Tema 2. Inserción de código en páginas Web

DESARROLLO EN ENTORNO SERVIDOR 2º DAW

MARÍA CRIADO DOMÍNGUEZ

Índice

Elemento del lenguaje PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa CGI, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies.

https://www.php.net/manual/es/intro-whatis.php

Elemento del lenguaje PHP

PHP puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, macOS y probablemente otros más. PHP admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. Esto incluye cualquier servidor web que pueda utilizar el binario de PHP FastCGI, como lighttpd y nginx. PHP funciona tanto como módulo como procesador de CGI.

De modo que con PHP, se tiene la libertad de elegir el sistema operativo y el servidor web. Además, se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos (POO), o una mezcla de ambas.

Elemento del lenguaje PHP

Con PHP no se está limitado a generar HTML. Entre las capacidades de PHP se incluyen la creación de imágenes, ficheros PDF ...

También se puede generar fácilmente cualquier tipo de texto, como HTML y cualquier otro tipo de fichero XML. PHP puede autogenerar estos ficheros y guardarlos en el sistema de ficheros.

Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos. Escribir una página web con acceso a una base de datos es increíblemente simple utilizando una de las extensiones específicas de bases de datos o utilizar una capa de abstracción como PDO, o conectarse a cualquier base de datos que admita el estándar de Conexión Abierta a Bases de Datos por medio de la extensión ODBC.

Elemento del lenguaje PHP

- Como PHP se ejecuta del lado del servidor sólo puede tener acceso a los datos del propio servidor.
- No puede acceder a los recursos del cliente
- No puede saber qué hora es en el cliente
- No puede acceder a los archivos del cliente
  - Salvo la excepción de las Cookies

Elemento del lenguaje PHP

Fichero php.ini

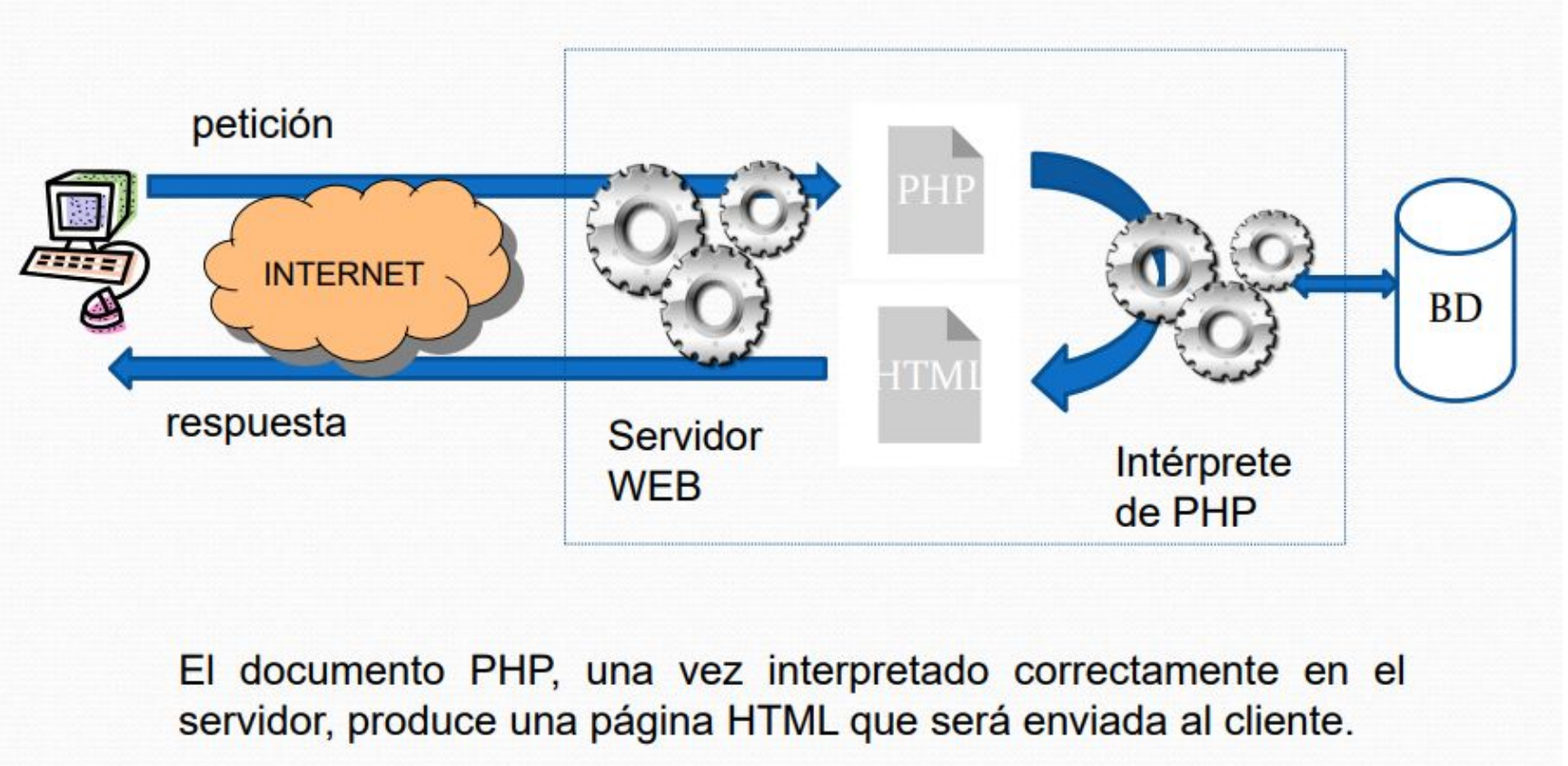
- Indica una serie de valores que determinan el comportamiento del intérprete PHP.
- Se encuentra en el directorio raíz de PHP
- Las instrucciones del fichero se denominan directivas
  - PHP: Listado de directivas de php.ini – Manual
- Las directivas están formadas por una pareja de clave y valor.
- Las directivas que comienzan por ; están comentadas y son ignoradas por el motor del intérprete.
- El fichero php.ini se lee cada vez que se arranca el servidor web.

