

1. FUNCIONES INTEGRADAS Y DE USUARIO.

Una función se define con la siguiente sintaxis:

```
function foo ($arg_1, $arg_2, ..., $arg_n) {  
    echo "Función de ejemplo.\n";  
    return $retval;  
}
```

Cualquier instrucción válida de PHP puede aparecer en el cuerpo de la función, incluso otras funciones y definiciones de clases

PHP no soporta la sobrecarga de funciones, y tampoco es posible redefinir u ocultar funciones previamente declaradas.

a. Parámetros de las funciones

La información puede suministrarse a las funciones mediante la lista de parámetros, una lista de variables y/o constantes separadas por comas. PHP soporta pasar parámetros por valor (el comportamiento por defecto), por referencia, y parámetros por defecto.

b. Pasar parámetros por referencia

Por defecto, los parámetros de una función se pasan por valor (de manera que si cambias el valor del argumento dentro de la función, no se ve modificado fuera de ella). Si deseas permitir a una función modificar sus parámetros, debes pasarlos por referencia.

Si quieres que un parámetro de una función siempre se pase por referencia, puedes anteponer un ampersand (&) al nombre del parámetro en la definición de la función:

```
function add_some_extra(&$string) {  
    $string .= ' y algo más.';  
}  
  
$str = 'Esto es una cadena, ';  
add_some_extra($str);  
echo $str; // Saca 'Esto es una cadena, y algo más.'
```

Si deseas pasar una variable por referencia a una función que no toma el parámetro por referencia por defecto, puedes anteponer un ampersand al nombre del parámetro en la llamada a la función:

```
function foo ($bar) {  
    $bar .= ' y algo más.';  
}  
  
$str = 'Esto es una cadena, ';  
foo (&$str);
```

```
echo $str; // Saca 'Esto es una cadena, '  
foo (&$str);  
  
echo $str; // Saca 'Esto es una cadena, y algo más.'
```

c. Parámetros por defecto

Una función puede definir valores por defecto para los parámetros escalares estilo C++:

```
function makecoffee ($type = "cappucino") {  
    return "Hacer una taza de $type.\n";  
}  
echo makecoffee ();  
echo makecoffee ("espresso");
```

La salida del fragmento anterior es:

```
Hacer una taza de cappucino.  
Hacer una taza de espresso.
```

d. Devolver valores

Los valores se retornan usando la instrucción opcional return. Puede devolverse cualquier tipo de valor, incluyendo listas y objetos.

```
function square ($num) {  
    return $num * $num;  
}  
  
echo square (4); // saca '16'.
```

No puedes devolver múltiples valores desde una función, pero un efecto similar se puede conseguir devolviendo una lista.

```
function small_numbers() {  
    return array (0, 1, 2);  
}  
  
list ($zero, $one, $two) = small_numbers();
```

A continuación un ejemplo de la función doble, que recibe un número y nos devuelve su doble

```
<?  
function doble($numero)  
{  
    return ($numero*2);  
}
```

```
$b=7;

$c=doble($b);

echo "el numero $b, y su doble $c";
?>
```

e. Include

Este tipo de ficheros pueden contener: texto, etiquetas HTML y funciones.

Si no contiene funciones se podrá insertar tantas veces como se invoque y se insertará, además, todo su contenido tal como puedes ver en el ejemplo.

Si el fichero contiene funciones solo podrá ser invocado una vez ya que si se hiciera una segunda llamada se produciría un error por duplicidad en los nombres de las funciones

Como verás en el ejemplo, es posible incluir cualquier tipo de funciones. Vamos a ver en primer lugar el fichero con las funciones, y en segundo lugar su utilización en una página PHP. Para diferenciar los ficheros a incluir del resto de páginas php se puede optar por añadir antes de la extensión .php las iniciales .inc (de fichero para incluir), no es obligatorio, pero sí aconsejable.

```
<?
/* Fichero funciones.inc.php */

function doble($numero) {
return ($numero*2); }

function mitad($numero) {
return (int)($numero/2); }
?>

<?
/* fichero prueba.php */
include("funciones.inc.php"); # utilizamos include al principio del
fichero

$a=6;
echo $a;
echo "<br>";
echo doble($a);
echo "<br>";
echo mitad($a);
?>
```

Definidas las funciones en el fichero a incluir y colocado al comienzo de la página un script que contenga el include y la ruta de ese fichero, bastará con invocar

cualquiera de las funciones, en cualquier punto del documento, para que esa llamada sea sustituida por el resultado de la ejecución de la función.

Como es lógico, solo serán visualizados en el navegador del cliente los resultados de la ejecución de las funciones que hayan sido invocadas.

f. Require

Tiene la misma sintaxis que include y una funcionalidad similar, aunque con algunas diferencias.

Igual que ocurría con aquél, cuando un fichero es invocado por require esa llamada lo que hace es sustituirse a sí misma por el contenido del fichero especificado.

A diferencia de include, la etiqueta require lee y escribe –en el documento a partir del que es invocada– el archivo referenciado completo y no acepta condicionales que sí son aceptados por incluye

g. include_once() y require_once()

Tanto en el caso de usar la instrucción include como con require, si se intenta incluir dos o más veces un fichero que contenga funciones, se producirá un error (PHP no permite que dos funciones tengan el mismo nombre) y se interrumpirá la ejecución del script. Los errores de ese tipo pueden evitarse usando las funciones:

```
include_once("fichero")  
require_once("fichero")
```

Que a diferencia de incluye() y require() van a impedir que un mismo fichero pueda incluirse dos veces.