PHP BASIC

BEFORE WE GET STARTED

1. Using Windows 10 or Ubuntu 16.04.
2. Installed Git.
3. Installed Chrome Web Browser.
4. Installed VS Code.
5. Installed XAMPP for Windows PHP 7.0.
6. Checkout : https://github.com/namnh06/aptech-php-course

SET UP ENVIROMENT

1. Turn on Git Bash (on Windows).
2. Move to C:/xampp/htdocs/ by command line : **cd /c/xampp/htdocs** -> Enter.
3. Clone Repository by Git Hub, using this command line : git clone https://github.com/namnh06/aptech-php-course.git -> Enter.
4. Using this repository to learn and get homework, exercise from there.
5. Create your repository with convention code : Aptech-php-*-your-name. E.G : Aptech-php-12-nam-nh.
6. Push your repository to Git Hub and do your home work.

SYNTAX, ECHO, COMMENT

- PHP files have “.php” extension.
- PHP script always starts with <?php and end with ?>, in some case no need to close ?> in file .php. The end of statement of code always need “;”.
- Comment starts with “//”, “#” or “/* */”.
- Output into HTML Dom by using “echo”, “print”, “print_r” or “var_dump”.
- Can write HTML, CSS, Javascript inside “.php” file.

```php
<?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
*/

echo "Hello World";
```

VARIABLE, CONSTANT

- Variable starts with “$” (dollar sign), no need to declare type of variable.
- Variable name must start with a letter or underscore character “_” and can not start with number.
- Variable name can only contain alpha-numeric characters and underscores (A-z, 0-9 and _).
- Using “global”, “static” keyword to make variable “special”.
- Using function “define” to create a constant.

```php
<?php
$currentYear = 2018;

function forecast()
{
    global $currentYear;
    $nextYear = ++$currentYear;
    echo "Next Year is $nextYear";
}

forecast();
```

```php
<?php
define("CURRENT_YEAR", 2018);

function forecast()
{
    $nextYear = CURRENT_YEAR + 1;
    echo "Next Year is $nextYear";
}

forecast();
```

DATA TYPES

- String : a sequence of characters.
- Integer : non-decimal number between -2,147,483,648 and 2,147,483,647.
- Float : decimal number.
- Boolean : TRUE/FALSE.
- Array : stores multiple values.
- Object : stores data and information or properties and function.
- NULL : is NULL.

```php
<?php

$string = "Hello World";

$integer = 2018;

$float = 99.99;

$boolean = true;

$array = [1, 2, 3, 4, 5];
```

OPERATORS

- Arithmetic Operators

Operator	Name	Example	Result
+	Addition	$x + $y	Sum of $x and $y
-	Subtraction	$x - $y	Difference of $x and $y
*	Multiplication	$x * $y	Product of $x and $y
/	Division	$x / $y	Quotient of $x and $y
%	Modulus	$x % $y	Remainder of $x divided by $y
**	Exponentiation	$x ** $y	Result of raising $x to the $y'th power (Introduced in PHP 5.6)

```php
<?php
$three = 3;
$nine = 9;

// Additional
$twelve = $three + $nine;

// Subtraction
$six = $nine - $three;

// Multiplication
$twentySeventh = $nine * $three;

// Division
$anotherThree = $nine / $three;

// Modulus
$zero = $none % $three;

// Exponentiation
$sevenHundredAndTwentyNinth = $nine ** $three;
```

OPERATORS (cont. 1)

- Assignment Operators

Assignment	Same as...	Description
x = y	x = y	The left operand gets set to the value of the expression on the right
x += y	x = x + y	Addition
x -= y	x = x - y	Subtraction
x *= y	x = x * y	Multiplication
x /= y	x = x / y	Division
x %= y	x = x % y	Modulus

OPERATORS (cont. 2)

- Comparison Operators

Operator	Name	Example	Result
==	Equal	$x == $y	Returns true if $x is equal to $y
===	Identical	$x === $y	Returns true if $x is equal to $y, and they are of the same type
!=	Not equal	$x != $y	Returns true if $x is not equal to $y
<>	Not equal	$x <> $y	Returns true if $x is not equal to $y
!==	Not identical	$x !== $y	Returns true if $x is not equal to $y, or they are not of the same type
>	Greater than	$x > $y	Returns true if $x is greater than $y
<	Less than	$x < $y	Returns true if $x is less than $y
>=	Greater than or equal to	$x >= $y	Returns true if $x is greater than or equal to $y
<=	Less than or equal to	$x <= $y	Returns true if $x is less than or equal to $y

