EJERCICIOS PHP
RELACIÓN III - FUNCIONES, RECURSIVIDAD Y USO AVANZADO DE ARRAYS

1- Haz una función PHP esPrimo($num) que devuelva un booleano que indique si el número pasado como parámetro es primo o no. Haz un programa que pida un número natural, e incluyendo esta función, la utilice para mostrar todos los números primos entre 1 y el introducido. Todo en el mismo archivo php

2- Retoma el ejercicio del cálculo del factorial, y conviértelo en una función. Haz otra versión de esta función, pero esta vez utiliza **recursividad**

3- Retoma el ejercicio del cálculo del MCD y de la división por los dos algoritmos de Euclides, y cambia la operativa por dos funciones en el mismo programa. Haz una **versión recursiva** de estas funciones

4- Introduce todas las funciones de los ejercicios del 1 al 3 de esta relación en una librería denominada **relacion3.php** y prueba a incluirla e invocarla desde una nueva versión de alguno de los ejercicios anteriores, anulando con comentarios la declaración

5- Vuelve al ejercicio 12 de la relación anterior, y añade la validación en JS del email. Pide también en el formulario qué tipo de documento de identidad tiene la persona y comprueba su validez:

- dni:consultar en la web del Ministerio del Interior
- nie: consultar, la misma dirección
- TIE: el número del NIE y un "número de soporte" que empieza por la letra E, seguido de 8 dígitos, que a veces se rellenan con ceros a la izquierda, por ejemplo, E00235566

6- Como todos los lenguajes de alto nivel, PHP tiene una librería matemática. Consultala en PHP Math Functions.

Haz un programa de PHP que, a partir de un número de tiradas elevado que se introducirá mediante formulario, simule los lanzamientos de un dado equiprobable y de otro trucado (el 6 debe ser 3 veces más probable que los demás resultados). Contará las frecuencias tanto del dado correcto como del dado trucado, y las visualizará. Todo en un mismo archivo PHP

7- Haz un script PHP que practique varias funciones de fecha y hora. Para ello, consulta la dirección: PHP: Fecha/Hora - Manual Explora así mismo la aritmética de date-time

Crea **tus propias funciones** que devuelvan, una el nombre del día de la semana en español y otra, el nombre del mes en español

8- Haz un programa PHP que pida un texto por formulario y las opciones de mostrarlo en mayúsculas y/o en minúsculas. Haz otra versión del formulario, para que sea una verdadera disyunción (o una u otra) y compruebe que se elija alguna de ellas validando con JS en ambos casos. Cada uno de ellos en un mismo archivo (uno para cada uno)

9- Haz un programa PHP que pida un texto por formulario y a continuación extraiga y muestre la palabra más larga. Todo en el mismo archivo

10- Haz un script que pida un texto como entrada y lo muestre con las palabras en orden inverso. Todo en un mismo archivo php

11- Haz un programa PHP que incluya una función swap(n1, n2) que intercambie los valores de los dos parámetros. Haz también una función que invierta el orden de los componentes de un array usando swap. Luego, pasa estas dos funciones a la librería **functionsRel3.php**.

12- Implementa en PHP una función que ordene los componentes de una array de strings en desorden utilizando el famoso algoritmo de "**ascenso de burbujas**" y haz un programa que la incluya. Vamos a utilizar el envío de parámetros por referencia: el resultado de la ordenación, se verá reflejado en el propio parámetro array. El array utilizado podría ser (por ejemplo) éste:

$datos = ['Pérez','García','López','Márquez','Álvarez','Domínguez','Ruíz','Díaz'];

Observa que, al no indicarse tipos de datos, también sirve la función para ordenar arrays de números. Pruébalo

13- PHP dispone de una amplia variedad de funciones de manejo de strings, como podrás comprobar en la documentación oficial: https://www.php.net/manual/es/ref.strings.php

Algunas nos vienen acompañando desde relaciones anteriores: echo, print, printf, trim, ord y chr. Otras no las hemos utilizado aún:

- strlen, strtolower, strtoupper, strcmp, stripos, stristr,…: son habituales en otros lenguajes también en manejo básico de cadenas
- crypt, md5 y sha1: las utilizaremos para encriptar passwords cuando las almacenemos en BBDD
- strip_tags, htmlentities, htmlspecialchars, htmlspecialchars_decode, nl2br: son funciones que trabajan con caracteres distinguiendo etiquetas html y php

Vamos a utilizar algunas de ellas para el siguiente programa. Pide una cadena de texto por formulario de entrada y muestra, en el interior de un alert de Bootstrap diferente, cada uno de los siguientes resultados:

- La misma cadena de caracteres, del revés, indicando además si es palíndroma o no
- La misma cadena de caracteres, con las palabras del revés

- La misma cadena de caracteres toda en mayúsculas y toda en minúsculas
- Un recuento de los caracteres (incluido los espacios en blanco) y de las palabras que contiene
- Como curiosidad, el resultado de aplicarle crypt, md5 y sha1

14- PHP permite el uso de funciones anónimas. Una función anónima, como es de esperar, no tiene un nombre asignado, en su lugar, se puede usar asignando su descripción a una variable, y a partir de ahí, invocarlas utilizando el nombre de la variable, y pasándole el/los parámetros necesarios, si es que los tuviera

Vamos a hacer un programa en el que declaremos una serie de funciones anónimas del primer tipo:

- $circunferencia = function ($n) { ….. } (calcula la longitud de una circunferencia)
- $circulo = function ($n) { …….} (calcula el área de un círculo)
- $esfera = function ($r) {…….} (calcula el volumen de una esfera)

