

# PHP : tableaux

**Achref El Mouelhi**

Docteur de l'université d'Aix-Marseille  
Chercheur en programmation par contrainte (IA)  
Ingénieur en génie logiciel

[elmouelhi.achref@gmail.com](mailto:elmouelhi.achref@gmail.com)



- 1 Introduction
- 2 Tableaux indexés
- 3 Tableaux associatifs
- 4 Tableaux multidimensionnels
- 5 Tableaux de constante

## Quatre types de tableau

- **tableaux indexés** : tableaux avec indices numériques
- **tableaux associatifs** : tableaux avec indices personnalisés
- **tableaux multidimensionnels** : tableaux contenant d'autres tableaux
- **tableaux de constantes** : tableaux contenant des constantes

## Pour créer un tableau indexées d'entiers

```
$nombres = array(  
    2,  
    5,  
    8  
) ;
```

# PHP

## Pour créer un tableau indexées d'entiers

```
$nombres = array(  
    2,  
    5,  
    8  
) ;
```

Ou aussi

```
$nombres = [  
    2,  
    5,  
    8  
];
```

Pour connaître le nombre d'éléments d'un tableau

```
echo count($nombres);  
/* affiche 3 */
```

© Achref EL MOUELLI

Pour connaître le nombre d'éléments d'un tableau

```
echo count($nombres);  
/* affiche 3 */
```

Pour accéder à un élément du tableau, on utilise son indice

```
$nombres[1] = 6;  
echo $nombres[1];  
/* affiche 6 */
```

Les éléments d'un tableau ne doivent pas forcément avoir des indices consécutifs

```
$nombres[5] = 6;  
var_dump($nombres);  
/* affiche  
C:\wamp64\www\premier-cours-php\index.php:11:  
array (size=4)  
  0 => int 2  
  1 => int 5  
  2 => int 8  
  5 => int 6  
 */
```

## Remarque

- PHP ne supporte pas les indices négatifs dans les tableaux. Par conséquent, tenter d'accéder à `$tableau[-1]` génère un message d'erreur de type : `Warning: Undefined array key -1.`
- Le message d'erreur `Warning: Undefined array key` s'affiche également lorsqu'on essaie d'accéder à un indice positif inexistant dans le tableau.

Pour ajouter un nouvel élément dans le tableau tout en gardant la succession des indices

```
$nombres[] = 6;  
var_dump($nombres);  
/* affiche  
C:\wamp64\www\premier-cours-php\index.php:11:  
array (size=4)  
  0 => int 2  
  1 => int 5  
  2 => int 8  
  3 => int 6  
 */
```

Pour mieux afficher un tableau, on peut utiliser la fonction `print_r()`

```
$nombres[5] = 6;  
print_r($nombres);  
/* affiche  
Array ([0] => 2[1] => 5[2] => 8[5] => 6 )  
*/
```

Depuis PHP 7.4, il est possible de décomposer un tableau avec l'opérateur . . .

```
$tab = [0, 1, ...$nombres, 7];
print_r($tab);
// affiche Array ( [0] => 0 [1] => 1 [2] => 2 [3] => 5 [4] => 8
                  [5] => 7 )
```

# PHP

Depuis PHP 7.4, il est possible de décomposer un tableau avec l'opérateur . . .

```
$tab = [0, 1, ...$nombres, 7];
print_r($tab);
// affiche Array ( [0] => 0 [1] => 1 [2] => 2 [3] => 5 [4] => 8
     [5] => 7 )
```

Depuis PHP 7.1, il est possible de déstructurer un tableau

```
[$first, $second] = $nombres;
echo "$first, $second";
// affiche 2, 5
```

# PHP

Pour parcourir un tableau, avec une boucle, élément par élément

```
for($i = 0; $i < count($nombres); $i++) {  
    echo $nombres[$i], "<br>";  
}
```

© Achref EL MOUELHI ©

# PHP

Pour parcourir un tableau, avec une boucle, élément par élément

```
for($i = 0; $i < count($nombres); $i++) {  
    echo $nombres[$i], "<br>";  
}
```

