# PHP - Data Types

The term "data types" refers to the classification of data in distinct categories. In PHP, the following built-in data types are defined −

- integer
- double
- bool
- string
- null
- array
- object
- resource

In this chapter, let's discuss in detail about these built-in data types of PHP.

## Integer

A whole number, without a decimal point (like 4195) is of int type in PHP.

- An int is a number of the set Z = {..., -2, -1, 0, 1, 2, ...}.
- An int can be represented in a decimal (base 10), hexadecimal (base 16), octal (base 8) or binary (base 2) notation.

To use octal notation, a number is preceded with "0o" or "0O". To use hexadecimal notation, precede the number with "0x". To use binary notation, precede the number with "0b". Given below are some examples −

- Decimal integer: 201, 4195, -15
- Octal integer: 0010, 0O12, -0O21
- Hexadecimal integer: 0x10, -0x100
- Binary integer: 0b10101, -0b100

## double

Floating point numbers (also known as "floats", "doubles", or "real numbers") are the numbers with a fractional component. The fractional component follows after the integer component separated by the decimal symbol (.)

```
Standard float notation - 1.55, 21.2468, -123.0
```

PHP also allows the use of scientific notation to represent a floating point number with more digits after the decimal point. The symbol "E" or "e" is used to separate the integer and fractional part.

```
Scientific float notation - 1.2e3, 2.33e-4, 7E-10, 1.0E5
```

## bool

The bool type only has only two values; it can either be True or False. The bool type is used to express a truth value. Typically, the result of an operator which returns a bool value is passed on to a control structure such as "if", "while" or "do-while".

## String

A string is a sequence of characters, for example, 'PHP supports string operations.'

- In PHP, a character is the same as a byte. This means that PHP only supports a 256-character set, and hence does not offer native Unicode support.
- PHP supports single quoted as well as double quoted string formation. Both the representations 'this is a simple string' as well as "this is a simple string" are valid.
- PHP also has Heredoc and Newdoc representations of string data type.

## null

In PHP, null represents a special type that only has one value: NULL. Undefined and unset() variables will resolve to the value "null".

## array

An array in PHP is an ordered map, a key is associated with one or more values. A PHP array is defined using the array() function, or with the use of a shorter notation where the data is pit in square brackets. These are the examples of associative arrays –

## array() function

```
$arr = array(
   "foo" => "bar",
   "bar" => "foo",
);
```

## Short notation

```
$arr = [
   "foo" => "bar",
   "bar" => "foo",
];
```

An array in PHP can also be defined with the "key-value pair" syntax. It is called an **indexed array**.

```
$arr = array("foo", "bar", "hello", "world");
```

In a multi-dimensional array, each element in the main array can also be an array. And, each element in the sub-array can be an array, and so on. Values in the multi-dimensional array are accessed using multiple index.

## object

An object type is an instance of a programmer-defined class, which can package up both other kinds of values and functions that are specific to the class.

To create a new object, use the **new** statement to instantiate a class –

```
class foo {
   function bar() {
      echo "Hello World.";
   }
}
```

```
$obj = new foo;
$obj->bar();
```

### resource

Resources are special variables that hold references to resources external to PHP (such as a file stream or database connections).

Here is an example of file resource −

```
$fp = fopen("foo.txt", "w");
```

Data belonging to any of the above types is stored in a variable. However, since PHP is a dynamically typed language, there is no need to specify the type of a variable, as this will be determined at runtime.

## Example: The gettype() Function

The gettype() function is helpful to find out the type of data stored in a variable.

```
<?php
   $x = 10;
   echo gettype($x) . "\n";

   $y = 10.55;
   echo gettype($y) . "\n";

   $z = [1,2,3,4,5];
   echo gettype($z) . "\n";
?>
```

When you run this code, it will produce the following **output** −

```
integer
double
array
```