

PHP 06

Arrays

NUMERIC

```
NAME      KEYWORD  ARRAY KEYS  ARRAY VALUES
$name = array( [0][1][2][3][4][5]
               4,3,5,2,1,0 );
```

ASSOCIATIVE

```
KEY NAME  KEY VALUE
$name = array( 'Sam' => 23,
               'Bob' => 35, 'Jim' => 42 );
                        |
                        ROCKET
```

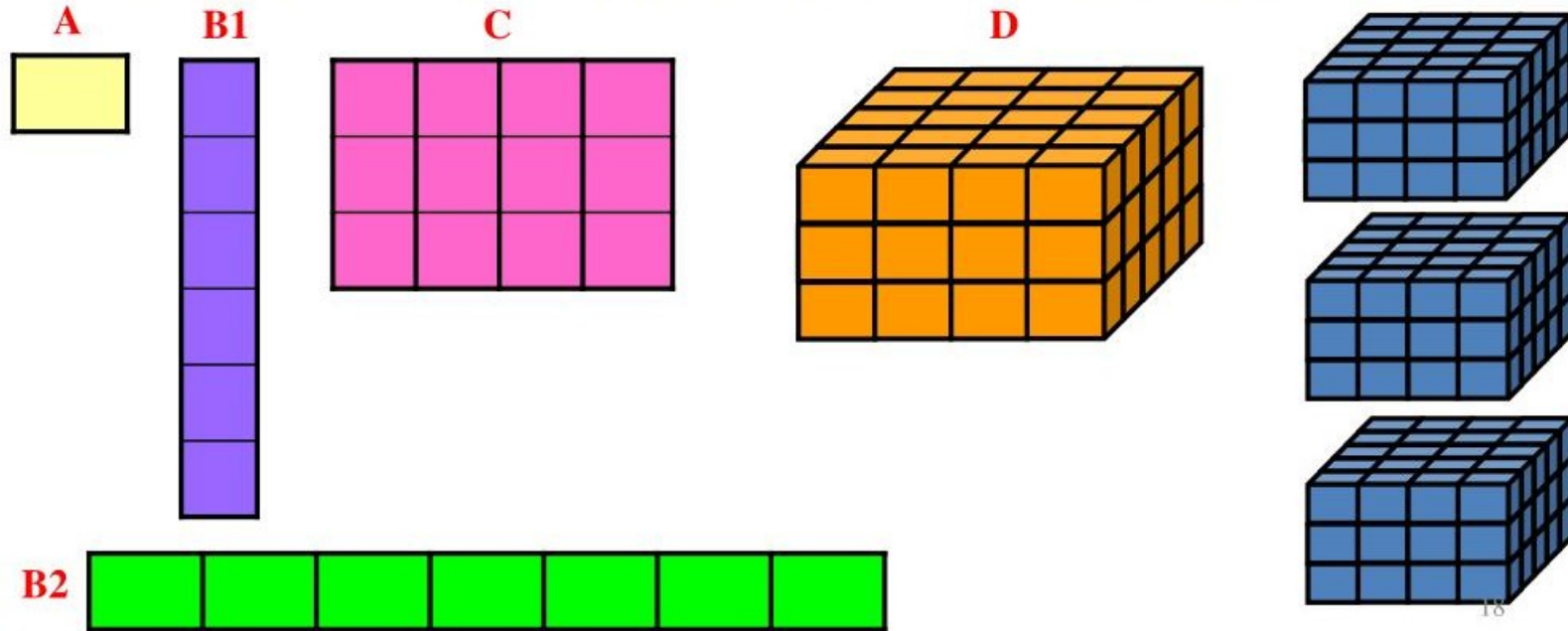
9)Arrays

- **Array**: conjunt de **valors independents** guardats en una **mateixa variable**.
- En **PHP**: **no és necessari** que siguin totes del **mateix tipus**:
 - Però el més **habitual** és que **sí que ho siguin**.
- **Accés** als **elements individuals** d'un array: nombre **enter** anomenat **índex**.
 - **Primer** element: 0
 - **Últim** element: $n-1$
 - **n** : **dimensió**, **grandària**, **mida** o **longitud** de l'array.
 - **\$empleat[23]** : l'empleat en la posició 23 de l'array.