Lenguajes embebidos en HTML

Los programas escritos en PHP, además encontrarse estructurados normalmente en varias páginas (ya veremos más adelante cómo se pueden comunicar datos de unas páginas a otras), suelen incluir en una misma página varios bloques de código.

Cada bloque de código debe ir entre delimitadores, y en caso de que genere alguna salida, ésta se introduce en el código HTML en el mismo punto en el que figuran las instrucciones en PHP.

Lenguajes embebidos en HTML



El documento PHP, una vez interpretado correctamente en el servidor, produce una página HTML que será enviada al cliente.

Etiquetas para inserción de código

- **Comentarios**

PHP admite comentarios al estilo de 'C', 'C++' y de consola de Unix (estilo de Perl).

```
// Esto es un comentario al estilo de c++ de una sola
línea
/* Esto es un comentario multilínea
       y otra lína de comentarios */

 # Esto es un comentario al estilo de consola de una s
ola línea
```

Etiquetas para inserción de código

Hay dos tipos de etiquetas para delimitar bloques de código PHP:

- 1ª Forma. Para servir documentos HTML, XHTML o XML:

```
<?php
  Instrucciones PHP
?>
```

- 2ª Formato corto:

```
<?
  Instrucciones PHP
?>
```

Etiquetas para inserción de código

Sólo la primera forma asegura portabilidad.

- Para usar el formato corto debemos activar en el fichero php.ini la directiva:

```
short_open_tag = on
```

Etiquetas para inserción de código

Existen varias formas incluir contenido en la página web a partir del resultado de la ejecución de código PHP.

La forma más sencilla es usando echo, que no devuelve nada (void), y genera como salida el texto de los parámetros que recibe.

```
void echo (string $arg1, ...);

echo “Hola mundo”;
echo “Hola “, “mundo”;
```

Etiquetas para inserción de código

Otra posibilidad es print, que funciona de forma similar. La diferencia más importante entre print y echo, es que print sólo puede recibir un parámetro y devuelve siempre 1.

```
int print (string $arg);


print “Hola mundo”;
print “Hola “ . “mundo”;
```

Etiquetas para inserción de código

Generación de código HTML.

printf es otra opción para generar una salida desde PHP. Puede recibir varios parámetros, el primero de los cuales es siempre una cadena de texto que indica el formato que se ha de aplicar. Esa cadena debe contener un especificador de conversión por cada uno de los demás parámetros que se le pasen a la función, y en el mismo orden en que figuran en la lista de parámetros.

```
printf(string $format, mixed $args = ?, mixed $... = ?
```

```
printf("%d", "17,999")
printf("%f", "17,999")
printf("%s", "17,999")
```

Etiquetas para inserción de código

Generación de código HTML.

