

PHP

04 - Les tableaux

I. Introduction

Un tableau est une variable contenant une liste de valeurs regroupées sous forme d'un tableau en lignes et en colonnes.

Les tableaux ont une grande importance dans la création de script PHP, comme dans bien d'autres langages d'ailleurs.

Dans le cas d'accès à des bases de données par exemple certaines fonctions retournent des va-leurs sous forme de tableaux.

1. Déclaration de tableaux

Tableaux simples

Exemple :

```
<?php
$tableau[0] = "Pomme";
$tableau[1] = "Poire";
$tableau[2] = "Banane";
```

ou

```
$tableau = array("Pomme", "Poire", "Banane");
```

Ces deux exemples sont en fait identiques dans le résultat.

Tableaux à plusieurs dimensions

```
<?php
$tab1[] = array(1, "janvier", "2016");
$tab1[] = array(2, "février", "2017");
$tab1[] = array(3, "mars", "2018");
$tab1[] = array(4, "avril", "2019");
```

Exemple :

```
// Affiche : 3 mars 2018
echo $tab1[2][0]." ".$tab1[2][1]." ".$tab1[2][2]."<br>;
```

Tableaux associatifs

Comme son nom l'indique, le tableau associatif permet d'associer une valeur à une autre plus représentative que celle-ci.

Imaginons que vous vouliez représenter ce que vous avez payé comme facture de téléphone par mois sur une année. Si vous utilisez un tableau classique vous allez associer chaque ligne du tableau aux mois et la colonne aux valeurs.

[illegible]

Grâce aux tableaux associatifs vous allez représenter le même tableau comme suit :

```
$facture["Janvier"] = 500;
$facture["Février"] = 620;
$. . . . .
$facture["décembre"] = 300;
```

Vous pouvez aussi l'écrire en utilisant l'opérateur de l'association \Rightarrow (signe égal et signe supérieur à)

```
$facture = array("Janvier"=>500,"Février"=>620,...,"Décembre"=>300);
```

II. Manipulations de tableaux

1. Manipulation des tableaux associatifs

Après avoir créé un tableau associatif, il est bien sûr indispensable de pouvoir manipuler et trier les valeurs de celui-ci. Pour cela PHP nous offre une multitude de fonctions.

Exemple :

```
$facture = array("Janvier"=>500, "Février"=>620, "Mars"=>300, "Avril"=>130,
"Mai"=>560, "Juin"=>350);
$facture_sixmois=0;

foreach ($facture as $mois => $valeur)
{
    echo "Facture du mois de $mois : $valeur Euros<br />";
    $facture_sixmois += $valeur;
}

echo "Facture total de six mois : <b>$facture_sixmois Euros</b>";
?>
```

L'instruction `foreach` permet d'extraire un couple (clé, valeur) à chaque tour de boucle.

2. Tri dans les tableaux

PHP propose un ensemble complet de fonctions pour le tri de tableaux.

La fonction `sort()`

Cette fonction vous permet de trier dans l'ordre alphabétique ou numérique les données d'un tableau.

```
$nom = array("franck","laurent","caroline","magali","veronique");
sort($nom);

for ($nb1=0;$nb1<=count($nom)-1; $nb1++)
{
    echo "$nom[$nb1]<br>";
}
```

Le résultat affiche *caroline, franck, laurent, magali, veronique*.

La fonction `rsort(array)` ;

Tri décroissant d'un tableau

```
$nom = array("franck","laurent","caroline","magali","veronique");
rsort($nom);

for ($nb1=0;$nb1<=count($nom)-1; $nb1++)
{
    echo "$nom[$nb1]<br>";
}
```

Le résultat affiche *veronique magali laurent franck caroline*.

3. Instructions de manipulation de tableaux

La fonction `arsort()`

Tri décroissant sur un tableau associatif, l'indexation des clefs est conservée. Le tri se fait sur les valeurs, pas sur les clés :

```
$tableau = array("a" => "Lundi",
                 "b" => "Mardi",
                 "c" => "Mercredi",
                 "d" => "Jeudi");
```

La fonction `arsort()`

```
foreach($tableau as $cle => $valeur)
{
    echo "Jour[$cle] = ".$tableau[$cle]."<br>";
}
```

Le résultat donne : *Jour[c] = Mercredi Jour[b] = Mardi Jour[a] = Lundi Jour[d] = Jeudi*.

