APRENDERAPROGRAMAR.COM

FUNCIÓN COUNT. RECORRIDO DE ARRAYS UNIDIMENSIONALES Y MULTIDIMENSIONALES EN PHP CON FOR Y FOR-EACH. EJEMPLOS . (CU00826B)

Sección: Cursos
Categoría: Tutorial básico del programador web: PHP desde cero

Fecha revisión: 2029

Resumen: Entrega nº26 del Tutorial básico “PHP desde cero”.

Autor: Enrique González Gutiérrez

FUNCIÓN COUNT

Antes de poder recorrer un array, debemos saber cuál es su tamaño para poder recorrerlo. Supongamos que un array tiene tres elementos: tenemos que dar una instrucción para que se extraiga el valor asociado a cada uno de los elementos del array, que normalmente será algo similar a "para cada uno de los tres elementos del array, extraer su valor".

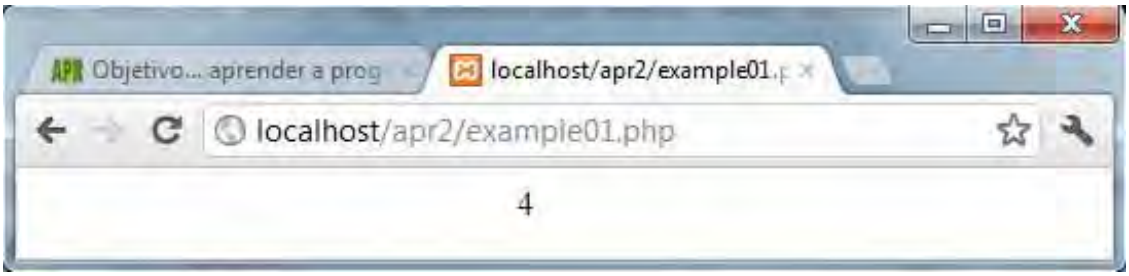
Sin embargo, si el array tiene 24 elementos, la instrucción será del tipo "para cada uno de los veinticuatro elementos del array, extraer su valor". Como vemos, muchas veces nos es necesario saber el número de elementos que hay en el array para poder recorrerlo, y para ello nos va a resultar útil la función count.

La función count devuelve el número de elementos que hay en el array. Es decir para un array de 4 elementos, la función count devolverá el número 4. Recuerda que si se trata de valores numéricos de índices, los cuatro valores numéricos serán normalmente 0, 1, 2 y 3 en lugar de 1, 2, 3 y 4.

Escribe ahora este código y guárdalo con un nombre de archivo como ejemplo2.php. A continuación, sube el fichero al servidor y visualiza el resultado.

```php
<?php //Ejemplo count aprenderaprogramar.com
$estacion[0] = "Primavera";
$estacion[1] = "Verano";
$estacion[2] = "Otoño";
$estacion[3] = "Invierno";

echo count($estacion);
?>
```



Puedes comprobar que se muestra el número 4 porque el array tiene 4 elementos.

RECORRIDOS DE ARRAYS UNIDIMENSIONALES

Ahora que ya conocemos qué son los arrays, debemos conocer cómo recorrerlos para extraer o comprobar los valores que contiene cada uno de los elementos del array. Esta es una tarea habitual en programación que en general va a tener gran utilidad para nosotros.

Lo arrays se pueden recorrer de muchas formas, nosotros no las explicaremos todas. Trataremos de ver en principio las más habituales y sencillas.

Recorrido de arrays mediante for

Escribe ahora este código y guárdalo con un nombre de archivo como ejemplo3.php. A continuación, sube el fichero al servidor y visualiza el resultado.

```php
<?php //Ejemplo arrays aprenderaprogramar.com
$array[0] = "Uno";
$array[1] = "Dos";
$array[2] = "Tres";
$array[3] = "Cuatro";
$array[4] = "Cinco";
$array[5] = "Seis";
$array[6] = "Siete";
$array[7] = "Ocho";

for($i=0;$i<count($array);$i++) {
        echo $array[$i].'<br />';
}
?>
```

Como ejercicio y teniendo en cuenta lo explicado en las entregas anteriores sobre los distintos tipos de instrucciones de repetición, escribe el código que dé lugar al mismo resultado que el ejemplo anterior pero usando la instrucción while y comprueba que obtienes el mismo resultado.

