



# Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [Les structure de données](#) ► [Le tableau associatif array](#)

## Le tableau associatif array

PHP nous propose une structure de donnée appelée tableau ([array](#)) qui va nous permettre de stocker des données de n'importe quel type. Pour créer un tableau vide en PHP nous pouvons utiliser deux syntaxes, voici la plus ancienne :

```
1 <?php
2
3 // tableau_1.php
4 $tab = array();
```

Et à partir de PHP 5.4 :

```
1 <?php
2
3 // tableau_2.php
4 $tab = [];
```

Communément appelée tableau, cette structure de donnée est en fait un dictionnaire : elle associe une clé à une valeur : le tableau PHP associe des clés de type integer ou string aux valeurs stockées. Nous pouvons tout aussi bien utiliser le tableau PHP comme une simple liste en y insérant des valeurs sans nous préoccuper des clés. Dans ce cas, PHP associe automatiquement une clé de type entier à la valeur insérée. La première de ces clés est 0, puis elle est incrémentée de 1 à chaque insertion de valeur. Nous allons insérer quatre valeurs dans notre tableau en utilisant l'opérateur []. La première valeur sera donc à l'index 0, la deuxième à l'index 1, etc..

```
1 <?php
2
3 // tableau_3.php
4 $tab = []; // création d'un tableau vide
5 $tab[] = 'azerty'; // insertion à l'index 0
6 $tab[] = 456; // insertion à l'index 1
7 $tab[] = 78.9; // insertion à l'index 2
8 $tab[] = true; // insertion à l'index 3
```

Nous pourrions aussi créer un tableau initialisé, c'est-à-dire contenant des valeurs avec la syntaxe :

```
1 <?php
2
3 // tableau_4.php
4 $tab = array('azerty', 456, 78.9, true);
```

ou la syntaxe :

```
1 <?php
2
3 // tableau_5.php
4 $tab = ['azerty', 456, 78.9, true];
```

Pour récupérer une valeur stockée dans le tableau j'utilise la clé qui lui est associée. Par exemple pour récupérer la deuxième valeur du tableau j'utilise la clé 1 car la première clé est 0 :

```

1 <?php
2
3 // tableau_6.php
4 $tab = ['azerty', 456, 78.9, true];
5 echo $tab[1]; // affiche 456

```

Si je souhaite récupérer toutes les valeurs de mon tableau, je peux très bien utiliser les clés comme compteur dans une boucle for puisque ces clés s'incrémentent de 1 à chaque fois.

```

1 <?php
2
3 // tableau_7.php
4 $tab = ['azerty', 456, 78.9, true];
5 for ($i = 0; $i < 4; $i++) {
6     echo $tab[$i] . PHP_EOL;
7 }

```

Dans cette boucle *i* prendra successivement les valeurs 0, 1, 2, 3. Au premier tour de boucle *i* vaudra 0, nous afficherons donc `$tab[0]`, puisque nous remplaçons *i* par sa valeur. Au deuxième tour de boucle *i* vaudra 1, nous afficherons donc `$tab[1]`, puisque nous remplaçons *i* par sa valeur. Au troisième tour de boucle *i* vaudra 2, nous afficherons donc `$tab[2]`, puisque nous remplaçons *i* par sa valeur. Au quatrième tour de boucle *i* vaudra 3, nous afficherons donc `$tab[3]`, puisque nous remplaçons *i* par sa valeur.

La boucle `for` est utilisable lorsque nous utilisons le tableau PHP comme une simple liste, c'est-à-dire sans assignation explicite de clés. Sinon, il faudra utiliser autre chose...

Si nous souhaitons utiliser le tableau PHP comme un dictionnaire, nous devons assigner une clé de type integer ou string à chaque valeur.

```

1 <?php
2
3 // tableau_8.php
4 $tab['oranges'] = 12;
5 $tab['navets'] = 1;
6 $tab['urgent'] = true;
7 $tab[45] = 'azerty';

```

Nous pouvons aussi initialiser le tableau lors de sa création :

```

1 <?php
2
3 // tableau_9.php
4 $tab = ['oranges' => 12, 'navets' => 1, 'urgent' => true, 45 => 'azerty'];

```

Pour récupérer une valeur nous utilisons toujours sa clé :

```

1 <?php
2
3 // tableau_10.php
4 $tab = ['oranges' => 12, 'navets' => 1, 'urgent' => true, 45 => 'azerty'];
5 echo $tab['oranges'] . PHP_EOL;
6 echo $tab['navets'] . PHP_EOL;
7 echo $tab['urgent'] . PHP_EOL;
8 echo $tab[45] . PHP_EOL;

```

Pour afficher toutes les valeurs dans une boucle, nous allons utiliser une boucle spéciale : la boucle `foreach`. La boucle `foreach` pourrait être décrite comme cela : pour chaque élément contenu dans le tableau, faire quelque chose avec l'élément. Par exemple si je souhaite afficher les valeurs d'un tableau simple avec `foreach` je pourrais écrire :

```

1 <?php

```

```

2
3 // tableau_11.php
4 $tab = ['azerty', 456, 'uiop'];
5 foreach ($tab as $element) {
6     echo $element . PHP_EOL;
7 }

```

Au premier tour de boucle, la variable element sera "azerty", au deuxième element sera 456, au troisième "uiop". La variable element prendra successivement toutes les valeurs du tableau tab.