On peut aussi utiliser `foreach`

```
foreach ($nombres as $value) {  
    echo $value, "<br>";  
}
```

# PHP

Pour parcourir un tableau, avec une boucle, élément par élément

```
for($i = 0; $i < count($nombres); $i++) {  
    echo $nombres[$i], "<br>";  
}
```

On peut aussi utiliser `foreach`

```
foreach ($nombres as $value) {  
    echo $value, "<br>";  
}
```

Pour récupérer l'indice avec `foreach`

```
foreach ($nombres as $key => $value) {  
    echo "$key => $value <br>";  
}
```

Étant donné le tableau suivant

```
$voitures = ['peugeot', 'ford', 'fiat', 'mercedes', 'seat'];
```

Étant donné le tableau suivant

```
$voitures = ['peugeot', 'ford', 'fiat', 'mercedes', 'seat'];
```

### Exercice

Écrire un script **PHP** qui permet de calculer le nombre total de caractères de toutes les chaînes présentes dans le tableau `$voitures`.

## Correction

```
$voitures = ['peugeot', 'ford', 'fiat', 'mercedes', 'seat'];
$total_caracteres = 0;

foreach ($voitures as $value) {
    $total_caracteres += strlen($value);
}
echo $total_caracteres;
```

Étant données les deux variables suivantes

```
$voiture = 'ford';
$voitures = ['peugeot', 'ford', 'fiat', 'mercedes', 'seat', 'ford', 'fiat', 'ford'];
```

© Achref EL MOUL

Étant données les deux variables suivantes

```
$voiture = 'ford';
$voitures = ['peugeot', 'ford', 'fiat', 'mercedes', 'seat', 'ford', 'fiat', 'ford'];
```

### Exercice

Écrire un script **PHP** qui permet de calculer nombre d'occurrence de `$voiture` dans `$voitures`.

# PHP

## Première solution

```
$occurrence = 0;  
foreach ($voitures as $value) {  
    if ($value == $voiture) {  
        $occurrence++;  
    }  
}  
echo $occurrence;
```

# PHP

## Première solution

```
$occurrence = 0;  
foreach ($voitures as $value) {  
    if ($value == $voiture) {  
        $occurrence++;  
    }  
}  
echo $occurrence;
```

## Deuxième solution

```
$occurrence = 0;  
foreach ($voitures as $value) {  
    $occurrence += ($value == $voiture);  
}  
echo $occurrence;
```

## Pour chercher un élément dans un tableau indexé

```
echo(array_search(8, $nombres));  
/* affiche 2 */
```

© Achref EL MOUELHI ©

# PHP

Pour chercher un élément dans un tableau indexé

```
echo(array_search(8, $nombres));  
/* affiche 2 */
```

On peut aussi comparer des types différents

```
echo(array_search("8", $nombres));  
/* affiche 2 */
```

# PHP

Pour chercher un élément dans un tableau indexé

```
echo(array_search(8, $nombres));  
/* affiche 2 */
```

On peut aussi comparer des types différents

```
echo(array_search("8", $nombres));  
/* affiche 2 */
```

Pour comparer les valeurs et les types, on ajoute un troisième paramètre optionnel

```
echo(array_search("5", $nombres, true));  
/* n'affiche rien */
```

**La fonction `in_array()` retourne un booléen**

```
echo(in_array("8", $nombres));  
/* affiche 1 */
```

```
echo(in_array(145, $nombres));  
/* n'affiche rien */
```

La fonction `in_array()` retourne un booléen

```
echo(in_array("8", $nombres));  
/* affiche 1 */
```

```
echo(in_array(145, $nombres));  
/* n'affiche rien */
```

Pour comparer les valeurs et les types, on ajoute un troisième paramètre optionnel

```
echo(in_array("8", $nombres, true));  
/* n'affiche rien */
```

## Quelques fonctions sur les tableaux indexés

- `range($min, $max)` : retourne un tableau de valeurs séquentielles comprises entre \$min et \$max
- `array_pop(tableau)` : supprime et retourne le dernier élément d'un tableau
- `array_shift(tableau)` : supprime et retourne le premier élément d'un tableau
- `array_push(tableau, $valeur1, $valeur2...)` : ajoute plusieurs valeurs à la fin du tableau
- `array_sum(tableau)` : retourne la somme des éléments d'un tableau
- `array_product(tableau)` : retourne le produit des éléments d'un tableau
- ...