CONOCER COUNT

Ejecuta este código:

```
<?php //Ejemplo arrays aprenderaprogramar.com
$array[5] = "Uno";
$array[6] = "Dos";
$array[7] = "Tres";
$array[8] = "Cuatro";
$array[9] = "Cinco";
$array[10] = "Seis";
$array[11] = "Siete";
$array[12] = "Ocho";
echo 'Elementos inicializados en el array: '.count($array).'<br/>';
for($i=0;$i<count($array);$i++) {
	echo $array[$i].'...<br/>';
}
echo $array[15].'aaa<br/>';
echo count($otroArray[14][33]).' elementos<br/>';
?>
```

El resultado obtenido será el siguiente:

```
Elementos inicializados en el array: 8
...
...
...
...
...
Uno...
Dos...
Tres...
aaa
0 elementos
```

Analicemos lo que está ocurriendo. El array $array no tiene definidos cuáles son sus elementos de índice 0, 1, 2, 3 y 4. En cambio sí tiene definidos valores para sus elementos de índice 5, 6, 7, 8, 9, 10, 11 y 12. Hay 8 elementos con valores definidos pero sus índices no son 0, 1, 2, 3, 4, 5, 6, 7 como es lo habitual, sino que son 5, 6, 7, 8, 9, 10, 11 y 12.

Al ejecutar el for comenzando con un valor de la variable de control igual a cero, se ejecuta echo $array[0].'...
'; Al no tener un valor asignado, $array[0] devuelve vacío y simplemente se muestran tres puntos por pantalla. Lo mismo ocurre con los índices 1, 2, 3, 4 y sólo es al llegar al índice cinco cuando se muestran valores por pantalla.

Sin embargo, sólo se mostrarán los elementos con índice 5 a 8 del array, quedando el resto de elementos sin ser mostrado.

Además comprobamos que podemos invocar a elementos del array por encima del índice máximo definido sin obtener error, y que incluso podemos invocar a un nombre de array que no hemos declarado ni utilizado, con cualquier índice, sin obtener error.

En este ejemplo podríamos mostrar todos los elementos del array realizando un "pequeño cambio":

for($i=5;$i<count($array)+5;$i++)

Sin embargo esta solución no es satisfactoria porque no la podemos aplicar como solución para recorrer cualquier array.

A modo de resumen diremos lo siguiente:

count nos devuelve el número de elementos inicializados del array.

Si los índices del array no son secuencialmente 0, 1, 2, 3 , etc. el recorrido con un for tradicional puede resultar no satisfactorio, por lo que habremos de pensar en otras alternativas para recorrer el array. Hablaremos de esto más adelante.

COUNT CON ARRAYS MULTIDIMENSIONALES

En el caso de arrays de más de una dimensión, la función count devuelve el número de elementos que hay en un nivel del array definido y para un índice definido. Llamamos nivel del array a cada uno de los corchetes existentes en el array. Por ejemplo si hemos definido $ejem[2][5][1][6][2] = 33; y $ejem[2][5][3][1][9] = 55; decimos que en el primer nivel del array, correspondiente al primer índice, hay un solo índice: el 2. En el segundo nivel del array para índice 2 hay un solo índice: el 5. En el tercer nivel del array con índice 1 hay un solo índice: el 6. En el tercer nivel del array con índice 3 hay un solo índice: el 1. En el cuarto nivel del array con índice 6 hay un solo índice: el 2. En el cuarto nivel del array con índice 1 hay un solo indice: el 9.

Con un ejemplo comprenderemos mejor el funcionamiento de count en arrays multidimensionales.

