

PRACTICA 11



PHP

FUNCIONES

php

Por Ramon Abramo

EJERCICIO 1

Desarrollar una función que acepte como parámetros:

- Valor mínimo
- Valor máximo
- Numero de valores

La función debe generar un vector con números aleatorios (que devolverá) ordenado de forma ascendente.

```
$salida=ejercicio1(1,10,10);
var_dump($salida);
```

```
array
0 => int 3
1 => int 1
2 => int 1
3 => int 6
4 => int 6
5 => int 7
6 => int 10
7 => int 10
8 => int 1
9 => int 3
```

La función debe estar completamente documentada.

Una posible solución sería:

```

/**
 * Esta funcion genera una serie de numeros aleatorios
 * @param int $minimo Este es el valor minimo
 * @param int $maximo Este es el valor maximo
 * @param int $numero Numero de valores a generar
 * @return int[] el conjunto de numeros solicitado
 *
 */

function ejercicio1($minimo,$maximo,$numero){
    $local=array(); // array donde voy a colocar el resultado

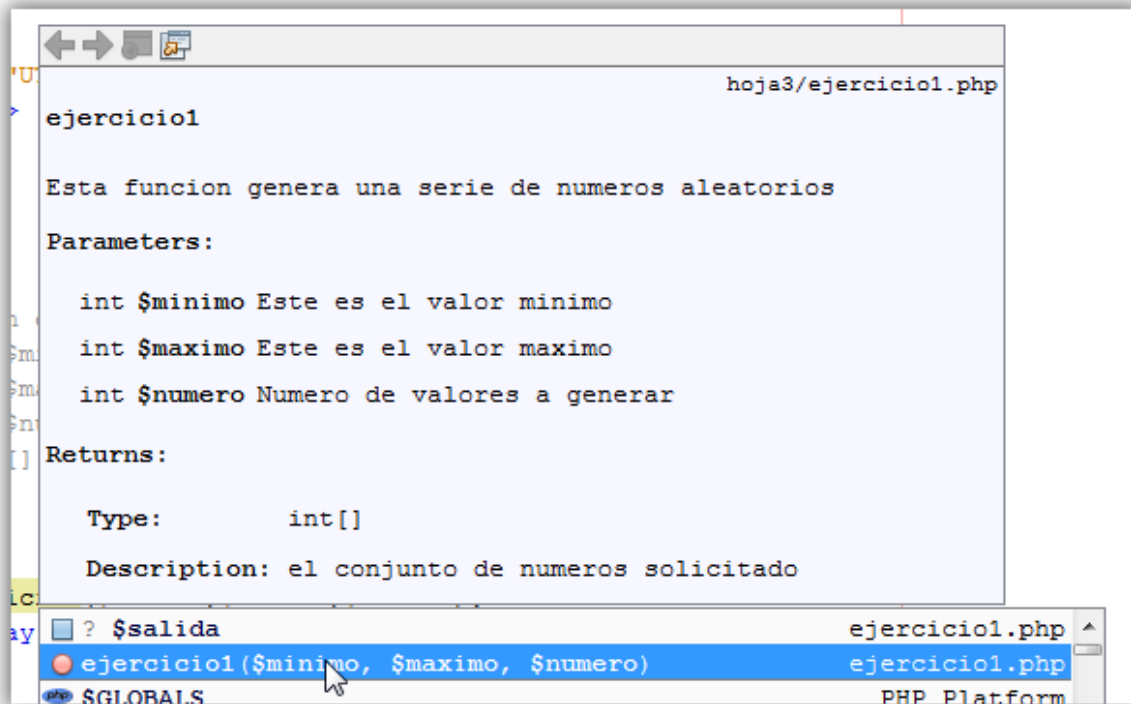
    /**
     * Bucle para rellenar el array
     */
    for($c=0;$c<$numero;$c++){
        $local[$c]=mt_rand($minimo,$maximo);
    }

    /**
     * Devolver el array
     */

    return $local;
}

```

Al utilizar la función Netbeans debe darte ayuda sobre la función y sus parámetros.

