Ejercicios PHP 2

1.Contador de Palabras

Crea un programa que cuente el número de palabras en una cadena de texto ingresada por el usuario.

· Pedir al usuario que ingrese una frase.

· Inicializar un contador de palabras en 0.

· Recorrer cada carácter de la frase.

· Si el carácter es un espacio, incrementar el contador de palabras.

· Al finalizar, sumar 1 al contador de palabras (para contar la última palabra).

· Imprimir el número total de palabras.

<?php

error\_reporting(E\_ALL);

//error\_reporting(0);

$contador = 0;

$frase = readline("Introduzca la  cadena de texto: ");

for ($i = 0; $i < strlen($frase); $i++) { // Cuenta cuantos espacios hay en la cadena de texto

    switch($frase[$i]) {

        case " ": $contador++;

        break;

        default:

        break;

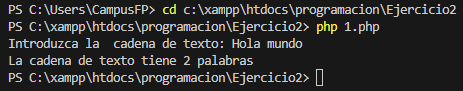
    }

}

$contador++;

echo ("La cadena de texto tiene $contador palabras") // Indica cuantas palabras hay en la cadena de texto

?>



2. Ordenamiento de array

Crea un programa que ordene un array de números de forma ascendente.

· Crear un array de números aleatorios.

· Utilizar un doble bucle para comparar cada elemento con los demás y realizar intercambios si están en el orden incorrecto (algoritmo de burbuja).

· Imprimir el array ordenado.

<?php

error\_reporting(E\_ALL);

//error\_reporting(0);

$array = array();

$numero\_valores = 5; // Indica la cantidad de valores que tendrá el array

for ($x = 0; $x < $numero\_valores; $x++) { // Introduce valores aleatorios en el array

    $numero\_aleatorio = rand(1,100);

    array\_push($array,$numero\_aleatorio);

}

echo("-Array desordenado")."\n"; // Muestra el array desordenado

for ($x = 0; $x < count($array); $x++) {

    echo $array[$x]."\n";

}

for ($y = 0; $y < count($array); $y++) { // Ordena el array

    for ($z = 0; $z < count($array); $z++) {

        if ($array[$z] > $array[$y]) { // Si uno de los números es mayor que otro intercambian sus posiciones

            $puente = $array[$z];

            $array[$z] = $array[$y];

            $array[$y] = $puente;

        }

    }

}

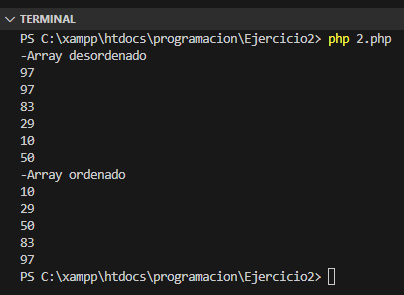
echo("-Array ordenado")."\n"; // Muestra el array ordenado

for ($x = 0; $x < count($array); $x++) {

    echo $array[$x]."\n";

}

?>



3. Validación de contraseña

Crea un programa que valide si una contraseña cumple con ciertos criterios.

Pedir al usuario que ingrese una contraseña.

Verificar si la contraseña tiene al menos 8 caracteres.

Verificar si la contraseña contiene al menos una letra mayúscula, una minúscula y un número.

Utilizar expresiones regulares para realizar estas verificaciones.

Si la contraseña cumple con todos los criterios, mostrar un mensaje de éxito.

<?php

error\_reporting(E\_ALL);

//error\_reporting(0);

$contador = 0;

$patron1 = "/[[:alnum:]]{8,}/";

$patron2 = "/[[:digit:]]/";

$patron3 = "/[a-z]/";

$patron4 = "/[A-Z]/";

$clave = readline("Introduzca la contraseña: ");

if (preg\_match($patron1, $clave)) { // Comprueba el número de caractéres de la contraseña

    echo("La contraseña tiene 8 o más caractéres")."\n";

    $contador++;

} else {

    echo("La contraseña tiene menos de 8 caractéres")."\n";

}

if (preg\_match($patron2, $clave)) { // Comprueba si lacontraseña incluye números

    echo("La contraseña contiene números")."\n";