Ejecuta este código:

```
<?php //Ejemplo arrays aprenderaprogramar.com
$miEspArray[6]=99;
$miEspArray[22]=87;
echo 'Numero de elementos de miEspArray es '.count($miEspArray).'<br/>';
echo "--------------------";
$array2[0][0][0] = "Prueba1";
$array2[0][0][1] = "Prueba2";
```

```
$array2[0][0][2] = "Prueba3";
$array2[0][1][0] = "Prueba4";
$array2[0][1][1] = "Prueba5";
$array2[1][0][1] = "Prueba6";
$array2[1][1][2] = "Prueba7";
$array2[1][2][1] = "Prueba8";
$array2[2][0][0] = "Prueba9";
$array2[2][0][1] = "Prueba10";
$array2[5][0][1] = "Prueba11";
echo '<br/>Numero de indices en el primer nivel: '.count($array2);
echo '<br/>Numero de indices en el segundo nivel de array[0]: '.count($array2[0]);
echo '<br/>Numero de indices en el segundo nivel de array[1]: '.count($array2[1]);
echo '<br/>Numero de indices en el segundo nivel de array[2]: '.count($array2[2]);
echo '<br/>Numero de indices en el segundo nivel de array[3]: '.count($array2[3]);
echo '<br/>Numero de indices en el segundo nivel de array[5]: '.count($array2[5]);
echo '<br/>Numero de indices en el tercer nivel de array[0][0]: '.count($array2[0][0]);
echo '<br/>Numero de indices en el tercer nivel de array[0][1]: '.count($array2[0][1]);
echo '<br/>Numero de indices en el tercer nivel de array[1][0]: '.count($array2[1][0]);
echo '<br/>Numero de indices en el tercer nivel de array[1][1]: '.count($array2[1][1]);
echo '<br/>Numero de indices en el tercer nivel de array[1][2]: '.count($array2[0][0]);
echo '<br/>Numero de indices en el tercer nivel de array[2][0]: '.count($array2[2][0]);
echo '<br/>Numero de indices en el tercer nivel de array[5][0]: '.count($array2[5][0]);
?>
```

El resultado obtenido será el siguiente:

```
Numero de elementos de miEspArray es 2
--------------------
Numero de indices en el primer nivel: 4
Numero de indices en el segundo nivel de array[0]: 2
Numero de indices en el segundo nivel de array[1]: 3
Numero de indices en el segundo nivel de array[2]: 1
Numero de indices en el segundo nivel de array[3]: 0
Numero de indices en el segundo nivel de array[5]: 1
Numero de indices en el tercer nivel de array[0][0]: 3
Numero de indices en el tercer nivel de array[0][1]: 2
Numero de indices en el tercer nivel de array[1][0]: 1
Numero de indices en el tercer nivel de array[1][1]: 1
Numero de indices en el tercer nivel de array[1][2]: 3
Numero de indices en el tercer nivel de array[2][0]: 2
Numero de indices en el tercer nivel de array[5][0]: 1
```

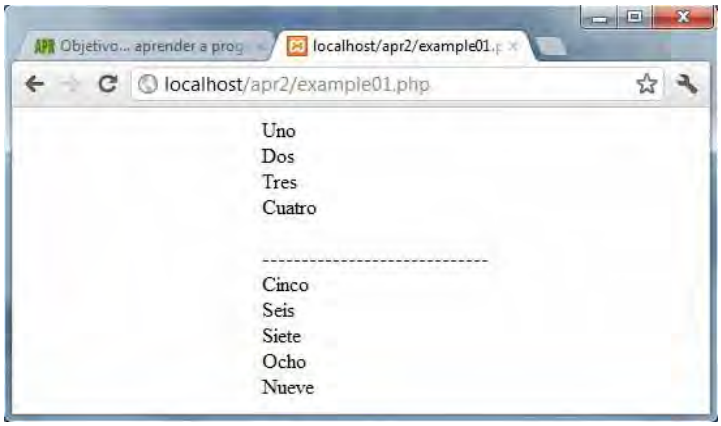
Fíjate que por ejemplo count($array2[0][1]) nos devuelve el número de elementos existentes de tipo $array2[0][1][x] donde x es cualquier índice, es decir, el número de elementos de tercer nivel del array para los índices iniciales 0 y 1.

