Algorithmique et développement web S2
– 3. Tableaux –

Christophe BLANC

– IUT MMI –
IUT d'Allier
Université Clermont Auvergne
www.christophe-blanc.fr

2016-2017

- ✔ En PHP, un tableau est une collection (liste d'éléments) de couples/valeur.
- ✔ La clé peut être de type entier (tableau numérique) ou de type chaîne (tableau associatif : les clés ne sont pas forcément consécutives, ni ordonnées, et ce tableau peut présenter des clés entières et des clés de type chaîne).
- ✔ La valeur associée à la clé peut être de n'importe quel type, et notamment de type tableau : dans ce cas, le tableau est dit multidimensionnel.

Indices ordonnés consécutifs

Clé/Indice	**Valeur**
0	zéro
1	un
2	deux
3	trois

Indices non ordonnés non consécutifs

Clé/Indice	**Valeur**
20	vingt
30	trente
10	dix

Clé/Indice	Valeur
0	zéro
zéro	0
un	1
1	un
deux	2
2	deux
trois	3
3	trois

Clé/Indice	Valeur	
FRANCE	**Clé/Indice**	**Valeur**
	0	Paris
	1	Lyon
	2	Nantes
ITALIE	**Clé/Indice**	**Valeur**
	0	Rome
	1	Venise

Une variable de type tableau peut être définie explicitement grâce à la fonction *array* ou implicitement en utilisant une notation à crochets ([]).

Tableaux

Création : notation à crochets ([])

- ✔ Une variable utilisée pour la première fois avec une notation de la forme *$variable[...]* est automatiquement créée avec le type tableau.
- ✔ La même opération effectuée sur une variable déjà définie, avec un type scalaire, provoque un message d'erreur.
- ✔ Le contenu d'un tableau peut être défini par plusieurs affectations du type *$tableau[...] = valeur*.
- ✔ Avec une affectation du type *$tableau[...] = valeur*, PHP recherche le plus grand indice entier utilisé et associe la valeur à l'indice supérieur. Si le tableau est vide, l'élément est placé à l'indice 0.
- ✔ Avec une affectation du type *$tableau[clé] = valeur*, PHP associe la valeur indiquée à la clé indiquée (qui peut être de type entier ou de type chaîne).

Tableaux

Exemple Création : notation à crochets ([])

```php
<?php
$nombres[] = "zero";//=> indice 0
$nombres[] = "un";//=> indice max (0) + 1 = 1
$nombres[] = "deux";//=> indice max (1) + 1 = 2
$nombres[] = "trois";//=> indice max (2) + 1 = 3
$nombres[5] = "cinq";//=> indice 5
$nombres[] = "six";//=> indice max (5) + 1 = 6
$nombres["un"] = 1;//=> indice "un"
$nombres[] = "sept";//=> indice max (6) + 1 = 7
$nombres[-1] = "moins␣un";//=> indice -1
?>
```

Clé/Indice	Valeur
0	zero
1	un
2	deux
3	trois
5	cinq
6	six
un	1
7	sept
-1	moins un

Ces notations peuvent être utilisées pour construire un tableau multidimensionnel, sous la forme *$tableau[...] = $tableau_intérieur* ou *$tableau[...][...] = valeur*. La première notation permet de stocker un tableau dans un emplacement d'un autre tableau, et la deuxième notation, de stocker une valeur directement dans un emplacement situé à l'intérieur d'un autre tableau.

Tableaux

Exemple Création : notation à crochets ([])

```php
<?php
//creation d'un tableau contenant
//des villes de France
$villes_france[] = "Paris";
$villes_france[] = "Lyon";
$villes_france[] = "Nantes";
//stockage du tableau des villes de France
//dans le tableau des villes
$villes["FRANCE"] = $villes_france;
//idem avec les villes d'Italie
$villes_italie[] = "Rome";
$villes_italie[] = "Venise";
$villes["ITALIE"] = $villes_italie;
?>
```

```php
<?php
//stockage direct des villes dans le tableau
//      -pour la France
$villes["FRANCE"][] = "Paris";
$villes["FRANCE"][] = "Lyon";
$villes["FRANCE"][] = "Nantes";
//      -pour l'Italie
$villes["ITALIE"][] = "Rome";
$villes["ITALIE"][] = "Venise";
?>
```

Clé/Indice	**Valeur**	
FRANCE	**Clé/Indice**	**Valeur**
	0	Paris
	1	Lyon
	2	Nantes
ITALIE	**Clé/Indice**	**Valeur**
	0	Rome
	1	Venise

La fonction *array* permet de créer un tableau à partir d'une liste d'éléments.