Notez que la syntaxe n'est pas "pour chaque élément présent dans tab" mais plutôt "pour chaque valeur de tab prise comme élément".

Nous avons affiché les valeurs de notre tableau, mais nous pouvons aussi afficher les clés associées aux valeurs :

```

1 <?php
2
3 // tableau_12.php
4 $tab = ['azerty', 456, 'uiop'];
5 foreach ($tab as $key => $value) {
6     echo "À la clé $key se trouve la valeur $value" . PHP_EOL;
7 }

```

Encore un exemple avec un tableau associatif :

```

1 <?php
2
3 // tableau_13.php
4 $tab = ['oranges' => 12, 'navets' => 1, 'urgent' => true, 45 => 'azerty'];
5 foreach ($tab as $key => $value) {
6     echo "À la clé $key se trouve la valeur $value" . PHP_EOL;
7 }

```

Pour vérifier l'égalité entre deux tableaux nous pouvons utiliser les [opérateurs](#) == et ===. L'opérateur == vérifiera si les ensembles clé/valeur sont les mêmes dans les deux tableaux, l'opérateur === vérifiera en plus si ces ensembles sont de même type et dans le même ordre. Les opérateurs != et !== sont aussi utilisables.

Fin

## NAVIGATION

[Accueil](#)

■ [Ma page](#)

[Pages du site](#)

[Mon profil](#)

[Cours actuel](#)

[Intro PHP](#)

[Participants](#)


[Le langage PHP : introduction](#)

[Les types et les variables](#)

[Les opérateurs](#)

[Les structures de contrôle](#)

[Les structure de données](#)

 [Le tableau associatif array](#)

 [Tableau multidimensionnel](#)

 [T.P. array](#)

[Les fonctions](#)

[Les erreurs](#)

[Les fichiers](#)

[Les expressions rationnelles](#)

[PHP et HTML](#)

[Petite application](#)

Mes cours

ADMINISTRATION



Administration du cours

Réglages de mon profil

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))  
[Intro PHP](#)



# Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [Les structure de données](#) ► [Tableau multidimensionnel](#)

## Tableau multidimensionnel

Puisqu'un tableau peut contenir n'importe quel type de données, il peut aussi contenir une donnée de type array, c'est-à-dire un autre tableau. Par exemple, créons un tableau vide et deux autres tableaux initialisés avec les valeurs 1, 2 et 3 pour le premier tableau, 'a', 'b', 'c', 'd' pour le deuxième.

```

1 <?php
2
3 // tableau_multi_1.php
4 $tab = [];
5 $sub_tab_1 = [1, 2, 3];
6 $sub_tab_2 = ['a', 'b', 'c', 'd'];

```

Maintenant ajoutons à tab les sous-tableaux sub\_tab\_1 et sub\_tab\_2 en utilisant l'opérateur [].

```

1 <?php
2
3 // tableau_multi_2.php
4 $tab = [];
5 $sub_tab_1 = [1, 2, 3];
6 $sub_tab_2 = ['a', 'b', 'c', 'd'];
7 $tab[] = $sub_tab_1;
8 $tab[] = $sub_tab_2;

```

À l'index 0 de tab, c'est-à-dire à tab[0] nous avons donc un sous tableau qui vaut [1, 2, 3]. À l'index 1 de tab, c'est-à-dire à tab[1] nous avons donc un sous tableau qui vaut ['a', 'b', 'c', 'd'].

1	2	3	'a'	'b'	'c'	'd'

Nous pourrions aussi initialiser le tableau à sa création. Deux syntaxes sont possibles. La première :

```

1 <?php
2
3 // tableau_multi_3.php
4 $tab = array(array(1, 2, 3), array('a', 'b', 'c', 'd'));

```

La deuxième :

```

1 <?php
2
3 // tableau_multi_4.php
4 $tab = [[1, 2, 3], ['a', 'b', 'c', 'd']];

```

Pour récupérer la première valeur du premier sous-tableau, c'est-à-dire 1, je dois aller dans la première case de tab : tab[0], puis dans la première case du tableau contenu dans la première case de tab : tab[0][0]. Le premier index entre [] correspond à la case de tab, le deuxième index entre [] correspond à la case du sous-tableau

contenu dans la case de tab.

```

1 <?php
2
3 // tableau_multi_5.php
4 $tab = [];
5 $sub_tab_1 = [1, 2, 3];
6 $sub_tab_2 = ['a', 'b', 'c', 'd'];
7 $tab[] = $sub_tab_1;
8 $tab[] = $sub_tab_2;
9 echo $tab[0][0];