RECORRIDO DE ARRAYS MULTIDIMENSIONALES

Para recorrer un array multidimensional, tendremos que ir anidando tantas estructuras repetitivas como dimensiones tenga el array. Con un ejemplo se verá todo mucho más claro.

Escribe ahora este código y guárdalo con un nombre de archivo como ejemplo4.php. A continuación, sube el fichero al servidor y visualiza el resultado.

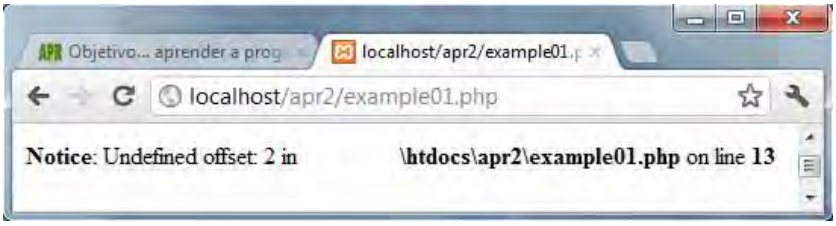
```php
<?php //Ejemplo arrays aprenderaprogramar.com
$array[0][0] = "Uno";
$array[0][1] = "Dos";
$array[1][0] = "Tres";
$array[1][1] = "Cuatro";
for($i=0;$i<count($array);$i++) {
	for($j=0;$j<count($array[$i]);$j++) {
		echo $array[$i][$j].'<br />';
	}
}
echo "--------------------";
$array2[0][0][0] = "Cinco";
$array2[0][0][1] = "Seis";
$array2[0][0][2] = "Siete";
$array2[0][1][0] = "Ocho";
$array2[0][1][1] = "Nueve";
for($i=0;$i<count($array);$i++) {
	for($j=0;$j<count($array[$i]);$j++) {
		for($k=0;$k<count($array[$i][$j]);$k++) {
			echo $array[$i][$j][$k].'<br />';
		}
	}
}
?>
```



Fíjate que para poder obtener el resultado deseado los elementos de los arrays deben estar definidos usando índices que comiencen en 0 y sean progresivamente 1, 2, 3, 4... Si no fuera así estos bucles, al estar definidos partiendo de 0, no funcionarían.

Fíjate también en la lógica de los bucles, por ejemplo el primer bucle lo podríamos leer así: para cada elemento desde 0 hasta el número de elementos de primer nivel, y para cada elemento desde 0 hasta el número de elementos de segundo nivel con el índice extraido previamente, mostrar el contenido del array. Si por ejemplo en el primer nivel tenemos dos índices que son 0 y 1, comenzaremos con el índice 0 y se verá para el índice 0 cuántos índices hay en el segundo nivel (son 2). De este modo lo primero que se mostrará son los índices [0][0] y [0][1]. Una vez completado el recorrido del índice 0 se pasa al índice 1 y se sigue el mismo proceso.

Nota: en algunas versiones o situaciones pedir un elemento no definido de un array puede dar lugar a que aparezca un error. Por ejemplo, si el elemento $array[2][3] no existe e hiciéramos una llamada a ese elemento como echo $array[2][3] podría aparecer un error tipo "undefined offset..." similar a éste:



Recorrido de arrays mediante foreach

PHP incorpora una forma "cómoda" para poder recorrer todos los elementos de un array. Esta forma se basa en el uso de la instrucción foreach.

Este tipo de recorridos se suele usar cuando obtenemos datos de una base de datos.

