

**PROGRAMA DE ESTUDIOS**

# **DESARROLLO DE SISTEMAS DE INFORMACIÓN**

**DESARROLLO DE  
BACKEND**

**Tema**

**MANEJO DE LOS DIFERENTES TIPOS  
DE DATOS, CONSTANTES Y  
VARIABLES EN EL LENGUAJE PHP.**

## MANEJO DE LOS DIFERENTES TIPOS DE DATOS, CONSTANTES Y VARIABLES EN PHP

En PHP, el manejo de datos se realiza a través de variables y constantes, cada una con sus propias características y reglas. Las variables almacenan valores que pueden cambiar durante la ejecución del script, mientras que las constantes mantienen valores fijos. PHP soporta diferentes tipos de datos, como enteros, flotantes, cadenas, booleanos, arrays y objetos, y maneja la conversión de tipos de forma dinámica.

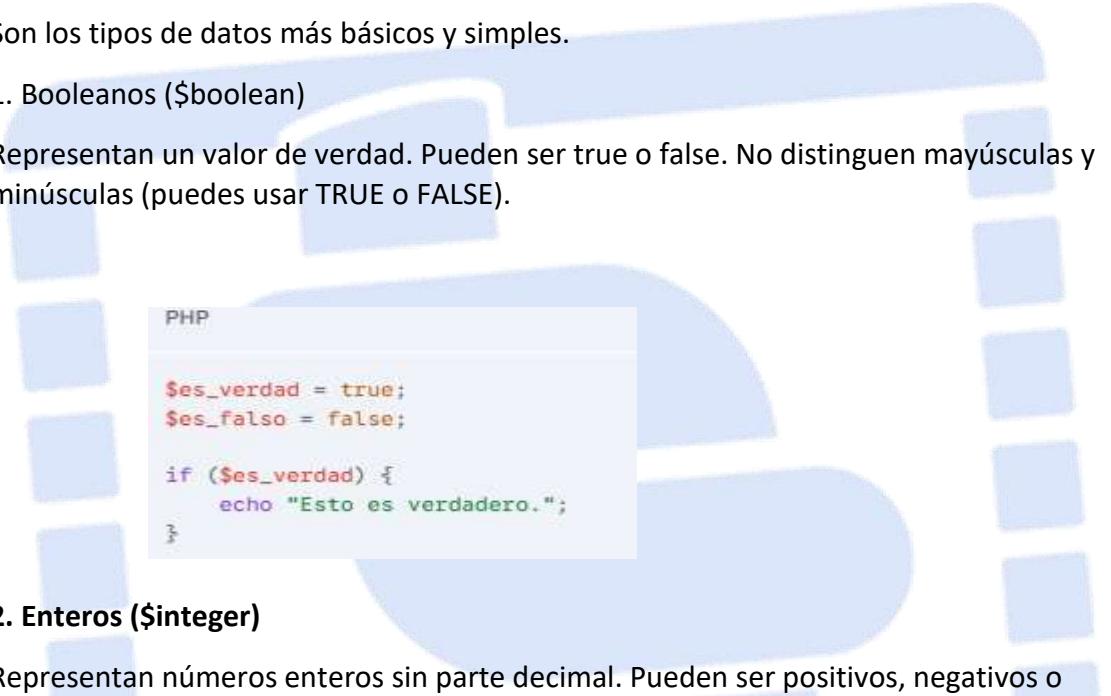
Aquí se presentan los tipos de datos en PHP, divididos en tipos escalares, compuestos y especiales:

### Tipos Escalares

Son los tipos de datos más básicos y simples.

#### 1. Booleanos (\$boolean)

Representan un valor de verdad. Pueden ser true o false. No distinguen mayúsculas y minúsculas (puedes usar TRUE o FALSE).

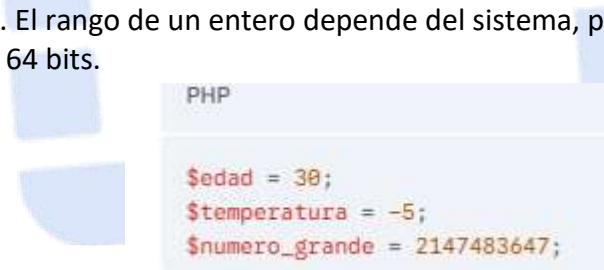


```
PHP
$es_verdad = true;
$es_falso = false;

if ($es_verdad) {
    echo "Esto es verdadero.";
}
```

#### 2. Enteros (\$integer)

Representan números enteros sin parte decimal. Pueden ser positivos, negativos o cero. El rango de un entero depende del sistema, pero generalmente es un número de 32 o 64 bits.



```
PHP
$edad = 30;
$temperatura = -5;
$numero_grande = 2147483647;
```

#### 3. Números de Punto Flotante (\$float / \$double)

Representan números con parte decimal. También se conocen como números de doble precisión o double.

```
$nombre = "Juan";
$saludo_dobles = "Hola, mi nombre es $nombre."; // Salida: "Hola, mi nombre es Juan.
$saludo_simples = 'Hola, mi nombre es $nombre.'; // Salida: "Hola, mi nombre es $nom
```

### Tipos Compuestos

Son tipos de datos que pueden contener múltiples valore

#### 1. Arrays (\$array)

Son estructuras de datos que almacenan múltiples valores en una sola variable. Los arrays pueden ser indexados (con índices numéricos) o asociativos (con claves de cadena).

```
// Array indexado
$frutas = array("manzana", "banana", "cereza");
echo $frutas[0]; // Salida: "manzana"

// Array asociativo
$persona = array(
    "nombre" => "Ana",
    "edad" => 25,
    "ciudad" => "Madrid"
);
echo $persona["nombre"]; // Salida: "Ana"
```

#### 2. Objetos (\$object)

Son instancias de clases. Una clase es un molde que define propiedades (variables) y métodos (funciones) para un objeto.

```
class Coche {
    public $marca;
    public $modelo;

    public function arrancar() {
        return "El coche está arrancando.";
    }
}

$mi_coche = new Coche();
$mi_coche->marca = "Toyota";
$mi_coche->modelo = "Corolla";

echo $mi_coche->arrancar(); // Salida: "El coche está arrancando."
```

### Tipos Especiales

Son tipos de datos con un propósito particular.

