50 ANIVERSARIO

IES ANTONIO SEQUEROS ALMORADÍ

TEMA

4

Arrays

2020/2021

Versión: 201025.1552

IES Antonio Sequeros

ÍNDICE

1. ARRAYS

Un *array* es sencillamente una tabla de valores donde cada uno de los elementos de esa *tabla* se identifica por medio de un **nombre** (común para todos) y un **índice** (que diferenciaría a cada uno de ellos) y mediante el que accedemos a ese elemento.

Cuando los *índices* de un array son *números* se dice que es **escalar** mientras que si fueran *cadenas* se le llamaría array **asociativo**

Tablas unidimensionales

Array escalar			Array asociativo		
Variable	**Indice**	**Valor**	**Variable**	**Indice**	**Valor**
$a[0]	0	Domingo	$a['Primero']	Primero	Domingo
$a[1]	1	Lunes	$a['Segundo']	Segundo	Lunes
$a[2]	2	Martes	$a['Tercero']	Tercero	Martes
$a[3]	3	Miércoles	$a['Cuarto']	Cuarto	Miércoles
$a[4]	4	Jueves	$a['Quinto']	Quinto	Jueves
$a[5]	5	Viernes	$a['Sexto']	Sexto	Viernes
$a[6]	6	Sábado	$a['Septimo']	Septimo	Sábado

1.1. ARRAYS ESCALARES

Los elementos de un *array* escalar pueden definirse con una de estas sintaxis:

Arrays Escalares

a[]=valor	PHP asigna los índices de forma automática atribuyendo a cada elemento el valor entero siguiente al último asignado. Si es el primero será el 0
$a[xx]=valor	Indicamos el elemento al que queremos acceder y le asignamos un valor.
$a=array(valor0, valor1, valor2,...)	Definimos todos los elementos del array dándoles su valor.

1.2. ARRAYS ASOCIATIVOS

Para definir arrays asociativos:

Arrays Asociativos

`$array = array(`
`  "color" => "rojo",`
`  "flor" => "amapola",`
`);`
	Un array puede ser creado con el constructor del lenguaje array(). Éste toma cualquier número de parejas *clave => valor* como argumentos.

`$array = [`
`  "color" => "rojo",`
`  "flor" => "amapola",`
`];`
	A partir de PHP 5.4 también se puede usar la sintaxis de array corta, la cual reemplaza *array()* con *[]*.

Los índices no tienen por qué ser correlativos

<?php $cuadrados = [3 => 9, 5 => 25, 10 => 100]; print " El cuadrado de 3 es $cuadrados[3] \n"; ?>	El cuadrado de 3 es 9
<?php $edades = ["Andrés" => 20, "Bárbara" => 19, "Camilo" => 17]; print " Bárbara tiene $edades[Bárbara] años \n"; ?>	Bárbara tiene 19 años

En este caso estamos obligados a escribir el nombre del índice que habrá de ser una cadena y debe ponerse entre comillas.

Tanto en este supuesto como en el anterior, es posible -y bastante frecuente- utilizar como índice el contenido de una variable. El modo de hacerlo sería:

$a[$ind]=valor

En este caso, sea cual fuere el valor de la variable $ind, el nombre de la variable nunca se pone entre comillas.

1.3. ARRAYS MULTIMENSIONALES

Los arrays pueden ser de más de una dimensión, es decir, arrays cuyos elementos son a su vez arrays. Para referirse a los elementos concretos, se necesitan utilizar varios índices (tantos como dimensiones -niveles de anidamiento- tenga el array).

Los *índices* pueden ser de tipo **escalar** -*equivalen al número de fila y columna que la* ***celda*** *ocupa en la tabla*- o puede ser **asociativos** lo que equivaldría en alguna medida a usar como índices los *nombres de la fila y de la columna*.

Los elementos de un *array* de dos dimensiones escalar pueden definirse con una de estas sintaxis:

Arrays de dos dimensiones	
a[][]=valor	PHP asigna los índices de forma automática atribuyendo a cada elemento el valor entero siguiente al último asignado. Si es el primero será el 0,0
$a[xx][]=valor	En el segundo de los casos, asignamos un valor al primer índice (xx) y será el segundo quien se incremente en una unidad respecto al de valor más alto de todos aquellos cuyo primer índice coincide con el especificado.
$a[][xx]=valor	En el segundo de los casos, asignamos un valor al segundo índice (xx) y será el primero quien se incremente en una unidad respecto al de valor más alto de todos aquellos cuyo segundo índice coincide con el especificado.

1.4. Imprimir todos los valores de una matriz: la función print_r()

La función print_r($variable [, $devolver]) permite imprimir todos los valores de una matriz de forma estructura. En general, print_r() imprime cualquier variable de forma legible.

Aunque print_r() genera espacios y saltos de línea que pueden verse en el código fuente de la página, print_r() no genera etiquetas html, por lo que el navegador no muestra esos espacios y saltos de línea.

