Les tableaux associatifs, les formulaires et les variables de session

Contenu

- Retour sur les tableaux
- Les tableaux associatifs
- Lire les données d’un formulaire

Retour sur les tableaux

Retour sur les tableaux

- **Au dernier cours, nous avons vu qu'un tableau est une liste d'éléments accessibles par des indices**
  - C'est-à-dire des numéros représentant les positions des éléments
  - Le premier indice d'un tableau est 0

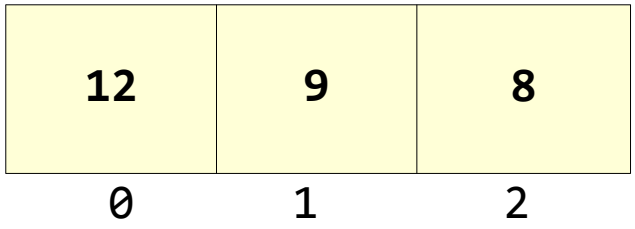
Indice	**0**	**1**	**2**	**...**	**99**
Valeur	42	17	35	...	76

Retour sur les tableaux

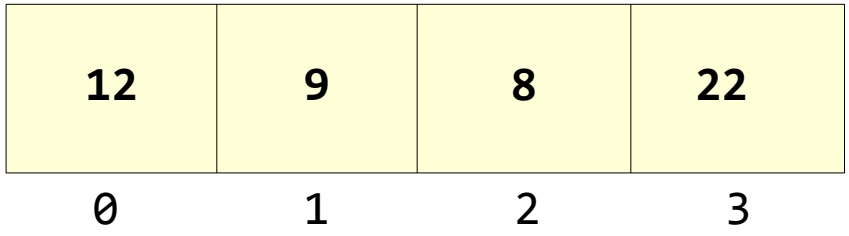
- Lorsqu'on ajoute un élément à la fin d'un tableau (avec array_push), il obtient son indice automatiquement

  - Ex:

```php
$monTableau = [12, 9, 8];
```




```php
array_push($monTableau, 22);
```



Retour sur les tableaux

- On peut utiliser une boucle For pour itérer sur les éléments d'un tableau, par exemple pour les afficher ou les modifier
- Ex:

```
/* Multiplier par 2 tous les éléments du tableau $monTableau
    sachant que le tableau a 10 éléments */
for ($i = 0; $i < 10; $i++) {
    $monTableau[$i] *= 2;
}
```

Retour sur les tableaux

- On peut aussi itérer sur un tableau à l'aide d'une boucle For-Each

```
// Afficher tous les éléments du tableau $monTableau
foreach ($monTableau as $element) {
    echo "$element <br />";
}
```

Les tableaux associatifs

Un tableau associatif, c'est quoi?

- **Jusqu'à maintenant, nous avons utilisé des tableaux numérotés**
  - Les indices sont des numéros attribués séquentiellement (le premier élément a l'indice 0, le deuxième a l'indice 1, etc.)
- **Avec un tableau associatif, les indices (qu'on appelle plutôt des clés) sont...**
  - Des chaînes de caractères
  - et/ou des nombres entiers arbitraires (non séquentiels)

Exemple 1

```php
$cours = [
    "420-74A-FE" => "Environnement Web",
    "420-7B4-FE" => "Exploitation d’une base de données",
    "420-715-FE" => "Programmation d’une application Web transactionnelle"
]; // on aurait aussi pu utiliser la syntaxe « array(...) »

echo $cours["420-7B4-FE"]; // Affiche « Exploitation d’une base de données »
```

Clé	Valeur
420-74A-FE	Environnement Web
420-7B4-FE	Exploitation d’une base de données
420-715-FE	Programmation d’une application Web transactionnelle

Exemple 2

```php
$indicatifsRegionaux = [
    450 => "Banlieue de Montréal",
    514 => "Montréal",
    819 => "Centre et ouest du Québec"
];

echo $indicatifsRegionaux[514]; // Affiche « Montréal »
```

Ici, même si les clés sont des nombre entiers, il s'agit d'un tableau associatif
et non d'un tableau numéroté, puisque les clés ne sont pas attribuées séquentiellement.

Clé	Valeur
450	Banlieue de Montréal
514	Montréal
819	Centre et ouest du Québec

Exemple 3

```
$mois = [
    1 => "Janvier", 2 => "Février", 3 =>  "Mars"
    4 => "Avril", 5 => "Mai", 6 => "Juin",
    7 => "Juillet", 8 => "Août", 9 => "Septembre",
    10 => "Octobre", 11 => "Novembre", 12 => "Décembre"
];
```

Ici, même si les clés sont des nombre entiers séquentiels, il s'agit d'un tableau associatif et non d'un tableau numéroté, puisque la première clé n'est pas 0.

