UT3. Desarrollo de aplicaciones web utilizando código embebido en PHP

DAW - Desarrollo Web Entorno Servidor
Fernando Galindo

Contenidos

1. Funciones
2. Tipos de datos compuestos
3. Ficheros

1. Funciones

Funciones

Las funciones nos permiten repetir un bloque de código, nos permiten asociar un bloque de código a una etiqueta

La función que está definida en todo sistema con PHP es la siguiente

```
phpinfo();
```

Que muestra información sobre la versión y configuración de PHP

1. Funciones

Si queremos definir nuestras funciones debemos usar la palabra function

```
function precio_con_iva($precio){
        return $precio*1.21;
}
```

Los argumentos de la función son definidos como variables

El resultado de la función se hace con la palabra `return`, fijaros que no hace falta indicar ningún tipo

```
echo precio_con_iva(37.2); // Llamada a la función
```

1. Funciones

Si queremos usar una función debe definirse con anterioridad

```
<?php
echo precio_con_iva(32.7) //ERROR
function precio_con_iva($precio){
        return $precio*1.21;
}
?>
```

Ámbito de las variables

```
$a=1

function prueba(){

// Dentro de la función no se tiene acceso a la variable
// $a anterior

    $b = $a;

// $a es una variable nueva que no tiene valor (null)

}
```

1. Funciones

Ámbito de las variables, con la palabra reservada global podemos acceder a las variables del nivel superior

```
$a = 1;

function prueba(){

   global $a;

   $b = $a;

// En este caso se le asigna a $b el valor 1

}
```

1. Funciones

Variables estáticas, son variables locales a la función que recuerdan su valor

```
function estatica(){
    static $cuenta=0;
    $cuenta++;
    echo "Esta es la llamada número $cuenta<br />";
}
for($i=1;$i<=10;$i++){
    estatica();
}
```

1. Funciones

El paso de valores de las funciones es por valor

```
function intercambia($a,$b){
        $aux=$a;
        $a=$b;
        $b=$aux;
}
        $num1=5;
        $num2=8;
        echo "<br/>".$num1." ".$num2;    // Muestra 5 y 8
        intercambia($num1,$num2);        // El paso de hace por valor
        echo "<br/>".$num1." ".$num2;    // Sigue mostrando 5 y 8
```

1. Funciones

El paso de valores de las funciones es por valor

```
function intercambia(&$a,&$b){
        $aux=$a;
        $a=$b;
        $b=$aux;
}
        $num1=5;
        $num2=8;
        echo "<br/>".$num1." ".$num2;    // Muestra 5 y 8
        intercambia($num1,$num2);        // El paso de hace por referencia
        echo "<br/>".$num1." ".$num2;    // Muestra 8 y 5
```

1. Funciones

Podemos crear nuestros propios ficheros de PHP con funciones y códigos para reutilizar en nuestro programa con las palabras `include` y `require`

```php
/*Fichero funciones.php*/
<?php
$PI=3.141516;
function precio_con_iva($precio){
        return $precio*1.21;
}
?>
```

```php
/*Fichero index.php*/
<?php
include 'funciones.php';
echo precio_con_iva(36);
echo $PI;
?>
```

Ambas instrucciones `include` y `require`, copian el código y está disponible dentro del programa principal

En el caso de que el fichero a importar no exista

- `include`, el script continua y no da error mientras no se usen las funciones
- `require`, da un error en tiempo de ejecución y termina el script

```
/*Fichero index.php*/
<?php
include 'noExiste.php';
echo $PI; //Da un warning
?>
```

```
/*Fichero index.php*/
<?php
require 'noExiste.php';
//Error no se ejecuta
echo $PI;
?>
```

1. Funciones

En lugar de usar `include` y `require` resulta conveniente utilizar `include_once` y `require_once` en lugar de las primeras.