Para mejorar la legibilidad una solución es añadir la etiqueta <pre>, que fuerza al navegador a mostrar los espacios y saltos de línea.

Si se añade un segundo argumento con el valor true, print_r() no imprime nada pero devuelve el texto que se imprimiría si no estuviera el argumento true. Se utiliza para guardar la respuesta en una variable que se puede imprimir posteriormente

La función unset() permite borrar una matriz o elementos de una matriz.

1.5. Funciones útiles con matrices

- La función count($matriz) permite contar los elementos de una matriz.
- La función max($matriz, ...) devuelve el valor máximo de una matriz (o varias). La función min($matriz, ...) devuelve el valor mínimo de una matriz (o varias).

Ordenar una matriz

Cuando los índices de la matriz que vamos a ordenar no son importantes y se pueden modificar, podemos utilizar las funciones sort($matriz, $opciones) y rsort($matriz, $opciones), que ordenan atendiendo únicamente a los valores de la matriz (no a sus índices), en orden creciente o decreciente, y reindexan la matriz:

- sort($matriz, $opciones): ordena por orden alfabético / numérico de los valores y genera nuevos índices numéricos consecutivos a partir de cero:
- rsort($matriz, $opciones): ordena por orden alfabético / numérico inverso de los valores y genera nuevos índices numéricos consecutivos a partir de cero:

Cuando los índices de la matriz que vamos a ordenar son importantes y no se deben modificar, podemos ordenar atendiendo únicamente a los valores de la matriz u ordenar atendiendo únicamente a los índices de la matriz y ordenar en orden creciente o decreciente, por lo que PHP dispone de cuatro funciones:

- asort($matriz, $opciones): ordena por orden alfabético / numérico de los valores
- arsort($matriz, $opciones): ordena por orden alfabético / numérico inverso de los valores
- ksort($matriz, $opciones): ordena por orden alfabético / numérico de los índices
- krsort($matriz, $opciones): ordena por orden alfabético / numérico de los índices

Para ver la diferencia entre estas funciones, la tabla siguiente resume los ejemplos anteriores:

Matriz Inicial	*sort()*	*rsort()*
Array ([5] => cinco [1] => uno [9] => nueve)	Array ([0] => cinco [1] => nueve [2] => uno)	Array ([0] => uno [1] => nueve [2] => cinco)

Matriz Inicial	*asort()*	*arsort()*
Array ([5] => cinco [1] => uno [9] => nueve)	Array ([5] => cinco [9] => nueve [1] => uno)	Array ([1] => uno [9] => nueve [5] => cinco)

Matriz Inicial	*ksort()*	*krsort()*
Array ([5] => cinco [1] => uno [9] => nueve)	Array ([1] => uno [5] => cinco [9] => nueve)	Array ([9] => nueve [5] => cinco [1] => uno)

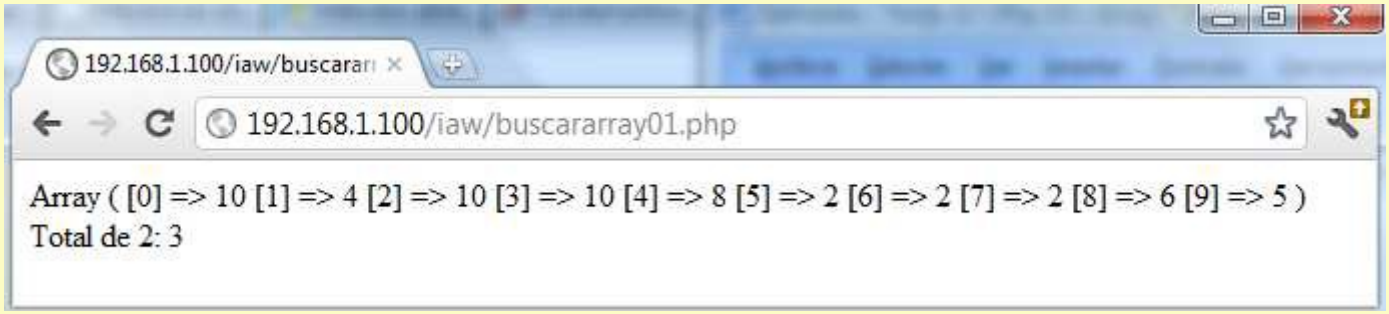
- La función booleana in_array($valor, $matriz[, $tipo]) devuelve true si el valor se encuentra en la matriz. Si el argumento booleano $tipo es true, in_array() comprueba además que los tipos coincidan.
- La función array_search($valor, $matriz[, $tipo]) busca el valor en la matriz y, si lo encuentra, devuelve el índice correspondiente, pero si hay varios valores coincidente sólo devuelve el primero que encuentra.
- La función array_keys($matriz[, $valor[, $tipo]) busca el valor en la matriz y, si lo encuentra, devuelve una matriz cuyos valores son los índices de todos los elementos coincidentes.
- La función array_values($matriz) devuelve los valores de una matriz en el mismo orden que en la matriz original, pero renumerando los índices desde cero
- La función shuffle($matriz) baraja los valores de una matriz. Los índices de la matriz original se pierden, ya que se renumeran desde cero.
- La función array_rand($matriz) extrae al azar uno de los índices de la matriz.

