



PHP variables

Variables are “containers” for storing information.

In PHP, a variable starts with the **\$** sign, followed by the name of the variable.

A variable can have a short name (like *x* and *y*) or more descriptive name (*username*, *password*)

Rules for PHP variables

- A variable starts with the **\$** sign, followed by the name of the variable.
- A variable 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.
- Variables names are case sensitive (**\$age** and **\$AGE** are two different variables).

Outputting Variables

The PHP **echo** and **print** statement are often used to output data to the screen.

```
<?php
    $company = "Team Piccolo Global Enterprises";
    echo $company;
?>
```

PHP is a loosely Typed Language

In the example above, when declaring the variable **\$company**; notice that we did not have to tell PHP which data type the variable is.

PHP automatically converts the variables to the correct data type depending on its value.

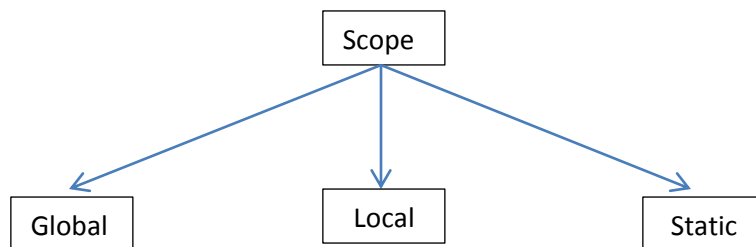
In other languages such as C, C++ and Java, the programmer must declare the name and type of the variable before using it.

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.



Global: A variable declared outside a function has a **GLOBAL SCOPE** and can only be accessed outside a function.

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

Static: Normally, when a function is completed/ executed, all of its variables are declared. 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.