

Variables and Data Types in PHP

Riya Jacob K

Dept of BCA

2020 - 21

Variables

- Variables: There are certain rules for 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)

User Defined Variables.

- PHP variables names must start with an underscore or an alphabetical character. Underscores start most predefined PHP variables, so starting user-defined variables with an alphabetical character is a good idea.
 - PHP variables were once always assigned by value or by reference.
 - You define a variable in PHP by prefacing the variable name with a \$ symbol.
 - Variables may be scalar or compound.
 - The difference between scalar and compound variables is scalar variables can hold one thing at a time, where as compound variables may hold more than one thing.
-

-
- When you define and assign a value to a variable in one step, this is called **declaring** a variable.
 - We can assign variables by reference to other variables. Assignment by value is done using a binary assignment operator, which is the equal (=) symbol between two operands. The left operand is the target variable, and the right operand is a source value or reference. Assignment statements are typically terminated by a semicolon to complete the expression.
 - We can assign by a reference using a combination of an assignment operator and ampersand (&) preceding the source variable name like **&\$variableName**.
-

PHP Datatypes

- Variables can store data of different types, and different data types can do different things.
- PHP supports the following data types:

String

Integer

Float (floating point numbers - also called double)

Boolean

PHP String

- A string is a sequence of characters, like "Hello world!".
- A string can be any text inside quotes. You can use single or double quotes:

- ```
<?php
$x = "Hello world!";
$y = 'Hello world!';

echo $x;
echo "
";
echo $y;
?>
```

# PHP Integer

---

- An integer is a whole number (without decimals). It is a number between -2,147,483,648 and +2,147,483,647.
- Rules for integers:
- An integer must have at least one digit (0-9)
- An integer cannot contain comma or blanks
- An integer must not have a decimal point
- An integer can be either positive or negative
- Integers can be specified in three formats: decimal (10-based), hexadecimal (16-based - prefixed with 0x) or octal (8-based - prefixed with 0)

- ```
<?php
$x = 5985;
var_dump($x);
?>
```

PHP Float

- A float (floating point number) is a number with a decimal point or a number in exponential form.
- In the following example \$x is a float. The PHP var_dump() function returns the data type and value

- ```
<?php
$x = 10.365;
var_dump($x);
?>
```



## PHP Boolean

---

- A Boolean represents two possible states: TRUE or FALSE.
- Booleans are often used in conditional testing. You will learn more about conditional testing in a later chapter of this tutorial.
- `$x = true;`  
`$y = false;`

# PHP Global Variables.

---

- Global variables are of two types. One is Environmental level and the other is Script level variables.
- Environmental level variables are available any where in your PHP progra,.
- Script level variables are available only externally to functions unless you pass them to a function as an actual parameter.

# PHP Predefined Variables.

---

- PHP predefined variables are also known as super global variables which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

- The PHP superglobal variables are:

`$GLOBALS`

`$_SERVER`

`$_REQUEST`

`$_POST`

`$_GET`

`$_FILES`

`$_ENV`

`$_COOKIE`

`$_SESSION`

# Predefined Variables.

---

| Variable Name          | Description                                                                                                                                                       |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\$GLOBALS</code> | The variable contains a reference to every variable with in the global scope of the script. The keys for these values are the names of the global variables.      |
| <code>\$_COOKIE</code> | The variable contains all HTTP cookies. It replaces the deprecated <code>\$\$HTTP_COOKIE_VARS</code> array.                                                       |
| <code>\$_ENV</code>    | The variable contains all inherited environment variables or those directly set within the script. It replaces the deprecated <code>\$HTTP_ENV_VARS</code> array. |
| <code>\$_GET</code>    | The variable contains all the URL query string values. It replaces the <code>\$HTTP_GET_VARS</code> array.                                                        |

# Predefined Variables.

| Variable Name           | Description                                                                                                                                                                       |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\$_SERVER</code>  | The variable contains variables set by the execution environment of the web server and current running scripts. It replaces the deprecated <code>\$HTTP_SERVER_VARS</code> array. |
| <code>\$_SESSION</code> | The variable contains variables bound to the current session. It replaces the deprecated <code>\$HTTP_SESSION_VARS</code> array.                                                  |
| <code>\$_FILES</code>   | The variable contains all variables provided by HTTP POST file uploads. It replaces the deprecated <code>\$HTTP_POST_FILES</code> array.                                          |
| <code>\$_POST</code>    | The variable contains all variables provided by HTTP POST it replaces the deprecated <code>\$HTTP_POST_VARS</code> array.                                                         |

# Predefined Variables.

---

- **\$\_REQUEST:** The variable contains all variables provided by GET, POST, and COOKIE inputs. The order of the variable is set by the PHP variable order configuration parameter. The values in this variable are a security risk because it makes a man-in-the-middle attack more likely, since \$\_GET is insecure. You should use the \$\_POST in lieu of the \$\_REQUEST predefined variable.