



Arreglos PHP

Se define a un arreglo como un grupo de elementos relacionados entre sí por medio de índices. Los arreglos pueden ser de una o más dimensiones, los de una dimensión, son llamados comúnmente "vectores".

A diferencia con el lenguaje C, en PHP, un vector puede tener elementos de distintos tipos.

Para hacer referencia a un elemento del vector, se utiliza un índice, que indica la dirección en donde se encuentra un determinado valor. El índice en un arreglo comienza siempre por cero. (Mas adelante se verá que el índice de un vector, no necesariamente debe ser un número entero, sino que también puede ser un texto).

Ejemplo Práctico 1:

Almacenar los nombres de los días de la semana en un vector y luego imprimirlos uno debajo de otro.

```
<Html>
<Title> Ejemplo 1 </Title>
<Body>
<?PHP
    // Inicializacion del Vector

    $dia[0] = "Domingo";
    $dia[1] = "Lunes";
    $dia[2] = "Martes";
    $dia[3] = "Miércoles";
    $dia[4] = "Jueves";
    $dia[5] = "Viernes";
    $dia[6] = "Sábado";

    // Impresion del vector

    for($i=0; $i<7; $i++)
    {
        echo ($dia[$i] . "<Br>") ;
    }
?>
</Body>
</Html>
```

Comentario:

Se inicializa el vector indicando el número que le corresponde a



cada posición entre corchetes [] y asignando el valor que se desea almacenar en dicha posición.

Un vector, en PHP, puede contener elementos de distintos tipos de datos, es decir, un elemento puede ser un número entero, otro una cadena, otro un número con decimales, etc.

Un modelo de este caso se puede observar en el siguiente ejemplo.

Ejemplo Práctico 2:

Almacenar en un vector los datos personales de un empleado y luego mostrarlos en pantalla.

```
<Html>
<Title> Ejemplo 2 </Title>
<Body>
<?PHP
    // Inicializacion del Vector

    $Empleado[0] = 4371;
    $Empleado[1] = "Martinez Leandro";
    $Empleado[2] = "27.643.742";
    $Empleado[3] = 1429.54;
    $Empleado[4] = "Arquitecto";

    // Impresion del vector

    echo ("Legajo: " . $Empleado[0] . "<Br>");
    echo ("Nombre: " . $Empleado[1] . "<Br>");
    echo ("DNI : " . $Empleado[2] . "<Br>");
    echo ("Sueldo: " . $Empleado[3] . "<Br>");
    echo ("Profesion: " . $Empleado[4] . "<Br>");
    ?>
</Body>
</Html>
```

Existen varias maneras de inicializar vectores en PHP. A continuación se describen algunos ejemplos.

```
Pais[] = "Argentina";
Pais[] = "Uruguay";
Pais[] = "Brasil";
Pais[] = "Chile";
```

En este caso se observa que no es necesario colocar el número de índice, ya que PHP lo asigna automáticamente para cada valor, comenzando siempre



desde cero.

Otra forma de inicializar un vector, es a través del constructor **array**, como se muestra en el siguiente ejemplo:

```
Pais =array("Argentina","Uruguay","Brasil","Chile");
```

También se puede definir un arreglo asociando explícitamente el índice a un valor, como se indica a continuación:

```
$Frutas = array(0 => "Manzana",  
                1 => "Naranja",  
                2 => "Pera",  
                3 => "Ananá");
```

Además, los índices, pueden no ser obligatoriamente consecutivos, ni tampoco comenzar de cero, ni tampoco ser un número. (Ver ejemplos mas adelante). Se puede conocer la cantidad de elementos que tiene un vector, para ello se utiliza la función **count(vector)**. Esta función acepta como parámetro el nombre del vector y devuelve la cantidad de elementos del mismo.

Ejemplo Práctico 3:

Cargar en un vector artículos de librería y luego imprimir la cantidad de ellos.

```
<Html>  
<Title> Ejemplo 3 </Title>  
<Body>  
<?PHP  
    // Inicializacion del Vector  
  
    $Articulos =array("Lápis","Goma","Hoja","Tinta");  
  
    // Impresion del vector  
  
    $Cantidad = count($Articulos);  
    echo ("La cantidad de Artículos son: " . $Cantidad);  
  
?>  
</Body>  
</Html>
```

Un vector en PHP puede tener elementos en cualquier posición, por lo tanto, se puede cargar un vector con posiciones **no** consecutivas, sino en forma



totalmente aleatoria. Para poder recorrer este tipo de vectores se utilizan las funciones **next()** y **prev()**.