Synonymes

- **Dans d'autres langages, on utilise parfois les termes suivants au lieu de tableau associatif:**
  - Dictionnaire
  - Map

Itérer sur un tableau associatif

- **On ne peut généralement pas utiliser une boucle For pour itérer sur un tableau associatif**
  - Puisque les clés ne sont pas des numéros séquentiels
- **On peut tout de même utiliser une boucle For-Each**

```
foreach ($cours as $titreCours) {
    echo "$titreCours <br />";
}
```

Boucle For-Each et clés

- Jusqu'à maintenant, nous avons utilisé la boucle For-Each uniquement pour accéder aux valeurs d'un tableau
- Les clés sont également accessibles
- Syntaxe:

```
foreach ($nomTableau as $cle => $valeur) {
    // ...
}
```

Exemple

```
foreach ($indicatifsRegionaux as $indicatif => $region) {
    echo "$indicatif: $region <br />";
}
```

Affichage:

450: Banlieue de Montréal
514: Montréal
819: Centre et ouest du Québec

Vérifier la présence d'une clé dans un tableau associatif

- `if (isset($nomTableau[clé]))`
  - Ex: `if (isset($cours["420-715-FE"]))`
  - Comme vous le faites déjà avec $_GET[...] ou toute autre variable

Conversion d’un tableau numéroté en tableau associatif

- Si on ajoute un indice non séquentiellement à un tableau numéroté, il sera converti en tableau associatif

```
$monTableau = ["Des pommes", "Des poires", "Des ananas"];
// Ici, $monTableau est un tableau numéroté
$monTableau[5] = "Des biscuits sodas";
// $monTableau est maintenant un tableau associatif
```

Clé	Valeur
0	Des pommes
1	Des poires
2	Des ananas
5	Des biscuits sodas

Question

Vous avez déjà utilisé un tableau associatif à plusieurs reprises durant les laboratoires. Quel est-il?

Lire les données d’un formulaire

Utilisation d’un formulaire

- **On peut utiliser un formulaire HTML pour transmettre des données à une page PHP**
- **Les formulaires sont omniprésents dans les applications Web transactionnelles**
  - Authentification
  - Inscription
  - Publication d’un commentaire
  - Paiement électronique
  - Etc.

Syntaxe

```
<form
    action="Nom du script PHP auquel envoyer les
données (ex: inscription.php)"
    method="GET ou POST"
>
  <input name="Nom qui sera utilisé par PHP pour
accéder à la valeur du champ" (...) />
</form>
```

Méthode GET

- **Si on choisit la méthode GET, les valeurs des champs seront passées directement dans l'URL**
  - Ce n'est généralement pas ce qu'on veut
- **On accède à ces données en PHP via le tableau associatif $_GET**

```
<form
    action="inscription.php"
    method="GET"
>
  <input type="text" name="prenom" />
</form>
```

Si l'utilisateur entre « Bob » dans le champ du formulaire, l'URL chargée sera « inscription.php?prenom=Bob »

La valeur saisie dans le champ texte sera accessible via **$_GET["prenom"]**

Méthode POST

- **Si on choisit la méthode POST, les valeurs des champs ne seront pas visibles dans l'URL**
  - Transmises dans les données (*corps*) de la requête HTTP
- **On accède à ces données en PHP via le tableau associatif $_POST**

```
<form
    action="inscription.php"
    method="POST"
>
  <input type="text" name="prenom" />
</form>
```

La valeur saisie dans le champ texte sera accessible via **$_POST["prenom"]**

Sécurité

- Il faut TOUJOURS valider les données qui proviennent d'un formulaire, même si on utilise des listes déroulantes, cases à cocher, boutons radio, etc.
- Rien n'empêche un utilisateur malveillant de modifier le code HTML du formulaire!
- Principe **NTUI**: *Never Trust User Input*

Sécurité

- **Attention lorsque vous affichez des données provenant de $_GET ou $_POST. Elles pourraient contenir du HTML!**
  - Un utilisateur malveillant pourrait même faire exécuter du JavaScript avec la balise *script*. On ne veut pas ça!
- **Solutions possibles:**
  - htmlspecialchars -> transforme certains caractères spéciaux d'une chaîne de caractères en entités HTML
    - Ex: < et > deviennent respectivement < et >
    - `$donnee = htmlspecialchars($_GET['donnee']);`
  - strip_tags -> retire toutes les balises HTML d'une chaîne de caractères
    - `$donnee = strip_tags($_GET['donnee']);`

Les variables de session

Les variables de session

- **Vous aurez remarqué qu'en PHP, nos variables n'existent que pour la durée de la génération de la page**
- **Nous voulons parfois conserver des données entre les exécutions d'une page, voire entre les pages**
  - Ex: identité de l'utilisateur connecté dans le cas d'un site Web avec une authentification
- **Les variables de session sont une façon d'y arriver**

Utilisation

- **Pour démarrer une session:**
  - `session_start();`
  - Doit être appelé avant tout envoi au navigateur (donc avant l'ouverture de la balise **html**)
- **Pour créer une variable de session:**
  - `$_SESSION['prenom'] = "Bob";`
- **Pour supprimer toutes les variables de session:**
  - `session_unset();`
- **Pour détruire la session:**
  - `session_destroy();`

Fonctionnement

- **Les données des variables de session sont stockées dans des fichiers temporaires sur le serveur**
  - Emplacement de sauvegarde déterminé par le paramètre session.save_path dans le fichier php.ini
- **Un identifiant de session (session ID) est conservé côté client dans un cookie et transmis avec chaque requête au serveur**
- **La session est automatiquement détruite après un certain temps (paramétrable sur le serveur)**

Attention!

- **Ne vous mettez pas à tout sauvegarder dans des variables de session!**
- **On ne les utilise que pour ce qui est strictement nécessaire**
  - Les variables qu'on a absolument besoin de conserver entre les exécutions d'une même page ou entre les pages
- **Ce n'est pas non plus un stockage permanent! (contrairement à une base de données)**

Fin de la présentation

Des questions?

Photo par Jon Tyson sur Unsplash