Introduction to PHP

Overview

- Introduction to PHP Fundamentals
- What is a Scripting Language?
- PHP Variable,Constant & Operators
- PHP Comments
- Include/Include_once,
- Require/Require_once
- Arrays

PHP (HyperText Pre-processor)

- PHP is a server side scripting language.
- PHP is open source and cross platform
- It is used to develop Static websites or Dynamic websites or Web applications.
- PHP scripts can only be interpreted on a server that has PHP installed.
- The client computers accessing the PHP scripts require a web browser only.
- A PHP file contains PHP tags and ends with the extension ".php".
- PHP code can be embedded into HTML code(**<HTML><PHP CODE></HTML>**).
- It can be used in combination with various web template systems, web content management system and web frameworks.

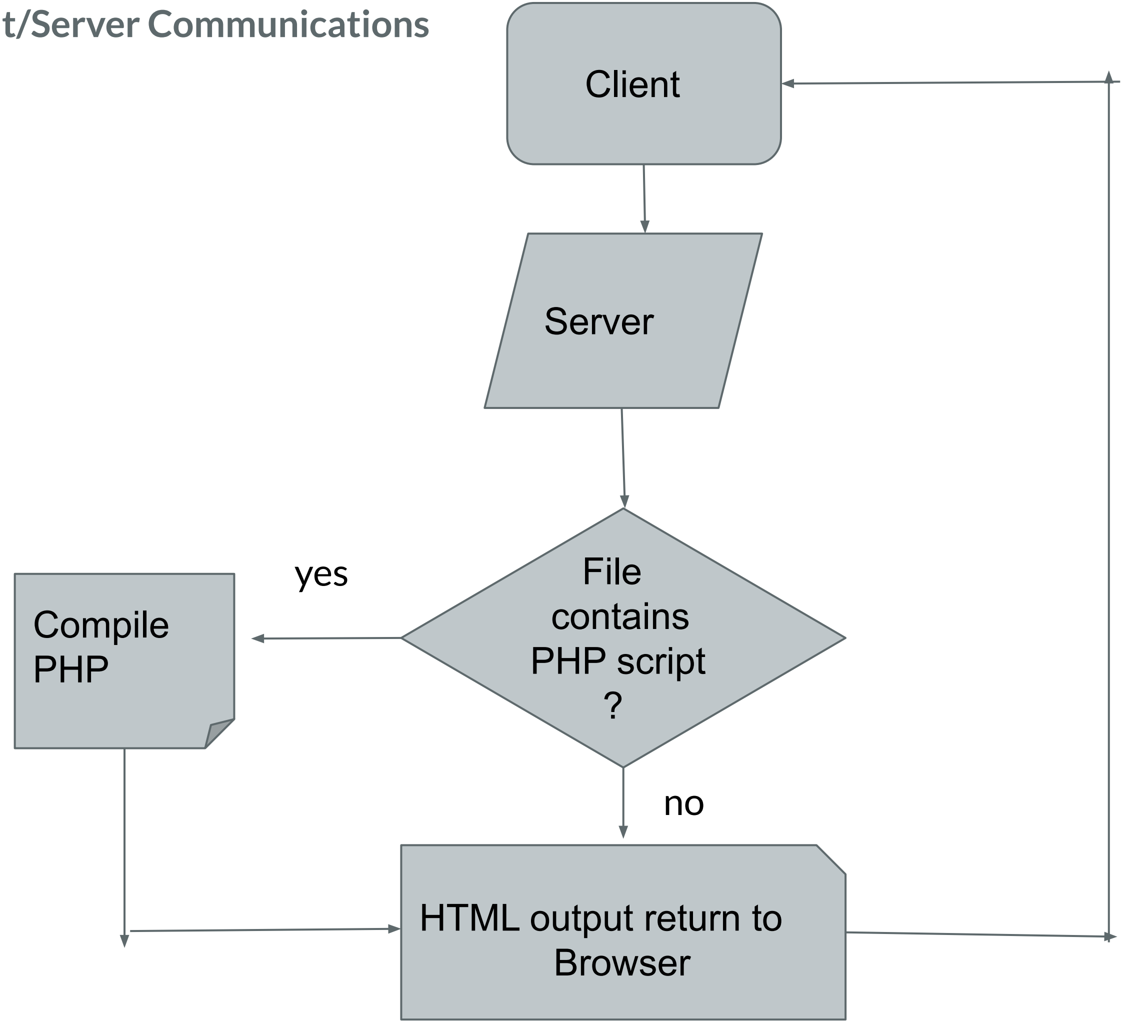
Programming language

- Has all the features needed to develop complete applications.
- The code has to be compiled before it can be executed.
- Does not need to be embedded into other languages

Scripting language

- Mostly used for routine tasks.
- The code is usually executed without compiling.
- Is usually embedded into other software environments.

