

UD2. Introducción al lenguaje PHP

Desarrollo Web en Entorno Servidor

Sistema de errores (I)
Comentarios
Variables
Tipos de datos
Constantes

Errores en PHP – sistema básico

Cuando hay un error, PHP genera un mensaje y se envía al cliente.

El error tiene: el número de línea y descripción.

Si estoy en mi casa o en preproducción esto está bien.

En caso contrario, lo suyo es deshabilitar los errores.

- La directiva `error_reporting` del fichero de configuración de PHP `php.ini`

Errores en PHP – sistema básico (tipos)

- **Errores sintácticos** (Parse Errors). Cuando no se encuentran los símbolos (tokens) esperados o encuentra otros en su lugar.
- **Advertencias** (Warnings). Indican que se ha producido un error de algún tipo que no impide la ejecución del resto del programa. Seguramente el script no funcionará como debiera.
- **Avisos** (Notices). Impiden la ejecución, Seguramente el script no funcionará como debiera.
- **Obsolescencias** (Strict/Deprecated). Se usa una función o característica que está en proceso de ser eliminada del lenguaje.

Comentarios

Existen tres tipos de comentarios: una línea (// y #) y múltiples líneas (/* */)

Recuerda: ayudan a depurar y a comprender el código.

```
<?php
    echo 'Esto es una prueba'; // Comentario de una sola línea
    /* Esto es un comentario multilínea
       y otra línea de comentarios */
    echo 'Esto es otra prueba';
    echo 'Una prueba final'; # Otro comentario
?>
```

Variables: declaración

tipo_dato.php

- Se usa el operador `$` para denotar una variable.
- En PHP no es necesario declarar las variables, se crean la primera vez que se les asigna un valor.
- **Débilmente tipado:** el programador no necesita indicar el tipo de datos. Se deduce automáticamente.
- Tipos de datos principales:
 - Escalares: int, float, boolean, string
 - Null
 - Arrays
 - Clases

Variables: asignación por valor y por referencia

copia.php

La asignación de una variable se hace **por valor**, es decir: `$a = $b;` crea una copia del valor de `$a` en `$b`.

Sin embargo, es posible hacer una **referencia** a una variable: `$a = &$b;` . En este caso `$a` es un alias de `$b` .

Variables: no inicializadas

no_init.php

Cuando una variable se define pero no se asigna, se pone al tipo de dato `null`.

Si se utiliza, se genera un aviso (E_NOTICE) pero **la ejecución continúa**.

PHP asigna entonces **un valor por defecto a la variable que no existe** en base al resto de variables.

Variables: no inicializadas

no_init.php

Cuando una variable se define pero no se asigna, se pone al tipo de dato `null`.

Si se utiliza, se genera un aviso (E_NOTICE) pero **la ejecución continúa**.

PHP asigna entonces **un valor por defecto a la variable que no existe** en base al resto de variables.

Variables: funciones útiles

Funciones útiles:

- `var_dump($var)`: permite volcar la información al completo de una variable.
- `isset($var)`: indica si la variable está definida y no es `null`.
- `unset($var)`: destruye una variable.
- `echo $var`: imprime el contenido.
- `gettype($var)`: indica el tipo de dato.

Variables: funciones útiles

Funciones útiles:

- `var_dump($var)`: permite volcar la información al completo de una variable.
- `isset($var)`: indica si la variable está definida y no es `null`.
- `unset($var)`: destruye una variable.
- `echo $var`: imprime el contenido.
- `gettype($var)`: indica el tipo de dato.

Constantes

Para definir una constante se utiliza la función `define()`, que recibe el nombre de la constante y el valor que queremos darle:

```
<?php
    define("FOO",      "something");
    define("FOO2",      5.222);
    define("FOO_BAR", 100);

    echo FOO_BAR;      // Imprime 100
?>
```

Tipos de datos: números

tipos_numericos.php

Número enteros: `integer`.

Números reales: `float`.

