# Syntax, Variables & Strings

## Basic PHP Syntax

You cannot view the PHP source code by selecting "View source" in the browser - you will only see the output from the PHP file, which is plain HTML.

This is because the scripts are executed on the server before the result is sent back to the browser.

A PHP scripting block always starts with **<?php** and ends with **?>**. A PHP scripting block can be placed anywhere in the document.

On servers with shorthand support enabled you can start a scripting block with **<?** and end with **?>**.

However, for maximum compatibility, we recommend that you use the standard form (<?php) rather than the shorthand form.

```
<?php

?>
```

## Basic PHP Syntax

A PHP file normally contains HTML tags, just like an HTML file, and some PHP scripting code.

Below, we have an example of a simple PHP script which sends the text "Hello World" to the browser:

```
<html>
<body>

<?php
echo "Hello World";
?>

</body>
</html>
```

Each code line in PHP must end with a semicolon. The semicolon is a separator and is used to distinguish one set of instructions from another.

There are two basic statements to output text with PHP: echo and print. In the example above we have used the echo statement to output the text "Hello World".

# Syntax, Variables & Strings

## Comments In PHP

In PHP, we use // to make a single-line comment or /* and */ to make a large comment block.

```
<html>
<body>

<?php

//This is a comment

/*
This is
a comment
block
*/

?>

</body>
</html>
```

## PHP Variables

Variables are used for storing a values, like text strings, numbers or arrays.

When a variable is set it can be used over and over again in your script

All variables in PHP start with a $ sign symbol.

The correct way of setting a variable in PHP:

```
$var_name = value;
```

# Syntax, Variables & Strings

## PHP Variables

New PHP programmers often forget the $ sign at the beginning of the variable. In that case it will not work.

Let's try creating a variable with a string, and a variable with a number:

```
<?php

$txt = "Hello World!";
$number = 16;

?>
```

## PHP is a Loosely Typed Language

In PHP a variable does not need to be declared before being set.

In the previous example, you see that you do not have to tell PHP which data type the variable is.

PHP automatically converts the variable to the correct data type, depending on how they are set.

In a strongly typed programming language, you have to declare (define) the type and name of the variable before using it.

In PHP the variable is declared automatically when you use it.

# Syntax, Variables & Strings

## Variable Naming Rules

- A variable name must start with a **letter** or an underscore "_"
- A variable name can only contain alpha-numeric characters and underscores (a-Z, 0-9, and _ )
- A variable name should not contain spaces. If a variable name is more than one word, it should be separated with underscore ($my_string), or with capitalization ($myString)

# Syntax, Variables & Strings

## Strings In PHP

String variables are used for values that contains character strings.

We are going to look at some of the most common functions and operators used to manipulate strings in PHP.

After we create a string we can manipulate it. A string can be used directly in a function or it can be stored in a variable.

Below, the PHP script assigns the string "Hello World" to a string variable called $txt:

```
<?php

$txt="Hello World";
echo $txt;

?>
```

## Strings In PHP

The output of the code will be:

```
Hello World
```

Now, lets try to use some different functions and operators to manipulate our string.

# Syntax, Variables & Strings

## The Concatenation Operator

There is only one string operator in PHP.

The concatenation operator (.)  is used to put two string values together.

To concatenate two variables together, use the dot (.) operator:

```
<?php

$txt1="Hello World";
$txt2="1234";

echo $txt1 . " " . $txt2;

?>
```

If we look at the code above you see that we used the concatenation operator two times. This is because we had to insert a third string.

Between the two string variables we added a string with a single character, an empty space, to separate the two variables.

# Syntax, Variables & Strings

## Strings In PHP

The output of the code will be:

```
Hello World 1234
```

## Using the strlen() function

The strlen() function is used to find the length of a string.

Let's find the length of our string "Hello world!":

```
<?php

echo strlen("Hello world!");

?>
```

The output of the code above will be:

```
12
```

The length of a string is often used in loops or other functions, when it is important to know when the string ends. (i.e. in a loop, we would want to stop the loop after the last character in the string)

## Using the strpos() function

The strpos() function is used to search for a string or character within a string.

If a match is found in the string, this function will return the position of the first match.
If no match is found, it will return FALSE.

Let's see if we can find the string "world" in our string:

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

The output of the code above will be:

```
6
```

As you see the position of the string "world" in our string is position 6. The reason that it is 6, and not 7, is that the first position in the string is 0, and not 1.

## Complete PHP String Reference

For a complete reference of all string functions, go to the W3C complete PHP String Reference.

http://w3schools.com/php/php_ref_string.asp

The reference contains a brief description and examples of use for each function!