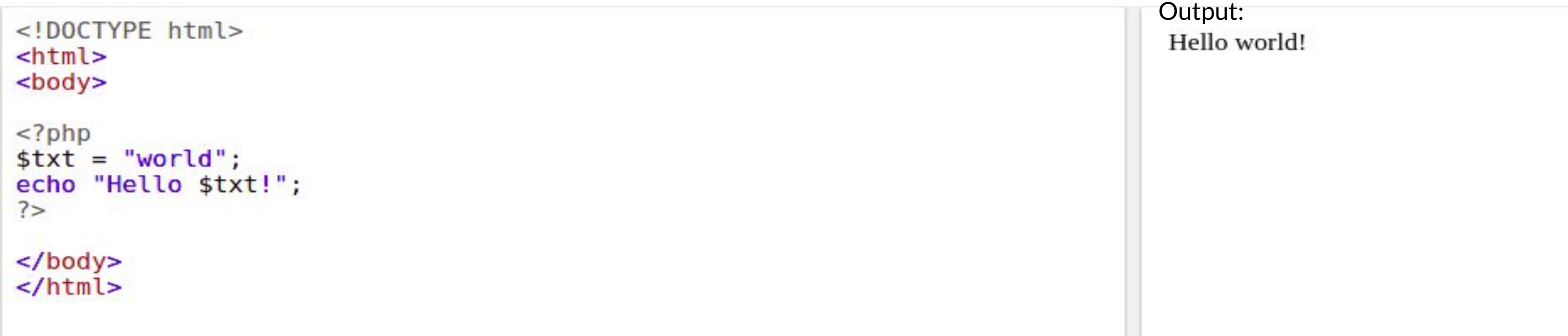
Client/Server Communications



PHP Variable

- A variable is a name given to a memory location that stores data at runtime.
- A Php global variable is accessible to all the scripts in an application.
- A local variable is only accessible to the script that it was defined in.
- All variable names must start with the dollar sign (e.g. $txt)
- Variable names are case sensitive; this means $my_var is different from $MY_VAR.
- All variables names must start with a letter follow other characters e.g. $my_var1.But $1my_var is not a legal variable name.
- Variable names must not contain any spaces, “$first name” is not a legal variable name

The program shown below is a basic PHP application that outputs the words “Hello World!” When viewed in a web browser.
PHP statements end with a semicolon (;).



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

<?php
$txt = "world";
echo "Hello $txt!";
?>

</body>
</html>
```

Output:

```
Hello world!
```

Variable scope

The user can declare variables anywhere in the PHP script, outside of a function or within one. The exact part in which a certain variable can be used is called PHP variable scope.
There are three main scopes:

- Local
- Global
- Static

Global Keyword: The global keyword imports variables from the global scope into the local scope of a function.

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.
Local Scope: A variable declared within a function has a LOCAL SCOPE and can only be accessed within that function.

PHP constant- A constant is a variable whose value cannot be changed at runtime.

```
define('PI',3.14);
```

PHP Operators:

The PHP **arithmetic operators** are used with numeric values to perform common arithmetical operations, such as addition, subtraction, multiplication etc.

Assignment operators are used with numeric values to write a value to a variable.

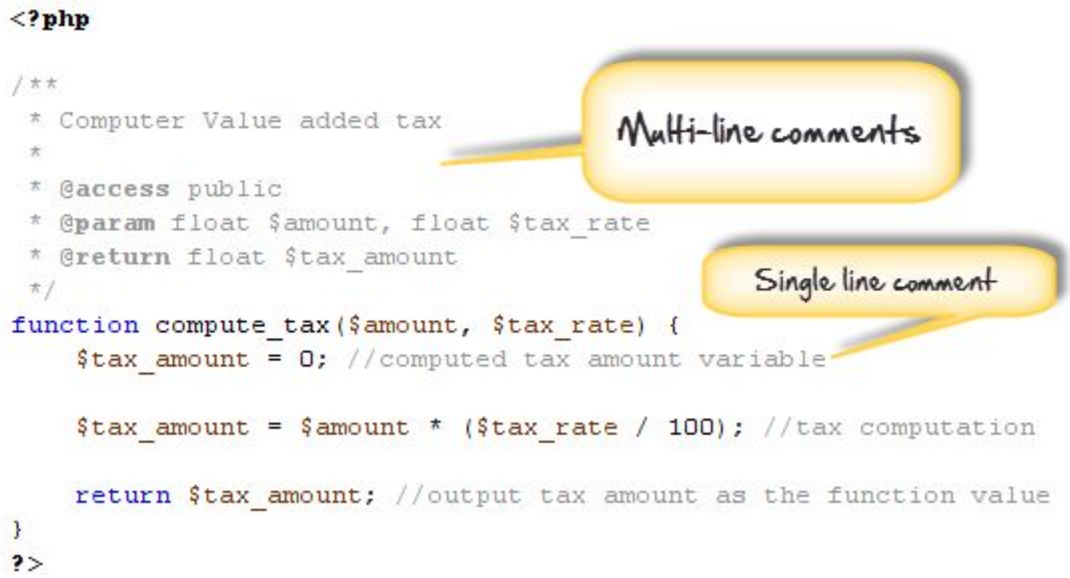
Comparison operators are used to compare two values (number or string):

Logical operators are used to combine conditional statements.

Ref. Link: https://www.w3schools.com/php/php_operators.asp

PHP Comments:

- Comments help us to understand the code.
- Comments are explanations that we include in our source code. These comments are for human understanding.
- Single line comments start with double forward slashes // and they end in the same line.
- Multiple line comments start with a forward slash followed by the asterisk /* and end with the asterisk followed by the forward slash */.



