

El bucle foreach

El bucle **foreach** es específico de los **array** y aplicable a ellos tanto si son *escalares* como si son de tipo *asociativo*.

Tiene dos posibles opciones. En una de ellas lee únicamente los valores contenidos en cada elemento del array. En el otro caso lee además los índices del array.

Lectura de valores

Utiliza la sintaxis:

```
foreach( array as var ){
...instrucciones...
}
```

donde *array* es el **nombre del array** (sin incluir índices ni corchetes), **as** es una *palabra* obligatoria y *var* el nombre de una variable (puede ser creada al escribir la instrucción ya que no requiere estar previamente definida).

Las instrucciones escritas *entre* las { } permiten el tratamiento o visualización de los valores obtenidos.

La variable *var* no podrá ser utilizada para *guardar* valores.

Hemos de tener en cuenta que su valor se *rescribe* en cada iteración del bucle y que al acabar este sólo contendrá el último de los valores leídos.

Lectura de índices y valores

Con una sintaxis como la que sigue se pueden leer no sólo los valores de un array sino también sus índices.

```
foreach( array as v1 => v2 ) {
...instrucciones...
}
```

donde *array* es el nombre de la matriz, **as** es una *palabra* obligatoria, *v1* es el nombre de la variable que recogerán los índices, los caracteres **=>** (son obligatorios) son el separador entre ambas variables y, por último, *v2* es el nombre de la variable que recoge el valor de cada uno de los elementos del array.

Tanto esta función como la anterior realizan una *lectura secuencial* que comienza en el **primer valor** del array.

Arrays bidimensionales

Cuando se trata de arrays bidimensionales la lectura de los valores que contienen sus elementos requiere el uso de **dos bucles anidados**.

Cuando un array de este tipo es sometido al bucle **foreach** se extrae como *índice* el **primero de ellos** y como *valor* un **array unidimensional** que tiene como *índice* el **segundo** del array original y como valor el de aquél.

La lectura de los valores de cada elemento requiere utilizar un segundo bucle que los vaya extrayendo a partir del array unidimensional obtenido por medio

El bucle foreach

```
<?
/* definimos un array escalar utilizando la sintaxis
   nombre del array=array (valores de los elemento separados por comas)
   si los valores son números no es necesario encerrarlos entre comillas */
$a=array("a","b","c","d","e");
/* definamos ahora un nuevo array, esta vez asociativo
   utilizando la sintaxis clave => valor tal como puedes ver */
$b=array(
    "uno" =>"Primer valor",
    "dos" =>"Segundo valor",
    "tres" =>"Tercer valor",
);
# establecemos el bucle que leerá el array $a
# recogiendo en la variable $pepe los valores extraídos
# y escribimos lo valores
foreach($a as $pepe) {
echo $pepe,"<br>";
};
# repetimos el mismo proceso, ahora con $b que es un array asociativo
foreach($b as $pepe) {
echo $pepe,"<br>";
};
?>
```

ejemplo51.php

```
<?
$a=array("a","b","c","d","e");
$b=array(
    "uno" =>"Primer valor",
    "dos" =>"Segundo valor",
    "tres" =>"Tercer valor",
);

# en este caso extraeremos índices y valores de ambos arrays
# usaremos $pepe para recoger los índices y $pepe para recoger los valores
# y separaremos ambas variables por => que es el separador obligatorio
# para estos casos

foreach($a as $pepe=>$pepa) {
    echo "Índice: ",$pepe," Valor: ",$pepa,"<br>";
};
foreach($b as $pepe=>$pepa) {
    echo "Índice: ",$pepe," Valor: ",$pepa,"<br>";
};
?>
```

ejemplo52.php

El bucle foreach en arrays bidimensionales

```
<?
# definamos un array bidimensional
$z=array(
    0 => array (
        0 => 34,
        1 => 35,
        2 => 36,
    ),
    1 => array (
        0 => 134,
        1 => 135,
        2 => 136,
    )
);
# intentemos leer índices y valores mediante un bucle foreach
# y veamos como los valores que extraemos son Arrays (unidimensionales)
# consecuencia del caracter bidimensional del array original

foreach($z as $pepe=>$pepa) {
    echo "Índice: ",$pepe," Valor: ",$pepa,"<br>";
};
/* anidemos dos bucles foreach de la siguiente forma
   en el primero extraemos un array que es tratado por
   el segundo foreach para extraer el segundo índice
   y el valor realmente contenido en ese elemento bidimensional */
foreach($z as $ind1=>$valor1) {
    foreach($valor1 as $ind2=>$valorReal) {
        echo "Ind. 1: ",$ind1,"Ind. 2: ",$ind2," Valor: ",$valorReal,"<br>";
    }
}
```

del bucle previo.

La sintaxis sería de este tipo:

```
foreach($a as $i1=>$na){
```

```
foreach($na as $i2=>$val){
```

..*\$i1 es el primer índice...*

..*\$i2 es el segundo índice...*

..*\$na nuevo array*

..*\$valor es el valor*

.....

```
}
```

```
}
```

En el caso de arrays con dimensiones superiores sería necesario proceder del mismo modo, y habría que *utilizar tantos bucles foreach como índices* contuviera el array.

```
};  
};  
?>
```

ejemplo52a.php

Ejercicio nº 25

Crea un array bidimensional que contenga los nombres de cinco alumnos y las calificaciones de tres materias (biología, física y latín por ejemplo). El primer índice puede ser el número de lista y los segundos pueden ser de tipo asociativo (*'nombre', 'biología', 'física', 'latín'*, por ejemplo). Asigna valores a los elementos del array y completa el script de forma que se visualice un listado con los nombres de los alumnos y las calificaciones de cada una de las materias.

Anterior Índice Siguiente