Syntaxe

```
tableau array([mixte valeur[,...]])
```

Les clés/indices ne sont pas spécifiés et c'est un tabelau numérique à indices consécutifs commençant à 0 qui est créé : le premier argument de la fonction étant stocké à l'indice 0, le deuxième à l'indice 1, etc.

ou

```
tableau array([{chaine / entier} cle => mixte valeur[,...]])
```

L'indice ou la clé sont spécifiés soit par un entier, soit par une chaîne ; et une valeur lui est associée par l'opérateur $=>$.

```php
<?php
$nombres = array("zero","un","deux","trois",5 => "cinq",
          "six", "un" => 1, "sept", -1 => "moins un");
?>
```

Clé/Indice	Valeur
0	zero
1	un
2	deux
3	trois
5	cinq
6	six
un	1
7	sept
-1	moins un

La fonction *array* accepte les données de type tableau (soit une variable, soit un appel imbriqué à *array*), ce qui permet de construire un tableau multidimensionnel.

```php
<?php
$villes_france = array("Paris","Lyon",
                       "Nantes");
$villes_italie = array("Rome","Venise");
$villes = array("FRANCE" => $villes_france,
                "ITALIE" => $villes_italie);
?>
```

```php
<?php
$villes = array("FRANCE" => array("Paris",
        "Lyon","Nantes"),
                "ITALIE" => array("Rome",
                        "Venise"));
?>
```

Clé/Indice	**Valeur**	
FRANCE	**Clé/Indice**	**Valeur**
	0	Paris
	1	Lyon
	2	Nantes
ITALIE	**Clé/Indice**	**Valeur**
	0	Rome
	1	Venise

Deux besoins "types" existent, relatifs à la manipulation d'un tableau :

- ✔ accéder à un élément individuel du tableau
- ✔ parcourir le tableau

La notation à crochets est utilisée pour accéder, en lecture ou en écriture, à un élément individuel du tableau :

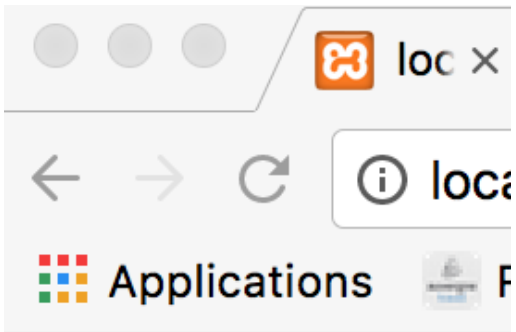
- ✔ *$tableau[i]* permet d'accéder à l'élément d'indice *i* du tableau (cas du tableau numérique, *i* étant un entier).
- ✔ *$tableau["x"]* permet d'accéder à l'élément de clé *"x"* du tableau (cas du tableau associatif, *"x"* étant une chaine).
- ✔ $tableau[] permet d'accéder, en écriture uniquement, à un nouvel élément du tableau qui se verra affecter un indice entier immédiatement supérieur au plus grand indice entier existant dans le tableau (0 s'il n'y en a pas).

Pour les tableaux multidimensionnels, plusieurs séries de crochets doivent être utilisées.

Tableaux

Manipulation exemple : accéder à un élément individuel du tableau

```php
<?php
$nombres = array("zero","un","deux","trois",5 => "cinq",
          "six", "un" => 1, "sept", -1 => "moins un");
echo $nombres[1],"</BR>";
echo $nombres["un"],"</BR>";
$villes = array("FRANCE" => array("Paris","Lyon","Nantes"),
                "ITALIE" => array("Rome","Venise"));
echo $villes["FRANCE"][0],"</BR>";
echo $villes["ITALIE"][1],"</BR>";
?>
```



un
1
Paris
Venise

De nombreuses méthodes peuvent être utilisées pour parcourir un tableau à l'aide des constructions suivantes :

- ✔ la structure de contrôle itérative *for*,
- ✔ la structure de contrôle itérative *while*,
- ✔ la structure de parcours de tableau *foreach*.

Nous allons étudier les deux méthodes les plus simples et finalement les plus efficaces. Elles ne nécessitent aucune connaissance particulière sur la nature du tableau (numérique, associatif, plage des indices/clés...)

