

Anexo V - Operaciones con arrays

1.1 Recuento de los valores contenidos en un array

Recuento de los elementos de un array:

```
$n=array_count_values(ar)
```

Genera un array, \$n, que tendrá como índices cada uno de los valores distintos *que contenga el array* ar, y como valores el resultado de contar el número de veces que se repite cada uno de los valores contenidos en el array inicial.

Búsqueda de elementos en un array

```
$clav=array_keys(arr)
```

Devuelve un array escalar (*clav*) que contiene como valores los *índices* del array inicial (*arr*).

```
$clav=array_keys(arr, valor)
```

Devuelve un array escalar (*clav*) cuyos valores son los índices de los elementos del array inicial cuyo valor coincide con el indicado mediante valor.

```
$valores=array_values(arr)
```

Esta función recoge en una nueva matriz (*valores*) todos los valores contenidos en otro array. Es una forma de conversión de un array asociativo en otro escalar.

1.2 Ordenación de arrays

Localización de valores en una matriz

```
in_array(valor,array)
```

La función in_array busca en la matriz (*array*) el valor (numérico o cadena) contenido en el parámetro *valor*. Si lo encuentra devuelve 1, y, si no existiera devolvería NUL.

Posicionamientos en una matriz

Mediante estas funciones se puede modificar la posición del *puntero interno* de una matriz y determinar los *índices* de los elementos a los que apunta en cada momento.

- **key(array)** Devuelve el índice del elemento de la matriz al que *apunta* en ese momento el *puntero interno* de la matriz.
- **reset(array)** Mueve el *puntero interno* a la posición del primer índice del array.
- **end(array)** Desplaza el *puntero interno* a la posición del último índice del array.
- **pos(array)** Mantiene el *puntero interno* en la posición del actual.
- **next(array)** Avanza el *puntero interno* en una posición respecto a la actual.
- **prev(array)** Retrocede el *puntero interno* en una posición respecto a la actual.

Ordenaciones de arrays

- **Ordenación por valores sin mantener índices:**
 - o **sort(array):** Ordena los valores del *array* en sentido *creciente* y lo *reindexa* asignando índice CERO al menor de los valores.
 - o **rsort(array):** Ordena la matriz en sentido *decreciente* de sus valores y la *reindexa* asignando índice CERO al mayor de estos.

- **Ordenación por índices:**
 - o **krsort(array):** Ordena la matriz según sus índices, en sentido creciente..
 - o **ksort(array):** Ordena la matriz por *índices* en sentido *decreciente*.
- Ordenación por valores manteniendo índices
 - o **asort(array):** Ordena la matriz según sus *valores* en sentido *creciente* y *mantiene* los índices del array original.
 - o **arsort(array):** Ordena la matriz por valores en sentido decreciente y sigue manteniendo los índices originales.
- Ordenación mediante función definida por usuario: PHP permite que el usuario pueda definir funciones en las que establezca sus criterios particulares de ordenación. Las funciones que permiten usar esta característica son:
 - o **uasort(array, funcion):** Ordena la matriz utilizando los criterios establecidos por la función definida por el usuario y mantiene los índices del array.
 - o **usort(array, funcion):** Ordena la matriz por valores utilizando los criterios definidos en la función de usuario y modifica los índices.
 - o **uksort(array, funcion):** Ordena la matriz por claves utilizando los criterios definidos en la función.

1.3 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.
- **shuffle(array):** Intercambia de modo *aleatorio* los valores de un array y los reindexa.
- **\$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.

1.3.1 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.
- **array_push(array, v1,v2,..):** Inserta al final de la matriz *array* los valores v1, v2,...(array_unshift y array_push dan a los nuevos elementos índices numéricos).
- **array_pad(array, n, var):** Inserta nuevos elementos en *array*, 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 n es positivo inserta los elementos al final del array, si es negativo los inserta al comienzo del mismo.
- **array_merge(\$a, \$b):** Crea un nuevo *array escalar* en el que se incluyen todos los elementos contenidos en los arrays \$a y \$b.

1.3.2 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):** Extrae una parte de los valores de una matriz. Si n y m son positivos, extrae m elementos a partir del que ocupa la posición enésima de *primero a último*. Cuando n es negativo y m es positivo extrae m elementos contados a partir del enésimo, recorriendo el array de *último a primero*. Si n es positivo y m negativo, extrae los comprendidos entre el enésimo contado de *primero a último* y el emésimo contado desde el *último hasta el primero*. Si ámbos son negativos, extrae los caracteres entre el enésimo contado de *último a primero* y el emésimo contado en el mismo sentido. En este caso requiere que valor absoluto de n sea mayor que el de m (o da un *array vacío*).

1.3.3 Invertir el orden de un array

- **array_reverse(array):** Devuelve un nuevo array cuyos elementos están en orden inverso al del original (*las posiciones iniciales de los elementos de un array no tienen relación con sus índices sino con la secuencia en la que fueron creados*).