Esta función muestra información estructurada sobre una o más expresiones incluyendo su tipo y valor. Las matrices y los objetos son explorados recursivamente con valores sangrados para mostrar su estructura.

```
var_dump("maria", 3.14);
```

Etiquetas para inserción de código

- Ejemplo
  - Hola Mundo

Etiquetas para inserción de código

Generación de código HTML

Uso de \n para generar código HTML legible

Código PHP

```
print ("<P>Párrafo 1</P>\n");
print ("<P>Párrafo 2</P>\n");
```

Código HTML

```
<P>Párrafo 1</P>
<P>Párrafo 2</P>
```

Salida

```
Párrafo 1

Párrafo 2
```

Etiquetas para inserción de código

Heredoc

- Cuando tenemos necesidad de escribir largos bloques de código HTML, incluso con variables intercaladas, podemos usar la construcción heredoc que nos permite escribir grandes cantidades de texto, sin necesidad de escapar caracteres en su interior.
- También podemos almacenarlo dentro de una variable

```
<<< TEXT
 Todo el texto que queremos <p>”con comillas” </p> O las variables… todo hasta
TEXT;
```

Variables

- Como en todos los lenguajes de programación, en PHP puedes crear variables para almacenar valores. Las variables en PHP siempre deben comenzar por el signo $.
- Los nombres de las variables deben comenzar por letras o por el carácter _, y pueden contener también números. Sin embargo, al contrario que en muchos otros lenguajes, en PHP no es necesario declarar una variable ni especificarle un tipo (entero, cadena,…) concreto.
- Para empezar a usar una variable, simplemente asígnale un valor utilizando el operador =.

```
$mi_variable = 7;
```

Variables

Los tipos de datos simples en PHP son:

- booleano (boolean). Sus posibles valores son true y false. Además, cualquier número entero se considera como true, salvo el 0 que es false.
- entero (integer). Cualquier número sin decimales. Se pueden representar en formato decimal, octal (comenzando por un 0), o hexadecimal (comenzando por 0x).
- real (float). Cualquier número con decimales. Se pueden representar también en notación científica.
- cadena (string). Conjuntos de caracteres delimitados por comillas simples o dobles.
- null. Es un tipo de datos especial, que se usa para indicar que la variable no tiene valor.
- Objetos. Variable de una clase

Variables

Functiones con variables

gettype(). Devuelve el tipo de la variable

settype() - Establece el tipo de una variable

is_array() - Comprueba si una variable es un array

is_bool() - Comprueba si una variable es de tipo booleano

is_float() - Comprueba si el tipo de una variable es float

is_int() - Comprueba si el tipo de una variable es integer

is_null() - Comprueba si una variable es null

is_numeric() - Comprueba si una variable es un número o un string numérico

is_object() - Comprueba si una variable es un objeto

is_string() - Comprueba si una variable es de tipo string

Variables

Si realizas una operación con variables de distintos tipos, ambas se convierten primero a un tipo común. Por ejemplo, si sumas un entero con un real, el entero se convierte a real antes de realizar la suma:

```
$mi_entero = 3;
$mi_real = 2.3;
$resultado = $mi_entero + $mi_real;
// La variable $resultado es de tipo real
```

Variables

La conversión automática que realiza PHP no siempre es lo que queremos.

PHP permite otras conversiones implícitas de tipos :

- (int) : Fuerza la conversión a entero
- (real), (double), (float): Fuerza la conversión a coma flotante.
- (string): Fuerza la conversión a cadena de caracteres.
- (array): Fuerza la conversión a matriz
- (object): Fuerza la conversión a un objeto.

Variables

Estas conversiones de tipo, que en el ejemplo anterior se lleva a cabo de forma automática, también se pueden realizar de forma forzada:

```
$mi_entero = 3;
$mi_real = 2.3;
$resultado = $mi_entero + (int) $mi_real;
// La variable $mi_real se convierte a entero (valor 2) antes de sumarse.
// La variable $resultado es de tipo entero (valor 5)
```

Variables

Variable de variables

Se pueden crear nombres de variables dinámicamente anteponiendo $$ a una variable.

