

Introducción a la programación PHP

1 - Introducción a Variables en PHP
Prof. Luis Eduardo Fagúndez

Qué son las variables?

- Una variable en PHP sirve para guardar datos en memoria mientras se ejecuta el programa, de forma que podamos usarlos, modificarlos y combinarlos más adelante en el código.
- Es el mecanismo básico para trabajar con información dinámica.

El concepto de la caja

- Una variable es como una caja con nombre, donde guardaremos un dato para usarlo después.
- Entonces, podemos tener una caja llamada: “nombre”, donde dentro de ella vamos a almacenar un dato.
- En los lenguajes de programación existe un concepto denominado sintaxis, que se refiere al conjunto de reglas y estructuras que determinan cómo debe escribirse el código.
- Estas reglas permiten que una herramienta especializada —como un **compilador** o un **intérprete**— pueda interpretar correctamente las instrucciones y transformarlas en acciones ejecutables por la computadora.

Sintaxis básica en todos nuestros archivos PHP

- Antes de comenzar a trabajar con PHP, es importante saber que todo el código debe escribirse en archivos cuya extensión sea **.php**, por ejemplo: **index.php**.
- Asimismo, el código PHP debe colocarse dentro de un bloque específico delimitado por etiquetas especiales: se inicia con **<?php** y se cierra con **?>**.
- Estas etiquetas indican al servidor qué partes del archivo deben interpretarse como código PHP.

Archivo index.php



```
<?php
```

```
//Todo el código de PHP va entre estas dos etiquetas
```

```
?>
```

Las dobles barras // se utilizan para escribir comentarios de una sola línea en el código. Los comentarios permiten incluir anotaciones o aclaraciones destinadas a los desarrolladores, y no son interpretados ni ejecutados por el sistema. Su función principal es documentar el propósito o funcionamiento de partes específicas del código para facilitar su comprensión y mantenimiento.

Sintaxis básica para crear una variable

- Para declarar una variable en PHP se utiliza una sintaxis básica como la siguiente:
 - `$variable = "Texto";`
 - `$nombre = "Luis Eduardo";`
 - `$edad = 33;`
- Analizando esta estructura:
 - El símbolo **\$** actúa como un identificador que le indica a **PHP** que el nombre que aparece a continuación corresponde a una **variable**. Es decir, todo identificador precedido por **\$** será interpretado por el lenguaje como un contenedor de datos cuyo valor puede utilizarse y modificarse durante la ejecución del programa.

El código en cuestión:



```
<?php
```

```
//Variables en PHP
```

```
$variable = "Texto";
```

```
$nombre = "Luis Eduardo";
```

```
$edad = 33;
```

```
?>
```

Cómo funcionan las variables

- El nombre de una variable siempre debe de comenzar por una letra (de la "a" a la "z" o "A" a "Z").
- El tipo de dato se determina en tiempo de ejecución.
 - Dependiendo del valor que se le asigne es el tipo de dato.
- En PHP una variable puede cambiar de tipo de dato según su contenido
- Pero: qué es un tipo de dato?

Tipos de variables

- **Enteras (Integer o int):**
 - Representan números sin parte decimal.
 - `$valorEntero = 15;`
- **Cadenas de texto (hasta 256 caracteres)**
 - Representan texto. Pueden definirse utilizando comillas dobles o comillas simples.
 - `$cadenaDeTexto = "Luis Eduardo";`
 - `$cadenaDeTextoDos = 'Soy un texto';`
- **Flotantes (float)**
 - Representan valores numéricos con parte decimal.
 - `$numeroReal = 3.141512;`

Y cómo mostramos el texto en pantalla?

- En PHP, la instrucción `echo` se utiliza para enviar información a la salida estándar, normalmente el navegador web.
- Su función es mostrar texto, valores de variables o resultados de expresiones en pantalla.
 - `echo "Texto a mostrar";`
 - `echo "Hola mundo";`
 - `$nombre = "Luis";`
 - `echo "El nombre guardado en la variable es: $nombre";`

Ejemplo



```
<?php

//Declramos una variable:
$nombre = "Luis";
//Imprimir datos en pantalla:
echo "Texto a mostrar";
echo "Hola mundo";
echo "El nombre guardado en la variable es: $nombre";

?>
```

Comillas simples en PHP

- Cuando se utilizan comillas simples en PHP, las variables y las secuencias especiales de escape no son interpretadas por el lenguaje.
- En estos casos, el contenido se muestra exactamente como fue escrito.
 - Ejemplo
 - `echo 'hola, soy $nombre';`
 - Resultado: hola, soy \$nombre
 - `echo 'Hola, esto es un carácter especial: \n';`
 - Resultado: Hola, esto es un carácter especial: \n

Comillas simples en PHP

- Esto ocurre porque las comillas simples tratan el contenido como una cadena literal, sin procesar variables ni secuencias de escape.
- En cambio, las comillas dobles sí permiten esa interpretación.

