

Tutorial 1

Introduction

PHP is a simple, elegant and powerful programming language that can be utilised for any type of web application, from simple home pages to full business solutions. The language basics are similar to those of any other programming language. This work sheet contains a summary of PHP syntax and constructs that will assist you in completing the accompanying exercises.

Convention

This type of text indicates a code example / listing. It is advisable to type the code to get familiar with the syntax.

Italic Text indicates code elements in line with a sentence. E.g.

The `<h1>` element

What You Need For This Tutorial

- Web Development Environment (e.g. WAMP Server or XAMPP etc)
- Familiarity with Web Authoring Tools (e.g. Adobe Dreamweaver or Microsoft Expression Web etc)
- Basic knowledge of HTML.
- General Understanding of programming concepts: Variables, Methods and branching structures (E.g. IF Statements)

Syntax & Structure

All script statements are contained within PHP brackets and a semi-colon terminates each statement, you can have as many PHP blocks in your mark-up as you wish:

```
<?php
echo "Just like this";
?>
```

White space is generally ignored (except when contained in a string variable). Comments are created using a double forward slash for single line and slash-star notation for multiline.

```
// This is a single line comment, next line is treated as code
/* Unless a new comment is used, this is a multi-line comment
so everything that is contained within it is disregarded by
PHP */
```

Precedence is determined by parentheses and follows standard mathematical order: Brackets, Division, Multiplication, Addition and Subtraction respectively.

PHP V

```
$sum = 5 + 3 * 6;
echo $sum; // prints 23
$sum = (5 + 3) * 6;
echo $sum; // prints 48
```

Variables are prefixed with a dollar sign (\$) and should not contain a number as their first character. Variable names are case sensitive. A variable is assigned a value by using the equal sign operator (=).

```
$name = "Trevor Adams";  
echo $Name; //prints nothing as the variable is undefined
```

There is no need to define an initial value when declaring variables. PHP is not a typed language and does not require that you specify the type with a variable. A variable's type will alter in how it is used.

PHP Constants

Constants are ideal for frequently used values that change infrequently (or sometimes never at all). They are declared using the define function and are used in code without a preceding dollar (\$).

```
<?php  
define("pi", 3.14159);  
$r = 4;  
$area = pi * $r *$r;  
?>
```

PHP Operators

Standard mathematical operators are similar to other languages: plus, minus, divide and multiply (+, -, /, *). The calculated remainder of a division, the modulus, is given by the percent symbol (%). String concatenation is performed using a period (.).

```
<?php  
$sum = 5 + 5;  
$difference = 10 - 5;  
$product = 5 * 5;  
$division = 10 / 5;  
$remainder = 10 % 4;  
$name = "Trevor Adams";  
echo "Hello, " . $name; // output: Hello, Trevor Adams  
?>
```

Conditional Branching

A logical *IF* statement is treated as a singular and does not require a semi-colon to terminate (same with all PHP block constructs). Each branch is contained within a set of brace brackets {}.

```
<?php  
if($name == "Trevor Adams") {  
    echo "Welcome, Trevor";  
} else {  
    echo "I do not know who you are!";  
}  
?>
```