

# Les tableaux

# Table des matières

<b>I. Contexte</b>	<b>3</b>
<b>II. Les tableaux indexés</b>	<b>3</b>
<b>III. Exercice : Appliquez la notion</b>	<b>6</b>
<b>IV. Les tableaux associatifs</b>	<b>6</b>
<b>V. Exercice : Appliquez la notion</b>	<b>8</b>
<b>VI. Les tableaux multidimensionnels</b>	<b>9</b>
<b>VII. Exercice : Appliquez la notion</b>	<b>12</b>
<b>VIII. Quelques fonctions utiles pour les tableaux</b>	<b>13</b>
<b>IX. Exercice : Appliquez la notion</b>	<b>15</b>
<b>X. Auto-évaluation</b>	<b>16</b>
A. Exercice final .....	16
B. Exercice : Défi .....	18
<b>Solutions des exercices</b>	<b>20</b>

## I. Contexte

**Durée :** 1 h

**Environnement de travail :** Repl.it

**Pré-requis :** Connaître les bases du langage PHP

### Contexte

Lors de nos développements PHP, nous avons souvent besoin de gérer des listes de valeurs (ex : une liste de produits, personnes, nombres, textes, etc). Nous avons donc besoin d'un type de variable particulier nous permettant facilement de stocker, parcourir, ajouter, modifier ou supprimer ces différentes valeurs.

Ce type particulier se nomme `array`, ou « tableau ». Ce cours va donc présenter les bases de l'utilisation des tableaux PHP, les différents types de tableaux, leur création, leur manipulation et quelques fonctions utiles pour les tableaux.

## II. Les tableaux indexés

### Objectifs

- Apprendre à déclarer un tableau.
- Apprendre à insérer, lire et modifier ses données.

### Mise en situation

Quand nous avons une liste de valeurs, nous avons besoin de les stocker quelque part afin d'y accéder et de les manipuler.

**Un tableau va nous permettre de gérer plusieurs éléments dans une seule variable.**

Nous allons voir comment déclarer et manipuler un tableau.

### Méthode Déclarer un tableau indexé

Quand on crée un tableau, on peut le déclarer vide dans le but d'y insérer des données par la suite, ou directement avec ses valeurs.

Pour déclarer un tableau vide, on va donc créer une variable (ex : `$myArray`) et lui affecter comme valeur des crochets : `[]`.

### Syntaxe Déclaration d'un tableau vide

```
1 $myArray = []
```

### Exemple Tableaux initialisés avec des valeurs (nombres ou texte)

Les valeurs d'un tableau sont séparées par une virgule suivie d'un espace (pour une meilleure lisibilité), quand il est déclaré sur une seule ligne.

```
1 <?php
2
3 $numbers = [5, 20, 16, 2, 99];
4 $animals = ['Lion', 'Chat', 'Chien', 'Cheval', 'Tigre', 'Ours'];
```

### Attention Syntaxe longue vs syntaxe courte

Il est possible de déclarer un tableau avec la syntaxe longue (ex : `$myArray = array();`), mais les bonnes pratiques PHP indiquent d'utiliser la syntaxe courte vue précédemment.

### Syntaxe Insertion de données

Pour insérer des données dans notre tableau, il faut reprendre la variable déclarée en lui accolant des crochets, pour signifier que l'on vient ajouter une valeur à la fin du tableau, puis lui affecter la valeur désirée.

```
1 $myArray[] = 'value'
```

### Exemple

Dans l'exemple ci-dessous, nous avons créé un nouveau tableau, puis nous avons ajouté la valeur `Voiture` et enfin la valeur `Camion`.

Le tableau contient donc au final les deux valeurs ajoutées.

```
1 <?php
2
3 $myArray = [];
4 $myArray[] = 'Voiture';
5 $myArray[] = 'Camion';
```

### Syntaxe Lecture de données

Pour lire une donnée, on indique sa position au sein du tableau, entre crochets.

```
1 echo $myArray[position]
```

### Exemple

L'exemple ci-dessous affiche la valeur `Voiture`, qui est la position 0 (c'est-à-dire la première position) du tableau.

```
1 <?php
2
3 $myArray = [];
4 $myArray[] = 'Voiture';
5 $myArray[] = 'Camion';
6
7 echo $myArray[0];
```

### Fondamental Clés (index) / Valeurs

Les tableaux PHP fonctionnent selon un système de valeurs associées à des clés (ou index).

Plus concrètement, quand vous créez un tableau comme : `$animals = ['Lion', 'Chat', 'Chien']`, celui-ci contient **3 valeurs** ('Lion', 'Chat' et 'Chien').

Par défaut, pour l'index PHP, on utilise un entier qui commence à 0 et qui s'incrémente à chaque valeur ajoutée. Dans le précédent tableau, la valeur 'Lion' a donc la clé 0, 'Chat' la clé 1 et 'Chien' la clé 2.

### Attention

La première valeur d'un tableau n'est donc pas à la clé 1 mais à la clé 0.

**Syntaxe**    **Modification de données**

Pour modifier une donnée d'un tableau, on utilise la variable en indiquant entre crochets la clé correspondant à la donnée à modifier.

```
1 $myArray[position] = 'new value'
```

**Exemple**

L'exemple ci-dessous va afficher la valeur : Moto.