Tendremos un formulario en el que habrá que introducir un valor de radio (positivo real) y a partir de ahí, calcularemos y mostraremos el resultado de la longitud, área y volumen. Haz ambas cosas en un único archivo (formulario y cálculo)

15- Investiga qué son las funciones arrow (flecha) en PHP e intenta redefinir las anteriores para que lo sean. Establece paralelismos entre switch y match (aunque en este caso, no son funciones sino estructuras de control, OJO)

16- Otro uso habitual de las funciones anónimas es utilizarlas como parámetros de otra función. Son las llamadas funciones callback. Es el caso de addEventListener utilizada en la relación anterior (pero OJO, eso era JavaScript, no PHP)

Estudia en la documentación oficial de PHP (https://www.php.net/manual/es/book.array.php)

las funciones para array que utilizan callbacks, por ejemplo:

array_all, array_any, array_filter, array_find, array_find_key, array_map, array_walk

Fíjate especialmente en el interfaz de las funciones (interfaz: nombre, parámetros y tipo de retorno)

Haz un pequeño programa PHP sin entrada de datos, con un array numérico definido con la función **range** (investigar) y un rango de valores entre 1 y 100. Después:

- Aplica array_all para comprobar si todos los números son positivos
- Aplica array_any para comprobar si hay algún múltiplo de 5
- Aplica array_filter para extraer los que sean primos

- Aplica array_find para que te de la primera ocurrencia de número de dos cifras idénticas
- Aplica array_map para obtener el cuadrado de cada valor
- Aplica array_walk para sustituir cada valor por su doble

Muestra cada resultado adecuadamente, en un alert de Bootstrap debidamente formateado (y coloreado en diferente color)

17- A continuación, estudia el resto de las funciones para array que se incluyen en esta dirección: https://www.php.net/manual/es/book.array.php

y estudia cómo funcionan al menos las siguientes:

array_combine, array_count_values, array_diff (y sus variantes), array_first (y array_last), array_flip, array_intersect ,arsort, rsort, shuffle, array_unique, array_values, array_search, array_reverse, array_slice, array_filter, array_pop, array_push, array_splice, compact, count, current, in_array, range, ….

Al igual que en el caso anterior, haz un pequeño programa PHP sin entrada de datos, con dos arrays numéricos: pares y multiplosDeTres . Introduce en uno de ellos los números impares entre 1 y 20, y en el otro, los múltiplos de 3 entre 1 y 40. Utiliza para ello **range y las funciones estudiadas en el ejercicio anterior**.

- Aplica array_count para comprobar cuántos pares tienen
- Aplica array_any para comprobar si hay algún múltiplo de 5
- Aplica array_filter para extraer los que sean primos
- Aplica array_find para que te de la primera ocurrencia de número de dos cifras idénticas
- Aplica array_map para obtener el cuadrado de cada valor
- Aplica array_walk para sustituir cada valor por su doble
- Aplica array_intersect para saber qué valores están en ambos arrays
-

Fíjate especialmente en los parámetros que tienen, de qué tipo, y el tipo o estructura devuelto (en algunos casos arrays, en otro booleanos, etc…)

18- Haz un programa php con una carta de platos:

```
$menu = ['entrante' => array('Ensalada César','Hummus','Boquerones al natural'),
        'primero' => array('Gazpachuelo','Salmorejo','Ajo Blanco'),
        'segundo' => array('Fritura Malagueña','Conejo al ajillo','Pisto con huevo'),
        'postre' => array('Helado 3 sabores','Flan','Tarta de Queso')];
```

Y un generador de n menús sugerencia. Los menús sugeridos, deberán aparecer formateados en una card de Bootstrap, en principio, sin imagen.

19- Haz otra versión en la que sea dos veces más probable que en la sugerencia aparezca la tercera opción de cada tipo de plato y además, aparezca una imagen del primer plato que haya salido al azar. Las imágenes estarán almacenadas en una subcarpeta img, y la url de las imágenes, estarán almacenadas en otro array asociativo de 3 valores (Gazpachuelo, Salmorejo, Ajo Blanco).

20- Vamos a utilizar controles de seguridad en los datos introducidos por formulario en una aplicación PHP. Para ello, vamos a utilizar las funciones:

- htmlspecialchars(...) : para evitar códigos malignos en las url al utilizarlas en el parámetro action del formulario
- la extensión Filter, para validar y "sanitizar" datos desde el lado servidor
- preg_match(...): para comprobar patrones complejos utilizando expresiones regulares

Vamos a actuar en dos líneas:

1. repasa **todos los ejercicios con formulario** hechos hasta el momento, y sustituye:

   action="loquesea.php" por

   action=<?php echo htmlspecialchars("loquesea.php") ?>

2. valida y sanitiza también en el lado servidor los datos del ejercicio 12 de la relación anterior. Lo haremos a partir de ahora siempre que tengamos un formulario de entrada cuyos datos vayan a ser almacenados/manipulados con una base de datos (cosa que ocurrira próximamente)

Lee los siguientes artículos y entenderás la necesidad de hacerlo:

- Evitar ataques XSS en PHP - Sutil Web
- Sanitización y Validación de Datos en Aplicaciones PHP: Guía Efectiva

Y documéntate en las siguientes urls:

- https://www.php.net/manual/es/filter.examples.validation.php
- https://www.w3schools.com/php/php_filter.asp
- How to use preg_match in PHP [Explained with examples] | CyberITHub
- Expresiones regulares - Regexp - Pattern matching