



PHP & SQL - Bases - 3 - Les Tableaux

▼ Cours	PHP
▼ Type	Guide
📎 Supports	https://www.php.net/manual/fr/language.types.array.php
▼ Difficulté	Facile

Imaginons que vous êtes en train d'écrire une histoire. Vos personnages ont évidemment plus de caractéristiques qu'un simple prénom. Au début, vous pourriez penser qu'il vous suffit de créer des variables telles que : \$character_one_name, \$character_one_age, \$character_two_name etc.

Cependant, vous remarquerez très vite que votre code devient difficile à organiser/manipuler et que vous avez du mal à retrouver vos personnages.

Pour pallier ce problème, vous pouvez faire appel aux tableaux !

Les tableaux sont très utiles pour grouper des variables dans un même endroit.

Je vous montre :

```
<?php  
$john = [0, "john", "doe", 35, "bleu", "brun"];
```

Grâce aux tableaux, vous avez réussi à stocker toutes les caractéristiques de votre personnage au même endroit !

Vous pouvez placer une infinité de valeurs au sein d'un tableau. C'est à vous de vous limiter au nombre d'éléments pour garder toujours de la clarté.

Si vous essayez de faire un `echo $john;` vous allez avoir une nouvelle erreur :

Warning: Array to string conversion in
C:\xampp\htdocs\Decouverte\index.php on line 4 Array

Cela signifie “Attention, vous avez essayé d'afficher une chaîne de caractères, mais vous me demandez d'afficher un tableau.”

Pour pouvoir afficher un élément du tableau, il faudra spécifier la valeur de l'indice de cet élément dans le tableau (en commençant par 0).

Ainsi, si j'écris :

```
<?php
$john = [0, "john", "doe", 35, "bleu", "brun"];

echo $john[1];
```

PHP me retournera `john`.

Pour afficher le tableau entier, il faudra utiliser `print_r()` :

```
<?php
$john = [0, "john", "doe", 35, "bleu", "brun"];

print_r($john);
```

`print_r()` permet d'afficher toutes les variables sans exception. Dans un tableau, il sera capable de reconnaître les clés et les éléments d'un tableau.

 “Vous avez dit clé ?”

Oui, dans un tableau, la clé correspond à l'indice de chaque élément. Comme vous l'avez découvert en copiant le bloc de code au-dessus, la clé de “john” est 1, la clé de “bleu” est 4 etc.

Il est possible de définir vous-mêmes les clés pour vous y retrouver plus facilement et faire ce que l'on appelle un tableau associatif :

```
<?php
$john = [
'id' => 0,
'name' => "john",
'lastname' => "doe",
'age' => 35,
'color' => "bleu",
```

```
'hairColor' => "brun"  
];
```

On a réussi à organiser notre tableau encore plus facilement ! On peut maintenant trouver l'âge de John encore plus facilement en faisant un simple `echo $john['age']` :

Exercice 3 :

Vous êtes actuellement recruté par un libraire pour alimenter son site Web. Les utilisateurs du site peuvent déposer des commentaires aux différents livres de la librairie pour inciter les autres utilisateurs à lire (ou non) les bouquins disponibles.

(Bien sûr, un vrai programme comme celui-ci sera plus complexe que sur cet exercice).

Copiez ce bloc de code dans votre fichier `index.php`.

```
<?php  
$john = [  
    'id' => 0,  
    'name' => "john",  
    'posts' => [  
        [  
            'id' => 0,  
            'title' => "Ce livre était incroyable.",  
            'content' => "J'ai adoré parcourir chaque page. J'ai dévoré ce livre en un rien de temps!"  
        ]  
    ]  
];
```

Votre mission si vous l'acceptez est de réussir à afficher le contenu du premier commentaire de John.

Pour vous aider, vous pouvez mener vos propres recherches avec le mot clé : [Multidimensional Array](#)