## 1. Nulo (\$null)

Una variable con el valor null no tiene ningún valor asignado. Es el único valor posible de este tipo.

```
$nombre = "Pepe";
var_dump($nombre); // Muestra string(4) "Pepe"

$nombre = null;
var_dump($nombre); // Muestra NULL
```

## 2. Recurso (\$resource)

Es una variable especial que contiene una referencia a un recurso externo, como un archivo abierto o una conexión a una base de datos. Se manejan con funciones específicas de PHP.

```
// Ejemplo de recurso (conexión a una base de datos)
$conexion = mysqli_connect("localhost", "usuario", "contraseña", "basededatos");
var_dump($conexion); // Muestra resource(3) of type (mysql link)
```

### Funciones Útiles para el Manejo de Tipos de Datos:

- `gettype($variable)` : Retorna el tipo de dato de una variable como una cadena de texto.
- `settype($variable, $tipo)` : Convierte una variable a un tipo de dato específico.
- `is_int()`, `is_string()`, `is_array()`, etc.: Funciones que comprueban si una variable es de un tipo de dato específico.
- `var_dump($variable)` : Muestra información estructurada sobre una o más expresiones, incluyendo su tipo y valor. Es muy útil para la depuración.

### Conversión de Tipos (Type Casting)

PHP permite la conversión de tipos de forma explícita. Puedes forzar que una variable sea de un tipo de dato diferente.

```
$numero_texto = "100";
$numero_entero = (int) $numero_texto; // $numero_entero ahora es un entero
var_dump($numero_entero); // Muestra int(100)

$booleano = (bool) "cadena"; // $booleano será true
var_dump($booleano); // Muestra bool(true)
```

### Tabla de conversión de tipos:

| Tipo a convertir | Ejemplo | Resultado |

```
|---|---|---|
| (int) | (int) "100" | 100 |
| (bool) | (bool) "" | false |
| (float) | (float) "1.23" | 1.23 |
| (string) | (string) 123 | "123" |
| (array) | (array) "abc" | array(0 => "abc") |
| (object) | (object) "abc" | object(stdClass)#1 (1) { ["scalar"]=> string(3) "abc" }
| |
```

Comprender y manejar estos tipos de datos es fundamental para escribir código PHP robusto y eficiente. El tipado dinámico de PHP ofrece flexibilidad, pero también requiere que el desarrollador sea consciente del tipo de información que está manejando en cada momento.

#### ¿CUÁLES SON LAS DIFERENCIAS ENTRE LAS CONSTANTES Y VARIABLES DE PHP?

**Constantes PHP:** Las constantes PHP son identificadores que permanecen invariables. Normalmente, no cambian durante la ejecución del script. Distinguen entre mayúsculas y minúsculas. Por defecto, los identificadores de constantes siempre se escriben en mayúsculas. Normalmente, el nombre de una constante comienza con un guion bajo o una letra, seguido de varias letras y números. No es necesario escribir una constante con el símbolo \$. La función `constante()` se utiliza para devolver el valor de una constante.

Ejemplo:

```
<?php
    define("Hello", "Welcome to geeksforgeeks.com!");
    echo Hello;
?>
```

Producción :

¡ Bienvenido a geeksforgeeks.com !

**Variables PHP:** Una variable PHP es el nombre que se le da a una dirección de memoria que contiene datos. El método básico para declarar una variable PHP es usar el símbolo \$ seguido del nombre de la variable. Una variable ayuda al código PHP a almacenar información durante el programa. Si se usa una variable antes de asignarla, ya tiene un valor predeterminado almacenado. Algunos tipos de datos que podemos usar para construir una variable son enteros, dobles y booleanos.

Ejemplo:

```
<?php
$txt = "¡Hola desde geeksforgeeks!";
$x = 5;
$y = 10.5;
echo $txt;
echo "<br>";
echo $x;
echo "<br>";
echo $y;
?>
```

Producción:

```
¡Hola desde geeksforgeeks!
5
10.5
```

### DIFERENCIA ENTRE CONSTANTES Y VARIABLES PHP

Constantes PHP	Variables PHP
En las constantes PHP no es necesario utilizar el signo \$.	En las variables PHP se utiliza el signo \$.
El tipo de datos de la constante PHP no se puede cambiar durante la ejecución del script.	El tipo de datos de la variable PHP se puede cambiar durante la ejecución del script.
Una constante PHP, una vez definida, no puede redefinirse.	Una variable PHP puede estar indefinida o redefinida.
No podemos definir una constante utilizando cualquier operador de asignación simple más bien, sólo se puede definir utilizando define().	Podemos definir una variable utilizando una operación de asignación simple (=).
Generalmente las constantes se escriben en números.	Por otro lado, las variables se escriben en letras y símbolos.
Las constantes PHP son automáticamente globales	Las variables PHP no son automáticamente

