# Hello World

<!DOCTYPE html>

<html>

<body>

<h1>My first PHP page</h1>

<?php

echo "Hello World!";

?>

</body>

</html>

# Echo in PHP

PHP script that uses a built-in PHP function "echo" to output the text "Hello World!" on a web page:

<!DOCTYPE html>

<html>

<body>

<h1>My first PHP page</h1>

**<?php**

echo "Hello World!";

**?>**

</body>

</html>

## **PHP Case Sensitivity**

keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are not case-sensitive.

### **Example**

ECHO is the same as echo:

<!DOCTYPE html>

<html>

<body>

**<?php**

ECHO "Hello World!<br>";

echo "Hello World!<br>";

EcHo "Hello World!<br>";

**?>**

</body>

</html>

Syntax for comments in PHP code:

// This is a single-line comment

# This is also a single-line comment

/\* This is a

multi-line comment \*/

# Variables in PHP

## **Creating (Declaring) PHP Variables**

a variable starts with the $ sign,

Think of variables as containers for storing data.

## **PHP Variables**

A variable can have a short name (like $x and $y) or a more descriptive name ($age, $carname, $total\_volume).

Rules for PHP variables:

* A variable starts with the $ sign, followed by the name of the variable
* A variable name must start with a letter or the underscore character
* A variable name cannot start with a number
* A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_ )
* Variable names are case-sensitive ($age and $AGE are two different variables)

Remember that PHP variable names are case-sensitive!

## **Output Variables**

The PHP echo statement is often used to output data to the screen.

$txt = "W3Schools.com";

echo "I love ". $txt. "!";

$x = 5;

$y = 4;

echo $x + $y;

$x = 5;// $x is an integer

$y = "John"; // $y is a string

echo $x;

echo $y;

PHP supports the following data types:

* String
* Integer
* Float (floating point numbers - also called double)
* Boolean
* Array
* Object
* NULL
* Resource

int(5)

string(4) "John"

float(3.14)

bool(true)

array(3) {

[0]=>

int(2)

[1]=>

int(3)

[2]=>

int(56)

}

NULL

* var\_dump(5);
* var\_dump("John");
* var\_dump(3.14);
* var\_dump(true);
* var\_dump([2, 3, 56]);
* var\_dump(NULL);

## **PHP Variables Scope**

In PHP, variables can be declared anywhere in the script.

The scope of a variable is the part of the script where the variable can be referenced/used.

PHP has three different variable scopes:

* local
* global
* static

## **Global and Local Scope**

A variable declared **outside** a function has a GLOBAL SCOPE and can only be accessed outside a function:

$x = 5; // global scope

function myTest() {

// using x inside this function will generate an error

echo "<p>Variable x inside function is: $x</p>";

}

myTest();

echo "<p>Variable x outside function is: $x</p>";

variable declared **within** a function has a LOCAL SCOPE and can only be accessed within that function:

### **Example**

Variable with local scope:

function myTest() {

$x = 5; // local scope

echo "<p>Variable x inside function is: $x</p>";

}

myTest();

// using x outside the function will generate an error

echo "<p>Variable x outside function is: $x</p>";

## **PHP the static Keyword**

Normally, when a function is completed/executed, all of its variables are deleted. However, sometimes we want a local variable NOT to be deleted. We need it for a further job.

To do this, use the static keyword when you first declare the variable:

### **Example**

function myTest() {

static $x = 0;

echo $x;

$x++;

}

myTest();

myTest();

myTest();

# **PHP echo and print Statements**

there are two basic ways to get output: echo and print.

The differences are small: echo has no return value while print has a return value of 1 so it can be used in expressions. echo can take multiple parameters (although such usage is rare) while print can take one argument. echo is marginally faster than print.

## **The PHP echo Statement**

echo statement can be used with or without parentheses: echo or echo().

### **Example**

echo "Hello";

//same as:

echo("Hello");