

Data Structures – Variables in PHP

Variables

Variables are memory locations in RAM which are used to hold the data required by a program. The contents of the memory location are not fixed and may be changed during the execution of a program. The 4 main basic types are integer, float, string and Boolean.

In PHP: All variables in PHP are prefixed with a dollar sign. The dollar sign is not technically part of the variable name, but it is required as the first character for the PHP parser to recognize the variable as such. Eg \$name

Declaring Variables

In some programming languages that are very structures eg Java, pascal you **must** declare the type of data the variable will hold BEFORE you use it (e.g. for QBASIC DIM age as INTEGER). This enables the language to set up the location (address) and the amount of memory (size) required for the data, before it is used.

In PHP: PHP does NOT require you to declare a variable and the data type that it will hold. A variable is created "on the fly" the moment you first assign a value to it and PHP will determine the best type of data which a variable holds. For example:

```
$name = "Fred";      //string
$age = 15;           //integer
$mark = 3.5;         //float
$paid = true         //Boolean
```

To see the type of data that the variable is holding you can use the function `gettype`, eg `echo gettype($paid);` //this would display Boolean

Php is different than other languages , you can declare a variable that contain a number and later replace it with a string or boolean.

Variable Scope

The scope of a variable is about where it can (or can not) be accessed from or used in a program.

LOCAL (specific to what's inside a function)

A local variable is one that is specific to a given instance of a given function. It is created when the function processing begins and is deleted as soon as the function is completed.

In PHP: Local scope is specific to functions.

Example:

In the example function below, the variable "\$name" is defined inside the function (it is called a "local" variable) so when the function is run the name will be displayed.

With the second output statement the variable "\$name" will **not** display Pete because the variable is being called outside the function.

```
function getPetersName(){
    $name = "Pete";
    echo $name;
}
```

Code	Output
Run the function using the following echo "Hi there " ; getPetersName();	Hi there Pete
Run the following statement which uses the \$name outside the function echo "Hi there \$name";	Hi there

GLOBAL (defined outside a function)

Global scope refers to any variable that is defined outside of any function. They can be accessed from any part of the program that is not inside a function.

In PHP: To access a global variable from within a function, you can call it into the function with the global keyword.

Eg `global $varToInclude;`

PHP also stores all global variables in an array called `$GLOBALS[]`. Its index is the name of the variable. This array is accessible from within functions and can be used to update global variables directly.

Eg `global $somVar; $someVar = 'abc'`

STATIC

Normally when a function terminates, all of its variables are also cleaned up. Sometimes you want a local variable to persist between instances of a given function.

In PHP: To do this you use the static keyword when you first declare the variable. Then each time you call the function, that variable will still have the information it contained from the last time the function was called.

Eg `static $aValueToRemember;`

Even though the value persists, the variable is still local to the function.

PARAMETERS

A parameter is a local variable whose value is passed to the function by the calling code. Unlike other variables, parameters are declared in a parameter list as part of the function declaration. Parameters are also sometimes called arguments.

E.g. `function myFunc($para1, $para2, [...]) { // function code }`

Environment Variables

Beyond the variables you declare in your code, PHP has a collection of environment variables, which are system defined variables that are accessible from anywhere inside the PHP code, inside of functions or out. We already encountered one in the \$GLOBALS array.

All of these environment variables are stored by PHP as arrays. Some you can address directly by using the name of the index position as a variable name. Other can only be accessed through their arrays. Which can be accessed which way is heavily dependent on how strict the security settings are for PHP on your server. We will discuss the methods of accessing these variables in more detail when we talk about processing user data, but using the arrays is a good coding practice.

Some of the environment variables include:

\$HTTP_SERVER_VARS[]

Contains information about the server and the HTTP connection.

\$HTTP_COOKIE_VARS[]

Contains any cookie data sent back to the server from the client. Indexed by cookie name.

\$HTTP_GET_VARS[]

Contains any information sent to the server as a search string as part of the URL.

Example for using Get

<pre><form name="form" method="GET" action="ProPage.php "> <input type="text" name="Name"> </form></pre>	<p>Once the form is submitted, the URL will change, appended with a list of each variable with the value. http://www.site.com/ProPage.php?Name=Peter</p>
<pre><?php \$fName = \$_GET['Name']; ?></pre>	<p>ProcessingPage.php extracts the variable value from the global array \$_GET</p>

\$HTTP_POST_VARS[]

Contains any information sent to the server as a POST style posting from a client form.

Example for using Post

<pre><form name="form" method="POST" action="ProPage.php"> <input type="text" name="firstName"> </form></pre>	<p>A form passes (POSTS) the variable firstName to the server.</p>
<pre><?php \$fName = \$_POST['firstName']; ?></pre>	<p>ProcessingPage.php extracts the variable value from the global array \$_POST</p>

\$HTTP_POST_FILES[]

Contains information about any uploaded files.

\$HTTP_ENV_VARS[]

Contains information about environmental variables on the server.

PHP Variable Questions:

1. Define the purpose of a variable

2. List 4 simple data types that a variable can hold

3. Identify the symbol that starts every variable in PHP_____

4. Identify the symbols used in an array in PHP _____

5. In PHP what is the purpose of Environmental variables, use an example to illustrate your answer

6. Outline the difference between a POST and GET, provide examples to illustrate their difference

7. Outline the difference between a global and local variable

7. Identify the name you call variables that get passed to a function
