

UT2

INSERCIÓN DE CÓDIGO EN PÁGINAS WEB

Índice

2

1. Inserción código de servidor
2. Ámbito de ejecución
3. PHP
4. Etiquetas de servidor
5. Sintaxis básica
6. Tipos de datos
7. Variables
8. Constantes
9. Expresiones y operadores

Inserción código de servidor


3

- Estructura páginas estáticas:
 - Sencilla
 - Con tres partes:
 - Cabecera (definición directivas)
 - Zona metadatos
 - Cuerpo del documento (con información a mostrar)
- Una vez creada la página se inserta el código a compilar e interpretar por el servidor.
- El código de servidor, se ejecutará al procesar la petición HTTP del cliente.

Ámbito de ejecución

4

- ❑ Los lenguajes de servidor sólo pueden tener acceso a los recursos alojados en el servidor o bien otros ofrecidos a través de la red.
- ❑ NO pueden acceder a recursos del propio cliente.
- ❑ Poseen archivos de configuración para definir el comportamiento de la aplicación (php.ini)

 **XAMPP for Windows**

English / Deutsch / Français / Nederlands / Polski / Italiano / Norwegian / Español / 中文 / Português (Brasil) / 日本語

XAMPP 1.8.2
[PHP: 5.4.19]

Bienvenido
Estado
Chequeo de seguridad
Documentación
Componentes
Applications

Php
Inicio
Administración de CD
Biontmo
Instant Art
Agenda de telefonos

Perl
perlinfo()
Libro de invitados

JZee
Info
Tomcat examples

Tools
phpMyAdmin
FileZilla FTP
Webalizer
Mail

©2002-2014
APACHE
FRIENDS...

Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	csconfig inolog configure.js "--enable-snapshot-build" "--disable-lsapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pdo-oci" "--with-pdo-oci=C:\php-sd\oracle\instantclient10sdk\shared" "--with-oci8=C:\php-sd\oracle\instantclient10\sdk\shared" "--with-oci8-11g=C:\php-sd\oracle\instantclient11sdk\shared" "--enable-object-out-dir=obj" "--enable-com-dotnet-shared" "--with-mcrypt-static" "--disable-static-analyze" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525.TS.VC9
PHP Extension Build	API20100525.TS.VC9
Debug Build	no
Thread Safety	enabled
Zend Signal Handling	disabled

□ **PHP** → *HyperText Preprocessor*

- Lenguaje de código **abierto**
- Permite su uso junto con documentos HTML mediante la **inserción de fragmentos** de código PHP acotados por etiquetas especiales
- El código PHP es **interpretado** por el servidor web donde se aloja el documento y se genera el correspondiente código HTML para mostrar por el navegador
- Las páginas tienen **extensión *.php**
- Tiene una interfaz de desarrollo basada en la **POO**
- Es compatible con diferentes tecnologías
- Admite los principales servidores web y soporte con **diferentes repositorios de BD** (para MySQL posee librerías específicas)

Etiquetas de servidor

6

- PHP es **sensible a las mayúsculas**
- La incrustación de código PHP en HTML se puede hacer de varias maneras:
 - Forma a) `<?php..... ?>` (**compatible con todas las plataformas**)
 - Forma b) `<? ?>`
 - Forma c) `<script language="php"> </script>`
 - Forma d) `<% %>`
- Sólo la **forma a)** es compatible y portable con todas las plataformas, el resto requiere hacer ajustes adicionales en el fichero de configuración.

Etiquetas de servidor

7

```
<html>
  <head>
    <title>
      Desarrollo PHP
    </title>
  </head>
  <body>
    <h2>Desarrollo en servidor</h2>
    Estas líneas están escritas directamente en HTML
    <br>Esta es una línea incluida directamente en el cuerpo de la página web<br/>
    <?php
      $expresion="1";
      if ($expresion == "1"){
        print("1. Empiezan líneas generadas por PHP <br>");
        print("2. El texto está por instrucción print de PHP");
      }
    ?>
  </body>
</html>
```

Etiquetas de servidor

8

Ejercicio

Realiza una página web que presente como título “Mi primer ejemplo” y un mensaje de texto “Primer ejemplo:” mediante etiquetas HTML y un mensaje “¡Hola mundo!” con un script PHP.

