

TEMA 2

INTRODUCCIÓN A PHP

E9_Bibliotecas

Ejercicios

BIBLIOTECAS DE FUNCIONES

1. Se pretende crear una librería que contenga funciones que realizan operaciones con el contenido de un array que se le pasa como argumento.

El archivo se llamará E9_libreriaListarArray.php y contendrá funciones, algunas que ya tengamos codificadas y otras, nuevas.

En concreto, requiere:

- Función *listarVectorTabla*

Recibe un array y visualiza su contenido en forma de tabla

Visualiza en forma de tabla:

ELEMENTO	VALOR
0	10
1	20
2	30
3	40

- Función *listarVectorNoOrdenada*,

Recibe un array y visualiza su contenido en forma de lista no ordenada

En forma de lista no ordenada:

- 10
- 20
- 30
- 40

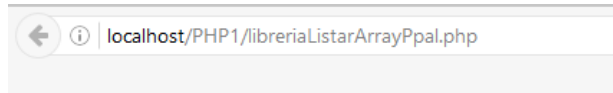
Nota: si no disponemos de los archivos con las funciones indicadas, habrá que crearlos previamente a la realización de este ejercicio.

2. Para hacer operativa la librería anterior, crearemos el archivo E9_libreriaListarArrayPpal.php

Se encargará de hacer:

- usar include("nombre_archivo")
- crear un array con valores numéricos
- invocar a cada una de las funciones de visualización de array que tenemos.
- invocar a cada una de las funciones implementadas en el ejercicio anterior.

Salida: Para un array de componentes 10, 20, 30 y 40



Invocamos a la función listarVectorTabla

Visualiza en forma de tabla:

ELEMENTO	VALOR
0	10
1	20
2	30
3	40

Invocamos a la función listarVectorNoOrdenada

En forma de lista no ordenada:

- 10
- 20
- 30
- 40

3. Crear archivo E9_libreriaArrays.php

Incluirá funciones que teníamos definidas en el tema de arrays.

En concreto:

- Función `vectoresUnordList`, que corresponderá al ejercicio que implementamos en el archivo *E5_vectoresUnordList.php*
- Función `vectoresUnordListRange`, que corresponderá al ejercicio que implementamos en el archivo *E5_vectoresUnordListRange.php*
- Función `vectoresWhile`, que corresponderá al ejercicio que implementamos en el archivo *E5_vectoresWhile.php*
- Función `vectoresInverso`, que corresponderá al ejercicio que implementamos en el archivo *E5_vectoresInverso.php*
- Función `vectoresIforeach`, que corresponderá al ejercicio que implementamos en el archivo *E5_vectoresForeach.php*

Crear el archivo `E9_libreriaArraysPpal.php`, que será el encargado de:

Definir el array de datos para cada una de las funciones

Invocar a las funciones definidas en la biblioteca de funciones en secuencia, siguiendo el orden en el que se han ido codificando en el archivo librería

4. Crear archivo E9_libreriaArraySuma.php

Incluirá funciones que teníamos definidas en el tema de arrays.

En concreto:

- Función arraySuma, que corresponderá al ejercicio que inplementamos en el archivo E5_arraySuma.php.php
- Función arraySumaTabla, que corresponderá al ejercicio que inplementamos en el archivo E5_arraySumaTabla.php

Crear el archivo E9_libreriaArraySumaPpal.php, que será el encargado de:

Definir el array de datos para cada una de las funciones

Invocar a las funciones definidas en la biblioteca de funciones en secuencia, siguiendo el orden en el que se han ido codificando en el archivo librería

5. Crear archivo E9_libreriaMediaAritmetica.php

Pretendemos que incluya todas las funciones de cálculo de media que tenemos codificadas hasta el momento.

- *mediaValores*, se encuentra en *funcionMediaAritmeticaVariosValores.php*.

Recibe cualquier cantidad de argumentos numéricos y calcula su media aritmética

Función MediaValores

=====

Número de argumentos utilizados: 3

Parámetro 0 ==> valor: 10

Parámetro 1 ==> valor: 20

Parámetro 2 ==> valor: 30

Valor de la media: 20

- *mediaArray*, se encuentra en *funcionMediaAritmeticaVector.php*.