En effet, nous avons écrasé la précédente valeur située à la clé 0, qui était 'Voiture', par la nouvelle : Moto.

```
1 <?php
2
3 $myArray = [];
4 $myArray[] = 'Voiture';
5 $myArray[] = 'Camion';
6
7 $myArray[0] = 'Moto' ;
8
9 echo $myArray[0];
```

**Remarque**

Si la clé précisée n'existe pas dans le tableau, alors la valeur est ajoutée au tableau et conserve la clé qui lui a été attribuée, même si elle n'est pas dans la suite logique.

Les nouvelles valeurs seront toujours ajoutées à l'incrément de la clé la plus élevée.

**Exemple**

L'exemple ci-dessous affiche bien la valeur : Moto.

La valeur Camion a bien comme clé : 6, même s'il n'y a pas autant de valeurs dans notre tableau.

```
1 <?php
2
3 $myArray = [];
4 $myArray[] = 'Voiture';
5 $myArray[5] = 'Moto';
6 $myArray[] = 'Camion';
7
8 echo $myArray[5];
9 echo $myArray[6];
```

**Syntaxe**    **À retenir**

Un tableau permet de gérer une liste dans une seule variable. Pour cela, il faut :

- Déclarer un tableau : `$myArray = [];`
- Déclarer un tableau avec des valeurs : `$myArray = ['value1', 'value2', 'value3']`
- Ajouter une nouvelle valeur à la fin du tableau : `$myArray[] = 'value'`
- Utiliser une valeur : `echo $myArray[position]`
- Modifier une valeur existante : `$myArray[position] = 'new value'`

La première valeur d'un tableau est à **la clé 0**.

### Complément

Les tableaux (documentation PHP)<sup>1</sup>

## III. Exercice : Appliquez la notion

Soit le tableau suivant :

```
1 <?php
2
3 $names = ['Lisa', 'Mathieu', 'Laure', 'Simon'];
```

Pour réaliser cet exercice, vous aurez besoin de travailler sur l'environnement de travail :



### Question 1

[solution n°1 p.21]

À l'aide de repl.it, écrivez le code permettant d'afficher "Bonjour Laure" (en tirant son prénom du tableau).

#### Indice :

N'oubliez pas que les indices commencent à 0 !

### Question 2

[solution n°2 p.21]

Ajoutez votre nom à ce tableau et affichez-le.

## IV. Les tableaux associatifs

### Objectifs

- Apprendre à déclarer un tableau associatif.
- Apprendre à insérer, lire et modifier ses données.

### Mise en situation

Un tableau peut utiliser des clés textuelles à la place des nombres afin de créer une association logique entre la clé et sa valeur, et ainsi améliorer la manipulation et la compréhension du tableau créé.

**Un tableau associatif est donc similaire à un tableau indexé, la seule différence réside dans le fait que ses clés sont des clés textuelles personnalisées.**

### Syntaxe

#### Déclarer un tableau associatif

Au lieu de lister les valeurs comme dans un tableau indexé (`$numbers = [5, 20, 16, 2, 99];`), il faut préciser la clé textuelle (entre apostrophes, car elle est au format texte) et lui affecter la valeur désirée grâce au symbole `=>` (symbole égal suivi du chevron supérieur).

```
1 $myArray=['key' => 'value', 'key2' => 'value2']
```

<sup>1</sup> <https://www.php.net/manual/fr/language.types.array.php>

<sup>2</sup> <https://repl.it/>

**Rappel**

Les couples clés / valeurs sont séparés par une virgule suivie d'un espace quand le tableau est déclaré sur une seule ligne.

**Exemple Déclarer un tableau associatif**

Dans l'exemple ci-dessous, nous avons créé un tableau représentant un utilisateur avec :

- Comme première clé, 'lastName', et sa valeur, 'Martin'.
- Puis le couple clé / valeur : 'firstName' / 'Jean'.
- Et enfin : 'age' / 36.

```
1 <?php
2
3 $user = ['lastName' => 'Martin', 'firstName' => 'Jean', 'age' => 36];
```

**Conseil Lisibilité**

Les tableaux associatifs permettent de faciliter la compréhension et la manipulation du code et de savoir à quoi correspondent exactement les valeurs (si les clés ont été bien choisies).

Par exemple, le même tableau en version indexée, ci-dessous, n'est pas compréhensible, car on ne sait pas à quoi correspondent exactement ces valeurs.

```
1 <?php
2
3 $user = ['Martin', 'Jean', 36];
```

**Méthode Insertion de données**

Pour insérer des données dans un tableau associatif, précisez la nouvelle clé entre crochets, puis affectez la valeur voulue.

```
1 $myArray['key'] = 'value'
```

**Exemple**

Dans l'exemple ci-dessous, nous avons repris l'utilisateur créé précédemment auquel nous ajoutons le couple clé / valeur : 'email' / 'jean.martin@email.com'.

Ce couple vient donc s'ajouter à la fin de notre tableau associatif.

```
1 <?php
2
3 $user = ['lastName' => 'Martin', 'firstName' => 'Jean', 'age' => 36];
4
5 $user['email'] = 'jean.martin@email.com';
```

**Syntaxe Lecture de données**

Pour lire une donnée, spécifiez sa clé entre crochets.

```
1 echo $myArray['key']
```

### Exemple

L'exemple ci-dessous va afficher (grâce à `echo`) la valeur : Martin.

```
1 <?php
2
3 $user = ['lastName' => 'Martin', 'firstName' => 'Jean', 'age' => 36];
4
5 echo $user['lastName'];
```

### Méthode Modification de données

Pour modifier une donnée d'un tableau, précisez entre crochets la clé correspondante.

```
1 $user['key'] = 'new value'
```

