PHP 7 Variables

- Variables are "containers" for storing information.

Creating (Declaring) PHP Variables

- In PHP, a variable starts with the $ sign, followed by the name of the variable:

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

- After the execution of the statements above, the variable $txt will hold the value Hello world!, the variable $x will hold the value 5, and the variable $y will hold the value 10.5.

Notes:

- 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.
- 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).

Output Variables

- The PHP echo statement is often used to output data to the screen.
- The following example will show how to output text and a variable:

PHP Code	Result
<!DOCTYPE html> <html> <body> <?php $txt = "PHP"; echo "I love $txt!"; ?> </body> </html>	I love PHP!

- The following example will produce the same output as the example above:

PHP Code	Result
<!DOCTYPE html> <html> <body> <?php $txt = "PHP"; echo "I love " . $txt . "!"; ?> </body> </html>	I love PHP!

The following example will output the sum of two variables:

PHP Code	Result
<!DOCTYPE html> <html> <body> <?php $x = 5; $y = 4; echo $x + $y; ?> </body> </html>	9

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

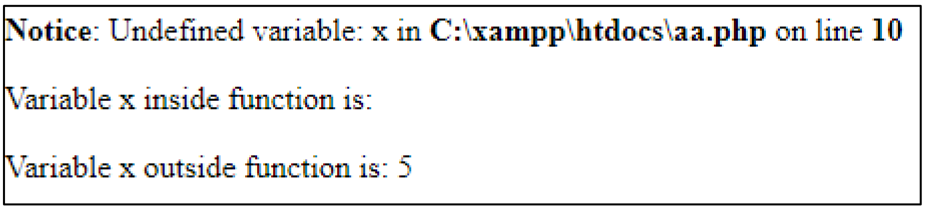
- In PHP, variables can be declared anywhere in the script.
- The scope of a variable is the part of the script where the variable can be referenced/used.
- PHP has three different variable scopes:
    - local
    - global
    - static

Global and Local Scope

- A variable declared outside a function has a GLOBAL SCOPE and can only be accessed outside a function:

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

Result:

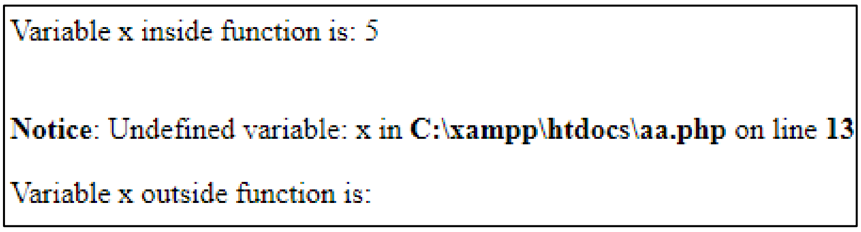


```
Notice: Undefined variable: x in C:\xampp\htdocs\aa.php on line 10

Variable x inside function is:

Variable x outside function is: 5
```

- A variable declared within a function has a LOCAL SCOPE and can only be accessed within that 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>";
?>
```

Result:



```
Variable x inside function is: 5


Notice: Undefined variable: x in C:\xampp\htdocs\aa.php on line 13

Variable x outside function is:
```

PHP The global keyword

- The global keyword is used to access a global variable from within a function.
- To do this, use the global keyword before the variables (inside the function):

```
<!DOCTYPE html>
<html>
<body>
<?php
$x = 5;
$y = 10;
function myTest() {
    global $x, $y;
    $y = $x + $y;
}
myTest();  // run function
echo $y; // output the new value for variable $y
?>
</body>
</html>
```

Result:

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.
- To do this, use the static keyword when you first declare the variable:

PHP Code	Result
`<?php` `function myTest() {` `    static $x = 0;` `    echo $x . " ";` `    $x++;` `}` `myTest();` `myTest();` `myTest();` `?>`	0 1 2

- Then, each time the function is called, that variable will still have the information it contained from the last time the function was called.

Note: The variable is still local to the function.

PHP 7 Constants

- Constants are like variables except that once they are defined they cannot be changed or undefined.
- A constant is an identifier (name) for a simple value. The value cannot be changed during the script.
- A valid constant name starts with a letter or underscore (no $ sign before the constant name).

Note: Unlike variables, constants are automatically global across the entire script.

Create a PHP Constant

- To create a constant, use the define() function.
- **Syntax:**

```
define(name, value, case-insensitive)
```

Parameters:

- name: Specifies the name of the constant
- value: Specifies the value of the constant
- case-insensitive: Specifies whether the constant name should be case-insensitive. Default is false

- The example below creates a constant with a case-sensitive name:

PHP Code	Result

`<!DOCTYPE html>`
`<html>`
`<body>`
`<?php`
`// case-sensitive constant name`
`define("GREETING", "Welcome to PHP!");`
`echo GREETING;`
`?>`
`</body>`
`</html>`
	Welcome to PHP!

– The example below creates a constant with a **case–insensitive** name:

PHP Code	Result

<!DOCTYPE html>
<html>
<body>
<?php
// case-insensitive constant name
define("GREETING", "Welcome to PHP!", true);
echo greeting;
?>
</body>
</html>
	Welcome to PHP!

Constants are Global

– Constants are automatically global and can be used across the entire script.

– The example below uses a constant inside a function, even if it is defined outside the function:

PHP Code	Result

<!DOCTYPE html>
<html>
<body>
<?php
define("GREETING", "Welcome to PHP!");
function myTest() {
    echo GREETING;
}
myTest();
?>
</body>
</html>
	Welcome to PHP!

PHP 7 output (echo and print Statements)

- 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.

1- The PHP echo Statement

- The echo statement can be used with or without parentheses: echo or echo().

Display Text

- The following example shows how to output text with the echo command (notice that the text can contain HTML markup):

PHP Code	<!DOCTYPE html> <html> <body> <?php echo "<h2>PHP is Fun!</h2>"; echo "Hello world! "; echo "I'm about to learn PHP! "; echo "This ", "string ", "was ", "made ", "with multiple parameters."; ?> </body> </html>
Result	**PHP is Fun!** Hello world! I'm about to learn PHP! This string was made with multiple parameters.

Display Variables

– The following example shows how to output text and variables with the echo statement:

PHP Code	Result
<?php $txt1 = "Internet Web Site Design";; $txt2 = "Learn PHP"; $x = 5; $y = 4; echo "<h2>" . $txt1 . "</h2>"; echo "I love to " . $txt2 . " "; echo $x + $y; ?> </body> </html>	**Internet Web Site Design** I love to Learn PHP 9

2– The PHP print Statement

– The print statement can be used with or without parentheses: print or print().

Display Text

The following example shows how to output text with the print command (notice that the text can contain HTML markup):

PHP Code	Result
<!DOCTYPE html> <html> <body> <?php print "<h2>PHP is Fun!</h2>"; print "Hello world! "; print "I'm about to learn PHP!"; ?> </body> </html>	**PHP is Fun!** Hello world! I'm about to learn PHP!

Display Variables

– The following example shows how to output text and variables with the print statement:

PHP Code	Result
<?php $txt1 = "Internet Web Site Design";; $txt2 = "Learn PHP"; $x = 5; $y = 4; print "<h2>" . $txt1 . "</h2>"; print "I love to " . $txt2 . " "; print $x + $y; ?> </body></html>	**Internet Web Site Design** I love to Learn PHP 9