Operadores de conversión: `(int)`, `(float)`.

```
$a = 1.4444;  
$resultado = (45 + $a);  
echo $resultado;
```

```
$a = 1.4444;  
$resultado = (int) (45 + $a);  
echo $resultado;
```

Tipos de datos: números

tipos_numericos.php

Número enteros: `integer`.

Números reales: `float`.

Operadores de conversión: `(int)`, `(float)`.

La conversión entre ambos es automática. OJO esto puede dar problemas:

- Si en una operación con enteros se recibe un `float` o excede sus límites (`PHP_INT_MAX` y `PHP_INT_MIN`), se convierte a `float`.
- Si se fuerza a `integer` y se recibe un `float`, se trunca.

```
$a = 1.4444;  
$resultado = (45 + $a);  
echo $resultado;
```

```
$a = 1.4444;  
$resultado = (int) (45 + $a);  
echo $resultado;
```

Tipos de datos: cadenas (I)

El tipo `string` almacena un conjunto de caracteres.

Se declara con:

- Comillas simples.
- Comillas dobles → se sustituye las variables que hay dentro por su contenido.

```
<?php
    $var = "azul";

    echo "El cielo es $var <br>";
    echo 'El cielo es $var';
?>
```

Tipos de datos: cadenas (I)

El tipo `string` almacena un conjunto de caracteres.

Se declara con:

- Comillas simples.
- Comillas dobles → se sustituye las variables que hay dentro por su contenido.

```
<?php
    $var = "azul";

    echo "El cielo es $var <br>";
    echo 'El cielo es $var';
?>
```



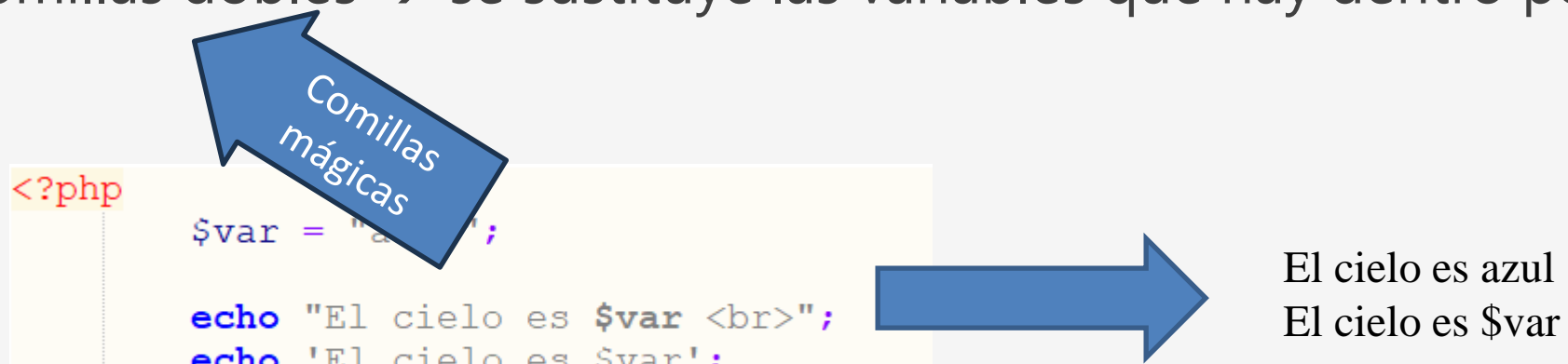
El cielo es azul
El cielo es \$var

Tipos de datos: cadenas (I)

El tipo `string` almacena un conjunto de caracteres.