Recibe un array unidimensional como argumento y devuelve la media de sus valores. El array recibido puede tener cualquier número de enteros

Función MediaArray

=====

Número de elementos del array recibido: 4

El contenido del array recibido es:

Array ([0] => 12 [1] => 13 [2] => 15 [3] => 16)

La media de dichos argumentos es: 14.00

- *mediaValoresRefer* se encuentra en *funcionMediaAritmeticaReferencia.php*.

Recibe dos argumentos, uno de entrada y otro de salida. Calcula su media aritmética y devuelve el resultado en el segundo argumento

Función MediaValoresRefer

=====

Valor de los argumentos utilizados:

25 y 45

No hace falta hacer Return

Damos formato al valor de la media mediante printf

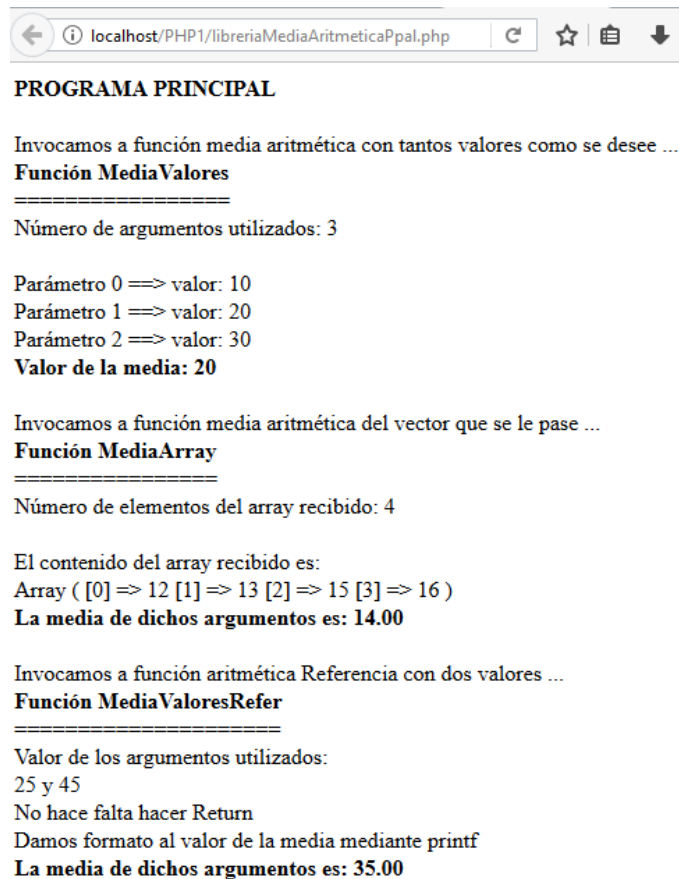
La media de dichos argumentos es: 35.00

6. Crear programa principal correspondiente al ejercicio anterior. Se llamará E9_libreriaMediaAritmeticaPpal.php

Se encarga de :

- Incluir la librería de funciones mediana (E9_libreriaMediaAritmetica.php). Usar INCLUDE
- Hacer llamamiento a cada una de estas funciones con valores que se generan en él.

Salida:



```

PROGRAMA PRINCIPAL

Invocamos a función media aritmética con tantos valores como se desee ...
Función MediaValores
=====
Número de argumentos utilizados: 3

Parámetro 0 ==> valor: 10
Parámetro 1 ==> valor: 20
Parámetro 2 ==> valor: 30
Valor de la media: 20

Invocamos a función media aritmética del vector que se le pase ...
Función MediaArray
=====
Número de elementos del array recibido: 4

El contenido del array recibido es:
Array ( [0] => 12 [1] => 13 [2] => 15 [3] => 16 )
La media de dichos argumentos es: 14.00

Invocamos a función aritmética Referencia con dos valores ...
Función MediaValoresRefer
=====
Valor de los argumentos utilizados:
25 y 45
No hace falta hacer Return
Damos formato al valor de la media mediante printf
La media de dichos argumentos es: 35.00

```


7. Crear archivo llamado E9_Funcionproducto.php

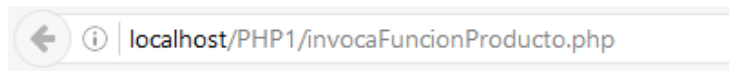
Contendrá una función, denominada producto, que recibirá los argumentos multiplicando y multiplicador y devolverá su producto.