```

Pour parcourir un tel tableau, il nous faudrait une double boucle : une boucle sur les cases de tab, une boucle sur les cases du sous-tableau. Nous pourrions l'exprimer ainsi : pour chaque sous-tableau de tab et pour chaque case du sous-tableau, afficher la case :

```

1 <?php
2
3 // tableau_multi_6.php
4 $tab = [[1, 2, 3], ['a', 'b', 'c', 'd']];
5 // pour chaque sous-tableau contenu dans tab
6 foreach ($tab as $sub_tab) {
7     // pour chaque élément contenu dans le sous-tableau sub_tab
8     foreach ($sub_tab as $element) {
9         // affichage de l'élément et d'un espace
10        echo $element . " ";
11    }
12    // affichage d'un saut de ligne
13    echo PHP_EOL;
14 }

```

Les tableaux multidimensionnels pourront aussi prendre la forme de tableaux associatifs plus ou moins complexes, par exemple :

```

1 <?php
2
3 // tableau_multi_7.php
4 $tab = [
5     'fruits' => ['oranges' => 3, 'poires' => 2, 'pommes' => 1.5],
6     'légumes' => ['carottes' => 1, 'navets' => 2, 'broccolis' => 1],
7     'unité de mesure' => 'kilo',
8     'devise' => 'euro'
9 ];
10 // affichage de la valeur liée à la clé 'navets'
11 echo $tab['légumes']['navets'];

```

Fin

## NAVIGATION



### Accueil

#### ■ Ma page

Pages du site

Mon profil

Cours actuel

#### Intro PHP

Participants




Le langage PHP : introduction

Les types et les variables

Les opérateurs

Les structures de contrôle

Les structure de données

-  [Le tableau associatif array](#)
-  **[Tableau multidimensionnel](#)**
-  [T.P. array](#)
- [Les fonctions](#)
- [Les erreurs](#)
- [Les fichiers](#)
- [Les expressions rationnelles](#)
- [PHP et HTML](#)
- [Petite application](#)

[Mes cours](#)

## ADMINISTRATION



[Administration du cours](#)

[Réglages de mon profil](#)

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))  
[Intro PHP](#)



# Introduction au langage de programmation PHP

[Accueil](#) ▶ [Mes cours](#) ▶ [Développement logiciel](#) ▶ [Intro PHP](#) ▶ [Les structure de données](#) ▶ [T.P. array](#)

## T.P. array

Implémentez un algorithme qui demandera à l'utilisateur de saisir 4 nombres. À chaque fois que l'utilisateur saisira un nombre, celui-ci sera ajouté à un tableau. L'utilisateur effectuera donc 4 saisies. Une fois les 4 saisies effectuées, l'algorithme affichera toutes les valeurs du tableau. Le nom du script sera de la forme "array\_1\_nom\_prenom.php".

Implémentez un algorithme qui demandera à l'utilisateur de saisir 5 nombres, après chaque saisie, la valeur sera ajoutée à un tableau sauf si cette valeur a déjà été saisie. Dans ce dernier cas, le programme affichera "*valeur* est un doublon". Une fois les saisies effectuées, l'algorithme affichera toutes les valeurs du tableau. Le tableau final ne devra donc contenir aucun doublon. Par exemple, si l'utilisateur saisit 5 fois le même nombre, alors le tableau final ne contiendra qu'une seule valeur.

Le nom du script sera de la forme "array\_2\_nom\_prenom.php".

Modifiez l'algorithme précédent pour que le tableau contiennent bien 5 valeurs : tant que le tableau n'est pas plein, vous redemanderez une saisie à l'utilisateur.

Le nom du script sera de la forme "array\_3\_nom\_prenom.php".

Vous remettrez une archive zip contenant ces fichiers, le nom de l'archive sera de la forme "array\_nom\_prenom.zip".

## État du travail remis

Numéro de tentative	Ceci est la tentative 2.
Statut des travaux remis	Rouvert
Statut de l'évaluation	Pas évalué
Dernière modification	jeudi 26 mars 2015, 18:57

Ajouter une tentative basée sur le travail remis précédent

Ceci copiera le contenu de le travail remis précédent pour vous permettre d'y travailler.

Ajouter une tentative

Ceci créera un travail remis vide pour vous permettre d'y travailler.

## Tentatives précédentes

### Tentative 1 : jeudi 26 mars 2015, 16:40

#### NAVIGATION

[Accueil](#)

■ [Ma page](#)

[Pages du site](#)

[Mon profil](#)





## Cours actuel

[Intro PHP](#)[Participants](#)[Le langage PHP : introduction](#)[Les types et les variables](#)[Les opérateurs](#)[Les structures de contrôle](#)[Les structure de données](#) [Le tableau associatif array](#) [Tableau multidimensionnel](#) **[T.P. array](#)**[Les fonctions](#)[Les erreurs](#)[Les fichiers](#)[Les expressions rationnelles](#)[PHP et HTML](#)[Petite application](#)[Mes cours](#)**ADMINISTRATION**[Administration du cours](#)[Réglages de mon profil](#)Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))[Intro PHP](#)