Se declara con:

- Comillas simples.
- Comillas dobles → se sustituye las variables que hay dentro por su contenido.



```
<?php
    $var = "azul";

    echo "El cielo es $var <br>";
    echo 'El cielo es $var';
?>
```

El cielo es azul
El cielo es \$var

Tipos de datos: cadenas (II)

comillas_magicas.php

Para concatenar varias cadenas → operador punto (.)

```
<?php
    $sujeto = "El cielo";
    $verbo = " es ";
    $predicado = "rojo.";
    echo "Mi frase:" . $sujeto . $verbo . $predicado;
?>
```

```
<?php
    $frase1 = "Esto es una frase";
    $frase2 = "Esto también";
    echo $frase1;
    echo $frase2;
?>
```

Tipos de datos: cadenas (II)

comillas_magicas.php

Para concatenar varias cadenas → operador punto (.)

```
<?php
    $sujeto = "El cielo";
    $verbo = " es ";
    $predicado = "rojo.";
    echo "Mi frase:" . $sujeto . $verbo . $predicado;
?>
```



Mi frase:El cielo es rojo.

Si necesitas un salto de línea debes generar la etiqueta


```
<?php
    $frase1 = "Esto es una frase";
    $frase2 = "Esto también";
    echo $frase1;
    echo $frase2;
?>
```



Esto es una fraseEsto también

Tipos de datos: booleanos

Solo pueden tomar dos valores: `TRUE` o `FALSE`.

Entre otros, se generan al evaluar las expresiones de bucles y condicionales.

OJO: ¿qué pasa si meto como condición de un `if` una variable que no es un booleano?

Tipos de datos: booleanos

Solo pueden tomar dos valores: `TRUE` o `FALSE`.

Entre otros, se generan al evaluar las expresiones de bucles y condicionales.

OJO: ¿qué pasa si meto como condición de un `if` una variable que no es un booleano? Que se hacen conversiones implícitas.

Tipos de datos: booleanos

Solo pueden tomar dos valores: `TRUE` o `FALSE`.

Entre otros, se generan al evaluar las expresiones de bucles y condicionales.

OJO: ¿qué pasa si meto como condición de un `if` una variable que no es un booleano? Que se hacen conversiones implícitas.

- `integer` se asume `0` como `FALSE` y en otro caso `TRUE`.
- `float` `0.0` es `FALSE` y en otro caso `TRUE`.
- `string` vacío es `FALSE` y en otro caso `TRUE`.
- `null` es `FALSE`.
- Si un `array` está vacío es `FALSE`, en otro caso `TRUE`.

Tipos de datos: resumen general

- `integer`
- `float`
- `string`
- `boolean`
- `array`
- `object`
- `null`: valor por defecto de una variable no inicializada.
- `callable`: para implementar funciones de callback (no lo vemos).
- `resource`: representa recursos externos.

Ámbito de las variables

- Una variable definida:
 - **Dentro en una función** → solo visible dentro de ella.
 - **Fuera de una función** → visible en su fichero y en los que se incluyan desde este.

```
<?php
$a = 1;

function test()
{
    $a = 8;
    echo $a;
}

echo $a;
?>
```

Ámbito de las variables

- Una variable definida:
 - **Dentro en una función** → solo visible dentro de ella.
 - **Fuera de una función** → visible en su fichero y en los que se incluyan desde este.

```
<?php
$a = 1;

function test()
{
    $a = 8;
    echo $a; // Se imprime "8"
}

echo $a; // Se imprime "1"
?>
```


Ámbito de las variables

¿Qué es una variable global?

Variables predefinidas

global_server.php

En cada página que se ejecute:

- El módulo de PHP me pone a disposición ciertas variables especiales.
- Con información sobre el servidor, el cliente, la petición, etc.
- La mayoría visibles cualquier punto del código (son **globales**).

NOMBRE	DESCRIPCIÓN
\$GLOBALS	Variables globales definidas en la aplicación
\$_SERVER	Información sobre el servidor
\$_GET	Parámetros enviados con el método GET (en la URL)
\$_POST	Parámetros enviados con el método POST (formularios)
\$_FILES	Ficheros subidos al servidor
\$_COOKIE	Cookies enviadas por el cliente
\$_SESSION	Información de sesión
\$_REQUEST	Contiene la información de \$_GET, \$_POST y \$_COOKIE
\$_ENV	Variables de entorno