La variable de variable toma su nombre del valor de otra variable previamente declarada.

```
<?php
 $var = "uno";
 $$var = "dos";
 print ($var);   // produce el texto: "uno"
 print ($uno);   // produce el texto: "dos"
 print ($$var);          // produce el texto: "dos"
```

Variables por referencia

La variable no contiene un valor sino la dirección de otra variable.

En PHP las variables se pasan por referencia al precederlas del símbolo &.

El signo & indica que se está almacenando la dirección de la variable y no su contenido.

```
$ref = &$var;
```

Variables Especiales

PHP incluye unas cuantas variables internas predefinidas que pueden usarse desde cualquier ámbito, por lo que reciben el nombre de variables superglobales. Ni siquiera es necesario que uses global para acceder a ellas. Cada una de estas variables es un array que contiene un conjunto de valores .

- $_SERVER. Contiene información sobre el entorno del servidor web y de ejecución.
- $_GET, $_POST y $_COOKIE contienen las variables que se han pasado al guión actual utilizando respectivamente los métodos GET (parámetros en la URL), HTTP POST y Cookies HTTP.
- $_REQUEST junta en uno solo el contenido de los tres arrays anteriores, $_GET, $_POST y $_COOKIE.
- $_ENV contiene las variables que se puedan haber pasado a PHP desde el entorno en que se ejecuta.
- $_FILES contiene los ficheros que se puedan haber subido al servidor utilizando el método POST.
- $_SESSION contiene las variables de sesión disponibles para el guión actual.

Variables
En la URL

Un mecanismo práctico aunque no muy seguro de intercambio de información entre una página y otra consiste en pasar las variables a través de un sufijo en la URL de la página llamada.

```
http://www.mipagina.es/pagina.php?nombre='maría'
```

El programa PHP recibe estas variables dentro de las matrices superglobales $_REQUEST o $_GET,

GET ['nombre']

Utilizar la función urlencode() cuando los valores de las variables en la URL contienen caracteres especiales.

Variables

Ámbito

En PHP puedes utilizar variables en cualquier lugar de un programa. Si esa variable aún no existe, la primera vez que se utiliza se reserva espacio para ella. En ese momento, dependiendo del lugar del código en que aparezca, se decide desde qué partes del programa se podrá utilizar esa variable. A esto se le llama visibilidad de la variable.

Si aparece una asignación fuera de la función, se le considerará una variable distinta.

```
$a = 1;
function prueba() {//no se tiene acceso a la variable $a
    $b = $a; // no tiene valor asignado (su valor es null)
}
```

Variables

Ámbito

Si en la función anterior quisieras utilizar la variable $a externa, podrías hacerlo utilizando la palabra global. De esta forma le dices a PHP que no cree una nueva variable local, sino que utilice la ya existente.

```
$a = 1;
function prueba() {
    global $a;
    $b = $a; // En este caso se le asigna a $b el valor 1
}
```

Variables

Ámbito

Las variables locales a una función desparecen cuando acaba la función y su valor se pierde. Si quisieras mantener el valor de una variable local entre distintas llamadas a la función, deberás declarar la variable como estática utilizando la palabra static.

```
function contador() {
  static $a=0;
  $a++; // Cada vez que se ejecuta la función, se incrementa el valor de $a
}
```

Las variables estáticas deben inicializarse en la misma sentencia en que se declaran como estáticas. De esta forma, se inicializan sólo la primera vez que se llama a la función.

Constantes

- Una constante es un identificador de un dato que no cambia de valor durante toda la ejecución de un programa.
- Las constantes no se asignan con el operador =, sino con la función define :

```
define(nombre_constante_entre_comillas, dato_constante);
define ("PI", 3.1416)
```

- No llevan $ delante
- La función defined("PI") devuelve TRUE si existe la constante.
- Son siempre globales por defecto.
- Sólo se pueden definir constantes de los tipos escalares (boolean, integer, double, string)
- No se puede cambiar su valor ,se generará un error

Constantes predefinidas

Dependen de las extensiones que se hayan cargado en el servidor, aunque hay constantes predefinidas que siempre están presentes :

- PHP_VERSION: Indica la versión de PHP que se está utilizando.
- PHP_OS: Nombre del sistema operativo que ejecuta PHP.
- TRUE
- FALSE
- E_ERROR: Indica los errores de interpretación que no se pueden recuperar.
- E_PARSE: Indica errores de sintaxis que no se pueden recuperar.
- E_ALL: Representa a todas las constantes que empiezan por E_.