Práctica 05

1. Letranif.php → Realizar un programa que dado un número de NIF calcule su letra correspondiente.

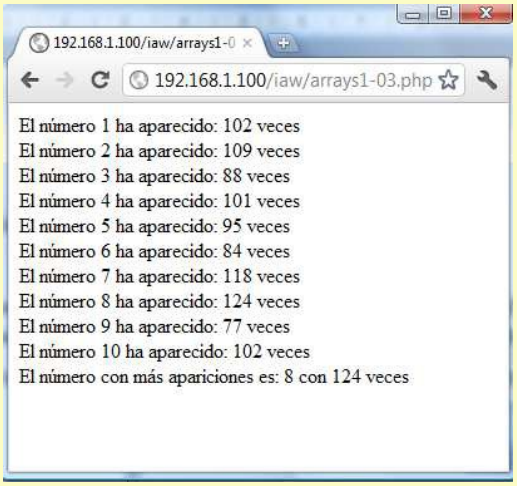
   La letra se obtiene calculando el resto de la división del número del DNI por 23. A cada resultado le corresponde una letra: 0=T; 1=R; 2=W; 3=A; 4=G; 5=M; 6=Y; 7=F; 8=P; 9=D; 10=X; 11=B; 12=N; 13=J; 14=Z; 15=S; 16=Q; 17=V; 18=H; 19=L; 20=C; 21=K; 22=E.

2. BuscarEnArray.php → Generar un array de 10 elementos con valores aleatorios entre 1 y 10. Pedir un número y mostrar el número de veces que aparece en el array. Ejemplo el 2:



3. Realizar un script en PHP llamado frecuencia.php que realice lo siguiente: Generar 1000 números aleatorios entre 1 y 100. Mostrar el número de veces que aparece un cada número, el número con más apariciones y el total de esas apariciones;

   Veamos que mostraría la solución para números entre 1 y 10:



   Muestra también el número con menos apariciones y el total de las mismas.

4. Realizar en un vector T de 10 elementos:

   4.a) Inicializar los elementos de un vector T con los números pares del 2 al 20

   4.b) Calcula la suma de los elementos del vector

   4.c) Calcula la media de los elementos del vector

5. Realizar a partir de un vector T de 10 elementos aleatorios entre 1 y 10:

   5.a) Programa que nos calcule un vector S sin elementos repetidos

   5.b) Programa que nos calcule un vector S con los elementos repetidos de T.

   5.c) Programa nos calcule un vector S de 10 elementos de la forma:

   $S(1) = T(1)$

   $S(2) = T(1) + T(2)$

   ……..

   $S(10) = T(1) + T(2) + \ldots + T(10)$

6. Programa que a partir de dos vectores S y T de 10 elementos calcule:

   6.a) S U T (elementos de S más los elementos de T, unión)

   6.b) S ∩ T (elementos pertenecientes a S y T, intersección)

   6.c) S – T (elementos de S que no están en T, diferencia)

   Ejercicios Arrays Bidimensionales

7. Dada una matriz de 4x5 de valores numéricos aleatorios entre 0 y 100.

   7.a) Mostrar todos los elementos por filas

   7.b) Mostrar todos los elementos por columnas

8. Dada una matriz de 5x6 de valores numéricos aleatorios entre 0 y 10. Mostrar la suma de todos sus elementos

9. Dada una matriz de 4x5 de valores numéricos aleatorios entre 0 y 100. Determinar la posición (fila y columna) del mayor número almacenado en la matriz.

10. Dada una matriz de 4x5 de valores numéricos aleatorios entre 0 y 100. Mostrar la columna cuyos elementos tiene la suma mayor de todas y el valor de esa suma.

11. Crear una matriz de 10x10 de manera que contenga los siguientes valores: En la diagonal principal un 1 y en el resto 0.

    Ejercicios Cadenas (strings)

12. Programa que a partir de una cadena de caracteres nos cuente las apariciones de cada uno de sus caracteres. Ejemplo:
    - Cadena = “programa”
    - El carácter 'a' ha aparecido: 2 veces
    - El carácter 'g' ha aparecido: 1 veces
    - El carácter 'm' ha aparecido: 1 veces
    - El carácter 'o' ha aparecido: 1 veces
    - El carácter 'p' ha aparecido: 1 veces
    - El carácter 'r' ha aparecido: 2 veces
    - El carácter con más apariciones es: 'r' con 2 veces