



- **Unidad Didáctica 2**
GENERACIÓN DE DOCUMENTOS
WEB CON GUIONES DE
SERVIDOR

2.1 SINTAXIS

La primera página en PHP

- Las páginas PHP pueden ser páginas web *normales a las que se cambia la extensión, poniendo **.php** en vez de **.htm** ó **.html**.*
- En ellas **se pueden** insertar instrucciones –escritas en lenguaje PHP

```
<html>
<head>
<title>Aprendiendo PHP</title></head>
<body>
Esta es una página supersimple
<?php
echo "¡Aparecerá esta línea?";
?>
</body>
</html>
```

Las primeras instrucciones PHP

- `echo "un texto..";`
- `print "un texto..";`
- `print ("un texto..");`



Hará que el PHP escriba en la página web resultante lo contenido en esa cadena de texto.

Realizar el ejercicio sintaxis I

Líneas de comentario

- **Una sola línea: // o #**

// Este comentario no se verá en la página

- **Varias líneas: /* ... */**

/* Cuidado con anidar

/* comentarios

multilinea con estos*/

al PHP no le gustan */

Realizar el ejercicio sintaxis2

Constantes

- **Una constante es un valor** –un número o una *cadena*– **que no va a ser modificado a lo largo del** proceso de ejecución de los *scripts* que contiene un documento.
- Se definen mediante la siguiente instrucción:

`define("Nombre", "Valor")`

```
define("EurPta",166.386)
```

```
define("Cadena2","12.54Constante con punto decimal");
```

Ampliando echo

- Echo puede presentar *de forma simultánea varias cadenas de caracteres y/o constantes y variables*
 - Basta con ponerlas **una a continuación de otra** utilizando **una coma o punto como separador entre cada** una de ellas.
 - **Cadenas entre comillas.**
 - **Nombres de constantes nunca** van entre comillas.

```
echo "Con echo los números no necesitan ir entre comillas: "  
    ,3,"<br>";
```

Ampliando *print*

- **print también** permiten concatenar cadenas en una misma instrucción.
 - sólo es posible usar el punto como elemento de unión
 - Cadenas entre comillas.
 - Nombres de constantes deben ir entre comillas.
- `print "En el caso de print si son necesarias: " . "7" . "
";`
- `print ("incluso entre paréntesis necesitan las comillas: ". "45" . "
");`
- `print "Solo hay una excepción en el caso de print. ";`
- `print "Si los números van en un print independiente no necesitan comillas ";`
- `print 23;`

Realizar los ejercicios sintaxis3 y sintaxis4

Variables

- Es un espacio de la *memoria RAM* que se *reserva* – *a lo largo del tiempo* de ejecución de un script– para almacenar un determinado tipo de datos cuyos valores son susceptibles de ser modificados por medio de las instrucciones contenidas en el propio programa.
- todos los nombres de variable tienen que empezar por el símbolo **\$**. Y han de **llevar una letra inmediatamente** después del símbolo **\$**
 - **\$sensor1** válido
 - **\$1sensor** no válido

Variables

- Tener en cuenta:
 - El nombre
 - Su definición
 - El tipo
 - El ámbito local, global o superglobal

Tipo, definición y ámbito

- No es necesario definir el **tipo de variable**
 - Una misma variable puede contener en un momento una cadena y en otro un valor numérico
- Para **definir** una variable →
variable=valor; \$Led1="LedRojo"
- **Ámbito**:
 - Variables dentro del script → se puede acceder desde todas partes, excepto dentro de una función.
 - Variables dentro de una función → sólo dentro de la función

Variables globales

- Las funciones pueden utilizar valores de *variables externas*:

Para ello, incluir *dentro de la propia función* la instrucción:

- **global nombre de la variable;**
- Por ejemplo:
 - **global \$sensor1;**

Variables superglobales

- Son variables capaces de *comportarse como globales sin necesidad de que se definan como tales.* (>php 4.1.0)
- *No pueden ser creadas por el usuario*
 - Por ejemplo **\$_SERVER**, **\$_POST**, **\$_GET** o **\$_ENV**
- *Recogen información de forma automática*

Realizar el ejercicio sintaxis5

Constantes predefinidas

- **__FILE__**

- **__LINE__**

- **PHP_OS**

- **PHP_VERSION**

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<?
# La constante del sistema __FILE__ nos devolverá
    echo "La ruta completa de este fichero es: ";
    echo __FILE__;