## Quelques fonctions sur les tableaux indexés

- `array_reverse(tableau)` : retourne les éléments d'un tableau dans le sens inverse
- `shuffle(tableau)` : désordonne les éléments d'un tableau
- `array_unique(tableau)` : supprime les doublons d'un tableau
- `sort(tableau)` : trie les éléments d'un tableau dans le sens ascendant
- `rsort(tableau)` : trie les éléments d'un tableau dans le sens descendant
- `array_count_values(tableau)` : compte le nombre d'occurrences de chaque valeur dans un tableau et retourne un tableau associatif où la clé est la valeur unique et la valeur est le nombre d'occurrences.
- `array_merge(...tableau)` : fusionne plusieurs tableaux en un seul. Les clés numériques sont réindexées, tandis que les clés associatives sont écrasées par les dernières valeurs.
- ...

## PHP

## Exemple avec splice

```
$sports = ["foot", "tennis", "basket", "volley"];
$new = ["rugby", "natation"];

$tab = array_splice($sports, 2, 0, $new);

print_r($sports);
/* affiche Array ([0] => foot[1] => tennis[2] => rugby[3] =>
   natation[4] => basket[5] => volley ) */

print_r($tab)
/* affiche Array ( ) */
```

### Exemple avec splice

```
$sports = ["foot", "tennis", "basket", "volley"];
$new = ["rugby", "natation"];

$tab = array_splice($sports, 2, 0, $new);

print_r($sports);
/* affiche Array ([0] => foot[1] => tennis[2] => rugby[3] =>
   natation[4] => basket[5] => volley ) */

print_r($tab)
/* affiche Array ( ) */
```

Aucun élément supprimé car le troisième paramètre = 0

## PHP

## Exemple avec splice

```
$sports = ["foot", "tennis", "basket", "volley"];
$new = ["rugby", "natation"];

$tab = array_splice($sports, 2, 1, $new);

print_r($sports);
/* affiche Array ([0] => foot[1] => tennis[2] => rugby[3] =>
   natation[4] => volley ) */

print_r($tab);
/* affiche Array ([0] => basket ) */
```

### Exemple avec splice

```
$sports = ["foot", "tennis", "basket", "volley"];
$new = ["rugby", "natation"];

$tab = array_splice($sports, 2, 1, $new);

print_r($sports);
/* affiche Array ([0] => foot[1] => tennis[2] => rugby[3] =>
   natation[4] => volley ) */

print_r($tab);
/* affiche Array ([0] => basket ) */
```

Un seul élément supprimé car le deuxième paramètre = 1

À ne pas confondre avec `slice` qui permet d'extraire un sous-tableau sans modifier le tableau d'origine

```
$sports = ["foot", "tennis", "basket", "volley"];  
  
$tab = array_slice($sports, 0, 2);  
  
print_r($sports);  
// affiche Array ([0] => foot[1] => tennis[2] =>  
 // basket[3] => volley )  
  
print_r($tab);  
/* affiche Array ([0] => foot[1] => tennis ) */
```

# PHP

La fonction `implode` permet de transformer un tableau en chaîne de caractères

```
$sports = ["foot", "tennis", "basket", "volley"];
$string = implode("-", $sports);

echo $string;
/* affiche foot-tennis-basket-volley */

echo gettype($string);
/* affiche string */
```

# PHP

La fonction `implode` permet de transformer un tableau en chaîne de caractères

```
$sports = ["foot", "tennis", "basket", "volley"];
$string = implode("-", $sports);

echo $string;
/* affiche foot-tennis-basket-volley */

echo gettype($string);
/* affiche string */
```