### Exemple

L'exemple ci-dessous va afficher la valeur : 37. En effet, nous avons écrasé la précédente valeur située à la clé `age`, qui était 36, par la nouvelle : 37.

```
1 <?php
2
3 $user = ['lastName' => 'Martin', 'firstName' => 'Jean', 'age' => 36];
4
5 $user['age'] = 37;
6
7 echo $user['age'];
```

### Syntaxe À retenir

Un tableau associatif est un tableau où les clés ont été personnalisées dans le but de créer des couples logiques de clés / valeurs. Pour cela, il faut :

- Déclarer un tableau associatif : `$myArray['key' => 'value', 'key2' => 'value2']`
- Ajouter une nouvelle valeur : `$myArray['key'] = 'value'`
- Utiliser une valeur : `echo $myArray['key']`
- Modifier une valeur existante : `$user['key'] = 'new value'`

### Complément

Les tableaux (documentation PHP)<sup>1</sup>

## V. Exercice : Appliquez la notion

Dans le monde des sorciers, la monnaie est un peu particulière : les sorciers payent en noises (*knuts*), mornilles (*sickles*) et gallions (*galleons*). Un gallion vaut 17 mornilles et une mornille vaut 29 noises. Le code ci-dessous permet de calculer l'équivalent en noises d'un sorcier.

```
1 <?php
2 $ron = ['name' => 'Ron Weasley', 'knuts' => 10, 'sickles' => 2, 'galleons' => 0];
3 $knuts = $ron['knuts'];
4 $knuts += $ron['sickles'] * 29;
5 $knuts += $ron['galleons'] * 17 * 29;
6 echo $ron['name'].' possède l\'équivalent de '.$knuts.' noises';
```

<sup>1</sup> <https://www.php.net/manual/fr/language.types.array.php>



Pour réaliser cet exercice, vous aurez besoin de travailler sur l'environnement de travail :



### Question 1

[solution n°3 p.21]

En vous basant sur ce code, combien de mornilles possède Ron ?

### Question 2

[solution n°4 p.21]

Exécutez ce code dans repl.it. Quel est l'équivalent en noises que Ron possède ?

### Question 3

[solution n°5 p.21]

La famille Weasley a gagné au loto, Ron se voit ainsi plus riche de 5 gallions ! Modifiez le code pour recalculer sa richesse en noises.

### Question 4

[solution n°6 p.21]

Afin d'empocher cet argent, il faut prouver que Ron soit majeur ! Modifiez le code afin de renseigner son âge dans le tableau.

### Question 5

[solution n°7 p.21]

Suite à son dernier passage chez les moldus, c'est-à-dire les non-sorciers, il reste également à Ron 15 euros. Sachant qu'un euro vaut 61 noises, modifiez le code pour prendre en compte ses euros dans le tableau et le calcul. Quel est l'équivalent en noises qu'il possède ?

## VI. Les tableaux multidimensionnels

### Objectif

- Apprendre à créer et à manipuler un tableau multidimensionnel

### Mise en situation

Un tableau multidimensionnel est un tableau ayant pour valeurs d'autres tableaux, ayant eux-mêmes leurs propres valeurs (qui peuvent également être des tableaux).

Par exemple, pour avoir une liste d'utilisateurs ayant chacun leurs propres données, nous pourrions avoir un seul tableau contenant tous les utilisateurs et, pour chaque utilisateur, un tableau avec ses données.

#### Méthode Déclarer un tableau multidimensionnel

Pour déclarer un tableau à plusieurs dimensions, nous allons déclarer un tableau avec la méthode habituelle et nous allons lui insérer comme valeurs d'autres tableaux.

#### Exemple

Dans l'exemple ci-dessous, nous avons créé un tableau `$users` contenant trois valeurs, qui sont trois tableaux associatifs (chacun de ces tableaux associatifs ayant ses propres couples clés / valeurs contenant les données sur un utilisateur).

Le tableau a été déclaré sur plusieurs lignes afin d'améliorer la lisibilité du code.

1 <https://repl.it/>

```
1 <?php
2
3 $users = [
4   ['lastName' => 'Martin', 'firstName' => 'Jean'],
5   ['lastName' => 'Doe', 'firstName' => 'Jane'],
6   ['lastName' => 'Dupont', 'firstName' => 'Marc'],
7 ];
```

### Méthode

Pour utiliser des tableaux multidimensionnels, on utilise récursivement la syntaxe utilisée pour les tableaux monodimensionnels.

### Remarque

On peut combiner les tableaux associatifs et indexés par des entiers.

### Exemple Insertion de données

Dans l'exemple ci-dessous, nous ajoutons dans le tableau `$users` le tableau associatif d'un nouvel utilisateur (Durant Clara).

```
1 <?php
2
3 $users = [
4   ['lastName' => 'Martin', 'firstName' => 'Jean'],
5   ['lastName' => 'Doe', 'firstName' => 'Jane'],
6   ['lastName' => 'Dupont', 'firstName' => 'Marc'],
7 ];
8
9 $users[] = ['lastName' => 'Durant', 'firstName' => 'Clara'];
```

### Exemple Lecture de données

Dans l'exemple ci-dessous, nous affichons (grâce à `var_dump()`) le tableau associatif de l'utilisateur Martin / Jean, car il est à la clé 0 du tableau `$users`.

```
1 <?php
2
3 $users = [
4   ['lastName' => 'Martin', 'firstName' => 'Jean'],
5   ['lastName' => 'Doe', 'firstName' => 'Jane'],
6   ['lastName' => 'Dupont', 'firstName' => 'Marc'],
7 ];
8
9 var_dump($users[0]);
```

Le résultat affiché sera donc :

```
1 array(2) {
2   ["lastName"]=>
3   string(6) "Martin"
4   ["firstName"]=>
5   string(4) "Jean"
6 }
```

Nous pouvons également accéder directement aux valeurs des tableaux situés dans le tableau multidimensionnel.