- Depenent de la **quantitat de dimensions** que tinguen els arrays poden ser:
 - D'**una dimensió** (**vectors**). Per **exemple**:
 - `$colors[5]`
 - De **dues dimensions** (**matrius**). Per **exemple**:
 - `$butaca[3][14]`
 - De **tres o més dimensions** (**multidimensionals**). Per **exemple**:
 - `$data[2021][12][1]`

- Dimensions** dels arrays, gràficament:

`int A, B1[6], B2[6], C[3][4], D[3][4][5], E[3][4][5][3];`



- Per **accedir** als **elements de l'array**: **claudàtors []**.
 - Dins hi haurà un **localitzador** o **índex** (nombre **sencer**).
- **Tipus** primitius que hi podem guardar:
 - **string**, **enter**, **decimal** (**float**), **booleà**.
- En un **mateix array** podem guardar valors de **diferents tipus** simultàniament.
- Per **exemple**:
 - `$valors[0] = "Xàtiva";`
 - `$valors[1] = true;`
 - `$valors[2] = 7.5;`

- En PHP **no és necessari declarar els arrays** abans de poder utilitzar-los.
- Quan es **defineixen elements** d'un array:



- PHP **reconeix automàticament** que es tracta d'un **array** sense necessitat de declaració prèvia.

Vectors (Arrays duna dimensió)

- **Vectors**: arrays que només contenen **una dimensió** (un índex).

- Forma general d'**ús d'un array** :

- `$nomDeVariableArray [índex] = valorAssignat;`

- **Exemple:**

<?php

```
$estacio[0] = "Primavera";  $estacio[1] = "Estiu";  
$estacio[2] = "Tardor";    $estacio[3] = "Hivern";  
echo $estacio[2]; // Mostra Tardor  
$nom[0] = 7;  $nom[1] = 11; $nom[2] = 15;  
echo $nom[1]; // Mostra 11
```

Assignació de
valors array:
Mètode 1

- PHP permet **usar arrays sense especificar** els números dels **índexs**:
 - Els pot posar **automàticament**.
- **Exemple 02:**

```
<?php
```

```
$ciutat[] = "Xàtiva" ; // Li assigna l'índex 0  
$ciutat[] = "Canals" ; // Li assigna l'índex 1  
$ciutat[] = "Genovés" ; // Li assigna l'índex 2  
$ciutat[] = "La Llosa" ; // Li assigna l'índex 3  
echo $ciutat[3]; //Mostra La Llosa
```

```
?>
```

Assignació de
valors array:
Mètode 2

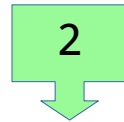
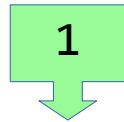
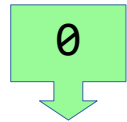
- **Altra forma** d'assignar valors a un array vector és la següent:
`$nomDeLArray = array(valorElement0, valorElement1,..., valorElementn);`
- **array** : **paraula clau** en PHP equivalent a dir:
 - "Els elements que apareixen a la llista a continuació són elements d'un array".

