

# PHP - Les tableaux

Date de dernière mise à jour : 27/06/2007 à 19:36

Source : <http://www.vulgarisation-informatique.com/tableaux-php.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)

Définition et présentation

Les tableaux vont vous changer la vie ! en fait, il s'agit de variables qui peuvent contenir plusieurs valeurs. On peut donc imaginer leur représentation sous forme d'un tableau. Pour différencier les valeurs, pour savoir comment on va dire à PHP "je veux telle valeur de cette variable" ; il faut bien distinguer les valeurs par des indices. Ces indices. Voilà comment on peut représenter tout ça :

Indice

Valeur

Maurice

1

Jean

2

Pierre

3

Paul

On peut voir ici qu'il s'agit d'une liste de prénoms. On appelle cela un **tableau**, aussi appelé **array**. La syntaxe de définition du tableau en PHP (pour dire "voilà, cette variable sera un tableau") sera de la forme suivante :

```
<?php $prenoms
= array();
//on déclare le tableau de prénoms, remarquez que cela se fait avec la fonction array() /*
    On va initialiser les valeurs pour les indices 0, 1, 2 et 3 */
$prenoms
[
] =
'Maurice'
;
$prenoms
[
1
] =
'Jean'
;
$prenoms
[
2
] =
'Pierre'
;
$prenoms
```

```
[
3
]=
'Paul'
;
?>
```

Les tableaux peuvent avoir des indices sous forme de nombres entiers ou de chaînes de caractères. On parle alors respectivement en PHP de tableaux indexés et de tableaux associatifs. Pour les tableaux indexés, le premier indice commence toujours à 0 lorsque le tableau est généré par une fonction. Vous pouvez cependant forcer les indices aux valeurs que vous souhaitez, mais cela peut être source d'erreurs. Faites-y donc attention ;)

Les tableaux à indices numériques (indexés)

En PHP, les tableaux indexés diffèrent uniquement des tableaux associatifs par leurs indices. Ici, ils sont numériques. Nous avons vu dans le premier exemple de ce cours comment affecter différentes valeurs à un tableau :

```
<?php $prenoms
= array();
//on déclare le tableau de prénoms, remarquez que cela se fait avec la fonction array() /*
    On va initialiser les valeurs pour les indices 0, 1, 2 et 3 */
$prenoms
[
]=
'Maurice'
;
$prenoms
[
1
]=
'Jean'
;
$prenoms
[
2
]=
'Pierre'
;
$prenoms
[
3
]=
'Paul'
;
?>
```

Voyons une syntaxe équivalente :

```
<?php $prenoms
= array(
0
=>
'Maurice'
,
```

```

1
=>
'Jean'
,
2
=>
'Pierre'
,
3
=>
'Paul'
);
//on initialise le tableau de prénoms
?>

```

Dans le cas où vous souhaitez garder le premier indice à zéro (et les suivants incrémentés de 1 à chaque fois) vous pouvez également utiliser la syntaxe suivante :

```

<?php $prenoms
= array(
'Maurice'
,
'Jean'
,
'Pierre'
,
'Paul'
);
//on initialise le tableau de prénoms
?>

```

C'est bien beau tout ça, mais comment afficher la valeur du tableau pour un indice spécifié ? bah on utilise toujours echo, mais cette fois-ci voilà comment on accède au tableau :

```

<?php
echo
$prenoms
[

];
//Va afficher "Maurice"
echo
$prenoms
[
1
];
//Va afficher "Jean"
...
?>

```

Les tableaux associatifs

PHP peut également vous permettre de distinguer les différentes valeurs de vos tableaux par des indices associatifs. Ils sont constitués de chaînes de caractères. Prenons cette fois l'exemple d'un article. On va le différencier par plusieurs choses :

- Son numéro unique
- Son nom
- Son prix

Voilà comment on pourrait représenter ça avec un tableau associatif :

```
<?php
/* Première possibilité */
$article
= array(
'Numero'
=>
77
,
'Nom'
=>
'Nom de l'article'
,
'Prix'
=>
7
);
/* Autre possibilité */
$article
= array();
$article
[
'Numero'
]=
77
;
$article
[
'Nom'
] =
'Nom de l'article'
;
$article
[
'Prix'
] =
7
;
?>
```

Si on veut afficher le prix de l'article, on utilisera toujours echo, mais de cette façon :

```
<?php
echo
$article
[
'Prix'
];
```

```
//Affiche "7"
```

```
?>
```

#### Parcourir un tableau

Les tableaux offrent une possibilité assez intéressante qui est de stocker plusieurs données dans une seule variable. PHP vous permet d'appliquer de nombreuses opérations sur les tableaux, donc celle de les parcourir. En effet, vous aurez par exemple besoin de lister tous les articles d'un tableau pour les afficher. On peut utiliser pour faire cette opération la fonction `foreach()`, dont nous allons voir la syntaxe de base ici :

```
<?php $tableau
= array(...); foreach(
$tableau
AS
$indice
=>
$valeur
) {
instructions
; }
?>
```

Voici ce que cela donne avec un exemple :

```
<?php $tableau
= array(
0
=>
'Chiffre 0'
,
1
=>
'Chiffre 1'
,
2
=>
'Chiffre 2'
); foreach(
$tableau
AS
$cle
=>
$valeur
) { echo
$cle
.
'|'
.
$valeur
.
"
; }
?>
```

Exécutez ce code et regardez ce que cela affiche ;) La variable `$cle` va récupérer l'indice du tableau, et la variable `$valeur` sa valeur associée. Si vous

n'avez pas besoin de récupérer l'indice, vous pouvez également utiliser la fonction `foreach()` comme ceci :

```
<?php $tableau
= array(
0
=>
'Chiffre 0'
,
1
=>
'Chiffre 1'
,
2
=>
'Chiffre 2'
); foreach(
$tableau
AS
$valeur
) {      echo
$valeur
.
"
; }
?>
```

Ceci fonctionne également pour les tableaux associatifs, vous récupèrerez alors la chaîne de caractères correspondant à l'indice dans la variable `$cle`.

Chapitre suivant



[Les dates](#)

[Présentation](#)

[La fonction date\(\)](#)

[Le timestamp PHP](#)

[Conversions : de date vers timestamp et inversement](#)

Source : <http://www.vulgarisation-informatique.com/tableaux-php.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)