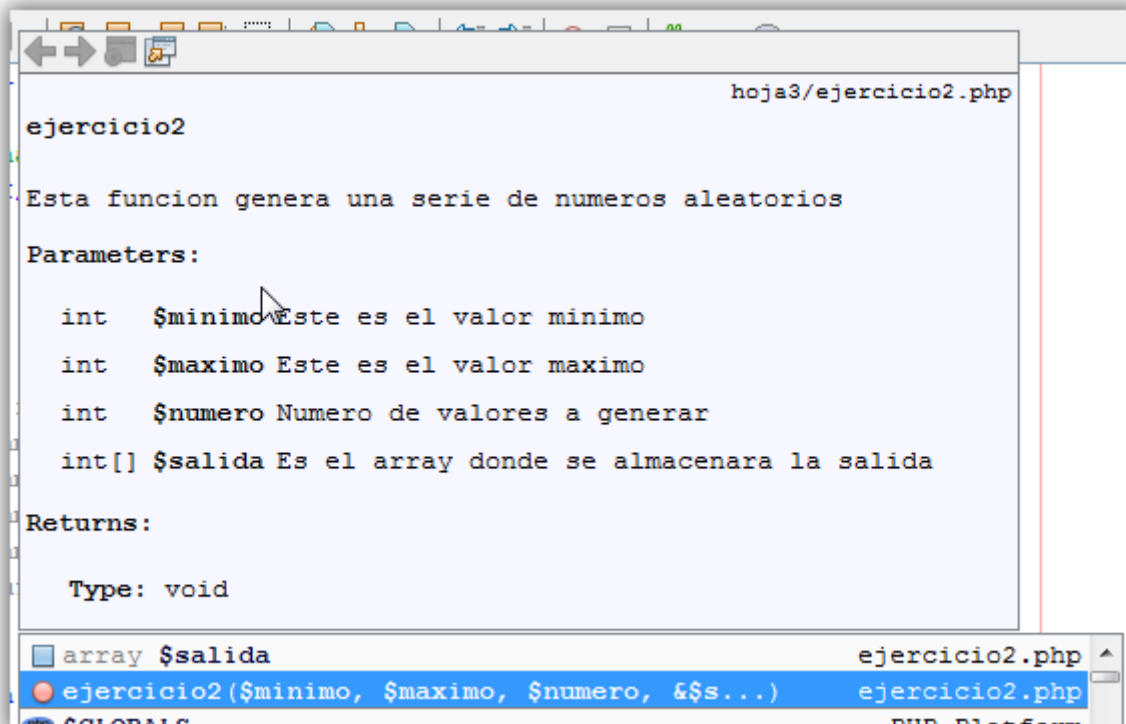


EJERCICIO 2

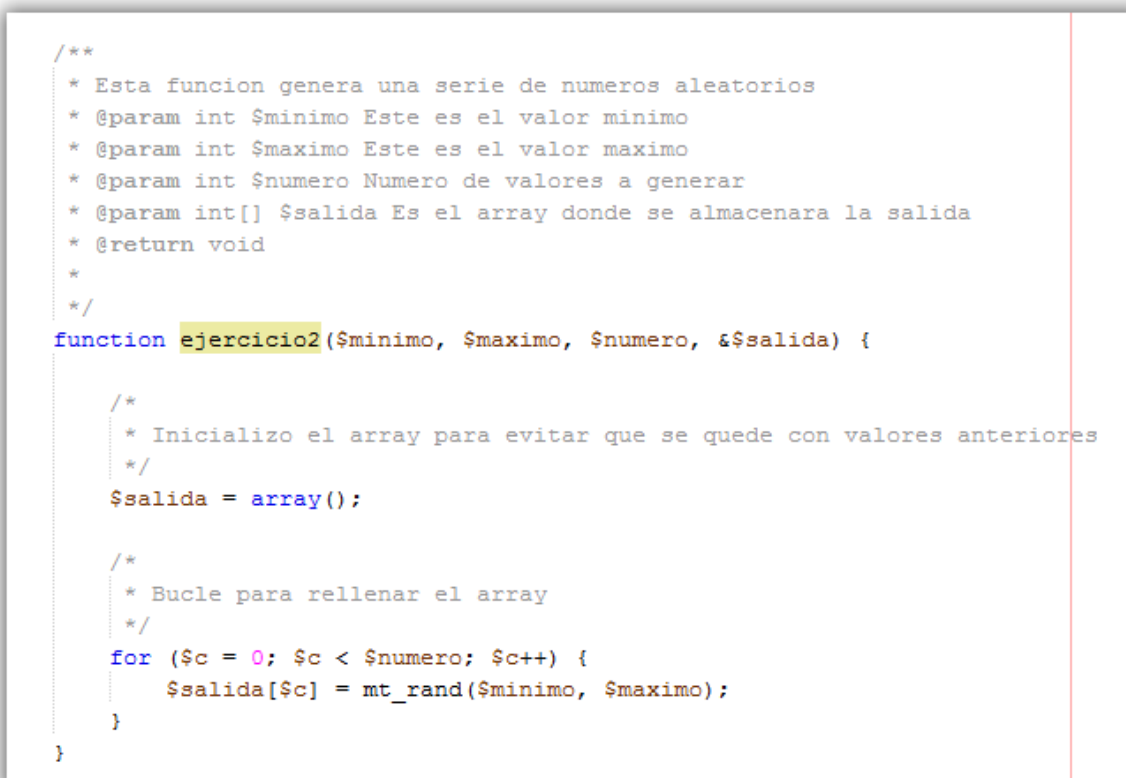
Modificar la función anterior para que acepte 4 argumentos:

- Valor mínimo
- Valor máximo
- Número de elementos
- Array donde se almacenaran los resultados.

La función no devolverá nada.



Una posible solución sería la siguiente:



EJERCICIO 3

Desarrollar una función que nos permita crear un color aleatorio en RGB. A la función le debemos pasar el numero de colores a generar y devolverá un array con esos colores.

```

hoja3/ejercicio3.php

generaColores

Funcion que genera colores

Parameters:

    int $numero El numero de colores a generar

Returns:

    Type:      array
    Description: los colores solicitados en un array de
                  cadenas

colores = generaColores($numero);

```

```

array
0 => string '#c6c826' (length=7)
1 => string '#bfe577' (length=7)
2 => string '#7f6d16' (length=7)
3 => string '#123252' (length=7)
4 => string '#84fe4d' (length=7)
5 => string '#319c78' (length=7)
6 => string '#d5f0ba' (length=7)
7 => string '#73ab3d' (length=7)
8 => string '#f9d383' (length=7)
9 => string '#786991' (length=7)

```

Una posible solución es:

```

/**
 * Funcion que genera colores
 * @param int $numero El numero de colores a generar
 * @return array los colores solicitados en un array de cadenas
 */

function generaColores($numero) {
    $colores=array();
    for($n=0;$n<$numero;$n++){
        $colores[$n]="#";
        for($c=1;$c<7;$c++){
            $colores[$n]..=dechex(mt_rand(0,15));
        }
    }
    return $colores;
}

```

EJERCICIO 4

Modificar el ejercicio anterior para que acepte un segundo parámetro opcional. Este parámetro funcionara de la siguiente forma:

- Si pasamos true coloca la # al principio del color. Si colocamos false no colocara la #. Por defecto el valor es true.

```

generaColores
Funcion que genera colores
Parameters:
    int $numero          El numero de colores a generar
    bool $almohadilla=true con valor true nos indica que coloquemos la almohadilla
Returns:
    Type:                array
    Description:          los colores solicitados en un array de cadenas
es(10,false));
generaColores($numero)          ejercicio4.php
generaColores($numero, $almohadilla = true...) ejercicio4.php

