

# Desarrollo de sitios web con PHP y MySQL



## Tema 4: Matrices

Ana M<sup>a</sup> Feroso García  
afermosoga@upsa.es

# Tema 4: Matrices

1. Sintaxis básica
2. Tratamiento de arrays con foreach
3. Funciones para trabajar con matrices
4. Arrays multidimensionales

# Matrices

- **Sintaxis:**

```
array ([clave =>] valor, ...)
```

- La clave es una cadena o un entero no negativo. El valor puede ser de cualquier tipo válido en PHP, incluyendo otro array

- **Ejemplos:**

```
$color = array ('rojo'=>101, 'verde'=>51, 'azul'=>255);  
$medidas = array (10, 25, 15);
```

- **Acceso:**

```
$color['rojo'] // No olvidar las comillas  
$medidas[0]
```

- El primer elemento es el 0

# Tratamiento de arrays con foreach

- La estructura de control **foreach** permite iterar sobre arrays

- Sintaxis:

```
foreach (expresión_array as $valor)  
    sentencia
```

```
foreach (expresión_array as $clave => $valor)  
    sentencia
```

- Ejemplos:

```
foreach ($color as $valor)
```

```
    print "Valor: $valor<BR>\n";
```

```
foreach ($color as $clave => $valor)
```

```
    print "Clave: $clave; Valor: $valor<BR>\n";
```

- Salida:

```
Valor: 101
```

```
Valor: 51
```

```
Valor: 255
```

```
Clave: rojo; Valor: 101
```

```
Clave: verde; Valor: 51
```

```
Clave: azul; Valor: 255
```

# Funciones para trabajar con arrays

- Sintaxis: `nom_funcion(array)`
  - `current`: Devuelve el valor del elemento actual
  - `each`: devuelve el actual y luego avanza uno
  - `next`: devuelve y avanza la siguiente
  - `prev`: devuelve y avanza al anterior al actual
  - `reset, end`: posicionar y devolver el primero y último elemento respectivamente
  - `count` y `sizeof`: ambas devuelven el número de elementos de la matriz
  - `sort`: ordenación ascendente
  - `asort` y `krsort`: ordenar matriz asociativa por valor o por clave respectivamente
  - `rsort, arsort, krsort`: igual que las anteriores pero en orden inversor
  - `usort($array, 'func_ord')`: ordenar aplicando la función de usuario que se pasa como segundo parámetro.
  - `array_count_values()`: Calcula la frecuencia de cada uno de los elementos de un array
  - `array_search()`: Busca un elemento en un array

# Arrays multidimensionales

- Aquellas tablas en que cada elemento es otro array

- Ejemplo declaración array multidimensional:

- `$calendario=array(array(1, "enero", 31), array(2, "febrero", 28))`
- `$calendario[]=array(1, "enero", 31)`  
`$calendario[]=array(2, "febrero", 28)`
- `$calendario[0]=array(1, "enero", 31)`  
`$calendario[1]=array(2, "febrero", 28)`

- Ejemplo:

```
$calendario[]=array(1, "enero", 31)
$calendario[]=array(2, "febrero", 28)
while (list($clave, $valor)=each($calendario)) //foreach ($calendario as $valor)
{
    $cadena=$valor[1];
    $cadena.=" ES EL MES NÚMERO ". "$valor[0];
    $cadena.=" y tiene ".$valor[2]. "días<br>";
    echo=$cadena;
}
```

- **list(\$clave,\$valor):** es un operador de asignación que asigna valores a una lista de variables. En este caso a los extraídos de una tabla con la función each().

# Ejercicio

- Implementar una agenda como un array en que cada posición de la agenda corresponda a un contacto con nombre y fecha de nacimiento (dd-mm-aa) .
- Implementar una función que pasándole la fecha de nacimiento de una persona, nos calcule su edad.
- El programa ha de mostrar por pantalla el nombre, edad y teléfono de aquellos contactos de la agenda cuyo nombre comience y termine por “A”