El tipo de array que nos suele devolver una consulta a la base de datos es similar al siguiente:

```
$datosArray = array(
                array('nombre' => 'Antonio', 'apellidos' => 'Gómez Gómez', 'telefono' => '675832145'),
                array('nombre' => 'Pedro', 'apellidos' => 'Guillén Gastón', 'telefono' => '674562178'),
                array('nombre' => 'Dolores', 'apellidos' => 'Candela Quema', 'telefono' => '689765432'),
                .
                .
                .
                .
                array('nombre' => 'Rubén', 'apellidos' => 'Guardia Jurado', 'telefono' => '654213896'),
);
```

Recuerda que la anterior forma de expresión es equivalente a esta otra:

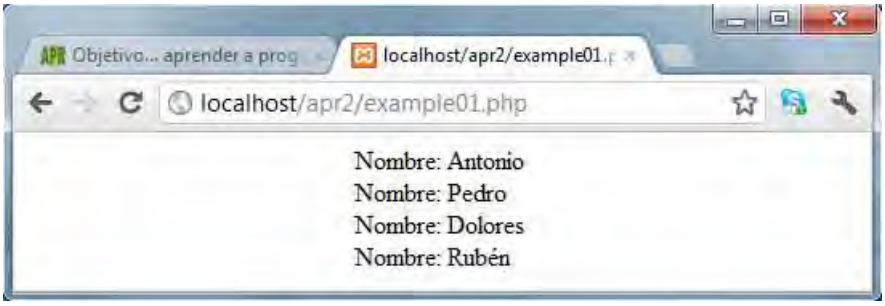
```
$datosArray[0] ['nombre'] = 'Antonio'
$datosArray[0] ['apellidos'] = 'Gómez Gómez'
$datosArray[0] ['telefono'] = '675832145'
$datosArray[1] ['nombre'] = 'Pedro'
$datosArray[1] ['apellidos'] = 'Guillén Gastón'
$datosArray[1] ['telefono'] = '674562178'
$datosArray[2] ['nombre'] = 'Dolores'
$datosArray[2] ['apellidos'] = 'Candela Quema'
$datosArray[2] ['telefono'] = '689765432'
```

La diferencia está en que en el primer caso los índices del array son en algunos casos cadenas (se trata de arrays asociativos) mientras que en el segundo caso los índices del array son números (se trata de arrays tradicionales). En el primer caso los valores 0, 1 y 2 son asignados automáticamente por php porque no se ha especificado otra cosa.

Para recorrer arrays de forma cómoda se usa la instrucción foreach.

Escribe ahora este código y guárdalo con un nombre de archivo como ejemplo5.php. A continuación, sube el fichero al servidor y visualiza el resultado.

```php
<?php //Ejemplo foreach aprenderaprogramar.com
$rows = array(
        array(
          'nombre' => 'Antonio', 'apellidos' => 'Gómez Gómez', 'telefono' => '675832145'),
        array(
          'nombre' => 'Pedro', 'apellidos' => 'Guillén Gastón', 'telefono' => '674562178'),
        array(
          'nombre' => 'Dolores', 'apellidos' => 'Candela Quema', 'telefono' => '689765432'),
        array(
          'nombre' => 'Rubén', 'apellidos' => 'Guardia Jurado', 'telefono' => '654213896')
);
foreach($rows as $valor) {
        echo 'Nombre: ' . $valor['nombre'] . '<br />';
}
?>
```



Como podemos observar, vamos a ir recorriendo todas las filas y mostrando el nombre.

$valor es una variable temporal que sólo existe durante la ejecución de la instrucción foreach, y esta variable va tomando en cada repetición o iteración el valor del siguiente elemento dentro del array principal. En este caso, el elemento que hay dentro del array es otro array. A su vez, indicamos que para cada uno de los arrays extraidos en el recorrido, se nos muestre el item del array cuyo índice es 'nombre'.

En este caso podríamos obtener el mismo resultado usando este código:

```
echo $rows[0]['nombre']. '<br />';
echo $rows[1]['nombre']. '<br />';
echo $rows[2]['nombre']. '<br />';
echo $rows[3]['nombre']. '<br />';
```

Pero piensa que para recorrer un array de varios cientos de elementos no resultará práctico escribirlos uno a uno...

