6. Funciones

6.1 Sintaxis.

Observa la siguientes funciones en PHP:

```php
<?php


// Genera un array de números aleatorios, de tamaño
// $numeroElementos, con un valor entre $min y $max

function arrayAleatorio($numeroElementos, $min, $max){
    for ($i=0; $i<$numeroElementos; $i++){
        $array[$i]=rand($min, $max);
    }
    return $array;
}

// Dado un array, devuelve otro array solo con los
// elementos pares del array introducido

function pares($arrayElementos){
    $i= 0;
    foreach($arrayElementos as $a){
        if ($a%2 ==0){
            $pares[$i]=$a;
            $i++;
        }
    }

    return $pares;


}

// Genera un array de 20 elementos y muestra los números pares
$array = arrayAleatorio(20, 1, 100);
$arrayPares = pares($array);

foreach($arrayPares as $a){
    echo $a."<br>";
}



?>
```

6.2 Organizando el código.

No debemos de programar las funciones donde exista aspecto gráfico debemos de hacerlo en archivos php separados.

Ejemplo: funcionesArrays.php

```php
<?php


// Genera un array de números aleatorios, de tamaño
// $numeroElementos, con un valor entre $min y $max

function arrayAleatorio($numeroElementos, $min, $max){
    for ($i=0; $i<$numeroElementos; $i++){
        $array[$i]=rand($min, $max);
    }
    return $array;
}

// Dado un array, devuelve otro array solo con los
// elementos pares del array introducido

function pares($arrayElementos){
    $i= 0;
    foreach($arrayElementos as $a){
        if ($a%2 ==0){
            $pares[$i]=$a;
            $i++;
        }
    }

    return $pares;


}

?>
```

Página principal:

```php
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>
<?php

include_once('funcionesArrays.php');

$array = arrayAleatorio(20, 1, 100);
$arrayPares = pares($array);

foreach ($arrayPares as $a) {
    echo $a."<br>";
}

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

A direrencia de ***include***, que básicamente copia y pega el archivo .php incluído, ***include_once*** sólo puede ser llamado una vez en la página, con lo que es la forma habitual de importar un .php con funciones.

Ejercicio 6.1

Convierte en función el ejercicio 8.5, que devuelva el signo zodiacal en función de una fecha.

Ejercicio 6.2

Convierte en función el ejercicio 8.4, haciendo que la función devuelva el resultado de una ecuación de segundo grado.

Ejercicio 6.3

Convierte en función el ejercicio 9.12, haciendo que la función nos devuelva si un entero es primo o no como valor booleano.

Ejercicio 6.4

Convierte en función el ejercicio 9.20, devolviendo un valor booleano diciendo si el número es capicúa o no.

Ejercicio 6.5

Convierte en función el ejercicio 9.21, devolviendo el número con el dígito insertado.

Ejercicio 6.6

Realiza una función que pinte la letra L por pantalla hecha con asteriscos. A la función se le pasará la altura como parámetro. El palo horizontal de la L tendrá una longitud de la mitad (división entera entre 2) de la altura más uno. (Ejercicio 9.17)

Ejercicio 6.7

Crea una biblioteca de funciones matemáticas que contenga las siguientes funciones. Recuerda que puedes usar unas dentro de otras si es necesario.

Observa bien lo que hace cada función ya que, si las implementas en el orden adecuado, te puedes ahorrar mucho trabajo. Por ejemplo, la función esCapicua resulta trivial teniendo voltea y la función siguientePrimo también es muy fácil de implementar teniendo esPrimo. Se permite hacer “trampa” con las funciones de strings.

1. esCapicua: Devuelve verdadero si el número que se pasa como paráme#tro es capicúa y falso en caso contrario.

2. esPrimo: Devuelve verdadero si el número que se pasa como parámetro es primo y falso en caso contrario.

3. siguientePrimo: Devuelve el menor primo que es mayor al número que se pasa como parámetro.

4. potencia: Dada una base y un exponente devuelve la potencia.

5. digitos: Cuenta el número de dígitos de un número entero.

6. voltea: Le da la vuelta a un número.

7. digitoN: Devuelve el dígito que está en la posición n de un número entero. Se empieza contando por el 0 y de izquierda a derecha.

8. posicionDeDigito: Da la posición de la primera ocurrencia de un dígito dentro de un número entero. Si no se encuentra, devuelve -1.

9. quitaPorDetras: Le quita a un número n dígitos por detrás (por la derecha).

10. quitaPorDelante: Le quita a un número n dígitos por delante (por la izquierda).

11. pegaPorDetras: Añade un dígito a un número por detrás.

12. pegaPorDelante: Añade un dígito a un número por delante.

13. trozoDeNumero: Toma como parámetros las posiciones inicial y final

dentro de un número y devuelve el trozo correspondiente.

14. juntaNumeros: Pega dos números para formar uno

Ejercicio 6.8

Muestra los números primos que hay entre 1 y 1000.

Ejercicio 6.9

Muestra los números capicúa que hay entre 1 y 99999.

Ejercicio 6.10

Crea una biblioteca de funciones para arrays (de una dimensión) de números enteros que contenga las siguientes funciones:

1. generaArrayInt: Genera un array de tamaño n con números aleatorios cuyo intervalo (mínimo y máximo) se indica como parámetro.

2. minimoArrayInt: Devuelve el mínimo del array que se pasa como parámetro.

3. maximoArrayInt: Devuelve el máximo del array que se pasa como parámetro.

4. mediaArrayInt: Devuelve la media del array que se pasa como parámetro.

5. estaEnArrayInt: Dice si un número está o no dentro de un array.

6. posicionEnArray: Busca un número en un array y devuelve la posición (el índice) en la que se encuentra.

7. volteaArrayInt: Le da la vuelta a un array.

8. rotaDerechaArrayInt: Rota n posiciones a la derecha los números de un array.

9. rotaIzquierdaArrayInt: Rota n posiciones a la izquierda los números de un array.

Ejercicio 6.11

Crea una biblioteca de funciones para arrays bidimensionales (de dos dimensiones) de números enteros que contenga las siguientes funciones:

1. generaArrayBiInt: Genera un array de tamaño n x m con números aleatorios cuyo intervalo (mínimo y máximo) se indica como parámetro.

2. filaDeArrayBiInt: Devuelve la fila i-ésima del array que se pasa como parámetro.

3. columnaDeArrayBiInt: Devuelve la columna j-ésima del array que se pasa como parámetro.

4. coordenadasEnArrayBiInt: Devuelve la fila y la columna (en un array con dos elementos) de la primera ocurrencia de un número dentro de un array bidimensional. Si el número no se encuentra en el array, la función devuelve el array {-1, -1}.

5. diagonal: Devuelve un array que contiene una de las diagonales del array bidimensional que se pasa como parámetro. Se pasan como parámetros fila, columna y dirección. La fila y la columna determinan el número que marcará las dos posibles diagonales dentro del array. La dirección es un booleano. Si es true indica que se elige la diagonal que va del noroeste hacia el sureste, mientras que si es false indica que se elige la diagonal que va del noreste hacia el suroeste.