Syntaxe :

- ✔
```
foreach (tableau as variable_valeur)
        { instructions}
```

  Cette première syntaxe permet de parcourir le tableau du début à la fin : à chaque itération la valeur courante du tableau est stockée dans la variable *variable_valeur* et les instructions entre accolades sont exécutées. Cette syntaxe est suffisante si le traitement n'a pas besoin de faire référence aux valeurs de la clé.

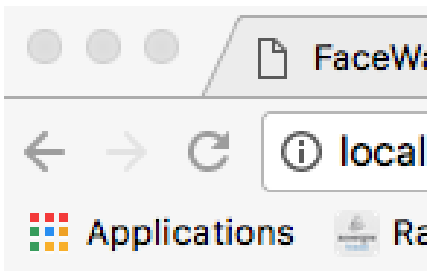
- ✔
```
foreach (tableau as variable_cle => variable_valeur)
        { instructions}
```

  La deuxième syntaxe fonctionne sur le même principe, mais à chaque itération, la clé courante est stockée dans la variable *variable_cle* et la valeur dans la variable *variable_valeur*. Cette syntaxe est pratique si le traitement a besoin de faire référence aux valeurs de la clé.

Tableaux

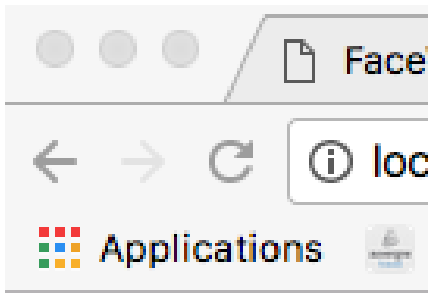
Manipulation : parcourir le tableau

```php
<?php
$nombres = array("zero","un","deux",
"zero" => 0, "un" => 1, "deux" => 2);
foreach ($nombres as $nombre)
{
        echo "$nombre</BR>";
}
?>
```



zero
un
deux
0
1
2

```php
<?php
$nombres = array("zero","un","deux",
"zero" => 0, "un" => 1, "deux" => 2);
foreach ($nombres as $cle => $nombre)
{
        echo "$cle␣=>␣$nombre</BR>";
}
?>
```



0 => zero
1 => un
2 => deux
zero => 0
un => 1
deux => 2

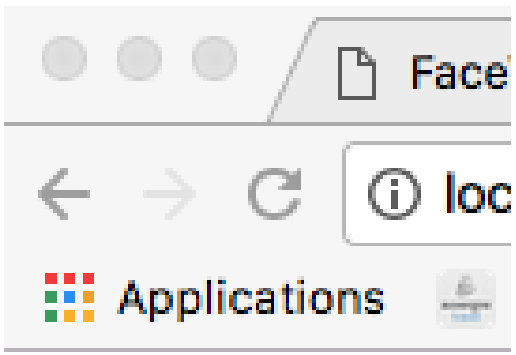
Syntaxe :

✔
```
while (list(variable_cle,variable_valeur) = each (tableau))
        { instructions}
```

Cette construction permet de parcourir du début à la fin : à chaque itération, la clé courante du tableau est stockée dans la variable *variable_cle*, la valeur dans la variable *variable_valeur* et les instructions entre accolades sont exécutées.
Si le tableau a été préalablement été manipulé, il faut faire appel à la fonction reset pour remettre le pointeur interne du tableau sur la première case et ainsi être certain de repartir du début.

```
mixte reset(tableau);
```

```php
<?php
$nombres = array("zero","un","deux",
"zero" => 0, "un" => 1, "deux" => 2);
while(list($cle,$nombre) = each($nombres))
{
        echo "$cle => $nombre</BR>";
}
?>
```



0 => zero
1 => un
2 => deux
zero => 0
un => 1
deux => 2

Nom	Rôle
count() ou sizeof()	compte le nombre d'éléments d'un tableau
in_array()	teste si une valeur est présente dans un tableau
array_search()	recherche une valeur dans un tableau
sort()	trie un tableau en ordre croissant
rsort()	trie un tableau en ordre décroissant
explode()	découpe une chaîne selon un séparateur et stocke les éléments dans un tableau
implode()	regroupe les éléments d'un tableau dans une chaîne à l'aide d'un séparateur
print_r()	affiche le tableau
array_push()	ajoute un élément à la fin du tableau
array_pop()	supprime un élément à la fin du tableau
array_shift()	supprime un élément au début du tableau
array_unshift()	ajoute un élément au début du tableau