La fonction `asort()`

Même chose que `arsort()` mais ici le tri se fait en ordre croissant.

```
$tableau = array("a"=>"Lundi",
                 "b"=>"Mardi",
```

```
"c"=>"Mercredi",  
"d"=>"Jeudi");
```

La fonction `asort()`;

```
foreach($tableau as $key => $valeur)  
{  
    echo "Jour[$key] = ".$valeur."<br>";  
}
```

Le résultat donne : *Jour[d] = Jeudi Jour[a] = Lundi Jour[b] = Mardi Jour[c] = Mercredi.*

La fonction `count()`;

Retourne le nombre d'éléments d'un tableau.

```
$tableau = array(1,2,3,4);  
$nbr = count($tableau);  
echo $nbr;
```

Renvoi 4 (4 éléments dans le tableau).

La fonction `sizeof()` fait strictement la même chose; il s'agit d'un *alias* de `count()`

La fonction `array_push()`;

Cette instruction permet d'ajouter un élément à la fin du tableau.

```
$tableau = array("Lundi", "Mardi", "Mercredi");  
array_push ($tableau, "Jeudi");
```

La fonction `array_pop()`

Extrait un élément de la fin du tableau :

```
$tableau = array("Lundi", "Mardi", "Mercredi");  
$jour = array_pop($tableau);
```

Après cette opération, `$tableau` ne contient plus que *Lundi* et *Mardi*.

La fonction `array_unshift()`

Ajoute un élément en début de tableau

```
$tableau = array("Jeudi", "Vendredi");  
array_unshift($tableau, "Lundi", "Mardi", "Mercredi");
```

Après cette opération `$tableau` contient *Lundi, Mardi, Mercredi, Jeudi, Vendredi*, dans cet ordre.

La fonction `array_shift()`

Dépile un élément (et un seul) en début du tableau

```
$tableau = array("Jeudi", "Vendredi");  
$jour = array_shift($tableau);
```

Après cette opération, `$tableau` contient *Vendredi* et `$jour` contient *Jeudi*.

III. Références

- [array_change_key_case](#) — Change la casse de toutes les clés d'un tableau
- [array_chunk](#) — Sépare un tableau en tableaux de taille inférieure
- [array_column](#) — Retourne les valeurs d'une colonne d'un tableau d'entrée
- [array_combine](#) — Crée un tableau à partir de deux autres tableaux
- [array_count_values](#) — Compte le nombre de valeurs d'un tableau
- [array_diff_assoc](#) — Calcule la différence de deux tableaux, en prenant aussi en compte les clés
- [array_diff_key](#) — Calcule la différence de deux tableaux en utilisant les clés pour comparaison
- [array_diff_uassoc](#) — Calcule la différence entre deux tableaux associatifs, à l'aide d'une fonction de rappel
- [array_diff_ukey](#) — Calcule la différence entre deux tableaux en utilisant une fonction de rappel sur les clés pour comparaison
- [array_diff](#) — Calcule la différence entre des tableaux
- [array_fill_keys](#) — Remplit un tableau avec des valeurs, en spécifiant les clés
- [array_fill](#) — Remplit un tableau avec une même valeur
- [array_filter](#) — Filtre les éléments d'un tableau grâce à une fonction utilisateur
- [array_flip](#) — Remplace les clés par les valeurs, et les valeurs par les clés
- [array_intersect_assoc](#) — Calcule l'intersection de deux tableaux avec des tests sur les index
- [array_intersect_key](#) — Calcule l'intersection de deux tableaux en utilisant les clés pour comparaison
- [array_intersect_uassoc](#) — Calcule l'intersection de deux tableaux avec des tests sur les index, compare les index en utilisant une fonction de rappel
- [array_intersect_ukey](#) — Calcule l'intersection de deux tableaux en utilisant une fonction de rappel sur les clés pour comparaison
- [array_intersect](#) — Calcule l'intersection de tableaux
- [array_key_exists](#) — Vérifie si une clé existe dans un tableau
- [array_key_first](#) — Récupère la première clé d'un tableau
- [array_key_last](#) — Récupère la dernière clé d'un tableau
- [array_keys](#) — Retourne toutes les clés ou un ensemble des clés d'un tableau
- [array_map](#) — Applique une fonction sur les éléments d'un tableau
- [array_merge_recursive](#) — Combine un ou plusieurs tableaux ensemble, récursivement
- [array_merge](#) — Fusionne plusieurs tableaux en un seul
- [array_multisort](#) — Trie les tableaux multidimensionnels
- [array_pad](#) — Complète un tableau avec une valeur jusqu'à la longueur spécifiée
- [array_pop](#) — Dépile un élément de la fin d'un tableau
- [array_product](#) — Calcule le produit des valeurs du tableau
- [array_push](#) — Empile un ou plusieurs éléments à la fin d'un tableau
- [array_rand](#) — Prend une ou plusieurs clés, au hasard dans un tableau
- [array_reduce](#) — Réduit itérativement un tableau
- [array_replace_recursive](#) — Remplace récursivement dans le premier tableau les éléments des autres tableaux fournis
- [array_replace](#) — Remplace les éléments d'un tableau par ceux d'autres tableaux
- [array_reverse](#) — Inverse l'ordre des éléments d'un tableau
- [array_search](#) — Recherche dans un tableau la clé associée à la première valeur
- [array_shift](#) — Dépile un élément au début d'un tableau

