

Modificaciones en arrays

var= range(*inf,sup*)

Crea una **nueva matriz** (*var*) **escalar** en la que los valores de los elementos serán los números enteros (ordenados) pertenecientes al intervalo comprendido entre los valores **inf** y **sup**, incluidos estos.

Los valores **inf** y **sup** deben ser números enteros.

shuffle(array)

Intercambia de modo *aleatorio* los valores de un array y los reindexa.

Igual que ocurría en caso de los números aleatorios, la función **shuffle** deberá ir precedida de una **semilla** del tipo **srand**.

En el ejemplo hemos usado como **semilla** la función: **srand(time()**).

var= array_flip(array)

Devuelve un **array** (*var*) que contiene como **valores** los **índices** de la matriz *array* y como **índices** los **valores** de aquella.

Como quiera que los **valores** pueden estar **repetidos** y **no es posible** que lo estén los **índices**, esta función, en caso de valores repetidos, toma cada uno de esos **valores una sola vez**, lo utiliza como índice del nuevo array y asigna como **valor** del nuevo elemento el mayor de los índices – del array original– de los elementos que contuvieran ese valor.

Insertando elementos en un arrays

array_unshift(*arr, v1,v2,..*)

Inserta **al principio de la matriz** *arr* los valores **v1**, **v2**, etcétera que pueden ser tantos como se deseen y deben estar separados por comas.

array_push(array, v1,v2,..)

Inserta **al final de la matriz** *array* los valores **v1**, **v2**, etcétera, que igual que en el caso anterior, pueden ser tantos como se deseen y deben estar separados por comas.

Tanto **array_unshift** como **array_push** asignan a los nuevos elementos **índices numéricos**.

array_pad(array, n, var)

Inserta nuevos elementos en *array* y les asigna el valor contenido en **var**. Insertará tantos nuevos elementos como sea necesario para que el **array** alcance una longitud de **n** elementos.

Si el valor de **n** es **positivo** inserta los elementos **al final del array**, si fuera **negativo** los insertaría **al comienzo** del mismo.

A los nuevos elementos del array se les asignan índices numéricos.

array_merge(\$a, \$b)

Crea un nuevo *array escalar* en el que se incluyen todos los **elementos** contenidos en los arrays **\$a** y **\$b**.

Modificación de arrays

```
<?
$a=array(1,2,3,1,1,2,3,3,4,4,4,0,1);
$b=array("blanco","azul","blanco","blanco","azul","Blanco","Azul");
$c=array(
    "b" =>"verde",
    "c" =>"rojo",
    "e" =>"verde",
    "f" =>"Rojo",
    "g" =>"Verde",
    "a"=>"rojo",
    "d" =>"rojo");
$c=array(
    "b" =>"verde",
    "c" =>"rojo",
    "e" =>"verde",
    "f" =>"Rojo",
    "g" =>"Verde",
    "a"=>"rojo",
    "d" =>"rojo");

echo "<h3>Crea una matriz de números enteros</h3>";

$r=range(7,11);

foreach($r as $clave=>$valor){
    echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}

echo "<h3>Intercambia aleatoriamente elementos en una matriz</h3>";

srand (time());
shuffle ($r);

foreach($r as $clave=>$valor){
    echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}

echo "<h3>Intercambia valores e índices</h3>";

$p=array_flip($a);

foreach($p as $clave=>$valor){
    echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}
echo "<br>";

$q=array_flip($c);

foreach($q as $clave=>$valor){
    echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}

echo "<h3>Inserta elementos al principio de una matriz</h3>" ;

array_unshift($a,97,"Pepe",128);

foreach($a as $clave=>$valor){
    echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}
echo "<br>";

array_unshift($c,97,"Pepe",128);

foreach($c as $clave=>$valor){
    echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}

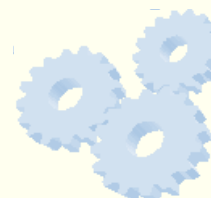
echo "<h3>Inserta elementos al final de una matriz</h3>";

array_push($a,3.4,"Luis",69);

foreach($a as $clave=>$valor){
    echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}
echo "<br>";

array_push($c,3.4,"Luis",69);

foreach($c as $clave=>$valor){
    echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}
```



Quitar elementos de un array

array_shift(\$a)

La función **array_shift** extrae el **primer elemento** del array **\$a**.

array_pop(\$a)

La función **array_pop** extrae el **último elemento** del array **\$a**.

array_slice(\$a,n)

La función **array_slice** extrae **n** elementos del array **\$a**.

