

Desarrollo de Portales WEB *Lenguaje PHP*

INSTITUTO SUPERIOR TECNOLÓGICO



R.M. N° 420 - 94 - ED

Gonzalo Anchante Hurtado

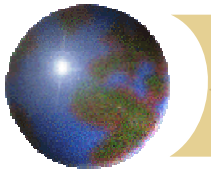
Docente Carrera Profesional

Computación e Informática



Trabajo de Arrays

- ❖ Un Array es una colección de valores. Los array pueden ser unidimensionales (vectores), bidimensionales (matrices) y multidimensionales (más de dos dimensiones)
- ❖ Los arrays se utilizan ampliamente en el lenguaje PHP. Se utiliza el delimitador [] para acceder a los diferentes elementos del vector.



Trabajo de Arrays

- Se puede crear sin tener que declararlo:

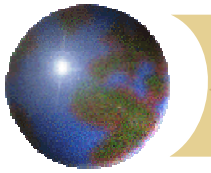
```
$dias[0]=31;
```

```
$dias[1]=28;
```

- Luego de estas dos líneas, tenemos creado un vector de dos elementos, a los cuales accedemos por un subíndice que comienza a numerarse desde cero.

```
echo $dias[0];      //31
```

```
echo $dias[1];      //28
```



Trabajo de Arrays

- El vector, como podemos ver, puede ir creciendo en forma dinámica, es decir que si ahora hacemos:

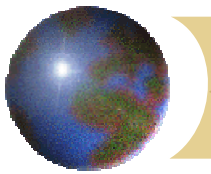
```
$dias[2]=31;
```

- el vector tiene 3 componentes.
- También podemos obviar el subíndice cuando asignamos los valores:

```
$dias[]=31;
```

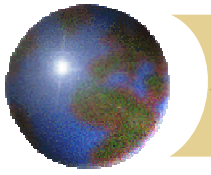
```
$dias[]=28;
```

```
$dias[]=31;
```



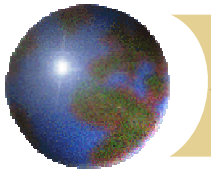
Trabajo de Arrays

```
<?php
    $nombres[]="juan";
    $nombres[]="pedro";
    $nombres[]="ana";
    for($f=0;$f<count($nombres);$f++) {
        echo $nombres[$f];
        echo "<br>";
    }
?>
```



Trabajo de Arrays

- Arrays escalares: denominado también, arreglo simple. Esta formado por un conjunto de valores ordenados respecto a un índice de tipo entero. Este índice indicará la posición del elemento dentro de esta colección ordenada.



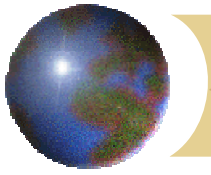
Complemento de Arrays

✿ Forma2: crear una matriz usando el constructor `array()`.

▣ Sintaxis: `nombre=array(elementos);`

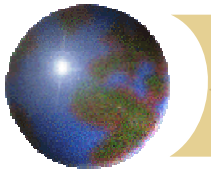
Ejemplo:

```
$matriz=array("isabel","ford",null,165);
```



Array asociativo

- ❖ Creando un array asociativo. Están formados por un conjunto de valores que están ordenados respecto a un índice de tipo string.
- ❖ Se puede crear en base al constructor Array o indicándolo mediante un índice fijo.



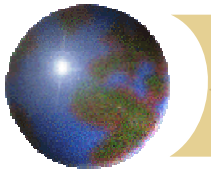
Array asociativo

- ⊕ Este tipo de vectores no es común a otros lenguajes, pero en PHP son de uso indispensable en distintas situaciones (ya lo empleamos cuando recuperamos información de un formulario accediendo al vector `$_REQUEST` que crea PHP en forma automática, cuando accedamos a datos de una base de datos también lo emplearemos etc.)
- ⊕ Un vector asociativo permite acceder a un elemento del vector por medio de un subíndice de tipo string.



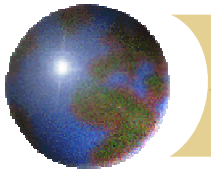
Array asociativo

- ❖ Inicialmente uno piensa que esto nos complica las cosas, como veremos más adelante la realidad nos demuestra lo contrario.
- ❖ Como ejemplo, consideremos que deseamos guardar en un vector el DNI, nombre y dirección de una persona. Empleando un vector con subíndice entero la solución sería:



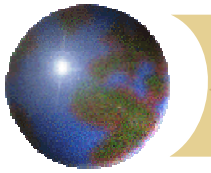
Array asociativo

```
<?php  
$registro[]="20456322";  
    $registro[]="Martinez Pablo";  
    $registro[]="Colon 134";  
?>
```



Array asociativo

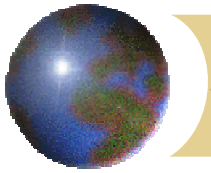
- ❖ De esta forma debemos recordar que cuando deseamos mostrar el nombre de la persona debemos acceder al subíndice 1. Esto es sencillo si tenemos un vector con tres elementos, pero que sucede si debemos almacenar 20 datos relacionados en un vector?



