently on the server side

What's new in PHP 7

- PHP 7 is much faster than the previous popular stable release (PHP 5.6)
- PHP 7 has improved Error Handling
- PHP 7 supports stricter Type Declarations for function arguments
- PHP 7 supports new operators (like the spaceship operator: `<=>`)

PHP Installation

- What Do I Need?
- To start using PHP, you can:
- Find a web host with PHP and MySQL support
- Install a web server on your own PC, and then install PHP and MySQL
- Use a Web Host With PHP Support
- If your server has activated support for PHP you do not need to do anything.
- Just create some .php files, place them in your web directory, and the server will automatically parse them for you.

- You do not need to compile anything or install any extra tools.
- Because PHP is free, most web hosts offer PHP support.
- Set Up PHP on Your Own PC
- However, if your server does not support PHP, you must:
- install a web server
- install PHP
- install a database, such as MySQL
- The official PHP website (PHP.net) has installation instructions for PHP: <http://php.net/manual/en/install.php>
- PHP Online Compiler / Editor

- PHP Syntax
- A PHP script is executed on the server, and the plain HTML result is sent back to the browser.
- Basic PHP Syntax
- A PHP script can be placed anywhere in the document.
- A PHP script starts with `<?php` and ends with `?>`:
- `<?php`
 `// PHP code goes here`
 `?>`

- The default file extension for PHP files is ".php".
- A PHP file normally contains HTML tags, and some PHP scripting code.
- Below, we have an example of a simple PHP file, with a PHP script that uses a built-in PHP function "echo" to output the text "Hello World!" on a web page:
- **Note:** PHP statements end with a semicolon (;).
- PHP Case Sensitivity
- In PHP, keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are not case-sensitive.
- In the example below, all three echo statements below are equal and legal:

- **Note:** However; all variable names are case-sensitive!
- Look at the example below; only the first statement will display the value of the \$color variable! This is because \$color, \$COLOR, and \$coLOR are treated as three different variables:

Ex:

https://www.w3schools.com/php/php_variables.asp

An array stores multiple values in one single variable:

- What is an Array?
- An array is a special variable, which can hold more than one value at a time.
- If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:


```
$cars1 = "Volvo";  
$cars2 = "BMW";  
$cars3 = "Toyota";
```
- However, what if you want to loop through the cars and find a specific one? And what if you had not 3 cars, but 300?
- The solution is to create an array!
- An array can hold many values under a single name, and you can access the values by referring to an index number.

- Create an Array in PHP
- In PHP, the array() function is used to create an array:
- array();
- In PHP, there are three types of arrays:
- **Indexed arrays** - Arrays with a numeric index
- **Associative arrays** - Arrays with named keys
- **Multidimensional arrays** - Arrays containing one or more arrays
- Get The Length of an Array - The count() Function
- The count() function is used to return the length (the number of elements) of an array:

- PHP Indexed Arrays
- There are two ways to create indexed arrays:
- The index can be assigned automatically (index always starts at 0), like this:
 - `$cars = array("Volvo", "BMW", "Toyota");`
 - or the index can be assigned manually:
 - `$cars[0] = "Volvo";`
 - `$cars[1] = "BMW";`
 - `$cars[2] = "Toyota";`

- PHP Associative Arrays
- Associative arrays are arrays that use named keys that you assign to them.
- There are two ways to create an associative array:
- `$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");`
- or:
- `$age['Peter'] = "35";`
`$age['Ben'] = "37";`
`$age['Joe'] = "43";`

- PHP - Multidimensional Arrays
- A multidimensional array is an array containing one or more arrays.
- PHP supports multidimensional arrays that are two, three, four, five, or more levels deep. However, arrays more than three levels deep are hard to manage for most people.
- **The dimension of an array indicates the number of indices you need to select an element.**
- For a two-dimensional array you need two indices to select an element
- For a three-dimensional array you need three indices to select an element

- PHP - Two-dimensional Arrays
- A two-dimensional array is an array of arrays (a three-dimensional array is an array of arrays of arrays).
- First, take a look at the following table:
- | Name | Stock | Sold |
|------------|-------|------|
| Volvo | 22 | 18 |
| BMW | 15 | 13 |
| Saab | 5 | 2 |
| Land Rover | 17 | 15 |
- We can store the data from the table above in a two-dimensional array, like this:
- ```
$cars = array (
 array("Volvo",22,18),
 array("BMW",15,13),
 array("Saab",5,2),
 array("Land Rover",17,15)
);
```

- Now the two-dimensional \$cars array contains four arrays, and it has two indices: row and column.
- To get access to the elements of the \$cars array we must point to the two indices (row and column):