

UT2 INSERCIÓN DE CÓDIGO EN PÁGINAS WEB

Índice

- UT2 INSERCIÓN DE CÓDIGO EN PÁGINAS WEB
 - Índice
 - Inserción de código de servidor
 - Ámbito de ejecución
 - PHP- Hypertext Preprocessor
 - Etiquetas de servidor
 - Sintaxis básica
 - Comentarios:
 - Imprimir : echo y print
 - variables y tipos de datos
 - Expresiones y operadores
 - Precedencia de operadores (de mayor a menor):
 - Combinación de etiquetas html y código PHP
 - Insertar etiquetas dentro de un script PHP
 - Insertar todas las etiquetas dentro de un script PHP
 - Ámbito de las variables
 - Cadenas de texto
 - Concatenación de cadenas
 - Funciones relacionadas con los tipos de datos
 - Operador ternario

Inserción de código de servidor

- Estructura páginas estáticas. Una página sencilla tiene tres partes:
 - Cabecera (definición directivas)
 - Zona metadatos
 - Cuerpo del documento (con la 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

- 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)

Etiquetas de servidor

La incrustación de código PHP en HTML se puede hacer de varias maneras:

- Forma a)(compatible con todas las plataformas)

```
<?php
...
?>
```

- 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.
- Ejemplo

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>
      Desarrollo web
    </title>
  </head>
  <body>
    <h1>Desarrollo en servidor</h1>
    <h2> estas líneas están escritas en HtML</h2>
    <p> esta es una línea incluida en el cuerpo de la página</p>
```

```

<?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>

```

- 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

Sintaxis básica

- 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.
- Manual de PHP [manual de php](#)

```

<!DOCTYPE html>
<html>

```

```

<?php

    $salida="contenido php";

?>

```

```

<head>
    <meta charset="UTF-8">
    <title>

```

```

<?php
    echo $salida;

```

```
?>
```

```
    </title>
</head>
<body>
    <h1>Otro ejemplo</h1>
    <h2>
```

```
<?php
    echo $salida;
?>
```

```
    </h2>
</body>
<html>
```

Comentarios:

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

Imprimir : echo y print

- echo: muestra **una o más cadenas**

```
echo cadena1 [, cadena2...];
```

Ejemplo

```
echo "Hola Mundo";
echo "Hola", "Mundo";
```

- print: muestra **una cadena**

```
print cadena ;
```

Ejemplo

```
print "Hola mundo";  
print "Hola "."mundo";
```

variables y tipos de datos

- 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";
```

- 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)
- 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.

Ejercicios:

1. 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;
```

2. ¿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

- 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**: and (&&), or (||), !, xor
- Operadores de **cadena**:
 - concatenación: . (punto)
 - asignación con concatenación: .=

Precedencia de operadores (de mayor a menor):

- ++,-- (operadores unarios)
- *, /, %
- +, -
- <, <=, >, >=
- ==, !=
- &&
- ||
- and
- or

Combinación de etiquetas html y código PHP

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8">  
    <title>  
      Hola mundo  
    </title>  
  </head>  
  <body>  
    <h1>
```

```
<?php
    echo "Hola mundo";
?>
```

```
</h1>
</body>
<html>
```

Insertar etiquetas dentro de un script PHP

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>
      Hola mundo
    </title>
  </head>
  <body>
```

```
<?php
    echo "<h1> Hola mundo </h1>";
?>
```

```
</body>
<html>
```

Insertar todas las etiquetas dentro de un script PHP

```
<?php
    print ("<!DOCTYPE html>");
    print ("<html>");
    print ("<head>");
    print ("<title> hola mundo</title>");
    print ("</head>");
    print ("<body>");
    print ("<h1>Hola mundo</h1>");
    print ("</body>");
    print ("</html>");
?>
```


Ámbito de las variables

- 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

- 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**

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

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

Funciones para comprobar y establecer el tipo de datos de una variable

- **gettype()**: obtiene el tipo de variable

```
gettype($variable);
```

- **settype()**: convierte una variable al tipo indicado por parámetro.

```
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.

```
isset($variable);
```

- **unset ()**: destruye una variable.

```
unset($variable);
```



Hoja02_PHP_02

Funciones para **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.

Más información sobre [funciones de fechas](#)

Operador ternario

El operador condicional (ternario) es el único operador que tiene tres operandos.

Sintaxis:

```
condición ? expr1 : expr2
```

- condición , una expresión que se evalúa como true o false.
- expr1, expr2 , expresión con valores de algún tipo

```
<?php
    $a=1;
    $b=3;
    $imprimir=($a<$b)? "a es menor que b":(($a>$b)?"a es mayor que b": "a es igual
a b");
    print $imprimir;
?>
```