Array asociativo

- ❖ Un vector asociativo se define de la siguiente forma:

```
<?php  
$registro['dni']="20456322";  
$registro['nombre']="Martinez Pablo";  
$registro['direccion']="Colon 134";  
echo $registro['nombre'];  
?>
```

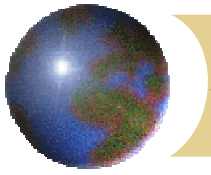


Arreglo asociativo

- `Matriz1=array("modelo"=>"cougar","marca"=>"ford","cc"=>250);`
- `Matriz2['modelo']="cougar";`
- `Matriz2['marca']="ford";`
- `Matriz2['cc']=250;`

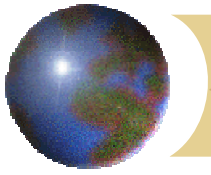
Para visualizar:

- ▣ `echo "matriz1['modelo'];`
- ▣ `echo "matriz2['modelo'];`



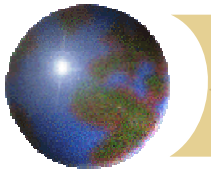
Arreglo asociativo

● EJERCICIO: **MATRICESASOC.PHP**



ORDENAR MATRICES.

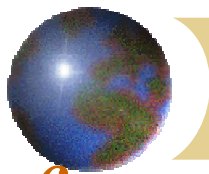
- ✚ A menudo resulta útil ordenar datos relacionados que estén almacenados en una matriz. es sencillo tomar una matriz y ordenarla



función sort()

✚ Esta función permite ordenar los datos de una matriz de manera ascendente alfabético.

- sintaxis: `sort($matriz);`

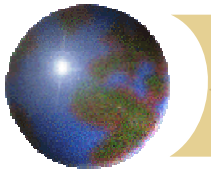


función sort()

```
<?php
```

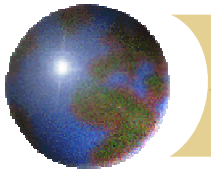
```
echo "<body bgcolor='orange'>";
echo "<h2>ESTAMOS EN MATRICES</h2>";
$matriz =array(5,4,6,"ahora","seguimos","clase","en","flash","asp");
for ($cont=0;$cont<=count($matriz);$cont++)
{
    echo $matriz[$cont]."<br>";
}

echo "<hr>";
echo "<h2>ahora ordenada</h2>";
sort($matriz);
for ($cont=0;$cont<=count($matriz);$cont++)
{
    echo $matriz[$cont]."<br>";
}
```



Función sort

- ❖ Ten en cuenta que la función sort discrimina entre Mayúsculas y minúsculas al momento de ordenar y que lo hará de la forma siguiente: primero mayúsculas y luego minúsculas.

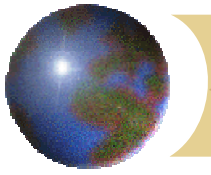


Parámetros del sort

- Existe un segundo parámetro que permite indicar el tipo de ordenamiento a efectuar. Es opcional.

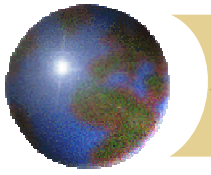
- SORT_NUMERIC
- SORT_STRING
- SORT_REGULAR (PRED)

• sintaxis: `sort(matriz, tipo);`



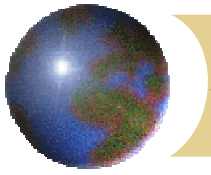
Función asort

- ✚ Si quisiéramos utilizar una matriz asociativa para almacenar artículos y sus precios necesitaríamos utilizar diferentes tipos de funciones de ordenación para mantener agrupados las claves y sus valores.



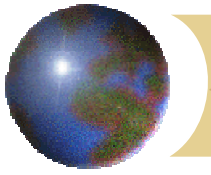
Función asort

- se ordena la matriz en función del valor de cada elemento. En la matriz los valores son las notas numéricas y las claves son las notas textuales.



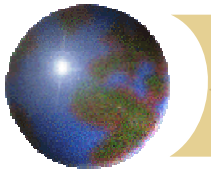
Función ksort

- ✚ Esta función ordena la matriz por la clave, lo inverso a la función asort.
- ✚ Se puede aplicar para cualquier estructura de vector. Unidimensional, multidimensional, etc.



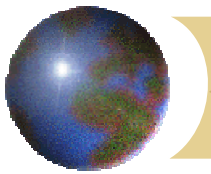
Función shuffle

- ⊕ Esta función ordena la matriz de forma aleatoria. Sin considerar criterio, simplemente hace un random



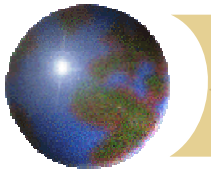
Ordenación inversa

- las funciones `sort()`, `asort()`, `ksort()` tienen funciones inversas que se establecen colocando la letra `r` antes del nombre de la función



Ordenación inversa

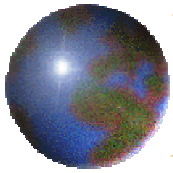
```
27 echo "<hr>";
28 echo "<h2>ahora ordenada descendientemente</h2>";
29 rsort ($matriz);
30 for ($cont=0;$cont<=count ($matriz);$cont++)
31 {
32     echo $matriz[$cont]. "<br>";
33 }
34
```



Unir los valores de la matriz

✚ Para unir los valores de la matriz se debe de utilizar la función join. Esta función permite unir los elementos de un vector y retornarlos o almacenarlos dentro de una variable.

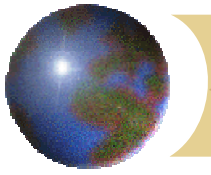
- Sintaxis: `$variable = join($matriz,"separador");`



Join

```
$precio=array("alicate" =>20,"catorce"=>14,"ochenta y cinco"=>85);  
$to=join($precio," / ");  
echo $to."<br>;
```

?>



Ejercicios

✚ Ejercicio: **formulariomatriz.php**

✚ Ejercicio: **EJECICIOMATRIZ.PHP**