PHP Programming

COMP203TH

Lecture: 5

Data types in PHP

PHP data types are used to hold different types of data or values. Data types specify the size and type of values that can be stored.

PHP is a loosely typed language so there is no need to define data type.

PHP supports 8 primitive data types that can be categorized further in three types:

1. Scalar Types (predefined)
2. Compound Types (user-defined)
3. Special Types

1. **Scalar Types:**
    a. **Integer:** Integer hold only whole numbers including positive and negative numbers i.e. numbers without fractional part or decimal point.
        - Integers can be decimal (base 10), octal(base 8) or hexadecimal (base 16).
        - The octal integers can be declared with leading 0 and the hexadecimal can be declared with leading 0x.
        - The range of an integer must lie between -2^31 to 2^31.

    b. **Double/Float:** A floating point number is a number with a decimal point. Unlike integer, it can hold numbers with a fractional or decimal point, including a negative or positive sign.

    c. **String:** Strings are sequences of characters. A string can hold letters, numbers and special characters. String values must be enclosed either within single quotes or in double quotes.

    d. **Boolean:** A Boolean represents two possible states: True or False. Booleans are often used in conditional testing.

2. **Compound Types:**
    a. **Array:** Array is a compound data type which can store multiple values of same date type. An array is formally defined as an indexed collection of data values. Each index (also known as the key) of an array is unique and references a corresponding value.

    b. **Objects:** Objects are defined as instances of user defined classes that can hold both values and functions.
        - A class is a template for objects and an object is an instance of a class. When the individual objects are created, they inherit all the properties and behaviors from the class, but each object will have different values for the properties.

- E.g. we have a class named Car. A Car can have properties like model, color etc. We can define variables like $model, $color and so on to hold the values of these properties. When the individual objects (maruti, tata, hond, etc.) are created, they inherit all the properties and behaviors from class, but each object will have different values for the properties.

3. **Special Types:**
   a. **Resource:** Resources in PHP are used to store references to some function call or to external PHP resources e.g. a database call.
   b. **Null:** Null is a special data type that has only one value: NULL. There is a convention of writing it in capital letters as it is case sensitive. A variable of data type NULL is a variable that has no value assigned to it.
      - If a variable is created without a value, it is automatically assigned a value of NULL. Variables can also e emptied by setting the value to NULL.

Unlike other programming languages, where a variable's data type must be explicitly defined by the programmer, PHP automatically determines a variable's data type from the content it holds. And if he variable's content changes over the duration of a script, the language will automatically set the variable to the appropriate data type.

- **PHP functions to test variable data types:**

gettype()	returns the type of a variable. Syntax: gettype(variable);
var_dump()	dumps information about one or more variables. The information holds type and value of the variable(s). Syntax: var_dump(var1,var2,...);
is_bool()	Tests if a variable holds a Boolean value
is_numeric()	Tests if a variable holds a numeric value
is_int()	Tests if a variable holds an integer
is_float()	tests if a variable holds a floating value.
is_string()	tests if a variable holds a string value.
is_null()	tests if a variable holds a null value.
is_array()	tests if a variable is an array.
is_object()	tests if a variable is an object.

Example:

```
<?php
// define string variable
$whoami = 'Hello';
// output: 'string'
echo gettype($whoami);
echo "</br>";
// assign new integer value to variable
$whoami = 99.8;
// output: 'double'
echo gettype($whoami);
?>
```

- **Destroying variables:**
  - To destroy a variable, pass the variable to PHP's unset() function.
  - Alternatively, you may also empty a variable of its contents by assigning PHP's special NULL value to it.

- **Using Constants:**

  Constants are used when you need to store a fixed value i.e. one that remains static over the course of a script. Constants are PHP containers for values that remain constant and never change. They are mostly used for data that is known well in advance and that is used unchanged in multiple places in an application.

  Constants **are defined using PHP's define () function** which accepts two arguments: the name of the constant and its value.

  Constant names must follow the same rules as variable names, with one exception that the $ prefix is not required for constant names.

  By conversion, **constant names are usually entirely uppercased**. This is to enable their easy identification and differentiation from regular variables in a script.

Example:

```
<?php

// define constants

define ('PROGRAM', 'The Matrix');

define ('VERSION', 11.7);

// use constants

// output: 'Welcome to The Matrix (version 11.7)'

echo 'Welcome to ' . PROGRAM . ' (version ' . VERSION . ')';

?>
```