Estas funciones comprueban si ya se ha importado el fichero previamente y no vuelven a recargar todas las funciones, evitando duplicados, redefiniciones de métodos o asignación de variables

2. Tipos de datos compuestos

Un programa utiliza más tipos que los simples que hemos visto, por lo que debemos conocer los tipos complejos como arrays y objetos

Los arrays tenemos de dos tipos:

- Numéricos
- Asociativos

```
// array numérico
$modulos1 = array(0 => "Programación", 1 => "Bases de
datos", ..., 9 => "Desarrollo web en
entorno servidor");
// array asociativo
$modulos2 = array("PR" => "Programación", "BD" => "Bases
de datos", ..., "DWES" => "Desarrollo
web en entorno servidor");
```

2. Tipos de datos compuestos

Para hacer referencia a cada uno de los elementos

```
//Array númerico
$modulos1 [9]
//Array asociativo
$modulos2 ["DWES"]
```

No es necesario especificar el tamaño del array

No estamos obligados a especificar todos los elementos del array

```
$modulo[1]= "Programación"
$modulo[100]= "Desarrollo Web Entorno Servidor"
```

2. Tipos de datos compuestos

Tampoco estamos obligados a indicar la clave, en ese caso se irá llenando el array desde la posición 0 hasta la siguiente posición numérica disponible

```
$modulos1 [ ] = "Programación";
$modulos1 [ ] = "Bases de datos";
...
$modulos1 [ ] = "Desarrollo web en entorno servidor";
```

Recorrer arrays

```
foreach ($modulos as $clave => $valor) {
print "El código del módulo ".$valor." es ".$clave."<br
/>"
}
```

Ejercicio

Haz una página PHP que utilice foreach para mostrar todos los valores del array $_SERVER en una **tabla con dos columnas**.

La primera columna debe contener el nombre de la variable, y la segunda su valor.

2. Tipos de datos compuestos

Array bidimensional

```
$ciclos = array(
   "DAW" => array ("PR" => "Programación", "BD" => "Bases
de datos", ..., "DWES" => "Desarrollo web en entorno
servidor"),
   "DAM" => array ("PR" => "Programación", "BD" => "Bases
de datos", ..., "PMDM" => "Programación multimedia y de
dispositivos móviles"));
```

Acceso

```
$ciclos ["DAW"]["DWES"]
```

Arrays creación de otra forma

```
$heroes = [
    ['Spiderman',   'Peter Parker', 'Nueva York'],
    ['Flash',    'Barry Allen', 'Ciudad Central'],
    ['Superman', 'Clark Kent', 'Krypton'],
       ['Hulk', 'Bruce Banner', 'Toronto'],
       ['Batman', 'Bruce Wayne', 'Gothan'],
       ['Deadpool', 'Wade Wilson', 'Nueva York'],
];

foreach ($heroes as [$heroe, $nombre, $ciudad]) {
    echo nl2br("$heroe nombre $nombre localidad $ciudad
\n");
}
//La función n12br("Texto") inserta una nueva línea
```

2. Tipos de datos compuestos

Funciones relacionadas con arrays:

`is_array($obj)` comprueba si una variable es un array

`count($array)` número de elementos de un array

`in_array($busqueda, $array)` comprueba si el elemento existe en el array

`array_search($busqueda,$array)` devuelve el primer índice del elemento buscado

`array_key_exists($clave, $array)` comprueba si la clave existe en el array

https://www.php.net/manual/es/function.in-array.php

2. Tipos de datos compuestos

Funciones relacionadas con arrays(bis):

`array_keys($arr)` Devuelve las claves del array en un array numérico

`array_column($arr)` Devuelve una columna del array

```
$notas = array(
    array(
        'DWES' => 7,
        'DWEC' => 8,
        'EIE' => 6,
    ),
    array(
        'DWES' => 5,
        'DWEC' => 6,
        'EIE' => 4,
    ));
```

```
echo $notas[1]['DWEC']; // Muestra 6

$notasDWES=array_column($notas,'DWEC'];

/* $notasDWES es un array numérico de tamaño 2
sería equivalente a $notasDWES=array(7,5); */
```

2. Tipos de datos compuestos

Funciones relacionadas con arrays(bis):

`explode($caracter,$cadena)` Devuelve un array a partir de una cadena utilizando el carácter como separador

`implode($caracter,$array)` Devuelve una cadena a partir de un array utilizando el carácter como separador

2. Tipos de datos compuestos (Variables Globales)

Variables reservadas
https://www.php.net/manual/es/reserved.variables.server.php

`$_SERVER['PHP_SELF']` guión que se está ejecutando actualmente.