• **Exemple:**

<?php

```
$color = array("blau" , "verd" , "negre" , "marró");  
echo $color[1]; // Mostra verd
```

?>



Assignació de
valors array:
Mètode 3

- **Altra forma** d'assignar valors a un array equivalent a l'anterior:
`$nomDelArray= [valorElement0, valorElement1,..., valorElementn];`

- **Exemple:**

```
<?php
    $color = ["roig" , "blau" , "groc" , "verd"];
    echo $color[1]; // Mostra blau
?>
```

0
↓

1
↓

2
↓

3
↓

Assignació de
valors array:
Mètode 4

- **Altra forma d'assignar valors a un array:** associatiu amb índex numèric.
 - Especifiquem **explícitament l'índex** que tindrà cada element:

```
$nomDelArray = array[índex1 => valorElement0,  
                    índex2 => valorElement1,..., índexn => valorElementn];
```

- **Exemple:**

```
<?php  
    $color=array(0 => "roig", 1 => "blau", 2 => "groc", 3 => "verd");  
    echo $color[1]; // Mostra blau  
?>
```

Assignació de
valors array:
Mètode 5

- **var_dump**: Funció per poder **visualitzar** tot el **contingut d'un array**:
 - **Evita** haver de fer un **bucle** per recórrer-ne tots els elements.

- **Exemple:**

```
<?php
```

```
$alumnes=array("Salva" , "Rebeca" , "Anna" , "Maria");
```

```
//visualitzem el contingut de la variable $alumnes
```

```
var_dump($alumnes);
```

```
?>
```

- Resultat:

```
array(4) { [0]=> string(5) "Salva" [1]=> string(6) "Rebeca" [2]=> string(4) "Anna"  
[3]=> string(5) "Maria" }
```

Eliminar elements d'un array (I)

- `unset()`
- Quan s'utilitza `unset()` els **índex** de l'array **no canviaran ni es reindexaran**.
- Si volem **reindexar els elements** després d'aplicar `unset()`:
 - Utilitzem `array_values()`:
 - Convertirà tots els **índex a valors numèrics numerats a partir de 0**.
 - Si estem utilitzant **arrays associatius**:
 - **Canviarà** també els **noms a valors enters consecutius**.

- Exemple unset():

```
<?php
$array = array("a","b","c");
var_dump($array);
// també $array = [0 => "a", 1 => "b", 2 => "c"];
unset($array[1]); //Eliminem element amb índex 1, passat per referència
var_dump($array);
//reindexem i visualitzem
$array=array_values($array);
var_dump($array);
?>
```


array(2) { [0]=> string(1) "a" [1]=> string(1) "b" [2]=> string(1) "c" }

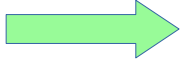

array(2) { [0]=> string(1) "a" [2]=> string(1) "c" }

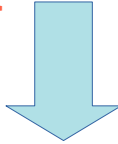
Reindexat

array(2) { [0]=> string(1) "a" [1]=> string(1) "c" }

- **Exemple** `unset()` amb arrays indexats:

```
<?php
$array = array("index1"=>"a","index2"=>"b","index3"=>"c");
var_dump($array);  array(2) { ["index1"]=> string(1) "a"
["index2"]=> string(1) "b" ["index3"]=> string(1) "c" }

unset($array["index2"]); //Eliminem element index2, passat per referència
var_dump($array);  array(2) { ["index1"]=> string(1) "a"
["index3"]=> string(1) "c" }
//reindexem i visualitzem
$array=array_values($array);
var_dump($array);  array(2) { [0]=> string(1) "a" [1]=> string(1) "c" }
?>
```

 Reindexat amb enters

Eliminar elements d'un array (II)

- `array_splice()`
- Utilitzant `array_splice()` els **índex** es tornaran a **indexar automàticament**.
- Si treballem amb **arrays associatius**: les **claus associatives no canviaran**:
 - A **diferència** del cas en què utilitzem `array_values()`:
 - **Convertiria** tots els **índex a valors numèrics**.

- `array_splice()` pot tindre fins a 4 **paràmetres**:
`array_splice(array_entrada, offset, longitud_a_eliminar, array_reemplaçaments);`
 - **array_entrada**: **array** sobre el que es **vol treballar**.
 - **offset**: **No és l'índex de l'element**:
 - És el **desplaçament respecte a l'inici de l'array** de l'**element** que volem **eliminar**.
 - Si és **positiu**: comença **des de l'inici**.
 - Si és **negatiu**: comença **des del final** de l'array.