```php
<?php

/**
 * Computer Value added tax
 *
 * @access public
 * @param float $amount, float $tax_rate
 * @return float $tax_amount
 */
function compute_tax($amount, $tax_rate) {
    $tax_amount = 0; //computed tax amount variable

    $tax_amount = $amount * ($tax_rate / 100); //tax computation

    return $tax_amount; //output tax amount as the function value
}
?>
```

Include/Include_once, Require/Require_once

- The “include” php statement is used to include other files into a PHP file.

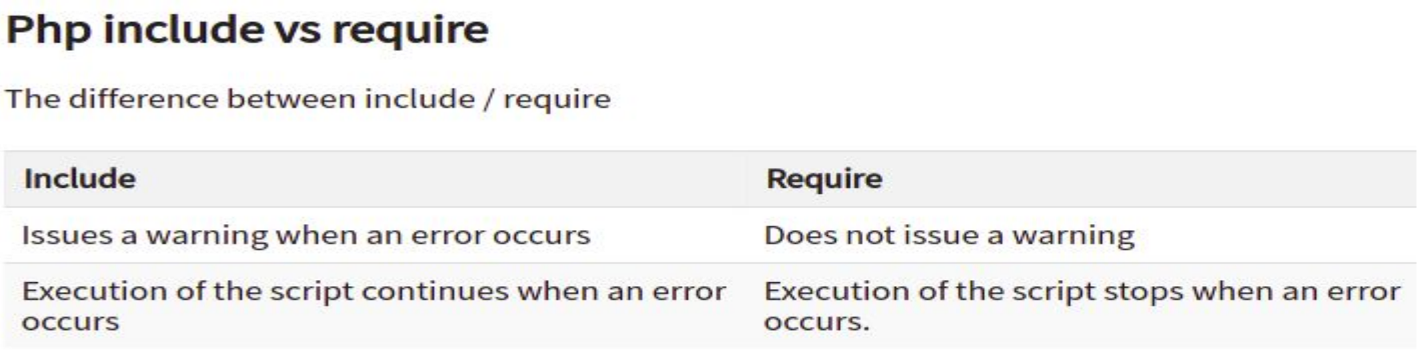
```
      <?php
       include 'file_name';
      ?>
```

- Include_once is ignored by the PHP interpreter if the file to be included.

```
      <?php
       include_once 'file_name';
      ?>
```

PHP Require & PHP require_once

- The require/require_once statement is used to include file.
- require_once is ignored if the required file has already been added by any of the four include statements.



Php include vs require

The difference between include / require

Include	Require
Issues a warning when an error occurs	Does not issue a warning
Execution of the script continues when an error occurs	Execution of the script stops when an error occurs.

Note: “Include/include_once” is the statement that includes file. “'file_name'” is the name of the file to be included.

Arrays

- A PHP array is a variable that stores more than one piece of related data in a single variable.

Numeric Arrays:

- Numeric arrays use number as access keys.
- An access key is a reference to a memory slot in an array variable.

```
<?php
$variable_name[n] = value;
?>
```

```
<?php
$variable_name = array(0 => ‘hello’, …);
?>
```

PHP Associative Array:

- Associative array differ from numeric array in the sense that associative arrays use descriptive names for id keys.

```
    <?php
     $variable_name['key_name'] = value;
     $variable_name = array('name' => ‘valuebound’);
    ?>
```

```
    <?php
      $persons = array("Mary" => "Female", "John" => "Male", "Mirriam" => "Female");
      print_r($persons);
      echo "Mary is a " . $persons["Mary"];
     ?>
```

PHP Multi-dimensional arrays:

- These are arrays that contain other nested arrays.

```
<?php
$movies =array(
      "comedy" => array("Pink Panther", "John English", "See no evil hear no evil"),
      "action" => array("Die Hard", "Expendables")
      );
print_r($movies);
?>
```

- A short array syntax exists which replaces array() with [].

Reference Links:

Online Compiler : https://www.w3schools.com/php/phptryit.asp?filename=tryphp_compiler
Online Exercise: https://www.w3schools.com/php/exercise.asp?filename=exercise_syntax1
PHP Operators: https://www.w3schools.com/php/php_operators.asp
PHP Arrays: https://www.w3schools.com/php/php_arrays.asp

Thanks!