Inclusión de ficheros externos

La inclusión de ficheros externos se consigue con:

- include()
- require()

Ambos incluyen y evalúan el fichero especificado

Diferencia: en caso de error include() produce un warning y require() un error fatal

Se usará require() si al producirse un error debe interrumpirse la carga de la página

- require_once()
- Include_one()

Tipos de datos especiales
Fechas

- Con frecuencia cuando creamos webs o apps tenemos que trabajar con fechas y calendarios. Por ejemplo, en la página web de un hotel o un restaurante es posible que trabajemos con fechas y horas de comienzo de reserva, de fin de reserva, etc. PHP dispone de funciones nativas para facilitar el trabajo con fechas, horas y tiempos.
- El manejo del “tiempo” es un aspecto controvertido en la programación.
- La primera dificultad y más obvia es que existen cientos de dispositivos electrónicos y no todos manejan la misma fecha.

Tipos de datos especiales
Fechas

Función time()

- La función time() devuelve la hora GMT actual medida como el número de segundos desde el 1 de enero de 1970 00:00:00 GMT (hora del meridiano de Greenwich) obtenidos a partir de la hora actual local del servidor.
- Cuando hablamos de “hora actual del servidor” nos referimos a una hora local, que es distinta según el país donde nos encontremos.
- La función time() devuelve un valor numérico entero largo, por ejemplo 1335169779.

Tipos de datos especiales
Fechas

Función date_default_timezone

date_default_timezone_get()

- informa de la zona horaria establecida en el servidor en el que se ejecuta el script

date_default_timezone_set('Europe/Madrid');

- Establezce la zona horaria donde se encuentra el servidor
- Ya tiene en cuenta horario de verano e invierno

Tipos de datos especiales
Fechas

Función date

Para transformar ese número en una fecha “entendible por las personas” usamos la función date, cuya sintaxis general es:

```
date (“formato de salida”, valorTimeValido)
```

https://www.php.net/manual/es/function.date.php

Tipos de datos especiales
Fechas

Función strtotime

- Esta función espera que se proporcione una cadena que contenga un formato de fecha en Inglés US
- Intentará convertir ese formato a marca de tiempo Unix (el número de segundos desde el 1 de Enero del 1970 00:00:00 UTC)
- Si no se le proporciona el parámetro now, tomará como fecha actual la que tengamos establecida en el script o en caso contrario la del servidor.

```
strtotime ( string $time [, int $now = time() ] ) : int
```

Tipos de datos especiales
Fechas

Función strtotime

Las fechas en los formatos m/d/y o d-m-y no son ambiguas al observar el separador entre los distintos componentes:

- si el separador es una barra (/), se asume el formato norteamericano m/d/y; mientras que si el separador es un guion (-) se asume el formato europeo d-m-y.
- Si el año se proporciona en un formato de dos dígitos y el separador es un guion (-), la cadena de la fecha se analiza como y-m-d.
- Para evitar esta ambigüedad potencial es mejor usar fechas ISO 8601 (YYYY-MM-DD)

Tipos de datos especiales
Fechas

Función mktime

Obtener la marca de tiempo Unix de una fecha

```
mktime(
    int $hour = date("H"),
    int $minute = date("i"),
    int $second = date("s"),
    int $month = date("n"),
    int $day = date("j"),
    int $year = date("Y"),
    int $is_dst = -1
)
```

Tipos de datos especiales Fechas

Función getdate

Devuelve un array asociativo que contiene la información de la fecha de timestamp, o el momento local actual si no se da timestamp.

```
getdate(int $timestamp = time())
```

```
Array
(
    [seconds] => 40
    [minutes] => 58
    [hours]   => 21
    [mday]    => 17
    [wday]    => 2
    [mon]     => 6
    [year]    => 2003
    [yday]    => 167
    [weekday] => Tuesday
    [month]   => June
    [0]       => 1055901520
)
```

Tipos de datos especiales
Fechas

DateTime y DateTimeInterval

Este método es un estilo orientado a objetos para obtener la diferencia entre dos fechas, este es también el más fácil ya que no requiere el cálculo manual de las fechas y recomendado ya que es de la más reciente versión de PHP.

```
$firstDate  = new DateTime("2019-01-01");
$secondDate = new DateTime("2020-03-04");
$intvl = $firstDate->diff($secondDate);
```