La fonction `explode` permet de transformer une chaîne de caractères en tableau de caractères

```
$string = "bonjour john wick";
print_r (explode(" ", $string));
/* affiche Array ([0] => bonjour[1] => john[2] => wick ) */
```

Pour transformer une chaîne de caractères en tableau de caractères, on utilise la fonction str\_split

```
$string = "bonjour john wick";
print_r(str_split($string));
/* affiche Array ([0] => b[1] => o[2] => n[3] => j[4] => o[5] => u[6]
   => r[7] => [8] => j[9] => o[10] => h[11] => n[12] => [13] => w[14] =>
   i[15] => c[16] => k ) */
```

Pour transformer une chaîne de caractères en tableau de caractères, on utilise la fonction str\_split

```
$string = "bonjour john wick";
print_r(str_split($string));
/* affiche Array ([0] => b[1] => o[2] => n[3] => j[4] => o[5] => u[6]
   => r[7] => [8] => j[9] => o[10] => h[11] => n[12] => [13] => w[14] =>
   i[15] => c[16] => k ) */
```

Pour transformer une chaîne de caractères en tableau de chaînes de caractères de taille fixe, on ajoute la taille à la fonction str\_split

```
$string = "bonjour john wick";
print_r(str_split($string, 2));
/* affiche Array ( [0] => bo [1] => nj [2] => ou [3] => r [4] => jo [5]
   => hn [6] => w [7] => ic [8] => k ) */
```

**Exemple avec** array\_count\_values

```
$numbers = [1, 2, 3, 1, 2, 1, 5, 4, 4];  
  
print_r(array_count_values($numbers));  
/*  
Array  
(  
    [1] => 3  
    [2] => 2  
    [3] => 1  
    [5] => 1  
    [4] => 2  
)  
*/
```

Étant donné le tableau suivant

```
$numbers = [1, 2, 3, 1, 2, 1, 5, 4, 4];
```

Étant donné le tableau suivant

```
$numbers = [1, 2, 3, 1, 2, 1, 5, 4, 4];
```

### Exercice

Écrire un script **PHP** qui permet de

- demander à l'utilisateur de saisir un nombre
- afficher le nombre d'occurrences du nombre saisi dans `$numbers`

# PHP

## Exemple avec array\_merge

```
$numbers1 = [1, 2, 3, 1];
$numbers2 = [1, 5, 4, 4];

print_r(array_merge($numbers1, $numbers2));
/*
Array
(
    [0] => 1
    [1] => 2
    [2] => 3
    [3] => 1
    [4] => 1
    [5] => 5
    [6] => 4
    [7] => 4
)
*/
```

Étant donné le tableau suivant

```
$numbers = [[1, 2], [3, 4], [5, 6, 7]];
```

Étant donné le tableau suivant

```
$numbers = [[1, 2], [3, 4], [5, 6, 7]];
```

### Exercice

Écrire un script **PHP** qui permet de créer un tableau contenant les valeurs de tous les sous-tableaux de \$numbers.

Étant donné le tableau suivant

```
$numbers = [[1, 2], [3, 4], [5, 6, 7]];
```

### Exercice

Écrire un script **PHP** qui permet de créer un tableau contenant les valeurs de tous les sous-tableaux de \$numbers.

### Résultat attendu

```
Array([0] => 1 [1] => 2 [2] => 3 [3] => 4 [4] => 5 [5] => 6 [6] => 7)
```

Pour créer un tableau associatif (ensemble clé  $\Rightarrow$  valeur)

```
$fcb = array(  
    "messi" => 10,  
    "umtiti" => 23,  
    "suarez" => 9  
) ;
```

© Achref EL MOUADJI

# PHP

Pour créer un tableau associatif (ensemble clé  $\Rightarrow$  valeur)

```
$fcb = array(  
    "messi" => 10,  
    "umtiti" => 23,  
    "suarez" => 9  
) ;
```

Ou

```
$fcb = [  
    "messi" => 10,  
    "umtiti" => 23,  
    "suarez" => 9  
] ;
```