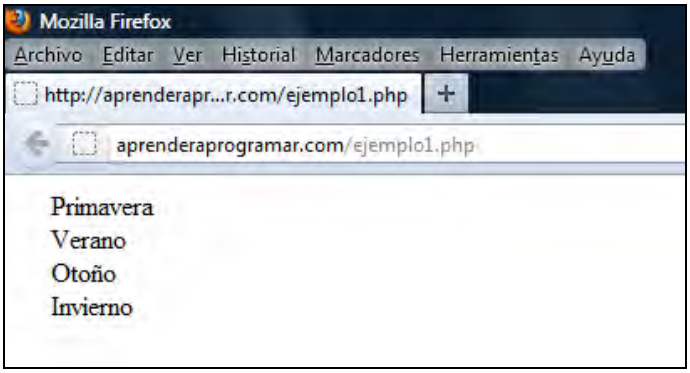
Como ejercicio escribe el código que muestre los nombres y apellidos del array anterior.

Si te resulta un poco complicado el ejercicio anterior, prueba con este código que es más sencillo, y luego vuelve a revisar el código anterior.

```php
<?php
$estacion[0] = "Primavera";
$estacion[1] = "Verano";
$estacion[2] = "Otoño";
$estacion[3] = "Invierno";

foreach($estacion as $valor) {
        echo '     '. $valor.'<br/>';
}
?>
```

El resultado que obtenemos es similar al que se ve en la siguiente imagen. Como puedes comprobar, la variable $valor (variable transitoria) va tomando en cada repetición del bucle el contenido de los elementos o items del array que estamos recorriendo.



EJERCICIO 1

Crea un array con nombre paisLimitrofe donde los elementos del array serán cadenas de texto con los países limítrofes (con frontera) con el tuyo. Por ejemplo si vivimos en Perú tenemos como países limítrofes Ecuador, Colombia, Brasil, Bolivia y Chile.

a) Utilizando un bucle for, muestra por pantalla los países limítrofes.

b) Utilizando un bucle for each, muestra por pantalla los países limítrofes.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

EJERCICIO 2

Escribe y ejecuta el siguiente código y responde a las siguientes preguntas:

```
$equipo = array(portero=>'Casillas', defensa=>'Hierro', medio=>'Ces', delantero=>'Ronaldo');

foreach($equipo as $posicion=>$jugador) {
    echo "El " . $posicion . " es " . $jugador;
    }
```

a) ¿$equipo es una variable normal, un array tradicional ó un array asociativo?

b) ¿portero es un índice de un array, un contenido de un elemento de un array o un contenido de una variable simple?

c) ¿Al ejecutar el código obtienes un resultado por pantalla u obtienes un error? ¿Qué es lo que hace el código?

EJERCICIO 3

Supón que quieres representar lo siguiente: hay 2 equipos españoles, en el primero el portero es Frank, el defensa Pepe, el medio Luis y el delantero Raul. En el segundo, el portero es Tiger, el defensa Mourin, el medio Katz y el delantero Alberto. Hay 1 equipo mexicano, donde el portero es Suarez, el defensa Koltz, el medio Fernandez y el delantero Ramirez. Hay 2 equipos argentinos. En el primero el portero es Higuita, el defensa Mel, el medio Rubens y el delantero Messi. En el segundo el portero es Kostenmeiner, el defensa Lenkins, el medio Marash y el delantero Juanes.

a) Representa los datos usando un array de tres dimensiones con índices numéricos donde el primer índice indica el país, el segundo el equipo y el tercero la posición del jugador. Presenta la información del país, equipo, posiciones y jugadores de cada equipo usando un bucle for.

b) Representa los datos usando un array de tres dimensiones con índices numéricos donde el primer índice indica el país, el segundo el equipo y el tercero la posición del jugador. Presenta la información del país, equipo, posiciones y jugadores de cada equipo usando un bucle for each.

c) Representa los datos usando arrays arrays asociativos donde el primer índice indica el país, el segundo el equipo y el tercero la posición del jugador (un ejemplo de cómo declarar un elemento sería por ejemplo: $equipos['Mexico']['Equipo1']['defensa']="koltz";). A continuación usando un bucle foreach recorre los elementos del array mostrando la información del país, equipo, posiciones y jugadores de cada equipo.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU00827B

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=70&Itemid=193