Si el valor de **n** es **positivo** extraerá **todos** los elementos a partir del que ocupa la posición **n** contando desde primero hasta el último según el orden de creación de los elementos.

Si el valor de **n** es **negativo** extraerá todos los elementos a partir del **enésimo**, esta vez, contando desde el último hasta el primero.

array_slice(\$a,n,m)

La función **array_slice** con *dos parámetros* permite extraer **una parte** de los valores de una matriz siguiendo estos criterios:

Si **n** y **m** son **positivos**, extraerá **m** elementos a partir del que ocupa la posición **enésima** de *primero a último*.

Cuando **n** es **negativo** y **m** es **positivo** se extraerán **m** elementos contados a partir del **enésimo**, esta vez recorriendo el array de *último a primero*.

En el caso en que **n** tenga valor **positivo** y **m** sea **negativo** extraerá los comprendidos entre el **enésimo** contado de *primero a último* y el **emésimo** contado desde el *último hasta el primero*.

Si **n** es **negativo** y **m** es también **negativo** extraerá los caracteres comprendidos entre el **enésimo** contado de *último a primero* y el **emésimo** contado en el mismo sentido.

En este caso se requiere que el valor absoluto de **n** sea mayor que el de **m**.

En caso de no cumplirse esta condición devolverá un *array vacío*.

Invertir el orden de un array

array_reverse(array)

Devuelve un nuevo array cuyos elementos están en orden inverso al del array original.

De esta forma el elemento que ocupaba la última posición pasa a ocupar la primera y así sucesivamente.

¡Cuidado!

Recuerda que *las posiciones iniciales de los elementos de un array* no tienen relación con sus índices sino con **la secuencia** en la que fueron creados.

Y otra cosa, mucho cuidado con la aplicación de todas estas funciones y con los **índices** de los arrays resultantes.

Fíjate en los ejemplos y verás que algunas estas funciones **reindexan** los resultados y los

```
}

echo "<h3>Inserta elementos iguales
      al principio o al final de una matriz</h3>";

$wz1=array_pad($a,25,"relleno");

foreach($wz1 as $clave=>$valor){
echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}

echo "<br>";

$wz2=array_pad($c,-17,"relleno");

foreach($wz2 as $clave=>$valor){
echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}

echo "<h3>Fusiona dos matrices</h3>";

$wz3=array_merge($a,$b);

foreach($wz3 as $clave=>$valor){
echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}
echo "<h3>Extrae el primer elemento de una matriz</h3>";

array_shift ($a);

foreach($a as $clave=>$valor){
echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}
echo "<br>";

array_shift ($c);

foreach($c as $clave=>$valor){
echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}
echo "<h3>Extrae el ultimo elemento de una matriz</h3>";

array_pop($a);

foreach($a as $clave=>$valor){
echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}
echo "<br>";

array_pop ($c);

foreach($c as $clave=>$valor){
echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}
echo "<h3>Extrae elementos de una matriz</h3>";

$zz1=array_slice($a,3);

foreach($zz1 as $clave=>$valor){
echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}
echo "<br>";

$zz2=array_slice($a,-3);

foreach($zz2 as $clave=>$valor){
echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}

echo "<br>";

$zz3=array_slice($b,3,4);

foreach($zz3 as $clave=>$valor){
echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}

echo "<br>";

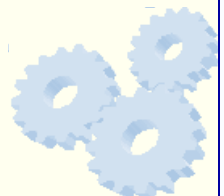
$zz4=array_slice($b,3,-2);

foreach($zz4 as $clave=>$valor){
echo "Clave: ",$clave," Valor: ",$valor,"<br>";
}

echo "<br>";

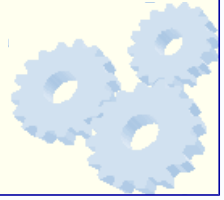
$zz5=array_slice($b,-5,-2);

foreach($zz5 as $clave=>$valor){
echo "Clave: ",$clave," Valor: ",$valor,"<br>";
```



convierten en escalares aún en el caso de que originalmente fueran asociativos.

```
}  
echo "<h3>Invierte los elementos de la matriz</h3>";  
  
$inv=array_reverse($C);  
  
foreach($inv as $clave=>$valor){  
echo "Clave: ",$clave," Valor: ",$valor,"<br>";  
}  
?>
```



ejemplo63.php

[Anterior](#) [Indice](#) [Siguiete](#)