## Pour ajouter un nouvel élément

```
$fcb["rakitic"] = 4;
```

© Achref EL MOUELHI ©

## Pour ajouter un nouvel élément

```
$fcb["rakitic"] = 4;
```

## Pour afficher les éléments d'un tableau associatif

```
var_dump($fcb);
/* C:\wamp64\www\premier-cours-php\index.php:11:
array (size=4)
  'messi' => int 10
  'umtiti' => int 23
  'suarez' => int 9
  'rakitic' => int 4 */
```

Pour connaître le nombre d'éléments d'un tableau associatif un nouvel élément

```
echo count($fcb);  
// affiche 4
```

Pour parcourir un tableau associatif élément par élément

```
foreach ($fcb as $key => $value) {  
    echo $key, " => ", $value, "<br>";  
}
```

© Achref EL MOUADJI

Pour parcourir un tableau associatif élément par élément

```
foreach ($fcb as $key => $value) {  
    echo $key, " => ", $value, "<br>";  
}
```

Le résultat est

```
messi => 10  
umtiti => 23  
suarez => 9
```

On peut aussi faire

```
foreach (array_keys($fcb) as $key) {  
    echo $key, " => ", $fcb[$key], "<br>";  
}
```

© Achref EL MOUADJI

On peut aussi faire

```
foreach (array_keys($fcb) as $key) {  
    echo $key, " => ", $fcb[$key], "<br>";  
}
```

Le résultat est le même

```
messi => 10  
umtiti => 23  
suarez => 9
```

# PHP

Les clés (respectivement les valeurs) peuvent ne pas avoir le même type

```
$mixte = array(  
    17 => 10,  
    "umtiti" => 23,  
    "suarez" => 9  
);  
$mixte[2] = true;  
  
var_dump($mixte);  
/* affiche C:\wamp64\www\premier-cours-php\index.php:11:  
array (size=4)  
    17 => int 10  
    'umtiti' => int 23  
    'suarez' => int 9  
    2 => boolean true  
*/
```

## Remarque

Le message d'erreur `Warning: Undefined array key` s'affiche lorsqu'on essaie d'accéder à une clé inexistante dans le tableau associatif.

Pour déstructurer un tableau associatif, on utilise `extract`

```
extract($fcb);  
  
echo "$messi, $umtiti";  
// affiche 10, 23
```

Pour déstructurer un tableau associatif, on utilise `extract`

```
extract ($fcb);  
  
echo "$messi, $umtiti";  
// affiche 10, 23
```

### Remarque

`extract ()` peut être risqué, car il va automatiquement créer des variables pour chaque clé présente dans le tableau.

## Quelques fonctions sur les tableaux associatifs

- `array_key_exists()` : vérifie si une clé existe
- `array_count_values()` : retourne un tableau associatif contenant chaque valeur et son occurrence dans le tableau
- `array_values()` : retourne la liste des valeurs
- `array_keys()` : retourne la liste des clés
- `array_flip` : transforme les clés en valeurs et les valeurs en clés
- `ksort()` : trie un tableau associatif selon les clés dans un ordre croissant
- `krsort()` : trie un tableau associatif selon les clés dans un ordre décroissant
- ...

## Étant données les variables suivantes

```
$cle = "messi";  
$valeur = 10;  
$fcb = array(  
    "messi" => 10,  
    "umtiti" => 23,  
    "suarez" => 9  
) ;
```

## Étant données les variables suivantes

```
$cle = "messi";
$valeur = 10;
$fcb = array(
    "messi" => 10,
    "umtiti" => 23,
    "suarez" => 9
);
```

### Exercice

Écrire un script **PHP** qui permet de vérifier si `$valeur` est associée à `$cle`.

