

Relación3:Funciones.

Para resolver estos ejercicios se creará la carpeta /aplicación/relacion3. En ella se colocarán los ficheros librería.php e index.php. Dentro de librería.php se colocarán todas las funciones pedidas. Librería.PHP es una API y como tal no tiene interfaz (ninguna función puede tener sentencias que generen salida como echo). En index.php se realizarán llamadas a las funciones creadas mostrando resultados (hacedlo directamente en la vista).

Se seguirán todas las normas definidas en las relaciones anteriores (modelo/vista/controlador, comentarios, barra de ubicación, enlace en el menú principal, etc).

NOTA: Cuando se crea una librería:

- Las funciones que realizan operaciones no muestran texto ni devuelven cadena con el texto formateado, es la aplicación principal quien se encarga de mostrar los resultados.
- Se comprueban argumentos. En caso de error se devuelve false. En la aplicación principal se comprueba si la llamada a una función es correcta o errónea.
- Se documentan las funciones indicando parámetros y operación: usar /**. Además se indican comentarios en todo lo que se deba aclarar de su funcionamiento.
- En todas las funciones se le indicará el tipo de los parámetros/valor de retorno.
- Cuando la función pueda devolver error, se hará la llamada comprobando si se ha producido o no el error.

Las funciones que se piden son:

1.- Función cuentaVeces(). A esta función se le pasan cuatro parámetros: un array, una cadena y dos números. La función se encargará de añadir en el array el primer número a la posición asociativa indicada por la cadena. Hay que tener en cuenta que el array debe quedar actualizado cuando se salga de la función. Además, cada vez que se llama a la función, se devuelve en el segundo número un valor que corresponde con el número de llamada actual (1 para la primera, 2 para la segunda llamada, etc). Se devolverá true si todo ha ido bien y false si se ha detectado algún error. (se debe dar error si se indica como posición "2daw" o "primera")

```
$vector=array();//el vector vacio
```

```
$numero=0;
```

```
cuentaVeces($vector,"1osición",7,$numero); // $vector ("1osición"=>7). En numero se devuelve 1 porque es la primera llamada...
```

```
cuentaVeces($vector,"otra",2,$numero);// $vector ("otra"=>9). En numero se devuelve 2 porque es la segunda llamada...
```

2.- Función generarCadena(). A esta función se le pasa un parámetro con valor por defecto 10 y se encarga de devolver una cadena de longitud la indicada cuyos caracteres son generados de forma aleatoria (caracteres válidos: letras mayúsculas, minúsculas y números). Devolverá false si se ha indicado un número negativo o igual a 0.

generarCadena(7); //crea la cadena de 7 caracteres "AcG_%15"

3.- Función operaciones(). Esta función recibe al menos tres parámetros. El primero (entero) es el tipo de operación. Los demás son valores numéricos (los operandos). El tipo de operación puede ser

- 1, suma todos los operandos
- 2, al primer operando se restan el resto
- 3, multiplicación de todos los operandos.

En cualquier otro caso se suman los operandos que ocupan las posiciones pares y se restan los de las posiciones impares (el primer operando es 1).

Operaciones(1,2,5,7,3,4); //haría la suma de 2+5+7+3+4

Operaciones(5,2,5,7,3,4); //Sumaria suma las posiciones pares (5+3), las posiciones impares (2+7+4), y se restan los dos.

4.- Función devuelve(). Esta función recibe como parámetros tres números (enteros) y devuelve la suma de los tres valores en el primer parámetro y como resultado de la función el producto de los tres valores. Los dos últimos parámetros tendrán definidos valor por defecto (el que queráis).

\$valor=7;

*devuelve(\$valor,4) // \$valor valdrá 7+4+10 y se devuelve 7*4*10 (10 es el valor por defecto que le he puesto al tercer parámetro.*

Hacer varias llamadas a la función indicando:

- Los tres parámetros
- El primer parámetro únicamente
- El primer y el segundo parámetro
- El primer y el tercer parámetro (ojo, el segundo parámetro no se indica)

5.- Se tienen las funciones suma, resta y multiplicacion que reciben cada una dos parámetros numéricos (entero) y que devuelven el resultado de la operación que indica su nombre sobre los parámetros.

Tenemos la Función hacerOperacion que recibe tres parámetros. El primero es una cadena que puede ser suma, resta o multiplicacion. Los otros dos son los operandos. Esta función se encargará de devolver el resultado de ejecutar la "cadena" con los operandos usando una variable-función para llamar a la operación. Hay que comprobar indicando una operación correcta (devuelve false si la operación no es válida).

hacerOperacion("resta",3,1) //devuelve 3-1. No se llama directamente a la función resta, se tiene una variable función para ello

6.- Crear la función llamadaAFuncion que recibe tres operandos (entero, entero y callable) y que devuelve un entero que es el resultado de llamar al parámetro callback con los otros parámetros.

Llamar a la función llamadaAFuncion 3 veces pasando como parámetro de callback funciones de

flecha distintas que tu definas

```
llamadaAFuncion(1,2,fn($a,$b)=>$a+$b)
```

Repetir las llamadas, definiendo las 3 funciones como anónimas (se asignan a variables)

7.- Función ordenar(): A esta función se le pasa un array y lo ordena mediante una llamada a usort, al que se indicará como función callback una función anónima que ordenará en orden descendente según el número de caracteres de cada elemento.

```
$vector=array(("uno", "grande", "camino", "a"))
```

```
Ordenar($vector); //el vector debe quedar ("camino", "grande", "uno", "a")
```