Dans l'exemple ci-dessous, nous affichons directement (avec `echo`) la valeur Martin.

```
1 <?php
2
3 $users = [
4     ['lastName' => 'Martin', 'firstName' => 'Jean'],
5     ['lastName' => 'Doe', 'firstName' => 'Jane'],
6     ['lastName' => 'Dupont', 'firstName' => 'Marc'],
7 ];
8
9 echo $users[0]['lastName'];
```

#### **Syntaxe** Accéder directement aux valeurs stockées en lecture (tableau à deux dimensions)

Pour manipuler un tableau à deux dimensions, on utilise deux fois les crochets.

```
1 echo $myArray[key1][key2];
```

#### **Syntaxe** Accéder directement aux valeurs stockées en écriture (tableau à deux dimensions)

```
1 $myArray[key1][key2] = 'value';
```

#### **Exemple** Modification de données

Dans l'exemple ci-dessous, le tableau associatif situé à la clé 0 (`['lastName' => 'Martin', 'firstName' => 'Jean']`) a été remplacé par `['lastName' => 'Durand', 'firstName' => 'Elyse']`.

Le résultat affiché avec `echo` est donc : Durand.

```
1 <?php
2
3 $users = [
4     ['lastName' => 'Martin', 'firstName' => 'Jean'],
5     ['lastName' => 'Doe', 'firstName' => 'Jane'],
6     ['lastName' => 'Dupont', 'firstName' => 'Marc'],
7 ];
8
9 $users[0] = ['lastName' => 'Durand', 'firstName' => 'Elyse'];
10
11 echo $users[0]['lastName'];
```

Dans l'exemple ci-dessous, 'Jane' a été modifié en 'Jeanne'.

```
1 <?php
2
3 $users = [
4     ['lastName' => 'Martin', 'firstName' => 'Jean'],
5     ['lastName' => 'Doe', 'firstName' => 'Jane'],
6     ['lastName' => 'Dupont', 'firstName' => 'Marc'],
7 ];
8
9 $users[1]['firstName'] = 'Jeanne';
10
11 echo $users[1]['firstName'];
```

### Syntaxe À retenir

Un tableau multidimensionnel est un tableau ayant pour valeurs d'autres tableaux, ayant eux-mêmes leurs propres valeurs (qui peuvent également être des tableaux).

### Complément

Les tableaux (documentation PHP)<sup>1</sup>

## VII. Exercice : Appliquez la notion

Soit le tableau multidimensionnel suivant :

```
1 <?php
2     $teams = [
3         [
4             'name' => 'Golden Guardians',
5             'members' => [
6                 ['firstName' => 'Kevin', 'lastName' => 'Yarnell'],
7                 ['firstName' => 'Can', 'lastName' => 'Çelik'],
8                 ['firstName' => 'Greyson', 'lastName' => 'Gilmer'],
9                 ['firstName' => 'Victor', 'lastName' => 'Huang'],
10                ['firstName' => 'Choi', 'lastName' => 'Jae-hyun'],
11                ['firstName' => 'Yuri', 'lastName' => 'Jew'],
12            ]
13        ],
14        [
15            'name' => 'Immortals',
16            'members' => [
17                ['firstName' => 'Paul', 'lastName' => 'Boyer'],
18                ['firstName' => 'Jake', 'lastName' => 'Puchero'],
19                ['firstName' => 'Jérémy', 'lastName' => 'Valdenaire'],
20                ['firstName' => 'Johnny', 'lastName' => 'Ru'],
21                ['firstName' => 'Nickolas', 'lastName' => 'Surgent'],
22            ]
23        ],
24    ];
```

Pour réaliser cet exercice, vous aurez besoin de travailler sur l'environnement de travail :



### Question 1

[solution n°8 p.22]

En vous aidant de repl.it, affichez le nom de la première équipe.

#### Indice :

Attention, les indices commencent à 0 !

<sup>1</sup> <https://www.php.net/manual/fr/language.types.array.php>

<sup>2</sup> <https://repl.it/>

**Question 2**

[solution n°9 p.22]

Affichez le prénom de M. Boyer.

**Indice :**

Il est dans la deuxième équipe, et c'est le première membre.

**Question 3**

[solution n°10 p.22]

On veut rajouter les informations du coach de chaque équipe. Rajoutez donc pour chaque équipe une clef "coach" qui pourra recevoir le nom et le prénom du coach, sachant que le coach des Golden Guardians est Nick Smith et que celui des Immortals est Thomas Si-Hassen.

Affichez le nom de famille des deux coachs pour tester votre ajout.

**Indice :**

Il n'y a qu'un coach par équipe, n'utilisez pas de tableau indexé. Par contre, un tableau associatif est nécessaire pour stocker les informations !

## VIII. Quelques fonctions utiles pour les tableaux

**Objectifs**

- Savoir utiliser les fonctions les plus utiles pour la manipulation des tableaux

**Mise en situation**

Il existe de nombreuses fonctions PHP pour les besoins les plus courants de manipulation de tableaux : les trier, connaître le nombre d'éléments, tester la présence d'une donnée, etc.

Nous allons en étudier quatre ici :

- L'affichage (`implode`)
- Tri simple (`sort`)
- Test de l'existence d'une donnée (`in_array`)
- Nombre d'éléments (`count`)

**Méthode**    **Affichage avec implode()**

Les tableaux étant des "blocs" de valeurs multiples, il n'est pas possible de les afficher avec l'instruction `echo` : PHP ne sait pas comment les afficher. Pour pallier ce problème, nous connaissons la fonction `var_dump`, mais elle n'est pas adaptée à l'affichage pour les utilisateurs.

Il existe cependant une autre fonction permettant d'afficher le contenu d'un tableau : `implode()`. Cette fonction va afficher tous les éléments d'un tableau, séparés par un séparateur choisi par le développeur.

**Exemple**

Imaginons une application de tirage de loto, pour laquelle nous devons afficher la liste des numéros gagnants séparés par une virgule et un espace :

```
1 <?php
2
3 $winningNumbers = [4, 5, 65, 78, 87];
4
5 echo implode(', ', $winningNumbers); // Affiche "4, 5, 65, 78, 87"
```

On pourrait aussi mettre un numéro par ligne, en séparant chaque numéro par un saut de ligne :

```
1 <?php
2
3 $winningNumbers = [4, 5, 65, 78, 87];
4
5 echo implode('<br>', $winningNumbers);
```

### Méthode Tri simple avec sort()

Cette fonction permet de trier les valeurs d'un tableau par ordre croissant.

### Exemple

```
1 <?php
2
3 $numbers = [99, 52, 104, 0, 16];
4
5 sort($numbers);
```

Le tableau est à présent trié du plus petit au plus grand : 0, 16, 52, 99, 104.

On peut également trier du texte, comme dans l'exemple ci-dessous, dans lequel le tableau sera égal après tri à : chat, koala, lion, zèbre.

```
1 <?php
2
3 $animals = ['zèbre', 'chat', 'lion', 'koala'];
4
5 sort($animals);
```

Il existe d'autres fonctions de tri, comme `rsort()` qui trie dans l'ordre décroissant, ou `krsort()`, qui permet de trier un tableau en fonction de ses clés et qui sera donc utile pour les tableaux associatifs.

### Méthode Test de l'existence d'une donnée avec in\_array()

La fonction `in_array()` permet de vérifier si une valeur appartient à un tableau : elle renvoie "vrai" ou "faux" (*true/false*).

Cette fonction est souvent utilisée dans une condition "SI" (*if*) afin d'exécuter ou non un bloc de code en fonction de la présence d'une donnée dans un tableau.

Cette fonction prend en paramètres :

1. La valeur recherchée.
2. Le tableau dans lequel la valeur est recherchée.

### Exemple

Dans l'exemple ci-dessous :

- `in_array('Bob', $users)` vérifie si 'Bob' est une valeur du tableau : comme c'est le cas, alors la fonction renvoie *true*.
- `in_array('Clara', $users)` vérifie si 'Clara' est une valeur du tableau : comme ce n'est pas le cas, alors la fonction renvoie *false*.

```
1 <?php
2
3 $users = ['John', 'Jane', 'Bob'];
4
5 var_dump(in_array('Bob', $users)); // true
6
7 var_dump(in_array('Clara', $users)); // false
```

Il existe d'autres fonctions similaires, comme `array_key_exists()` qui va vérifier si une clé d'un tableau existe, et va renvoyer "vrai" ou "faux" (*true/false*).

#### **Méthode** Nombre d'éléments avec `count()`

Cette méthode compte le nombre d'éléments dans le tableau.

#### **Exemple**

L'exemple ci-dessous va donc afficher 3.

```
1 <?php
2
3 $users = ['John', 'Jane', 'Bob'];
4
5 echo count($users); // affiche "3"
```

#### **Syntaxe** À retenir

- `implode ('', '$myArray)` ou `('<br>', $myArray)`
- `sort ($myArray)`
- `in_array('value', $myArray)`
- `count ($myArray)`

Il existe beaucoup d'autres fonctions décrites dans la documentation.

#### **Complément**

Fonctions sur les tableaux (documentation PHP)<sup>1</sup>

## Exercice : Appliquez la notion

[solution n°11 p.23]

Exercice

En vous aidant de la documentation PHP, quel est le nom de la fonction permettant de mélanger aléatoirement les éléments d'un tableau ?

Exercice

<sup>1</sup> <https://www.php.net/manual/fr/ref.array.php>

En vous aidant de la documentation PHP, que fait la fonction `array_chunk` ?

- ☐ Elle prélève un morceau du tableau donné
- ☐ Elle sélectionne une valeur aléatoire dans un tableau
- ☐ Elle remplit un tableau avec la valeur indiquée
- ☐ Elle découpe un tableau en plusieurs sous-tableaux de la taille donnée

Exercice

En vous aidant de la documentation PHP et de repl.it, quel est le résultat attendu du code ci-dessous ?

```
1 <?php
2
3 $numbers = [19, 21, 2, 15, 8];
4 sort($numbers);
5 array_shift($numbers);
6 echo $numbers[2];
```

## X. Auto-évaluation

### A. Exercice final

#### Exercice 1

[solution n°12 p.23]

Exercice

Quel est l'avantage des tableaux associatifs par rapport aux tableaux indexés ?

- ☐ Ne plus avoir à se rappeler que la première clef du tableau est égale à 0
- ☐ Pouvoir rajouter à la volée des éléments dans un tableau
- ☐ Pouvoir donner du sens aux clés

Exercice

Qu'est-ce qu'un tableau multidimensionnel ?

- ☐ Un tableau ayant pour valeurs d'autres tableaux
- ☐ Un tableau ayant pour clés d'autres tableaux
- ☐ Deux tableaux dans deux variables utilisés conjointement

Exercice

Qu'est-ce que le code suivant affiche ?

```
1 <?php
2 $car = ['brand' => 'Tesla', 'maxSpeed' => 250];
3 echo $car['brand'];
```

- ☐ Tesla
- ☐ 250
- ☐ brand
- ☐ maxSpeed



☐ Rien

#### Exercice

Qu'est-ce que le code suivant affiche ?

```
1 <?php
2
3 $men = ['Joe', 'Robert', 'Arthur'];
4 $women = ['Samantha', 'Simone'];
5
6 $people = array_merge($men, $women);
7 echo $people[2];
```

- ☐ Joe
- ☐ Robert
- ☐ Arthur
- ☐ Samantha
- ☐ Simone

#### Exercice

Qu'est-ce que le code suivant affiche ?

```
1 <?php
2 $names = ['John', 'Lisa', 'Robert', 'Alphonse'];
3 $numbers = [2, 1, 4, 3, 0];
4 $index = -1;
5 $index += $numbers[3];
6 $index--;
7 echo $names[$numbers[$index]];
```

- ☐ John
- ☐ Lisa
- ☐ Robert
- ☐ Alphonse

#### Exercice

Avec ce tableau :

```
1 <?php
2
3 $drinks = [
4     [
5         'name' => 'Menthe',
6         'price' => 2.0,
7         'ingredients' => [
8             ['name' => 'sirop', 'quantity' => 1],
9             ['name' => 'eau', 'quantity' => 5],
10        ]
11    ],
12    [
13        'name' => 'Mojito',
14        'price' => 4.0,
15        'ingredients' => [
```

```

16         ['name' => 'rhum', 'quantity' => 1],
17         ['name' => 'citron vert', 'quantity' => 0.5],
18         ['name' => 'eau gazeuse', 'quantity' => 5],
19         ['name' => 'menthe', 'quantity' => 10],
20         ['name' => 'sucre', 'quantity' => 1],
21     ]
22 ]
23 ];

```

Comment récupérer le nombre de feuilles de menthe d'un mojito ?

- ☐ \$drinks[0]['ingredients'][3]['quantity']
- ☐ \$drinks[1]['ingredients'][4]['quantity']
- ☐ \$drinks[2]['ingredients'][4]['quantity']
- ☐ \$drinks[1]['ingredients'][3]['quantity']
- ☐ \$drinks[0]['ingredients'][4]['quantity']

Exercice

Avec le tableau précédent, que retourne `array_pop($drinks) ['name']` ?

- ☐ Menthe
- ☐ eau
- ☐ Mojito
- ☐ sucre
- ☐ Cela soulève une erreur

## B. Exercice : Défi

Soit deux étudiants Hugo et Simone, représentés par les tableaux ci-après :

```

1 $student1 = [
2     'fullName' => 'Hugo',
3     'marks' => [
4         [
5             'name' => 'mathématiques',
6             'values' => [18, 12, 14, 8, 16]
7         ],
8         [
9             'name' => 'physique',
10            'values' => [11, 9, 7, 14, 18]
11        ],
12    ]
13 ];

1 $student2 = [
2     'fullName' => 'Simone',
3     'marks' => [
4         [
5             'name' => 'informatique',
6             'values' => [12, 11, 12, 10, 17]
7         ],
8         [
9             'name' => 'chimie',
10            'values' => [12, 8, 18, 14, 11]

```

```

11     ],
12     ]
13 ];

```

Pour réaliser cet exercice, vous aurez besoin de travailler sur l'environnement de travail :



### Question

[solution n°13 p.26]

Réalisez un programme qui :

- instancie `student1` et `student2`,
- crée un nouveau tableau multidimensionnel pour chaque étudiant, respectivement `$result1` et `$result2` qui contiennent pour chacun la moyenne de chacune des matières suivies.

La dernière instruction du programme sera `var_dump($result1, $result2)` et elle renverra le résultat ci-après :

```

1 array(2) {
2   [0]=>
3   array(2) {
4     ["name"]=>
5     string(14) "mathématiques"
6     ["average"]=>
7     float(13.6)
8   }
9   [1]=>
10  array(2) {
11    ["name"]=>
12    string(8) "physique"
13    ["average"]=>
14    float(11.8)
15  }
16 }
17 array(2) {
18   [0]=>
19   array(2) {
20     ["name"]=>
21     string(12) "informatique"
22     ["average"]=>
23     float(12.4)
24   }
25   [1]=>
26   array(2) {
27     ["name"]=>
28     string(6) "chimie"
29     ["average"]=>
30     float(12.6)
31   }
32 }

```

**Contrainte** : il est interdit dans le programme d'utiliser les mots "mathématiques", "physique", "informatique" ou "chimie" au sein des tableaux `result1` et `result2` (ces valeurs devront donc être récupérées depuis les tableaux `student1` et `student2`).

---

1 <https://repl.it/>

**Indice :**

```
1 <?php
2
3 $student1 = ...
4
5 $student2 = ...
6
7 $result1 = ...
8
9 $result2 = ...
10
11 var_dump($result1, $result2);
```

**Indice :**

Il n'y a pas de fonction toute faite en PHP pour calculer la moyenne d'un tableau. En revanche, il en existe une pour faire la somme, et une pour compter le nombre de valeurs d'un tableau.

**Solutions des exercices**

**p. 6 Solution n°1**

```

1 <?php
2
3 $names = ['Lisa', 'Mathieu', 'Laure', 'Simon'];
4 echo 'Bonjour' . ' ' . $names[2];

```

**p. 6 Solution n°2**

```

1 <?php
2
3 $names = ['Lisa', 'Mathieu', 'Laure', 'Simon'];
4 $names[] = 'John';
5 echo $names[4];

```

**p. 9 Solution n°3**

La valeur de la clef "sickles" est 2, il a donc deux mornilles.

**p. 9 Solution n°4**

Ron possède l'équivalent de 68 noises.

**p. 9 Solution n°5**

```

1 <?php
2 $ron = ['name' => 'Ron Weasley', 'knuts' => 10, 'sickles' => 2, 'galleons' => 5];
3 $knuts = $ron['knuts'];
4 $knuts += $ron['sickles'] * 29;
5 $knuts += $ron['galleons'] * 17 * 29;
6 echo $ron['name'] . ' possède l\'équivalent de ' . $knuts . ' noises';

```

Ce qui affiche : Ron Weasley possède l'équivalent de 2533 noises.

**p. 9 Solution n°6**

```

1 <?php
2 $ron = ['name' => 'Ron Weasley', 'knuts' => 10, 'sickles' => 2, 'galleons' => 5,
'euros' => 15];
3 $ron['age'] = 18;
4 echo 'Ron est majeur car il a ' . $ron['age'] . ' ans.';

```

**p. 9 Solution n°7**

```
1 <?php
2 $ron = ['name' => 'Ron Weasley', 'knuts' => 10, 'sickles' => 2, 'galleons' => 5, 'euros'
=> 15];
3 $knuts = $ron['knuts'];
4 $knuts += $ron['sickles'] * 29;
5 $knuts += $ron['galleons'] * 17 * 29;
6 $knuts += $ron['euros'] * 61;
7 echo $ron['name'].' possède l\'équivalent de '.$knuts.' noises';
```

Ron Weasley possède maintenant l'équivalent de 3448 noises.

#### p. 12 Solution n°8

```
1 <?php
2
3 echo $teams[0]['name'];
```

#### p. 13 Solution n°9

```
1 <?php
2
3 echo $teams[1]['members'][0]['firstName'];
```

#### p. 13 Solution n°10

```
1 <?php
2 $teams = [
3     [
4         'name' => 'Golden Guardians',
5         'members' => [
6             ['firstName' => 'Kevin', 'lastName' => 'Yarnell'],
7             ['firstName' => 'Can', 'lastName' => 'Çelik'],
8             ['firstName' => 'Greyson', 'lastName' => 'Gilmer'],
9             ['firstName' => 'Victor', 'lastName' => 'Huang'],
10            ['firstName' => 'Choi', 'lastName' => 'Jae-hyun'],
11            ['firstName' => 'Yuri', 'lastName' => 'Jew'],
12        ],
13        'coach' => ['firstName' => 'Nick', 'lastName' => 'Smith']
14    ],
15    [
16        'name' => 'Immortals',
17        'members' => [
18            ['firstName' => 'Paul', 'lastName' => 'Boyer'],
19            ['firstName' => 'Jake', 'lastName' => 'Puchero'],
20            ['firstName' => 'Jérémy', 'lastName' => 'Valdenaire'],
21            ['firstName' => 'Johnny', 'lastName' => 'Ru'],
22            ['firstName' => 'Nickolas', 'lastName' => 'Surgent'],
23        ],
24        'coach' => ['firstName' => 'Thomas', 'lastName' => 'Si-Hassen']
25    ],
26 ];
27 echo $teams[0]['coach']['lastName'];
28 echo '<br>';
29 echo $teams[1]['coach']['lastName'];
```


**Exercice p. 15 Solution n°11****Exercice**

En vous aidant de la documentation PHP, quel est le nom de la fonction permettant de mélanger aléatoirement les éléments d'un tableau ?

shuffle

**Exercice**

En vous aidant de la documentation PHP, que fait la fonction `array_chunk` ?


- ☐ Elle prélève un morceau du tableau donné  
*Non, ça, c'est `array_slice` (ou `array_splice`)*
  - ☐ Elle sélectionne une valeur aléatoire dans un tableau  
*C'est plutôt `array_rand`*
  - ☐ Elle remplit un tableau avec la valeur indiquée  
*C'est `array_fill`*
  - ☒ Elle découpe un tableau en plusieurs sous-tableaux de la taille donnée
-  `array_chunk` prend deux paramètres (un tableau et une taille) et découpe ce tableau en plusieurs sous-tableaux de la taille donnée.

**Exercice**

En vous aidant de la documentation PHP et de repl.it, quel est le résultat attendu du code ci-dessous ?


```
1 <?php
2
3 $numbers = [19, 21, 2, 15, 8];
4 sort($numbers);
5 array_shift($numbers);
6 echo $numbers[2];
```

19

-  `sort` trie les nombres par ordre croissant, ce qui donne : [2, 8, 15, 19, 21].  
`array_shift` retire le premier élément du tableau : [8, 15, 19, 21].  
`$numbers[2]` est donc 19.

**Exercice p. 16 Solution n°12****Exercice**

Quel est l'avantage des tableaux associatifs par rapport aux tableaux indexés ?

- ☐ Ne plus avoir à se rappeler que la première clef du tableau est égale à 0  
*Alors, techniquement, oui, mais ce n'est pas leur utilité première !*
  - ☐ Pouvoir rajouter à la volée des éléments dans un tableau  
*C'est aussi possible avec les tableaux associatifs*
  - ☒ Pouvoir donner du sens aux clés
-  Les tableaux associatifs permettent de donner du sens aux clés, et ainsi de pouvoir définir des structures à utiliser.

