Introduction aux tableaux en PHP

Pour créer un tableau nous utiliserons la fonction **array()**. Nous verrons par la suite que cette fonction peut être utilisée avec ou sans paramètres. Attention, un tableau en PHP correspond plus à une liste chaînée qu'à un tableau comme en Algo (on ne définit pas a priori par exemple le nombre de cases maximum du tableau).

1. Les tableaux presque comme en Algo

```
$tab = array() ;                // Création du tableau
$tab[0] = 3 ;                   // Affecte la valeur 3 à la 1ère case du tableau.
                                // Sans autre précision, l'indice de la première case est 0.
echo count($tab) ;              // Affiche le nombre d'éléments contenus dans le tableau
$i = 0 ;
while ($i < count($tab))        // Affiche tous les éléments contenus dans le tableau
{
        echo $tab[$i] ;
        $i++ ;
}
```

On peut ajouter un élément à la fin d'un tableau sans se soucier de sa position

$tab[] = 5 ; // Ajoute la valeur 5 à la fin du tableau

On peut également créer un tableau directement à partir des valeurs qu'il contient

$tab = array(3,5,4,7) ;

2. Les tableaux à plusieurs dimensions

En php, un tableau multi-dimensionnel est considéré comme un tableau de tableau.

$tab = array(array(3,2), array(5, 8) ,array(4,7)) ;

Permet de définir la matrice suivante :

3	**2**
5	**8**
4	**7**

Exemple d'utilisation d'une telle matrice :

```
$tab = array(array(3,2), array(5, 8) ,array(4,7)) ;

echo count($tab) ;              // Affiche le nombre d'éléments contenus dans le tableau

echo "<br>";

$i = 0 ;
while ($i < count($tab))        // Affiche tous les éléments contenus dans le tableau
{
        $j=0;
        while ($j <count($tab[$i]))
        {
                echo $tab[$i][$j] ;
                echo " ";
                $j++;
        }
        echo "<br>";
        $i++ ;
}
```

Tout comme précédemment, **$tab[0][]=6** permet d'ajouter la valeur 6 à la fin de la première ligne de la matrice.

3. Les tableaux associatifs

En php, on peut choisir la clé (l'indice) que l'on associe à chaque valeur. Cet indice n'est pas forcément de type entier et peut être par exemple de type chaîne.

Un tableau est donc, en php, une liste de couples **clé => valeur**.

Exemple pour un tableau simple : $tab = array(100, 87, 12) ;

$tab correspond à la liste (**0** => **100**, **1** => **87**, **2** => **12**) ;

Pour le vérifier utiliser la fonction **print_r(t)**.

De ce fait on peut donc choisir les clés que l'on associe à chaque valeur :

$tab = array(1 => 100, 4 => 87, 7 => 12) ;

Clé	***Valeur***
1	100
4	87
7	12

Ainsi seuls les accès $tab[1], $tab[4], $tab[7] retourneront une valeur. Tous les autres indices provoqueront une erreur.

Solution : Parcours du tableau avec les clés disponibles :

```
foreach($tab as $key => $valeur) // Pour chaque couple (key, valeur) du tableau tab
{
        echo "[".$key."] = ".$valeur. "<br>";
}
```

Autre exemple d'utilisation des clés :

Du fait qu'un tableau multi-dimensionnel soit un tableau de tableau, nous pouvons utiliser différents types dans les tableaux :

Exemple

```
$tab = array() ;

$tab["nom"] = array();
$tab["age"] = array();

$tab["nom"][1]="Chevalier";
$tab["age"][1]="25";
$tab["nom"][2]="Doe";
$tab["age"][2]="49";

$pos = array_search("Doe", $tab["nom"]) ;

print_r($pos);
```

<u>Quelques fonctions et procédures intéressantes :</u>

is_array(t) // Indique si t est un tableau (**true/false**)

array_search (v, t) // Recherche dans un tableau **t** la clé associée à une valeur **v**
// (renvoie la **clé** si l'élément existe sinon retourne **false**)

sort (t) // Trie le tableau selon les valeurs dans l'ordre croissant

rsort (t) // Trie le tableau selon les valeurs dans l'ordre inverse

ksort (t) // Trie le tableau selon les clés dans l'ordre croissant

krsort (t) // Trie le tableau selon les clés dans l'ordre inverse

…