Ejemplo Práctico 4:

Cargar los nombres de personas en cualquier posición.

```
<Html>
<Title> Ejemplo 4 </Title>
<Body>
<?PHP
    // Inicializacion del Vector

    $Nombre[10] = "Marcelo";
    $Nombre[20] = "Alicia";
    $Nombre[15] = "Alejandra";
    $Nombre[35] = "Mario";
    $Nombre[12] = "Alberto";

    // Impresion del vector

    reset($Nombre);
    echo("<H2>" . "Vector de Nombres");
    echo("<H3>" . "<Hr>");
    do
    {
        // Buscar Posicion especificada
        $i = key($Nombre);

        // Buscar Contenido en esa posicion
        $Valor = current($Nombre);
        echo("Posición: " . $i . " - ");
        echo("Contenido: " . $Valor);
        echo("<Br>");
    }
    while (next($Nombre));
?>
</Body>
</Html>
```

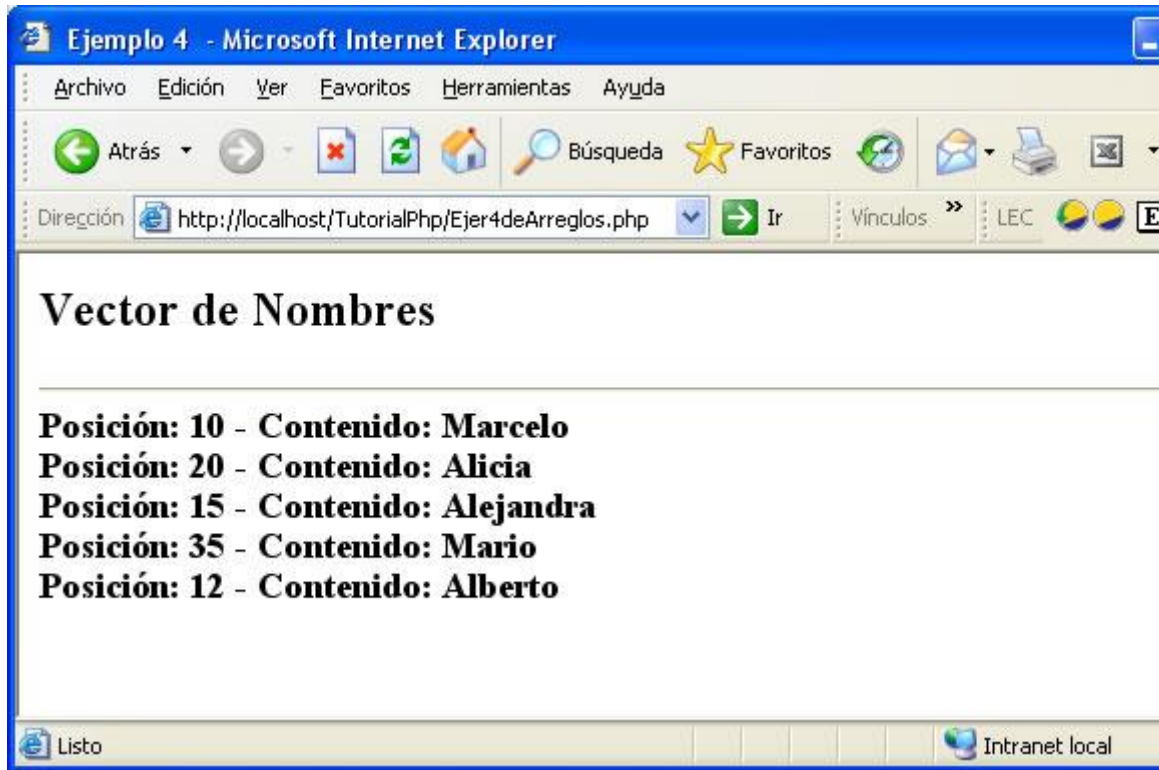
Comentario:

Aquí se observa que los nombres fueron cargados en posiciones aleatorias. Con la función **reset()**, se logra llevar el puntero al principio del vector y para poder encontrar la posición de un elemento se utiliza la función **key()**, la cual acepta como parámetro el vector y devuelve la posición. Luego para poder obtener el contenido en esa posición se utiliza la función **current()**, que acepta el vector y devuelve el valor almacenado. La función **next()** avanza el puntero a