- [array_slice](#) — Extrait une portion de tableau
- [array_splice](#) — Efface et remplace une portion de tableau
- [array_sum](#) — Calcule la somme des valeurs du tableau
- [array_udiff_assoc](#) — Calcule la différence entre des tableaux avec vérification des index, compare les données avec une fonction de rappel
- [array_udiff_uassoc](#) — Calcule la différence de deux tableaux associatifs, compare les données et les index avec une fonction de rappel
- [array_udiff](#) — Calcule la différence entre deux tableaux en utilisant une fonction de rappel
- [array_uintersect_assoc](#) — Calcule l'intersection de deux tableaux avec des tests sur l'index, compare les données en utilisant une fonction de rappel
- [array_uintersect_uassoc](#) — Calcule l'intersection de deux tableaux avec des tests sur l'index, compare les données et les index des deux tableaux en utilisant une fonction de rappel séparée
- [array_uintersect](#) — Calcule l'intersection de deux tableaux, compare les données en utilisant une fonction de rappel
- [array_unique](#) — Dédoublonne un tableau
- [array_unshift](#) — Empile un ou plusieurs éléments au début d'un tableau
- [array_values](#) — Retourne toutes les valeurs d'un tableau
- [array_walk_recursive](#) — Applique une fonction de rappel récursivement à chaque membre d'un tableau
- [array_walk](#) — Exécute une fonction fournie par l'utilisateur sur chacun des éléments d'un tableau
- [array](#) — Crée un tableau
- [arsort](#) — Trie un tableau en ordre inverse et conserve l'association des index
- [asort](#) — Trie un tableau et conserve l'association des index
- [compact](#) — Crée un tableau à partir de variables et de leur valeur
- [count](#) — Compte tous les éléments d'un tableau ou quelque chose d'un objet
- [current](#) — Retourne l'élément courant du tableau
- [each](#) — Retourne chaque paire clé/valeur d'un tableau
- [end](#) — Positionne le pointeur de tableau en fin de tableau
- [extract](#) — Importe les variables dans la table des symboles
- [in_array](#) — Indique si une valeur appartient à un tableau
- [key_exists](#) — Alias de array_key_exists
- [key](#) — Retourne une clé d'un tableau associatif
- [krsort](#) — Trie un tableau en sens inverse et suivant les clés
- [ksort](#) — Trie un tableau suivant les clés
- [list](#) — Assigne des variables comme si elles étaient un tableau
- [natcasesort](#) — Trie un tableau avec l'algorithme à "ordre naturel" insensible à la casse
- [natsort](#) — Trie un tableau avec l'algorithme à "ordre naturel"
- [next](#) — Avance le pointeur interne d'un tableau
- [pos](#) — Alias de current
- [prev](#) — Recule le pointeur courant de tableau
- [range](#) — Crée un tableau contenant un intervalle d'éléments
- [reset](#) — Remet le pointeur interne de tableau au début
- [rsort](#) — Trie un tableau en ordre inverse
- [shuffle](#) — Mélange les éléments d'un tableau
- [sizeof](#) — Alias de count
- [sort](#) — Trie un tableau
- [uasort](#) — Trie un tableau en utilisant une fonction de rappel
- [uksort](#) — Trie un tableau par ses clés en utilisant une fonction de rappel
- [usort](#) — Trie un tableau en utilisant une fonction de comparaison