Realízalo utilizando las distintas etiquetas PHP vistas para comprobar cuáles son compatibles con tu servidor.

```
<!DOCTYPE html>
<!-- Autor: Iván Lorenzo -->
<html>
  <head>
    <meta charset="UTF-8">
    <title>Mi primer ejemplo</title>
  </head>
  <body>
    <?php
      echo "¡Hola Mundo!";
    ?>
  </body>
</html>
```


Sintaxis básica

9

- Reglas básicas de PHP:
 - Es sensible a las mayúsculas
 - Los espacios en blanco dentro del código embebido no tienen ningún efecto
 - El final de instrucción se indica con ;
 - Los scripts embebidos pueden situarse en cualquier parte código HTML.
 - El número de scripts es indefinido.
 - Cuando se ejecuta un código embebido, el script se sustituye por el resultado de dicha ejecución, incluidas las etiquetas de inicio y fin.

Sintaxis básica

10

```
<!DOCTYPE html>
<!-- Autor: Iván Lorenzo -->
<html>
  <?php
    $salida = "Contenido PHP";
  ?>
  <head>
    <title> <?php      echo $salida;    ?>
    </title>
  </head>
  <body>
    <h1>Segundo ejemplo: </h1>
    <?php
      echo $salida;
    ?>
  </body>
</html>
```

Sintaxis básica

11

□ Comentarios:

- **Comentarios de una línea** utilizando //
- **Comentarios de una línea** utilizando #
- **Comentarios de varias líneas.** /* */

Sintaxis básica

12

- Para imprimir: ***echo*** y ***print***
 - ▣ ***echo***: muestra una o más cadenas
`echo cadena1 [, cadena2...];`
`echo "Hola mundo";`
`echo "Hola ", "mundo";`
 - ▣ ***print***: muestra una cadena
`print cadena;`
`print "Hola mundo";`
`print "Hola " . "mundo";`

Ejemplo

Realiza el primer ejemplo de "Hola mundo", pero en esta ocasión con la sentencia `print`

Variables y tipos de datos en PHP

13

- El tipo de una variable no se suele especificar.

```
$mi_variable = 7;
```

- Se decide en tiempo de ejecución en función del contexto y puede variar.

```
$mi_variable=7;
```

```
$mi_variable="cambio de tipo";
```

Variables y tipos de datos en PHP

14

- Reglas para nombrar una variable en PHP:
 - El nombre debe comenzar con una **letra** o con un **guión bajo** (“_”)
 - El nombre únicamente puede contener caracteres **alfanuméricos** y **guiones bajos**.
 - El nombre de una variable **no** debe contener **espacios** en blanco. Si queremos formar el nombre de una variable con más de una palabra lo que se suele hacer es utilizar un **guión bajo entre ellas** (`$mi_variable`) o poner **la primera letra de cada palabra en mayúsculas** (`$miVariable`)

Variables y tipos de datos en PHP

15

□ Tipos de datos:

- **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.

Variables y tipos de datos en PHP

16

Ejemplo

Comprueba el resultado de las siguientes expresiones visualizando por pantalla su valor:

```
$mi_booleano = false;  
$mi_entero = 0x2A;  
$mi_real = 7.3e-1;  
$mi_cadena = "texto";  
$mi_variable = null;
```

Ejemplo

¿De qué tipo es la variable \$resultado después de su ejecución?. Compruébalo visualizando su valor por pantalla:

```
$mi_entero = 3;  
$mi_real = 2.3;  
$resultado = $mi_entero + $mi_real;
```


Expresiones y operadores

17

- Operadores **aritméticos**: $+$, $-$, $*$, $/$, $\%$, $++$, $--$
- Operador de **asignación**: $=$
- Operadores de **comparación**: $==$, $!=$, $<$, $>$, $<=$, $>=$
- Operador de **control de error**: $@$. Antepuesto a una expresión, evita cualquier mensaje de error que pueda ser generado por la expresión.
- Operadores **lógicos**: $\text{and } (\&\&)$, $\text{or } (||)$, $!$, xor
- Operadores de **cadena**:
 - concatenación: $.$ (punto)
 - asignación con concatenación: $.=$

