

PHP and MySql - Seminar

By Petros Bournias
February 2010

Getting Started with PHP

Introducing PHP
Basic Syntax and Commands
Character sets, Constants, Expressions
Variables and Operators
Control Structures
Arrays
Functions and a simple explanation of Classes and Objects

PHP is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. As a general-purpose programming language, PHP code is processed by an interpreter application in command line mode performing desired operating system operations and producing program output on its standard output channel. It may also function as a graphical application. PHP is available as a processor for most modern web servers and as standalone interpreter on almost every operating system and computing platform.

What is the difference between a scripting language and a programming language?

The final result of a scripting language is not compiled whereas a programming language requires a compiler to be used before it can be executed, e.g. .com or .exe

<http://en.wikipedia.org/wiki/PHP>

The following site is basically your online book because it not only lists the commands but there are also special notes on many of them that will solve many problems for both Php and MySql. While it is better to use the online manual because it is constantly being updated like all technology, and with a variety of notes by professionals toward the bottom of pages, beginners can download a complete manual according to their version of Php so that they have a quick reference on their local system.

<http://www.php.net/>

Character Sets

As we are now in the era of UTF-8, I strongly suggest that you set your editor to use that as the default character set. This provides you with the ability to use of the majority of languages in the world today. This is the main reason that you want a UTF-8 editor. As also noted on the Elxis forum, some editors make use of a bom, but it is better to disable that if possible.

Php versions vary from 3 to the latest versions of 5.3, and their subsets.

This is a very important point because it may affect your programming as well especially between major version changes. The major version # is 3, 4, and 5 while the minor version # is the number after the period. Be aware that there may be a difference between your Internet web server version and the version that you use locally on your pc.

What are we doing with Php?

We write scripts. A script is a series of commands that when executed successfully produce results.

In working with Php, **you need to learn the basics first** just like in any programming or foreign language.

Php programs are files ending with .php even though the file ending maybe shown as .html due to htaccess parameters.

A line of Php code always begins with **<?php** ends with **?>** followed by a **semi-colon**; even if it spans multiple lines.

The Basics

The Php scripting language

<http://gr.php.net/manual/en/language.basic-syntax.php>

MySql functions available through Php

<http://gr.php.net/manual/en/ref.mysql.php>

Quotation marks

You will find that quotation marks can be a problem even at first but with experience you can learn to work with either double – “ “ or single ‘ ‘ quotes.

Variable types

Common variables can be string, numeric, Boolean, integer, arrays, objects, null, etc.. Variables have rules with regards to their creation and use a-z, A-Z, and 0-9 with some special characters.

<http://gr.php.net/manual/en/language.types.intro.php>

Constants

A variable that is pre-set to a value, whether that be string or numeric, etc..

An example is when you set a variable \$i = 0;

<http://gr.php.net/manual/en/language.constants.php>

Predefined Constants

A **protected constant** that is defined as a part of the **Php core**.

<http://gr.php.net/manual/en/array.constants.php>

Expressions

Assigning values to and from other **types**. An example is a function because you can provide values to variables in a function and have it return a result.

<http://gr.php.net/manual/en/language.expressions.php>

Operators

Use these to create your formulas. These are operators that help make decisions.

Examples are: < > != = == === && || AND OR i++ i--

<http://gr.php.net/manual/en/language.operators.php>

0 == false // this is true. The values are effectively the same in most contexts.

0 === false // this is false. One is an integer and the other is Boolean.

=== is used to check for equal values and both values must have same types.

Control Structures

Decision controls such as If else/if, while, do while, foreach, switch, etc.

<http://gr.php.net/manual/en/language.control-structures.php>

Arrays

One of the most important concepts that anyone should learn is how arrays work and how they can be used. Arrays can be simple and multi-dimensional and are a map of values and keys. An array in PHP is actually an ordered map. A map is a type that associates *values* to *keys*. This type is optimized for several different uses; it can be treated as an array, list (vector), hash table (an implementation of a map), dictionary, collection, stack, queue, and probably more. As array values can be other arrays, trees and multidimensional arrays are also possible.

<http://gr.php.net/manual/en/book.array.php>

In mathematics, the equivalent of an array is a matrix.

Functions

Programs that are written to do repetitive tasks and that provide a result.

An example of this can be something as simple as a javascript function to do simple mathematics within a program numerous times. Using a function also means reducing code since you are using a short script by just calling it and / or giving it some variable(s).

<http://gr.php.net/manual/en/language.functions.php>

Classes and Objects

These are in essence a combination of functions that can provide the functionality of a complete program.

An example of this is the combination such as:

```
library.php contains:
which can have various functions
    function users()
        which can have various functions
        newuser
        chguser

    function books()
        which can have various functions
        newbook
        chgbook
        borrow
        return
```

Please do not confuse this example with **Php libraries** that provide programmers with parts of programs that can be used to develop their own programs.

Another example of classes and objects here is Elxis, a CMS program that combines a variety of functions to provide a very complete program that gives the user an almost unlimited number of ways that it can be used without the user needing to write any programming code.

Classes and the information that follow on this php.net page is the knowledge that you want to learn if you wish to be able to use the language to the maximum.

<http://gr.php.net/manual/en/language.oop5.php>

An example of a Php program without database access

Help Forums

There are more in-depth areas of Php that require reading, analysis, and testing to better understand the openness and value of this language.

While the Elxis forum has made an effort to provide this information, questions regarding Php programming should only be regarding specific Elxis questions.

Help is available on both English and Greek **Php forums**. The clearer your question is, the better an answer you will receive. I have included some links to forums and for Php code in the first document but you will find many of them through the search engines.

Other areas of Php

There are many other areas of Php programming that deal with security, networking, sessions, cookies, global versus local variables, etc.

Other additions to Php include libraries such as Pear, GD for graphics, etc. They provide additional pre-packaged standardized script solutions. Be aware that pear may not be available on every web server.

http://en.wikipedia.org/wiki/List_of_PHP_libraries

A word of warning

One of the major requests by users is that of forms but the creation of a form should not be taken lightly because the **improper programming of a form can create a nightmare due to hackers**.

It is fairly easy to write a Php program but it requires a thorough understanding of the basics to protect your website from hackers breaking your forms, accessing your database, sending emails through your email server as spam, and even worse, accessing your server due to poor programming or the wrong use of file permissions.

What this means is to test your programs thoroughly and even ask someone else you know to test them on the Internet if you can before you release them to the public on the Internet. There will be a difference between using your local Php, MySql and Apache web server