- Increment/Decrement Operators

Operator	Name	Description
++$x	Pre-increment	Increments $x by one, then returns $x
$x++	Post-increment	Returns $x, then increments $x by one
--$x	Pre-decrement	Decrements $x by one, then returns $x
$x--	Post-decrement	Returns $x, then decrements $x by one

OPERATORS (cont. 3)

- Logical Operators

Operator	Name	Example	Result
and	And	$x and $y	True if both $x and $y are true
or	Or	$x or $y	True if either $x or $y is true
xor	Xor	$x xor $y	True if either $x or $y is true, but not both
&&	And	$x && $y	True if both $x and $y are true
||	Or	$x || $y	True if either $x or $y is true
!	Not	!$x	True if $x is not true

- String Operators

Operator	Name	Example	Result
.	Concatenation	$txt1 . $txt2	Concatenation of $txt1 and $txt2
.=	Concatenation assignment	$txt1 .= $txt2	Appends $txt2 to $txt1

OPERATORS (cont. 4)

- Array Operators

Operator	Name	Example	Result
+	Union	$x + $y	Union of $x and $y
==	Equality	$x == $y	Returns true if $x and $y have the same key/value pairs
===	Identity	$x === $y	Returns true if $x and $y have the same key/value pairs in the same order and of the same types
!=	Inequality	$x != $y	Returns true if $x is not equal to $y
<>	Inequality	$x <> $y	Returns true if $x is not equal to $y
!==	Non-identity	$x !== $y	Returns true if $x is not identical to $y

CONDITION STATMENTS

- If
- If … else
- If … else if … else
- Switch case

```php
<?php

$number = 10;

if ($number === 10) {
    echo "$number === 10";
} else {
    echo "$number is someone else";
}

if ($number > 10) {
    echo "$number is greater than 10.";
} else if ($number < 10) {
    echo "$number is less than 10.";
} else {
    echo "$number is equal 10.";
}
```

```php
<?php
define("SPRING", 5);
define("SUMMER", 10);
define("AUTUMN", 15);
define("WINTER", 20);
$yourCoupon = "spring";
$discountPercent = 0;

switch (strtoupper($yourCoupon)) {
    case "SPRING":
        $discountPercent = SPRING;
        break;
    case "SUMMER":
        $discountPercent = SUMMER;
        break;
    case "AUTUMN":
        $discountPercent = AUTUMN;
        break;
    case "WINTER":
        $discountPercent = WINTER;
        break;
    default:
        $discountPercent = 0;
}

echo $discountPercent;
```

LOOPS

- While
- Do ... While

```php
<?php

$number = 0;

do {
    echo "$number";
    $number++;
} while ($number < 5);
```

- For

```php
<?php

$number = 0;

for ($number; $number < 5; $number++) {
    echo "$number";
}
```

- Foreach

```php
<?php

$array = ["ten" => 10, "nine" => 9];

foreach ($array as $key => $value) {
    echo "$key is $value <br>";
}
```

FUNCTIONS

- A function name can start with a letter or underscore (not a number).
- Give the function a name that reflects what the function does!
- PHP has more than 1000 built-in functions.

```php
<?php

define("variable", 1);
$array = array(1, 2, 3, 4, 5);
count($array);
date("Y/m/d H:i:s");
rsort($array);
strotoupper("hello");
```

```php
<?php

function displayYourName($yourName = "Nam NH")
{
    echo "Here you are <br>";
    return $yourName;
}

displayYourName();
echo displayYourName("Smith");
```

SUPERGLOBALS

- Built-in variables that are always available in all scopes.
- Always accessible, regardless of scope, can access from any function or file.

- $GLOBALS
- $_SERVER
- $_REQUEST
- $_POST
- $_GET
- $_FILES
- $_ENV
- $_COOKIE
- $_SESSION

TIME TO PRACTICE.

EXERCISE

1. Write a PHP script to get the PHP version and configuration information.
2. Write a PHP script to display the following strings. E.g : ‘My name is Nam’.
3. $var = 'PHP Tutorial'. Put this variable into the title section, h3 tag and as an anchor text within an HTML document.
4. Do the operator : 10/5 and print the result.
5. Create a variable and use it into an string. E.g : ‘hello ’ + $var + ‘ world’.