**Conceptos generales sobre el uso de arrays**

Un **array** es un tipo de datos que nos permite almacenar varios valores. Cada miembro del **array** se almacena en una posición a la que se hace referencia utilizando un valor clave. Las claves pueden ser numéricas o asociativas.

**// array mediantes subíndices**

$moduloscurso1 = array( "Programación", "Bases de datos", ..., "Desarrollo web en entorno servidor");

**// array numérico/ asociativo**

$moduloscurso1 = array(0 => "Programación",1 => "Bases de datos",..., 9 => "Desarrollo web en entorno servidor",

**// array asociativo**

$moduloscurso2 = array("PR" => "Programación", "BD" => "Bases de datos", ..., "DWES" => "Desarrollo web en entorno servidor");

// **cadena de texto**

$modulo = "Desarrollo web en entorno servidor";

// $modulo[3] == "a";

**// array bidimensional**

$ciclos = array(

"DAW" => array ("PR" => "Programación", "BD" => "Bases de datos", ..., "DWES" => "Desarrollo web en entorno servidor"),

"DAM" => *array* ("PR" => "Programación", "BD" => "Bases de datos", ..., "PMDM" => "Programación multimedia y de dispositivos móviles")

);

Asignación:

/ array numérico

$moduloscurso1 [0] = "Programación";

$moduloscurso1 [1] = "Bases de datos";

...

$moduloscurso1 [9] = "Desarrollo web en entorno servidor";

// array asociativo

$moduloscurso2 ["PR"] = "Programación";

$moduloscurso2 ["BD"] = "Bases de datos";

...

$moduloscurso2 ["DWES"] = "Desarrollo web en entorno servidor";

Ni siquiera es necesario que especifiques el valor de la clave. Si la omites, el array se irá llenando a partir de la última clave numérica existente, o de la posición 0 si no existe ninguna:

$moduloscurso1 [ ] = "Programación";

$moduloscurso1 [ ] = "Bases de datos";

* **Para recorrer los elementos de un array,** en PHP puedes usar un bucle específico: **foreach.** Utiliza una variable temporal para asignarle en cada iteración el valor de cada uno de los elementos del arrays. Puedes usarlo de dos formas:

En **$key** pone la llave  
En **$value** pone el valor

* Recorriendo sólo los elementos: foreach (expresion\_array as $value)

$moduloscurso = arrays("PR" => "Programación", "BD" => "Bases de datos", ..., "DWES" => "Desarrollo web en entorno servidor");

foreach ($moduloscurso as $modulo) {

print "Módulo: ".$modulo. "<br />"

}

* Recorriendo los elementos y sus valores clave de **forma simultánea** foreach (expresion\_array as $key=>$value):

$moduloscurso = arrays("PR" => "Programación", "BD" => "Bases de datos", ..., "DWES" => "Desarrollo web en entorno servidor");

foreach ($moduloscurso as $codigo => $modulo) {

print "El código del módulo ".$modulo." es ".$codigo."<br />"

}

**Foreach** permite iterar sobre arrays sin saber cuantos elementos hay en el array.

Ver manual: <http://es.php.net/manual/es/ref.array.php>

*//Array lineal de 4 elementos*

$X[0]=1;

$X[1]=“una cadena”;

$X[2]=3;

$X[ ]=“ultimo elemento”;

***//mi cuenta bancaria***

$micuenta[“numero\_cuenta”]=20001;

$micuenta[“titular”]=“Jose Perez”;

$micuenta[“saldo”]=12000;

$micuenta[“fecha\_pago”]=”01012006”;

***//mi cuenta bancaria***

$micuenta=array(“numero\_cuenta”=>200, “titular”=>“Jose Perez”, “saldo”=>12000, “u\_pago”=>01012006) ;

***//Array de 2x2 elementos***

$X[0][0]=1;

$X[0][1]=0;

$X[1][0]=-1;

$X[1][1]=-2;

• La funcion “print\_r ( )” muestra los elementos de un array

• La manipulación de los elementos requiere uso de sentencias iterativas (foreach, for, etc.)

<http://www.w3schools.com/php/func_array.asp>