

# Cours 15 : Tableaux en PHP

Introduction à la programmation Web



# Le tableau à index numérique

- Rappel : Techniquement, les tableaux sont des listes de variables.
- Ces listes contiennent des valeurs qui sont accessibles par des index.
  - Les tableaux de base ont un index numérique.
- On les déclare comme suit
  - Ex : `$fruits = array("pomme", "raisin", "orange")`
- On doit accéder aux éléments en lecture par leur index.
  - `$fruit = $fruits[0]` // \$fruit sera "pomme"
  - `$fruit = $fruits[1]` // \$fruit sera "raisin"
- On peut accéder en écriture aux éléments du tableau par leur index, ou ajouter à la fin du tableau en ne spécifiant pas d'index.
  - `$fruits[ ] = "pamplemousse"` // ajoutera pamplemousse à la position 3
- **Ne pas oublier : les tableaux à index numérique sont à base ZÉRO.**

# Appel de fonctions prédéfinies :

## La fonction count()

Pour connaître la taille d'un tableau, on utilise généralement la fonction count(), décrite à l'adresse suivante :

<https://www.php.net/manual/fr/function.count.php>

Celle-ci nous permet, entre autres, de parcourir le tableau. C'est l'équivalent de la fonction Taille vue en algorithmique. Elle nous **retourne** la taille du tableau envoyé en **paramètres**.

Voir exemple d'utilisation.

# Parcours d'un tableau à index numérique

Il est très courant, en PHP, d'avoir à parcourir l'entièreté d'un tableau, pour en afficher le contenu ou pour y trouver un élément, par exemple.

Pour parcourir un tableau, on utilise généralement **une boucle** for bouclant de 0 jusqu'à la longueur du tableau. La fonction **count** est très souvent utilisée ici pour obtenir la longueur du tableau, puisque cette longueur pourrait varier selon les modifications effectuées au tableau.

Voir exemple.

# La boucle foreach()

Il est aussi possible de passer à travers tous les éléments d'un tableau avec la boucle foreach, **qui ne nécessite pas d'indice**.

Elle prend la forme suivante :

```
foreach($leTableau as $element)
{
    echo $element;
}
```

où \$element prend la prochaine valeur du tableau à chaque tour de boucle.

# Tableaux associatifs

Parfois, il n'est pas pratique pour un programmeur de stocker des informations dans un tableau avec un index. Par exemple, comment stocker plusieurs valeurs qui ont un lien ensemble?

Par exemple, si un programmeur stocke dans un tableau plusieurs informations sur un individu, comme un nom, un prénom et une ville, il faudra déterminer dans quelle case se trouve la ville pour pouvoir y accéder de façon systématique. Le nombre représentant l'index pour la ville (qu'il soit 0, 1, 2, etc..) n'a aucune signification particulière et le programmeur aurait pu choisir n'importe quel ordre!

# Tableaux associatifs en PHP

PHP offre la possibilité d'utiliser, à la place des index numériques, des index constitués d'une chaîne de caractères.

Lorsqu'un tableau utilise ces index, on dit que c'est un tableau associatif. Ce tableau devient un genre de **dictionnaire** où chaque **valeur** est associée à une **clé** dans ce que l'on appelle **un couple clé / valeur**.

**Les couples clés-valeurs sont parfois créés lors de la déclaration, en utilisant l'opérateur =>.**

Exemple :

```
$individu = array("nom" => "Harvey", "prenom" => "Guillaume", "ville" => "Montréal");
```

# Utilisation des tableaux associatifs

On utilise ensuite ce tableau en utilisant la **clé** comme index pour obtenir ou spécifier la valeur. Par exemple, avec le tableau \$individu donné comme exemple précédemment :

```
echo $individu["nom"]; //affichera Harvey
```

```
$individu["tel"] = 5149449444; //créera une nouvelle case  
dans l'individu pour la clé tel et la valeur 5149449444
```

```
$individu["ville"] = "Chicoutimi"; //remplacera la valeur  
"Montréal" par "Chicoutimi" dans la case ville
```



# Parcourir un tableau associatif avec foreach()

L'instruction foreach peut être utilisée pour obtenir à la fois les clés et les valeurs avec la syntaxe suivante :

```
foreach($tableau as $cle => $valeur) {  
    echo $cle . " = " . $valeur;  
}
```

À chaque tour de boucle, \$cle deviendra la prochaine clé et \$valeur deviendra la prochaine valeur.

# Fonctions spécifiques pour les tableaux

Il existe une foule de fonctions pour traiter des tableaux.

Par exemple :

- `sort()` permet de trier par ordre alphabétique les valeurs d'un tableau.
  - `rsort()` le fait dans l'ordre inverse.
- `print_r()` permet d'afficher rapidement le contenu du tableau (pour tester).
- `array_search()` permet d'effectuer une recherche dans un tableau, `array_replace` d'y faire un remplacement, `array_merge()` de fusionner plusieurs tableaux...

Plusieurs autres sont décrites ici :

<https://www.php.net/manual/fr/ref.array.php>

# Tableaux multidimensionnels

Chaque case d'un tableau, en PHP, peut contenir un tableau, ce qui permet la création de tableaux multidimensionnels dans le cas où chaque case contient un tableau!

Voir exemple... et exercice!