## Correction

```
echo ($fcb[$cle] == $valeur) ? "$cle est le numéro $valeur" : "  
$cle n'a pas le numéro $valeur";
```

Étant données les variables suivantes

```
$valeur = 10;  
$fcb = array(  
    "messi" => 10,  
    "umtiti" => 23,  
    "suarez" => 9,  
    "rivaldo" => 9,  
    "maradona" => 10,  
    "figo" => 9  
) ;
```

# PHP

Étant données les variables suivantes

```
$valeur = 10;  
$fcb = array(  
    "messi" => 10,  
    "umtiti" => 23,  
    "suarez" => 9,  
    "rivaldo" => 9,  
    "maradona" => 10,  
    "figo" => 9  
) ;
```

## Exercice

Écrire un script **PHP** qui permet de déterminer l'ensemble de clés associées à `$valeur`.

## Correction

```
$cles = [];
foreach ($fcb as $key => $value) {
    if ($valeur == $value) {
        $cles[] = $key;
    }
}
print_r($cles);
```

# PHP

## Pour créer un tableau multidimensionnel

```
$languages = array
(
    array("html", "css"),
    array("js", "ts"),
    array("php", "java", "csharp")
);
```

© Achref EL MOUADJI

# PHP

## Pour créer un tableau multidimensionnel

```
$langages = array
(
    array("html", "css"),
    array("js", "ts"),
    array("php", "java", "csharp")
);
```

## Pour accéder à un élément

```
echo ($langages [2] [0]);
/* affiche php */
```

# PHP

## Pour créer un tableau multidimensionnel

```
$langages = array
(
    array("html", "css"),
    array("js", "ts"),
    array("php", "java", "csharp")
);
```

## Pour accéder à un élément

```
echo ($langages [2] [0]);
/* affiche php */
```

## Pour ajouter un élément

```
$langages [3] = array("c", "c++");
```

Étant données les variables suivantes

```
$valeur = 5;  
$matrice = array(  
    array(2, 3, 5, 7),  
    array(1, 2, 5, 9, 5),  
    array(4, 2, 9, 3, 1, 5)  
,
```

Étant données les variables suivantes

```
$valeur = 5;  
$matrice = array(  
    array(2, 3, 5, 7),  
    array(1, 2, 5, 9, 5),  
    array(4, 2, 9, 3, 1, 5)  
,
```

### Exercice

Écrire un script **PHP** qui permet de calculer le nombre d'occurrence de \$valeur dans \$matrice par ligne.

## Correction

```
$compteurs = [0, 0, 0];
for ($i = 0; $i < count($matrice); $i++) {
    foreach ($matrice[$i] as $value) {
        if ($value == $valeur) {
            $compteurs[$i]++;
        }
    }
}
print_r($compteurs);
```

Étant données les variables suivantes

```
$valeur = 5;  
$matrice = array(  
    array(2, 3, 5, 7),  
    array(1, 2, 5, 9, 5),  
    array(4, 2, 9, 3, 1, 5)  
,
```

Étant données les variables suivantes

```
$valeur = 5;  
$matrice = array(  
    array(2, 3, 5, 7),  
    array(1, 2, 5, 9, 5),  
    array(4, 2, 9, 3, 1, 5)  
,
```

### Exercice

Écrire un script **PHP** qui permet de déterminer si \$valeur est présente dans chaque ligne de \$matrice.

## Correction

```
$trouver = true;
for ($i = 0; $i < count($matrice) && $trouver; $i++) {
    if (!in_array($valeur, $matrice[$i])) {
        $trouver = false;
    }
}
print($trouver ? "oui" : "non");
```

## Exercice

Écrire un script **PHP** qui calcule

- 1 la somme de deux matrices carrées,
- 2 le produit de deux matrices carrées.

## Remarque

Depuis **PHP 7**, on peut déclarer un tableau de constantes.

## Pour créer un tableau de constantes

```
const NOMBRES = array(  
    2,  
    5,  
    8  
) ;
```

© Achref EL MOUADJI

# PHP

## Pour créer un tableau de constantes

```
const NOMBRES = array(  
    2,  
    5,  
    8  
) ;
```

Ou aussi

```
define('NOMBRES', array(  
    2,  
    5,  
    8  
));
```

**L'instruction suivante ne peut être acceptée**

**NOMBRES [1] = 5;**