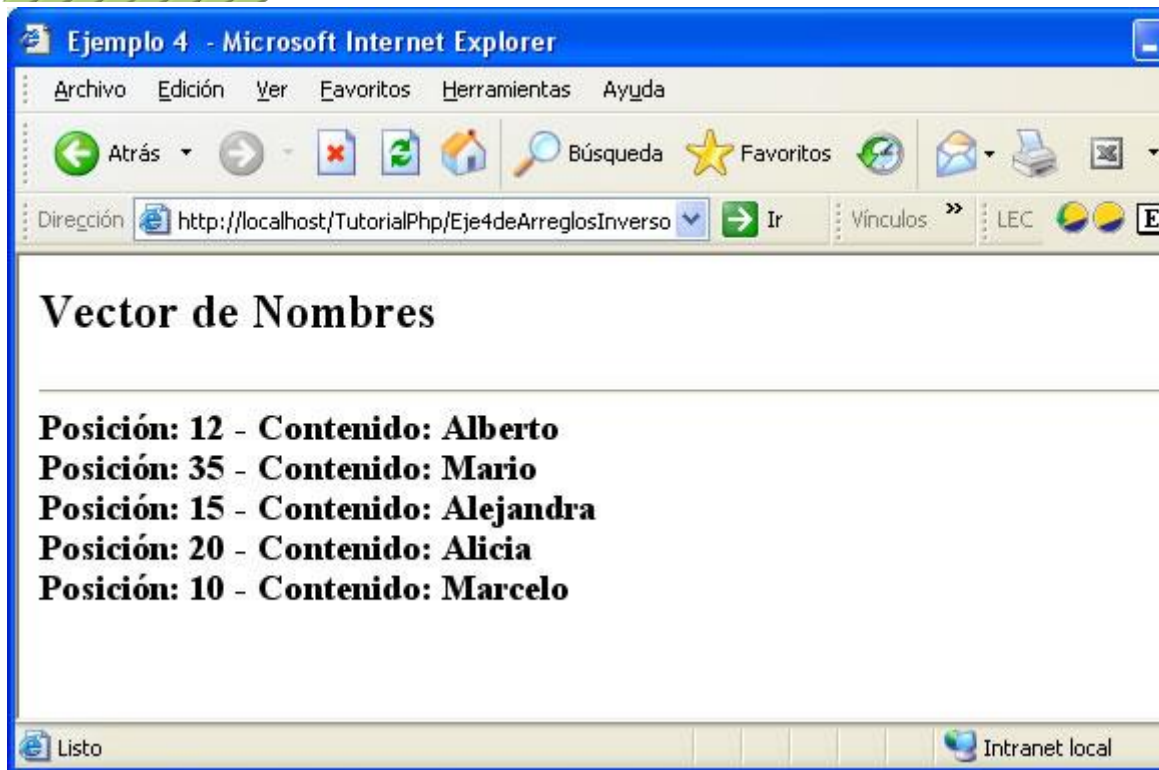
la posición siguiente, si se ha llegado al final del vector, esta función devuelve **false**.

La salida de este ejemplo es la siguiente:



En caso de querer imprimir los datos del vector en forma invertida, solamente se debe cambiar la instrucción **reset(\$Nombre)** por: **end(\$Nombre)** y la instrucción: **while (next(\$Nombre))** por : **while (prev(\$Nombre))**. Como se dijo anteriormente, las funciones **next()** y **prev()** devuelven **false** al llegar al final y principio del vector respectivamente, pero tambien devuelven **false** cuando el valor de un elemento es cero.

La salida quedará de la siguiente forma:



La función **end()** lleva el puntero al final del vector y con la función **prev()**, se retrocede el puntero hasta el anterior. Cuando se llegó al principio del vector, la función **prev()** devuelve **false** y sale del ciclo.

Otra forma de recorrer un vector de estas característica es utilizando las funciones **list()** y **each()**, de la siguiente manera:

Ejemplo Práctico 5:

Cargar Legajos de personas con sus respectivos Nombres

```
<Html>
<Title> Ejemplo5.php </Title>
<Body>
<?Php
    // Inicializacion del Vector

    $Nombre[100] = "Javier";
    $Nombre[200] = "Cintia";
    $Nombre[150] = "Ricardo";
    $Nombre[350] = "Raúl";
    $Nombre[120] = "Guillermo";

    // Impresion del vector
```



```
reset($Nombre);

echo "<H2>". "Vector de Nombres";
echo "<H3>". "<Hr>";
while (list($i,$Valor)=each($Nombre))
{
    echo "Legajo: " . $i . " - ";
    echo "Nombre: " . $Valor;
    echo "<Br>";
}
?>
</Body>
</Html>
```

La función **list()**, almacena en los parámetros (\$i, \$Valor), el índice y el valor devuelto por la función **each()**, que tiene como parámetro el vector puesto en juego. Además, la función **list()**, avanza automáticamente el puntero al siguiente elemento del vector, y en caso de que el vector haya llegado al final, la función devuelve false.

Vectores con índice de Texto

Un vector en PHP, no solamente debe contener índice numérico, sino también, puede ser una letra o un texto.

Ejemplo Práctico 6:

Cargar en un vector algunas ciudades del mundo, de manera que el índice del vector contenga los tres primeros caracteres de la ciudad almacenada.

```
<Html>
<Title> Ejemplo 6.php </Title>
<Body>
<?Php
    // Inicializacion del Vector
    $Ciudad = array("Par" =>
"Paris",
                                "Lon"
=> "Londres",
                                "Ate"
=> "Atenas",
                                "Ber"
=> "Berlin",
```



```
"Lim"
=> "Lima");

echo "<H2>". "Vector de
Ciudades";
echo "<H3>". "<Hr>";
while
(list($i,$Valor)=each($Ciudad))
{
    echo "Posición: " . $i . " -
";
    echo "Contenido: " . $Valor;
    echo "<Br>";
}
?>
</Body>
</Html>
```

La salida obtenida es:

