

Les tableaux

Qu'est-ce qu'un tableau ?

Un tableau est une liste d'éléments de n'importe quel type. Il peut être vide s'il ne contient aucun élément.

On va distinguer deux types de tableaux :

- Les tableaux dits « **simples** », qui contiennent simplement une liste de **valeurs**. Ces valeurs ne correspondront pas à une **clé** que nous aurons choisie, mais à un **index** qui sera attribué par défaut. Cet index commence à **0**, et s'incrémente autant de fois qu'il y a d'éléments dans le tableau. Exemple :

```
$arrayDonnees = array(  
    "Pomme", // 0  
    "Banane", // 1  
    "Kiwi", // 2  
    "Ananas", // 3  
    "Poire", // 4  
    "Litchi", // 5  
    "Fraise", // 6  
    "Framboise", // 7  
    "Myrtille" // 8  
);
```

Si l'on veut accéder à une des valeurs, alors on va pouvoir utiliser l'**index** du tableau pour cibler la valeur **souhaitée**, par exemple pour afficher « Litchi » :

```
echo $arrayDonnees[5];
```

- Les tableaux **associatifs**, consistent en une association de **clés** et de **valeurs**. Les tableaux associatifs ne comportent pas d'**index**, mais une clé suivie des caractères => pointant vers la **valeur associée à cette clé**. Exemple :

```
$user = array(  
    "pseudonyme" => "Pancake",  
    "email" => "pancake@boulangerie.net",  
    "date_inscription" => "2022-10-06"  
);
```

Si l'on veut accéder à une des valeurs, alors on va pouvoir utiliser la **clé** pour cibler la valeur **souhaitée**, par exemple pour afficher l'email :

```
echo $user['email'];
```

Les tableaux multi-dimensionnels

Nous avons vu plus haut qu'un tableau pouvait contenir des données de n'importe quel type. Parmi ces types figurent... Les tableaux 😊

Un tableau peut tout à fait contenir d'autres tableaux, comme par exemple ici où l'on a un tableau simple qui contient une liste de tableaux associatifs :

```
$users = array(
    array(
        "pseudonyme" => "Pancake",
        "email" => "pancake@boulangerie.net",
        "date_inscription" => "2022-10-06"
    ),
    array(
        "pseudonyme" => "Waffle",
        "email" => "waffle@boulangerie.net",
        "date_inscription" => "2022-05-14"
    ),
    array(
        "pseudonyme" => "Brioche",
        "email" => "brioche@boulangerie.net",
        "date_inscription" => "2021-02-16"
    ),
    array(
        "pseudonyme" => "Macaron",
        "email" => "macaron@boulangerie.net",
        "date_inscription" => "2024-03-26"
    ),
);
```

On dit alors de ce tableau qu'il est multidimensionnel, précisément à deux niveaux : Un tableau simple qui contient un autre niveau de tableau à l'intérieur.

Comment puis-je faire pour obtenir l'adresse e-mail de Brioche dans ce cas ?

On peut **cumuler les clés** comme ceci :

```
echo $users[2]['email'];
```

On a ici plusieurs fois nos crochets qui permettent de cibler une clé, et c'est normal : La première paire de crochets nous aide à obtenir l'**index** n°2 du tableau users, et la **clé** email nous permet d'obtenir la **valeur** correspondante dans le tableau renvoyé par `$users[2]`

Fonctions indispensables

- **array()** C'est une fonction basique qui sert à créer un tableau. Elle peut remplacer la notation entre crochets « [] » mais est un peu plus explicite, et surtout elle nous permettra de gérer les tableaux associatifs de manière beaucoup plus claire.

Exemple :

```
array(  
    "pseudonyme" => "Brioche",  
    "email" => "brioche@boulangerie.net",  
    "date_inscription" => "2021-02-16"  
)
```

- **array_push(\$tableau, \$truc_a_ajouter_au_tableau)** Est une fonction qui nous permet d'ajouter un élément à un tableau. Cet élément peut être de n'importe quel type, y compris un autre tableau. A le même effet que la notation `$tableau[] = $variable` ;
- **array_reverse(\$tableau)** Permet d'inverser l'ordre des objets dans un tableau. Attention, cela n'inverse l'ordre que du premier niveau
- **count(\$tableau)** Permet d'obtenir le nombre d'objets dans un tableau.
- **array_sum(\$tableau)** Permet d'obtenir la somme de tous les éléments d'un tableau. Attention, il faut que le tableau ne contienne que des éléments additionnables (Entiers ou décimaux), pas de chaînes de caractères
- **explode("-", \$chaîne_a_casser)** Permet d'obtenir un tableau en « cassant » une chaîne de caractères autour d'un caractère donné
- **implode("/", \$chaîne_a_casser)** Permet d'obtenir une chaîne de caractères en « collant » les éléments d'un tableau ensemble autour d'un caractère donné
- **sort(\$tableau)**, **rsort(\$tableau)**, **asort(\$tableau)**, **arsort(\$tableau)** permettent respectivement de trier un tableau par ordre croissant, décroissant, puis croissant en gardant les index, et décroissant en gardant les index.
- **unset(\$tableau["clé"])** Permet d'éliminer une variable. Si on lui passe un tableau avec une clé, il va simplement retirer l'association clé => valeur du tableau et elle ne sera plus accessible.
- **json_encode(\$tableau)** Permet de transformer un tableau en tableau JSON, à savoir une chaîne de caractère au format JSON
- **json_decode(\$chaîne_json)** Permet de transformer une chaîne de caractère au format JSON en Objet
- Lien complémentaire : <https://www.php.net/manual/fr/ref.array.php>

Fonctions de conditions

- `empty($tableau)`

Permet d'obtenir « true » ou « false » selon si le tableau contient quelque chose ou non.

- `array_key_exists("clé", $tableau)`

Permet d'obtenir « true » ou « false » selon si le tableau contient une clé telle que passée en premier paramètres.

- `is_array($tableau)`

Permet d'obtenir « true » ou « false » selon si notre variable est bien un tableau ou non