

DATA TYPES

In PHP, data types represent the type of data that a variable or an expression can hold. PHP is a loosely typed language, meaning that you don't need to explicitly declare the data type of a variable when you define it. PHP dynamically determines the data type based on the assigned value. However, understanding the different data types is crucial for proper variable usage and manipulation. Here are some common data types in PHP:

1. Integer:

An integer is a whole number without a decimal point.

```
$number = 42;
```

2. Float (Floating-point number or Double):

A float represents numbers with decimal points or in exponential form.

```
$floatNumber = 3.14;
```

3. String:

A string is a sequence of characters, enclosed in single or double quotes.

```
$name = "John";
```

```
$greeting = 'Hello, ' . $name . '!';
```

4. Boolean:

A boolean represents binary values, either true or false.

```
$isTrue = true;
```

```
$isFalse = false;
```

5. Array:

An array is an ordered map that stores multiple values.

```
$colors = array("red", "green", "blue");
```

6. Object:

An object represents instances of user-defined classes.

```
class Car {
```

```
    // Class definition
```

```
}
```

```
$myCar = new Car();
```

7. Resource:

A resource represents external resources (e.g., database connections, file handles) and is typically created and manipulated by special functions.

```
$fileHandle = fopen("example.txt", "r");
```

8. NULL:

NULL represents the absence of a value or a variable that has been explicitly set to null.

```
$nullVar = null;
```

9. Scalar Types (int, float, string, bool):

Scalar types are single-value data types that cannot be further subdivided.

```
$intValue = 42;    // Integer
```

```
$floatValue = 3.14; // Float
```

```
$stringValue = "Hello"; // String
```

```
$boolValue = true;  // Boolean
```

10. Compound Types (array, object):

Compound types are data types that can hold multiple values or subvalues.

Array:

```
$fruits = array("apple", "banana", "orange");
```

Object:

```
class Person {  
    public $name;  
    public $age;  
}
```

```
$person = new Person();
```

```
$person->name = "Alice";
```

```
$person->age = 30;
```

11. Mixed:

Mixed represents a value of any type. It is often used when a function or a method can accept different types of data.

```
function displayValue($value) {
```

```
    echo $value;
```

```
}
```

```
displayValue("Hello"); // String
```

```
displayValue(42);      // Integer
```

```
displayValue(true);    // Boolean
```

12. Iterable:

Iterable represents an iterable value, like arrays or objects implementing the Traversable interface. Introduced in PHP 7.1.

```
$iterableArray = ["apple", "banana", "orange"];
```

```
$iterableObject = new ArrayObject($iterableArray);
```

```
foreach ($iterableObject as $item) {
```

```
    echo $item . "\n";
```

```
}
```

Understanding these data types and their usage is fundamental to writing effective and flexible PHP code. It enables you to handle various kinds of data and perform operations accordingly in your scripts.

PHP Object

Classes and objects are the two main aspects of object-oriented programming.

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.

Let's assume we have a class named **Car** that 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 (Volvo, BMW, Toyota, etc.) are created, they inherit all the properties and behaviors from the class, but each object will have different values for the properties.

If you create a **__construct()** function, PHP will automatically call this function when you create an object from a class.

Example

```
class Car {  
    public $color;  
    public $model;  
    public function __construct($color, $model) {  
        $this->color = $color;  
    }  
}
```

```
$this->model = $model;
}

public function message() {
    return "My car is a " . $this->color . " " . $this->model . "!";
}
}

$myCar = new Car("red", "Volvo");
var_dump($myCar);
```

PHP NULL Value

Null is a special data type which can have only one value: NULL.

A variable of data type NULL is a variable that has no value assigned to it.

Tip: If a variable is created without a value, it is automatically assigned a value of NULL.

Variables can also be emptied by setting the value to NULL:

Example

```
$x = "Hello world!";
$x = null;
var_dump($x);
```

Change Data Type

If you assign an integer value to a variable, the type will automatically be an integer.

If you assign a string to the same variable, the type will change to a string:

Example

```
$x = 5;  
var_dump($x);  
  
$x = "Hello";  
var_dump($x);
```