Ahora creamos otro archivo llamado E9_invocaFuncionProducto.php, que contendrá el programa principal,

Se encargará de hacer:

- asignar los dos valores a las variables
- usar la instrucción include(nombre_archivo), donde nombre_archivo sea aquel en el que se encuentra codificada la función producto.
- invocar a la función anterior.

Salida:



Asignamos valores a las variables:

multiplicando :10

multiplicador :20

Invocamos a la función

Se encuentra en el archivo: **funcionProducto.php**

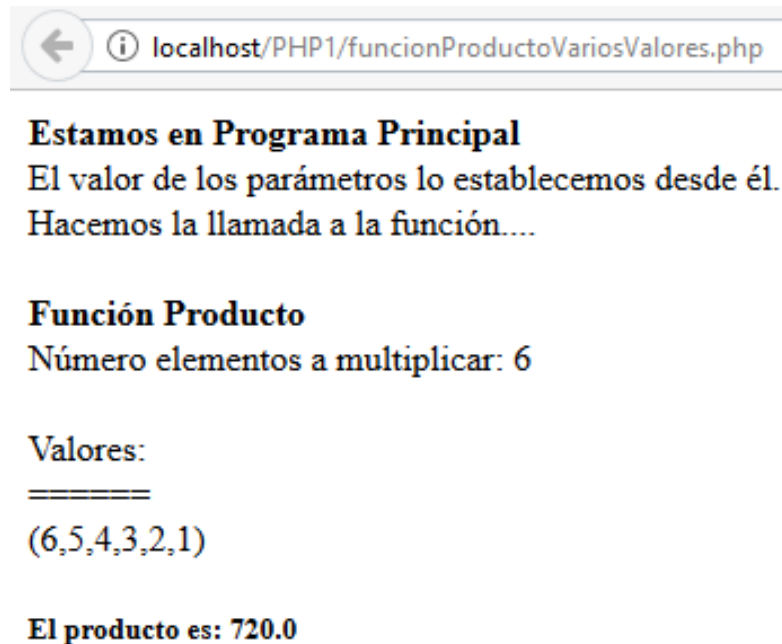
El resultado de 10 X 20 es **200**

8. Crea un archivo llamado E9_funcionProductoVariosValores.php.

Contendrá una función, denominada productoValores, que recibirá tantos argumentos como desee el usuario.

Visualizará como resultado el producto de todos ellos.

Salida:



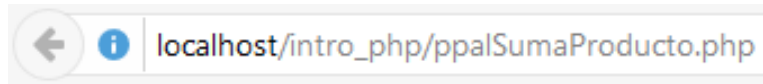
9. Crear un archivo E9_funcionSumaProducto.php que contenga la definición de una función Suma y una función Producto.

Ambas funciones reciben dos parámetros y realizan la operación correspondiente.

Para probar su funcionamiento se creará otro archivo, de nombre E9_ppalSumaProducto.php que será quien asigne valores a los datos y realice el llamamiento a ambas funciones.

Usaremos include para que la definición de ambas funciones sea accesible desde el programa principal

Ejecución similar a:



Asignamos valores a los dos valores

`$a= 10`

`$b= 20`

Llamada a Funcion Suma

La suma de 10 y 20 es 30

Llamada a Funcion Producto

El producto de 10 y 20 es 200

10. Crear archivo E9_libreriaProductos.php

Pretendemos que incluya todas las funciones de cálculo de producto que tenemos codificadas hasta el momento.

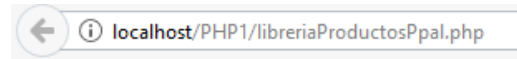
- Producto, se encuentra en funcionProducto.php.
Realiza el producto de dos valores que se le pasan como argumentos
- ProductoValores, se encuentra en funcionProductoVariosValores.php.
Recibe cualquier cantidad de argumentos numéricos y calcula su producto

11. Crear archivo E9_libreriaProductosPpal.php

Se encargará de:

- Incluir la librería que hemos definido en el archivo E9_libreriaProductos.php
- Establecer el valor de las variables a multiplicar
- Realizar la invocación a las funciones producto y productoValores que se encuentran en esta librería

Salida:



Estamos en Programa Principal

Producto

Producto de 12 y 10=**120**

Función Producto Varios

Número elementos a multiplicar: 6

Valores:(6,5,4,3,2,1)

El producto es: 720.0