Comillas dobles en PHP

- Las comillas dobles en PHP se utiliza para definir cadenas de texto que permiten la interpretación de variables y secuencias especiales de escape dentro del contenido.
- Es decir, cuando usás comillas dobles, PHP “lee” lo que hay dentro y reemplaza automáticamente ciertos elementos por su valor o efecto correspondiente.
- Si dentro del texto hay una variable, PHP la reemplaza por su valor.

Comillas dobles en PHP

- En las cadenas definidas con comillas dobles, PHP interpreta ciertos caracteres especiales denominados secuencias de escape. Estas permiten representar acciones o símbolos especiales dentro del texto.
 - `\n` = Salto de línea
 - `\r` = Retorno de carro
 - `\t` = Tabulador
 - `\v` = Tabulador vertical
 - `\e` = Escape (ESC)
 - `\\` = Barra invertida
 - `\$` = Signo de pesos
 - `\"` = Comillas dobles

Tipos de variables

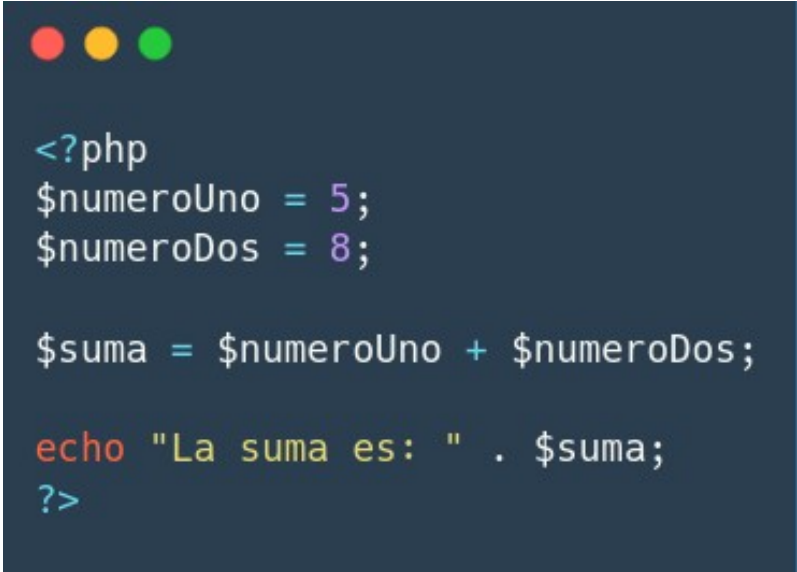
- **Variables booleanas:**
 - Guarda el valor verdadero o falso
 - `$existe = false;`
 - `$existeDos = true;`
- **En PHP es indiferente de mayúsculas a minúsculas.**
- **También es indiferente False, True, TruE, FaLse.**

Otros tipos de datos que veremos a futuro

- Arrays
- Object
- Callable
- Iterable
- Resource
- NULL

Bueno, qué puedo hacer con las variables?

- Con las variables aparte de guardar información podemos realizar operaciones arimetricas.
- Vamos a ver un ejemplo:



```
<?php
$numeroUno = 5;
$numeroDos = 8;

$suma = $numeroUno + $numeroDos;

echo "La suma es: " . $suma;
?>
```

Operadores aritméticos

+	suma
-	resta
*	multiplicación
/	división
%	Resto de división entera
**	potencia
=	asignación
\$a++	post-incremento
++\$a	pre-incremento
\$a+=n	Aumenta en N el valor de \$a

Constantes

- Una constante es un identificador que almacena un valor que no puede modificarse durante la ejecución del programa. A diferencia de las variables, una vez definida, su información permanece inalterable.
- Por convención, los nombres de las constantes se escriben en mayúsculas, para diferenciarlas visualmente de las variables.
- Ejemplos:
 - `define("TEXTO","soy un texto");`
 - `define("NUMERO", 99999);`
 - `define("RUTA","/home/usuario/Desktop")`

Características de las constantes:

- **Características importantes**
 - No llevan el símbolo \$
 - No cambian su valor
 - Se pueden usar en cualquier parte del script después de ser definidas
 - Se recomienda escribir sus nombres en mayúsculas por convención

Operadores aritméticos

$\$a -= n$	Disminuye en n el valor de $\$a$
$\$a *= n$	Multiplica en n el valor de $\$a$
$\$a /= n$	Divide en n el valor de $\$a$

Operadores de comparación

<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
==	Igual que
!=	Diferente que

Operadores lógicos

&&	Y lógico
	Or lógico
!	Negación
and	Y lógico
or	Or lógico