`$_SERVER['SERVER_ADDR']` dirección IP del servidor web.

`$_SERVER['SERVER_NAME']` nombre del servidor web.

`$_SERVER['DOCUMENT_ROOT']` directorio raíz bajo el que se ejecuta el guión actual.

`$_SERVER['REMOTE_ADDR']` dirección IP desde la que el usuario está viendo la página.

`$_SERVER['REQUEST_METHOD']` método utilizado para acceder a la página ('GET', 'HEAD', 'POST' o 'PUT')

2. Tipos de datos compuestos (Variables Globales)

Variables gloables
https://www.php.net/manual/es/language.variables.superglobals.php

`$_GET, $_POST` y `$_COOKIE` contienen y 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.

2. Tipos de datos compuestos

PHP es un lenguaje no tipado, pero podemos revisar el tipo de datos de las variables

`gettype($var)` es una función que nos devuelve el tipo de datos con el que estamos trabajando

`array, boolean, double, integer, object, string, null, resource o unknown type`

Funciones que validan si una variable es del tipo indicado

`is_array(), is_bool(), is_float(),is_integer(), is_null(), is_numeric(), is_object(), is_resource(), is_scalar() e is_string().`

2. Tipos de datos compuestos

`isset($var)` comprueba si una variable está definida

`unset($var)` destruye una variable

`empty($var)` comprueba una variable si está definida y su valor es vacio

- "" (una cadena vacía)
- 0 (0 como un integer)
- 0 (0 como un float)
- "0" (0 como un string)
- NULL
- FALSE
- array() (un array vacío)
- $var; (una variable declarada, pero sin un valor)

2. Tipos de datos compuestos

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

Caracter	Resultado
d	día del mes con dos dígitos.
z	día del año, comenzando por el cero (0 = 1 de enero).
N	día de la semana (1 = lunes, ..., 7 = domingo).
w	día de la semana (0 = domingo, ..., 6 = sábado).
l	texto del día de la semana, en inglés (Monday, ..., Sunday).
m	número del mes con dos dígitos.
Y	número del año

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

2. Tipos de datos compuestos

Ejercicio muestra la fecha actual con el siguiente formato

Lunes, 2 de Octubre de 2023

Crea un formulario, que pida una fecha al usuario, compruebe que es correcta y lo muestre con el formato anterior

Revisa las funciones checkdate y mktime

2. Tipos de datos compuestos

Resulta conveniente mostrar todo el contenido de una variable, cuando no conocemos todos sus detalles o índices para empezar a trabajar con ella

Nos podemos apoyar en las funciones var_dump y print_r

```php
<?php

$a = array(1, 2, array("a", "b", "c"));

var_dump($a);

?>
```

Ficheros

Abrimos el fichero con diferentes modos:

- r - lectura
- w - Escritura, sobreescribe el contenido
- a - Añade contenido al final
- r+ - Lectura y escritura, el puntero del fichero está al principio del fichero
- a+ - Lectura y escritura, el puntero del fichero está al final del fichero

```
$fp = fopen("miarchivo.txt", "w"); //Abre el fichero

rewind($fp) //El puntero del fichero se mueve al principio
```

Ficheros

```
/*Lectura del fichero*/

$fp = fopen("resultado.txt",
"r");

while (!feof($fp)){

    $linea = fgets($fp);

    echo $linea;

}

fclose($fp);
```

```
$fp = fopen("miarchivo.txt", "w");

fwrite($fp, "Hola que tal");

fwrite($fp, ", ¿Cómo estás? \n");

fclose($fp);

/*Sobrescribimos el contenido del fichero cada
vez que lo ejecutamos*/

if ($fp = fopen("miarchivo.txt", "a")){

    fwrite($fp, "Yo muy bien, ¿Y tú?");

}

fclose($fp);
```