- `array_splice()` pot tindre fins a 4 **paràmetres**:
`array_splice(array_entrada, offset, longitud_a_eliminar, array_reemplaçaments);`
- **longitud_a_eliminar** (opcional):
 - Si **no s'indica**: s'elimina tot des de la posició de l'offset fins al final.
 - Si **s'indica i és positiu**: s'elimina aquesta quantitat d'elements a partir de la posició de l'offset.
 - Si **s'especifica i és negatiu**: el final dels elements eliminats serà a tantes posicions del final de l'array com indique el número.
- **array_reemplaçaments** (opcional): Si s'especifica, els **elements eliminats** seran **substituïts per aquests** elements.

- Exemple `array_splice()`:

```
<?php
```

```
$array = array("a","b","c");  
var_dump($array);
```



array(2) { [0]=> string(1) "a" [1]=> string(1) "b"
[2]=> string(1) "c" }

//eliminem un element a partir de la posició 1 de l'array:

```
array_splice ($array, 1, 1); // passat per referència
```

```
var_dump($array);
```




array(2) { [0]=> string(1) "a" [1]=> string(1) "c" }

```
?>
```



Reindexat
automàticament

- `array_splice()` i `unset()` prenen el **valor** de l'array **per referència**:
 - **No cal assignar** els **valors de retorn** d'aquestes funcions **de nou a l'array**:
 - Es **modifica directament**.
-  **Esborrar diversos elements** sense cridar `unset()` o `array_splice()` diverses vegades, utilitzar:
 - `array_diff()` : si **coneguem els valors** dels elements a eliminar.
 - `array_diff_key()` : si **coneguem les claus** dels elements a eliminar.

Eliminar elements d'un array (III)

- `array_diff()`
- **Coneguem els valors** dels **elements** de l'array que volem suprimir.
- **No canviarà ni reindexarà els índex** de l'array.
- Cal **assignar el resultat de nou** a la **variable \$array inicial**:
 - Es pren la **variable \$array per valor**, **no per referència**.

- **Exemple** `array_diff($array_origen,$array_amb_valors_eliminar):`

```
<? php
```

```
$array = array("a","b","c");  
var_dump($array);
```



`array(2) { [0]=> string(1) "a" [1]=> string(1) "b" [2]=> string(1) "c" }`

```
// indiquem els valors que volem suprimir: "a", "c"
```

```
$array = array_diff($array, ["a", "c"]); // passat per valor  
var_dump($array);
```



`array(1) { [1]=> string(1) "b" }`

```
?>
```



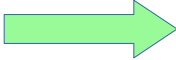
No reindexat


Eliminar elements d'un array (i IV)


- `array_diff_key()`
- **Coneguem els índex** dels **elements** de l'array que volem suprimir.
- **No canviarà ni reindexarà els índex** de l'array.
- Cal **assignar el resultat de nou** a la **variable \$array inicial**:
 - Es pren la **variable \$array per valor**, **no per referència**.
- Ens hem d'assegurar de **passar aquestes claus** a suprimir **com a claus en el segon array** que es passa com **paràmetre** i **no com a valors**.

- Exemple** `array_diff_key($array_origen,$array_amb_index_eliminar):`

```
<? php
$array = array("a","b","c");
var_dump($array);
// volem eliminar els elements amb index 0 i 2
// els valors "xy" són totalment irrelevantes.
$array= array_diff_key($array, [0 => "xy", "2" => "xy"]); //per valor
var_dump($array);
?>
```

 `array(2) { [0]=> string(1) "a" [1]=> string(1) "b" [2]=> string(1) "c" }`

 `array(1) { [1]=> string(1) "b" }`

 No reindexat

- Mètodes PHP per treballar amb arrays.

