

# 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 Syntax Sample**

```
<?php  
// PHP code goes here  
?>
```

- The default file extension for PHP files is ".php".

# PHP Syntax

- 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:
- PHP statements end with a semicolon (;).

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
    echo "Hello World!";
?>

</body>
</html>
```

# PHP Syntax

## ■ Comments in PHP

A comment in PHP code is a line that is not read/executed as part of the program. Its only purpose is to be read by someone who is looking at the code.

```
<?php
// This is a single-line comment
# This is also a single-line comment
/*
This is a multiple-lines comment block
that spans over multiple
lines
*/
// You can also use comments to leave out parts of a
code line
$x = 5 /* + 15 */ + 5;
echo $x;
?>
```

# PHP Syntax

## ■ PHP Case Sensitivity

In PHP, NO keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are case-sensitive.

```
<!DOCTYPE html>
<html>
<body>
<?php
    ECHO "Hello World!<br>";
    echo "Hello World!<br>";
    Echo "Hello World!<br>";
?>
</body>
</html>
```

 However; all variable names are case-sensitive.

# PHP Syntax

## ■ PHP Case Sensitivity

```
<!DOCTYPE html>
<html>
<body>

<?php
    $color = "red";
    echo "My car is " . $color . "<br>";
    echo "My house is " . $COLOR . "<br>";
    echo "My boat is " . $coLOR . "<br>";
?>

</body>
</html>
```

# PHP Variables

- Variables are "containers" for storing information.
- In PHP, a variable starts with the **\$** sign, followed by the name of the variable:

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

- ⚠ When you assign a text value to a variable, put quotes around the value.
- ⚠ Unlike other programming languages, PHP has no command for declaring a variable. It is created the moment you first assign a value to it.
- ⚠ 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

```
<?php
$txt = "Web Development";
echo "I love $txt!";
?>
```

```
<?php
$txt = "Web Development";
echo "I love " . $txt . "!";
?>
```

```
<?php
$x = 5;
$y = 4;
echo $x + $y;
?>
```

# PHP is a Loosely Typed Language

- In the example above, notice that we did not have to tell PHP which data type the variable is.
- PHP automatically converts the variable to the correct data type, depending on its value.
- In other languages such as C, C++, and Java, the programmer must declare the name and type of the variable before using it.

# 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:

```
<?php
$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>";
?>
```

# Global and Local Scope

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

```
<?php
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>";
?>
```

- ⚠ You can have local variables with the same name in different functions, because local variables are only recognized by the function in which they are declared.

# PHP The global Keyword

- The **global** keyword is used to access a global variable from within a function.

```
<?php
$x = 5;
$y = 10;

function myTest() {
    global $x, $y;
    $y = $x + $y;
}

myTest();
echo $y; // outputs 15
?>
```

# PHP The global Keyword

- PHP also stores all global variables in an array called `$GLOBALS[index]`. The index holds the name of the variable. This array is also accessible from within functions and can be used to update global variables directly.

```
<?php
$x = 5;
$y = 10;

function myTest() {
    $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
}

myTest();
echo $y; // outputs 15
?>
```

# 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.

```
<?php
function myTest() {
    static $x = 0;
    echo $x;
    $x++;
}

myTest();
myTest();
myTest();
?>
```

 The variable is still local to the function.

# PHP echo and print Statements

- In PHP there are two basic ways to get output:
  - *echo*
  - *print*
- *echo* and *print* are more or less the same. They are both used to output data to the screen.
- 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*.

# PHP echo Statement

- The **echo** statement can be used with or without parentheses: **echo** or **echo()**
- Display Text

```
<?php
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple
parameters.";
?>
```

# PHP echo Statement

## ■ Display Variables

```
<?php
$txt1 = "Learn PHP";
$txt2 = "Web Development";
$x = 5;
$y = 4;

echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
?>
```

# PHP print Statement

- The `print` statement can be used with or without parentheses: `print` or `print ()`
- Display Text

```
<?php  
print "<h2>PHP is Fun!</h2>";  
print "Hello world!<br>";  
print "I'm about to learn PHP!";  
?>
```

# PHP print Statement

## ■ Display Variables

```
<?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;

print "<h2>" . $txt1 . "</h2>";
print "Study PHP at " . $txt2 . "<br>";
print $x + $y;
?>
```

# PHP Data Types

- Variables can store data of different types, and different data types can do different things.
- PHP supports the following data types:
  - *String*
  - *Integer*
  - *Float (floating point numbers - also called double)*
  - *Boolean*
  - *Array*
  - *Object*
  - *NULL*
  - *Resource*

# PHP String

- string is a sequence of characters, like "Hello world!".
- A string can be any text inside quotes. You can use single or double quotes

```
<?php
$x = "Hello world!";
$y = 'Hello world!';

echo $x;
echo "<br>";
echo $y;
?>
```

# PHP Integer

- An integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647.
- Rules for integers:
  - An integer must have at least one digit
  - An integer must not have a decimal point
  - An integer can be either positive or negative
  - Integers can be specified in three formats: decimal (10-based), hexadecimal (16-based - prefixed with 0x) or octal (8-based - prefixed with 0)

```
<?php  
$x = 5985;  
var_dump($x);  
?>
```

. \* The PHP var\_dump() function returns the data type and value

# PHP Float

- A float (floating point number) is a number with a decimal point or a number in exponential form.

```
<?php  
$x = 10.365;  
var_dump($x);  
?>
```

# PHP Boolean

- A Boolean represents two possible states: TRUE or FALSE

```
$x = true;  
$y = false;
```

- Booleans are often used in conditional testing.

# PHP Array

- An array stores multiple values in one single variable.

```
<?php  
$cars = array("Volvo", "BMW", "Toyota");  
var_dump($cars);  
?>
```

- we will learn a lot more about arrays later.

# PHP Object

- An object is a data type which stores data and information on how to process that data.
- In PHP, an object must be explicitly declared.

```
<?php
class Car {
    function Car() {
        $this->model = "BMW";
    }
}

// create an object
$herbie = new Car();

// show object properties
echo $herbie->model;
?>
```

# PHP NULL Value

- Null is a special data type which can have only one value: NULL.
- A variable of data type NULL is a variable that has no value assigned to it.
- Variables can also be emptied by setting the value to NULL:

```
<?php
$x = "Hello world!";
$x = null;
var_dump($x);
?>
```

- ⚠ If a variable is created without a value, it is automatically assigned a value of NULL.

# PHP String Functions

## ■ Get The Length of a String

The PHP `strlen()` function returns the length of a string.

```
<?php
echo strlen("Hello world!"); // outputs 12
?>
```

## ■ Count The Number of Words in a String

The PHP `str_word_count()` function counts the number of words in a string.

```
<?php
echo str_word_count("Hello world!"); // outputs 2
?>
```

# PHP String Functions

## ■ Reverse a String

The PHP `strrev()` function reverses a string.

```
<?php
echo strrev("Hello world!"); // outputs !dlrow olleH
?>
```

## ■ Search For a Specific Text Within a String

The PHP `strpos()` function searches for a specific text within a string. If a match is found, the function returns the character position of the first match. If no match is found, it will return `FALSE`.

```
<?php
echo strpos("Hello world!", "world"); // outputs 6
?>
```

# PHP String Functions

## ■ Replace Text Within a String

The PHP `str_replace()` function replaces some characters with some other characters in a string.

```
<?php
echo str_replace("world", "Dolly", "Hello world!"); //
outputs Hello Dolly!
?>
```

PHP string functions complete List is on the website