    $contador++;

} else {

    echo("La contraseña no contiene números")."\n";

}

if (preg\_match($patron3, $clave)) { // Comprueba si lacontraseña incluye minúsculas

    echo("La contraseña contiene minúsculas")."\n";

    $contador++;

} else {

    echo("La contraseña no contiene minúsculas")."\n";

}

if (preg\_match($patron4, $clave)) { // Comprueba si lacontraseña incluye mayúsculas

    echo("La contraseña contiene mayúsculas")."\n";

    $contador++;

} else {

    echo("La contraseña no contiene mayúsculas")."\n";

}

if ($contador == 4) { // Indica si la contraseña cumple todas las condiciones

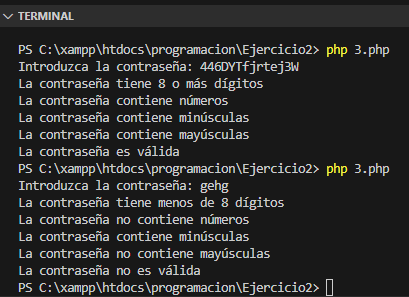
    echo ("La contraseña es válida");

} else {

    echo ("La contraseña no es válida");

}

?>



4. Generador de nombres aleatorios

Crea un programa que genere nombres aleatorios a partir de listas de nombres y apellidos.

Crear dos arrays, uno con nombres y otro con apellidos.

Generar un número aleatorio para seleccionar un nombre y otro para seleccionar un apellido.

Concatenar el nombre y el apellido para formar un nombre completo aleatorio.

<?php

error\_reporting(E\_ALL);

//error\_reporting(0);

$nombres = array( //Posibles nombres

    1 => "Juan ",

    2 => "Raúl ",

    3 => "María ",

    4 => "David ",

    5 =>"Luna ",

);

$apellidos = array( //Posibles apellidos

    1 => "Peréz ",

    2 => "Morales ",

    3 => "Casas ",

    4 => "García ",

    5 => "Guijo ",

);

$nombre\_random = rand(1, count($nombres)); //Elije uno de los nombres al azar

$apellido\_random = rand(1, count($apellidos)); //Elije uno de los apellidos al azar

echo("-Nombres: ")."\n";

for ($x = 1; $x <= count($nombres); $x++) { //Imprime los nombres

    echo $nombres[$x]."\n";

}

echo("-Apellidos: ")."\n";

for ($x = 1; $x <= count($apellidos); $x++) { //Imprime los apellidos

    echo $apellidos[$x]."\n";

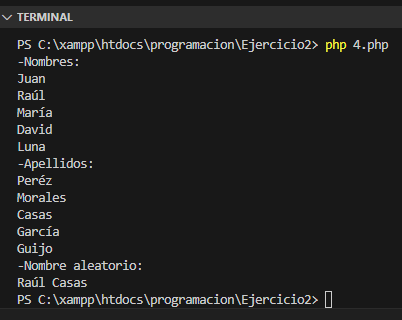
}

echo("-Nombre aleatorio:")."\n";

$nombre\_completo = $nombres[$nombre\_random] . $apellidos[$apellido\_random]; // Crea el nombrecompleto

echo $nombre\_completo //Imprime el nombre completo

?>



5. Simulador de dado

Crea un programa que simule el lanzamiento de un dado de seis caras.

Utilizar la función rand() para generar un número aleatorio entre 1 y 6.

Imprimir el número obtenido, simulando el resultado del lanzamiento del dado.

<?php

error\_reporting(E\_ALL);

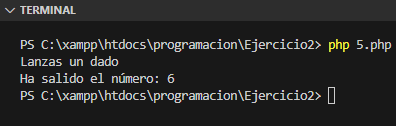
//error\_reporting(0);

echo("Lanzas un dado")."\n";

$numero = rand(1, 6); //Escoge un número entre 1 y 6

echo("Ha salido el número: $numero"); //Muestra el número

?>



<https://github.com/Rodrigo-Garcia-Ortiz/Programacion.git>