Expresiones y operadores

18

- Precedencia de operadores (de mayor a menor):

++, --

*, /, %

+, -

<, <=, >, >=

==, !=

&&

||

and

or

Combinación de etiquetas html y código PHP

19

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Hola Mundo</title>
```

```
  </head>
```

```
  <body>
```

```
    <strong><?php echo "Hola Mundo";?></strong>
```

```
  </body>
```

```
</html>
```

Insertar etiquetas dentro de un script PHP

20

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Hola Mundo</title>
```

```
  </head>
```

```
  <body>
```

```
    <?php echo "<strong>Hola Mundo</strong>" ;?>
```

```
  </body>
```

```
</html>
```

Insertar todas las etiquetas dentro de un script PHP

21

```
<?php
```

```
    print ("<!DOCTYPE html>");
```

```
    print ("<html>");
```

```
    print ("<head>");
```

```
    print ("<title>Hola Mundo</title>");
```

```
    print ("</head>");
```

```
    print ("<body>");
```

```
    print ("<strong>Hola Mundo</strong>");
```

```
    print ("</body>");
```

```
    print ("</html>");
```

```
?>
```

Ámbito de variables

22

- Se pueden utilizar en cualquier lugar del programa.
- Si la variable no existe se reserva espacio en memoria.
- Si una variable se define dentro de una función es una variable **local** a la función.
- Si aparece fuera de la función se considera distinta a la definida en la función.
- Si dentro de una función se quiere usar una variable definida fuera hay que usar la palabra **global**.
- Si se quiere mantener el valor de una función declara dentro al salir de esta, se debe definir como **static**

Cadenas de texto

23

- Se definen tanto con comillas simples como dobles.
- Cuando se pone una variable dentro de unas comillas dobles, se procesa y se sustituye por su valor.

```
<?php
```

```
$modulo="DWES";
```

```
echo ("<p>Módulo: $modulo</p>");
```

```
?>
```

- PHP al encontrar la variable la sustituye por DWES.
- Para que PHP distinga correctamente el texto que forma la cadena del nombre, a veces es necesario encerrarla entre llaves

```
echo "<p>Módulo: ${modulo}</p>";
```

Cadenas de texto

24

□ Concatenación de cadenas

- ▣ Concatenación punto (.)
- ▣ Operador de asignación y concatenación (.=): concatena al argumento del lado izquierdo la cadena del lado derecho

```
<?php
```

```
$a = "Módulo ";
```

```
$b = $a . "DWES";
```

```
// ahora $b contiene "Módulo DWES"
```

```
$a .= "DWES";
```

```
// ahora $a también contiene "Módulo DWES"
```

```
?>
```


Funciones relacionadas con los tipos de datos

25

- Funciones para comprobar y establecer el tipo de datos de una variable
 - **gettype()** → obtiene el tipo de variable
 - Ej: `gettype($variable);`
 - **settype()** → convierte una variable al tipo indicado por parámetro.
 - Ej: `settype($variable,"float");`
- Funciones para comprobar si es de un tipo concreto:
 - **is_array(), is_bool(), is_float(), is_integer()....**
 - **isset()** → indica si una variable está definida y no es null.
 - Ej: `isset($variable);`
 - **unset()** → destruye una variable
 - Ej: `unset($variable);`

Funciones relacionadas con los tipos de datos

26

□ **Fechas:** no hay un tipo específico. La información fecha y hora se almacena como un número entero y hay una serie de funciones en PHP para trabajar con ellas:

■ **date()** es una de las más útiles. Permite obtener una cadena de texto a partir de una fecha y hora con el formato que se elija.

`date (string $formato [, int $fechahora]);`

Si no se indica el segundo parámetro se utiliza la hora actual.

■ **date_default_timezone_set** establece la zona horaria donde me encuentro.

`date_default_timezone_set('Europe/Madrid');`

■ **getdate()** → devuelve un array con información sobre la fecha y hora actual.

Variables especiales en PHP

27

- PHP incluye variables internas predefinidas que pueden usarse desde cualquier ámbito. Se denominan **superglobales**.
- Cada una de estas variables es un array que contiene un conjunto de valores. Son las siguientes:
 - **\$_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
 - **\$_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.