

PRACTICA 3



PHP

ARRAYS



Por Ramon Abramo

EJERCICIO 1

Vamos a trabajar con arrays

```
<?php
/*
 * Crear arrays vacios
 */

$numero= [];
$textos= [];

// Inspeccionar el contenido de los arrays
var_dump($numero,$textos);
```

EJERCICIO 2

Crear un array con las vocales (introducir los valores a mano).

Mostrar el contenido de la variable utilizando la función var_dump.

Una posible solución.

```
<?php
```

```

/**
 * OPCION 1
 */

// crear un array directamente
$vocales=[
    "a","e","i","o","u"
];

var_dump($vocales);

/**
 * OPCION 2
 */

// crear un array con la funcion array
$vocales=array(
    "a","e","i","o","u"
);

var_dump($vocales);

```

EJERCICIO 3

Vamos a crear un array asociativo:

```
<?php
```

```
$numeros=[  
    "cer0" => 0,  
    "uno"  => 1,  
    "dos"  => 2,  
    "tres" => 3,  
    "cuatro" => 4,  
];  
  
var_dump($numeros);
```

EJERCICIO 4

Copiar al siguiente código.

```
<?php
```

```
$numeros=[
    "cer" => 0,
    "uno" => 1,
    "dos" => 2,
    "tres" => 3,
    "cuatro" => 4,
];

$vocales=[
    "a","e","i","o","u"
];
```

- 1 // leer el segundo elemento del array numeros y mostrarlo en pantalla (1)
- 2 // leer el segundo elemento del array vocales y mostrarlo en pantalla (e)
- 3 // añadir un elemento al final de vocales con la a con tilde
- 4 // añadir un elemento nuevo al array numeros con indice cinco con el valor 5

Completar el código realizando los 4 ejercicios

Una posible solución

```
// leer el segundo elemento del array numeros y mostrarlo en pantalla (1)
echo $numeros["uno"];

// leer el segundo elemento del array vocales y mostrarlo en pantalla (e)
echo $vocales[1];

// añadir un elemento al final de vocales con la a con tilde

// opcion 1
$vocales[]="á";
// opcion 2
array_push($vocales,"á");

var_dump($vocales);

// añadir un elemento nuevo al array numeros con indice cinco con el valor 5
$numeros["cinco"]=5;

var_dump($numeros);
```

EJERCICIO 5

Completar el siguiente código con los ejercicios solicitados

```
<?php
$datos=[
    [
        "nombre" => "Eva",
        "edad" => 50
    ],
    [
        "nombre" => "Jose",
        "edad" => 40,
        "peso" =>80
    ]
];
```

1 // mostrar el nombre del primer dato

2 // mostrar el nombre del segundo dato

3 // mostrar el peso

EJERCICIO 6

La función mt_rand devuelve un numero aleatorio entre dos argumentos pasados.

mt_rand

(PHP 4, PHP 5, PHP 7, PHP 8)

mt_rand — Genera un mejor número entero aleatorio

Descripción

```
mt_rand(): int
```

```
mt_rand(int $min, int $max): int
```

Muchos generadores de números aleatorios de libcs antiguas tienen características dudosas o desconocidas y son lentas. De manera predeterminada, PHP usa la libc generadora de números aleatorios con la función `rand()`. La función `mt_rand()` es un sustituto de dicha función. Utiliza un generador de números aleatorios con características conocidas usando » [Mersenne Twister](#), que produce números aleatorios cuatro veces más rápido que el promedio proporcionado por la libc `rand()`.

Si se emplea sin los argumentos opcionales `min` y `max`, `mt_rand()` devuelve un valor pseudoaleatorio entre 0 y `mt_getrandmax()`. Para obtener un número aleatorio entre 5 y 15 (incluidos), por ejemplo, use `mt_rand(5, 15)`.

Precaución Esta función no genera valores criptográficos fiables por lo que no debería emplearse para propósitos criptográficos. Si fuera necesario un valor criptográfico seguro, considérese utilizar `random_int()`, `random_bytes()`, o `openssl_random_pseudo_bytes()` en su lugar.

Parámetros

min

Opcionalmente, el menor valor a devolver (por defecto: 0)

max

Opcionalmente, el mayor valor a devolver (por defecto: `mt_getrandmax()`)

Valores devueltos

Un valor entero aleatorio entre `min` (o 0) y `max` (o `mt_getrandmax()`, incluidos), o `false` si `max` es menor que `min`.

La función `count` nos indica el número de elementos del array.

count

(PHP 4, PHP 5, PHP 7, PHP 8)

count — Cuenta todos los elementos de un array o algo de un objeto

Descripción

```
count(mixed $array_or_countable, int $mode = COUNT_NORMAL): int
```

Cuenta todos los elementos en un array o algo de un objeto.

En objetos, si se tiene [SPL](#) instalado, se puede enganchar a **count()** implementado la interfaz [Countable](#). Esta interfaz tiene solamente un método, [Countable::count\(\)](#), el cual devuelve el valor retornado por la función **count()**.

Véase la sección [Array](#) del manual para una explicación detallada sobre cómo se implementan y usan los arrays en PHP.

Parámetros

array_or_countable

Un array o un objeto [Countable](#).

mode

Si el parámetro opcional **mode** está establecido a **COUNT_RECURSIVE** (o 1), **count()** contará el array de forma recursiva. Esto es particularmente útil para contar todos los elementos de un array multidimensional.

Precaución **count()** puede detectar recursividad para evitar un bucle infinito, pero emitirá un mensaje **E_WARNING** cada vez que lo haga (en caso de que el array se contenga a sí mismo más de una vez) y devolverá una cuenta mayor de la que se podría esperar.

Valores devueltos

Devuelve el número de elementos en **array_or_countable**, Cuando el parámetro no es un array ni un objeto con la interfaz [Countable](#) implementada, devolverá 1. Existe una excepción, si **array_or_countable** es **null**, devolverá 0.

Copiar el siguiente código y completar los apartados.

```
<?php
```

```
$vocales=[
    "a","e","i","o","u"
];
```

1 // Mostrar el numero de elementos de vocales utilizando una funcion

2 // Mostrar una vocal aleatoria utilizando mt_rand

3 // Mostrar otra vocal aleatoria utilizando mt_rand

EJERCICIO 7

Completar el siguiente código

```
<?php
```

```
$datos=[
    [
        "nombre" => "Eva",
        "edad" => 50
    ],
    [
        "nombre" => "Jose",
        "edad" => 40,
        "peso" =>80
    ]
];
```

1 // Mostrar el numero de registros introducidos en datos (2)

2 // Mostrar el numero de elementos que tiene el registro de Eva (2)

EJERCICIO 8

Completar el siguiente codigo

```
<?php
```

```
$datos=[
    [
        "nombre" => "Eva",
        "edad" => 50
    ],
    [
        "nombre" => "Jose",
        "edad" => 40,
        "peso" => 80
    ]
];
```

1

```
// introducir a "Lorena" de 80 años y con una altura de 175
// Realizarlo directamente
```

2

```
// introducir a "Luis" de 20 años y con un peso de 90 y a "Oscar" de 23 años.
// Realizarlo mediante la funcion push
```

alpe
FORMACIÓN