BASICS OF PHP

PHP Syntax

A PHP script can be placed anywhere in the document.

A PHP script starts with <?php and ends with ?>:

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

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

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

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:

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


<h1>My first PHP page</h1>


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

```
</body>
</html>
```

PHP statements end with a semicolon (;).

PHP Case Sensitivity

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

In the example below, all three echo statements below are equal and legal:

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>
```

Look at the example below; only the first statement will display the value of the $color variable! This is because $color, $COLOR, and $coLOR are treated as three different variables:

Example

$COLOR is *not* same as $color:

```
<!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 Comments

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

Comments can be used to:

- Let others understand your code
- Remind yourself of what you did - Most programmers have experienced coming back to their own work a year or two later and having to re-figure out what they did. Comments can remind you of what you were thinking when you wrote the code
- Leave out some parts of your code

PHP supports several ways of commenting:

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 */
```

Single Line Comments:

Single line comments start with //.

Any text between // and the end of the line will be ignored (will not be executed).

You can also use # for single line comments, but in this tutorial we will use //.

The following examples uses a single-line comment as an explanation:

Example

A comment before the code:

```
// Outputs a welcome message:

echo "Welcome Home!";
```

Example

A comment at the end of a line:

```
echo "Welcome Home!"; // Outputs a welcome message
```

Comments to Ignore Code

We can use comments to prevent code lines from being executed:

Example

Do not display a welcome message:

```
// echo "Welcome Home!";
```

Multi-line Comments:

Multi-line comments start with /* and end with */.

Any text between /* and */ will be ignored.

The following example uses a multi-line comment as an explanation:

Example

Multi-line comment as an explanation:

```
/*
The next statement will
print a welcome message
*/
echo "Welcome Home!";
```

Multi-line Comments to Ignore Code:

We can use multi-line comments to prevent blocks of code from being executed:

Example

Multi-line comment to ignore code:

```
/*
echo "Welcome to my home!";
echo "Mi casa su casa!";
*/
echo "Hello!";
```

Comments in the Middle of the Code:

The multi-line comment syntax can also be used to prevent execution of parts inside a code-line:

Example

The + 15 part will be ignored in the calculation:

```
$x = 5 /* + 15 */ + 5;
echo $x;
```

Comments are crucial for improving code readability and maintainability. They help developers understand the purpose and functionality of different parts of the code, making it easier to collaborate and troubleshoot issues.

```php
<?php

// This is a single-line comment

# This is also a single-line comment

/*
 This is a multi-line comment
 spanning multiple lines
*/

/**
 * This is a PHPDoc comment.
 * It provides information about the purpose of a function, class, or variable.
 */

echo "Hello, World!"; // This is a comment at the end of a line

?>
```