# La constante del sistema __LINE__ nos devolverá
# el número de línea que se está interpretando
# también cuenta las líneas en blanco
# cuenta las líneas y verás que devuelve ... 16

    echo "<br>Esta es la línea: ",__LINE__, "del fichero";
    echo "<br>Estamos utilizando la versión: ",PHP_VERSION, " de PHP";
    echo "<br>El PHP se está ejecutando desde el sistema operativo:
    ",PHP_OS;
?>
</BODY>
</HTML>
```

Realizar el ejercicio sintaxis6

Variables predefinidas

- PHP genera automáticamente una serie de variables con diversa información sobre el cliente y el servidor.
 - **\$_REQUEST** es una matriz asociativa que contiene los datos enviado por los formularios y las cookies guardadas en el ordenador del cliente.
 - **\$_SERVER** es una matriz asociativa que contiene información sobre cabeceras, rutas y ubicaciones de scripts suministrada por el servidor (pero hay que tener en cuenta que no todos los servidores suministran todos los datos).
 - Documento sobre variables predefinidas
 - Ampliar en <http://php.net/manual/es/reserved.variables.php>

Otras variables I

- **Variables estáticas:**
 - conserva el último valor de una variable definida *dentro de una función*

```
static nombre = valor;  
static $a=3;
```

Realizar el ejercicio sintaxis7

Otras variables II

- **Variables de variables**

```
$$nombre_variable_previa;  
$color="verde";  
$$color="es horrible";
```

```
print $$color;  
o  
print ${$color};  
o también  
print $verde;
```

- **L a variable utilizada para definir una variable de variable al cambiar de valor *no se modifica ni el nombre de esta última ni tampoco su valor.***

Realizar el ejercicio sintaxis8

Tipos de variables

- En PHP **no es necesaria una** definición previa del tipo

Integer: Números enteros positivos y negativos

```
$MiVar = 123;
```

Double: Números decimales o de coma flotante

```
$MiVar = 1.3;
```

Strings: Cadena de caracteres

```
$MiVar = "Cadena de caracteres";
```

Boolean: Valores True o False

```
$MiVar = true;
```

Arrays

```
$MiVar[2] = 123;
```

Objetos

```
$MiVar = new MiClase();
```

Enteros cuyo valor esté comprendido entre $\pm 2^{31}$ (2147483647 -2147483647) será interpretado por PHP como de tipo **Integer**.

Decimal, o entero fuera del intervalo se convierten a Double

Determinación del tipo de variable utilizada

- **gettype**(*nombre de la variable*)

```
$a1=347          echo gettype($a1) devuelve integer  
$a5=3.1416      echo gettype($a5) devuelve double
```

Forzado de tipos

• Anteponemos el tipo al valor

Nuevos valores de la variable		
Valor inicial	Sintaxis	Nuevo valor
<code>\$a1=347</code>	<code>echo ((real)\$a1)</code>	347
<code>\$a2=2147483647</code>	<code>echo ((double)\$a2)</code>	2147483647
<code>\$a3=-2147483647</code>	<code>echo ((float)\$a3)</code>	-2147483647
<code>\$a4=23.7678</code>	<code>echo ((integer)\$a5)</code>	23
<code>\$a5="3.1416"</code>	<code>echo ((double)\$a6)</code>	3.1416
<code>\$a6="347"</code>	<code>echo ((int)\$a7)</code>	347
<code>\$a7="3.1416"</code>	<code>echo ((string)\$a7)</code>	3.1416
<code>\$a8="Solo literal"</code>	<code>echo ((int)\$a8)</code>	0
<code>\$a9="12.3 Literal con número"</code>	<code>echo ((double)\$a9)</code>	12.3
<code>\$a10=""</code>	<code>echo ((int)\$a9)</code>	0

Forzado de tipos usando *settype()*

Forzado de tipos con <i>settype()</i>		
Variable	Sintaxis	Devuelve
\$a1=347	echo (settype(\$a1,'double'))	1
	echo gettype(\$a1)	double
	echo \$a1	347
\$a2=2147483647	echo (settype(\$a2,'double'))	1
	echo gettype(\$a2)	double
	echo \$a2	2147483647
\$a3=-2147483647	echo settype(\$a3,'double')	1
	echo gettype(\$a3)	double
	echo \$a3	-2147483647
\$a4=23.7678	echo settype(\$a4,'integer')	1
	echo gettype(\$a4)	integer
	echo \$a4	23

2.1 Sintaxis

Fin de la presentación
Gracias