### Exercice

Qu'est-ce qu'un tableau multidimensionnel ?

- ☒ Un tableau ayant pour valeurs d'autres tableaux
- ☐ Un tableau ayant pour clés d'autres tableaux
- ☐ Deux tableaux dans deux variables utilisés conjointement
- ☐ Un tableau multidimensionnel est un tableau ayant pour valeurs d'autres tableaux.

### Exercice

Qu'est-ce que le code suivant affiche ?

```
1 <?php
2     $car = ['brand' => 'Tesla', 'maxSpeed' => 250];
3     echo $car['brand'];
```


- ☒ Tesla
- ☐ 250
- ☐ brand
- ☐ maxSpeed
- ☐ Rien

### Exercice

Qu'est-ce que le code suivant affiche ?

```
1 <?php
2
3 $men = ['Joe', 'Robert', 'Arthur'];
4 $women = ['Samantha', 'Simone'];
5
6 $people = array_merge($men, $women);
7 echo $people[2];
```

- ☐ Joe
- ☐ Robert
- ☒ Arthur
- ☐ Samantha
- ☐ Simone

 `array_merge` fusionne plusieurs tableaux passés en paramètres, donc `$people` contient `['Joe', 'Robert', 'Arthur', 'Samantha', 'Simone']`. L'élément à la clef 2 a donc pour valeur "Arthur".

### Exercice

Qu'est-ce que le code suivant affiche ?



```

1 <?php
2     $names = ['John', 'Lisa', 'Robert', 'Alphonse'];
3     $numbers = [2, 1, 4, 3, 0];
4     $index = -1;
5     $index += $numbers[3];
6     $index--;
7     echo $names[$numbers[$index]];

```

- ☐ John
- ☒ Lisa
- ☐ Robert
- ☐ Alphonse

🔍 On commence par calculer un indice, qui est égal à  $-1 + 3 - 1 = 1$ . On cherche ensuite, dans le tableau des nombres, l'élément à la clef 1 (donc le second), qui est 1. Enfin, on va chercher dans le tableau des noms l'élément à la clef 1, qui est "Lisa".

### Exercice

Avec ce tableau :

```

1 <?php
2
3 $drinks = [
4     [
5         'name' => 'Menthe',
6         'price' => 2.0,
7         'ingredients' => [
8             ['name' => 'sirop', 'quantity' => 1],
9             ['name' => 'eau', 'quantity' => 5],
10        ]
11    ],
12    [
13        'name' => 'Mojito',
14        'price' => 4.0,
15        'ingredients' => [
16            ['name' => 'rhum', 'quantity' => 1],
17            ['name' => 'citron vert', 'quantity' => 0.5],
18            ['name' => 'eau gazeuse', 'quantity' => 5],
19            ['name' => 'menthe', 'quantity' => 10],
20            ['name' => 'sucre', 'quantity' => 1],
21        ]
22    ]
23 ];

```

Comment récupérer le nombre de feuilles de menthe d'un mojito ?


- ☐ \$drinks[0]['ingredients'][3]['quantity']
- ☐ \$drinks[1]['ingredients'][4]['quantity']
- ☐ \$drinks[2]['ingredients'][4]['quantity']
- ☒ \$drinks[1]['ingredients'][3]['quantity']
- ☐ \$drinks[0]['ingredients'][4]['quantity']

## Exercice

Avec le tableau précédent, que retourne `array_pop($drinks) ['name']` ?

- ☐ Menthe
- ☐ eau
- ☒ Mojito
- ☐ sucre
- ☐ Cela soulève une erreur

*Cette syntaxe, bien que particulière, est valide : si une fonction retourne un tableau, on peut accéder à sa valeur directement*

 C'est "Mojito" qui est retourné.

### p. 19 Solution n°13

```

1 <?php
2
3 $student1 = [
4     'fullName' => 'Hugo Dubois',
5     'marks' => [
6         [
7             'name' => 'mathématiques',
8             'values' => [18, 12, 14, 8, 16]
9         ],
10        [
11            'name' => 'physique',
12            'values' => [11, 9, 7, 14, 18]
13        ],
14    ]
15 ];
16
17 $student2 = [
18     'fullName' => 'Simone Durand',
19     'marks' => [
20         [
21             'name' => 'informatique',
22             'values' => [12, 11, 12, 10, 17]
23         ],
24         [
25             'name' => 'chimie',
26             'values' => [12, 8, 18, 14, 11]
27         ],
28     ]
29 ];
30
31 $result1 = [
32     [
33         'name' => $student1['marks'][0]['name'],
34         'average' => array_sum($student1['marks'][0]['values'])/count($student1['marks'][0]
35     ['values']),
36     [
37         'name' => $student1['marks'][1]['name'],

```

```
38         'average' => array_sum($student1['marks'][1]['values'])/count($student1['marks'][1]
39         ['values']),
40     ];
41
42 $result2 = [
43     [
44         'name' => $student2['marks'][0]['name'],
45         'average' => array_sum($student2['marks'][0]['values'])/count($student2['marks'][0]
46         ['values']),
47     ],
48     [
49         'name' => $student2['marks'][1]['name'],
50         'average' => array_sum($student2['marks'][1]['values'])/count($student2['marks'][1]
51         ['values']),
52     ]
53 ];
54 var_dump($result1, $result2);
```