```

```

array
0 => string 'db8fb0' (length=6)
1 => string '430900' (length=6)
2 => string 'f891c3' (length=6)
3 => string 'd08c04' (length=6)
4 => string '640c92' (length=6)
5 => string 'add23d' (length=6)
6 => string '1392d0' (length=6)
7 => string '22cdac' (length=6)
8 => string 'afd268' (length=6)
9 => string 'd90a78' (length=6)

```

```

/**
 * Funcion que genera colores
 * @param int $numero El numero de colores a generar
 * @param bool $almohadilla=true con valor true nos indica que coloquemos la almohadilla
 * @return array los colores solicitados en un array de cadenas
 */

function generaColores($numero,$almohadilla=true){
    $colores=array();
    for($n=0;$n<$numero;$n++){
        $c=0;
        $limite=6;
        $colores[$n]="";
        if($almohadilla){
            $colores[$n]="#";
            $limite=7;
        }
        for($c<$limite;$c++){
            $colores[$n].=dechex(mt_rand(0,15));
        }
    }
    return $colores;
}

```

EJERCICIO 5

Analizar la siguiente función y colocar un ejemplo de funcionamiento. Se deben colocar los comentarios necesario a la función (argumentos, descripción y valores devueltos).


```
function elementosRepetidos($array, $devolverTodos = false) {
    $repeated = array();

    foreach ((array) $array as $value) {
        $inArray = false;

        foreach ($repeated as $i => $rItem) {
            if ($rItem['value'] === $value) {
                $inArray = true;
                ++$repeated[$i]['count'];
            }
        }

        if (false === $inArray) {
            $i = count($repeated);
            $repeated[$i] = array();
            $repeated[$i]['value'] = $value;
            $repeated[$i]['count'] = 1;
        }
    }

    if (!$devolverTodos) {
        foreach ($repeated as $i => $rItem) {
            if ($rItem['count'] === 1) {
                unset($repeated[$i]);
            }
        }
    }

    sort($repeated);

    return $repeated;
}
```

EJERCICIO 6

Crear una función que te devuelva en un array los ficheros existentes en un directorio ordenados alfabéticamente.

La función tiene las siguientes características:

- Como argumento de entrada le pasas un string con la ruta al directorio a analizar. Si no le pasas nada por defecto utiliza el directorio actual.
- La salida es un array con los ficheros ordenados alfabéticamente.


```
hoja3/ejercicio6.php

leerDirectorio

Funcion que te permite generar un array con el contenido
de un directorio. La salida esta ordenada

Parameters:

    type $handle cadena de caracteres con la ruta del
    directorio a listar

Returns:

    Type:      string[]
    Description: array con el nombre de los ficheros y
    directorios

leerDirectorio();
leerDirectorio();
leerDirectorio($handle = ".");
```

```
array
0 => string '.' (length=1)
1 => string '..' (length=2)
2 => string 'ejercicio1.php' (length=14)
3 => string 'ejercicio2.php' (length=14)
4 => string 'ejercicio3.php' (length=14)
5 => string 'ejercicio4.php' (length=14)
6 => string 'ejercicio5.php' (length=14)
7 => string 'ejercicio6.php' (length=14)
```

Una posible solución es :

```
/**
 * Funcion que te permite generar un array con el contenido de un directorio. La salida esta ordenada
 * @param type $handle cadena de caracteres con la ruta del directorio a listar
 * @return string[] array con el nombre de los ficheros y directorios
 */
function leerDirectorio($handle = ".") {
    $handle=opendir($handle);
    while (false !== ($archivo = readdir($handle))) {
        $archivos[] = strtolower($archivo);
    }
    closedir($handle);
    sort($archivos);
    return $archivos;
}
```