

TP Tableaux PHP (Array), utilisation avec « foreach »

Partie 1 :

Vous disposez du formulaire HTML (fichiers index.html et son css) qui envoie des informations en mode POST à une page PHP (tp_array.php) que vous allez compléter. Les fichiers sont fournis.

Il faudra modifier la page PHP pour qu'elle affiche à peu près ceci :

Résultat du sondage

Clé	Valeur
nom	ANTOINE
qualite	M
sport	natation
sport	cyclisme
sport	ski
animal	Chien
animal	Pingoin

Comment faire ?

Déjà bien comprendre que le mot « tableau » a plusieurs sens :

- Ci-dessus, le tableau est à prendre au sens « décoratif » : c'est un tableau HTML créé avec des balises <table> <tr> <th> <td> (voir tuto [ici](#))
- On appelle également « Tableau » une structure de données organisée en ligne/colonnes. Dans l'exemple de ce TP, les données sont contenues dans une variable de type tableau PHP (array) générée par le formulaire *tp_array.html*. Ce tableau nommé `$_POST[]` n'est pas « affichable » tel quel. Pour utiliser la commande **echo**, il faudra d'abord extraire les données de `$_POST` pour les intégrer au tableau HTML.

Préparez votre environnement de travail

1. Accédez par FileZila à votre espace de travail sur le serveur PHP, et créez un sous-dossier « tpPhp2 »
2. Copiez les fichiers fournis (html, css et php).



Premier essai : Modifiez le fichier tp_array.php fourni et testez le résultat :

3. **Le seul moyen d'avoir une idée du contenu du tableau PHP est d'utiliser la commande « print_r » :**
Affichez la variable tableau PHP telle qu'elle est stockée en mémoire :

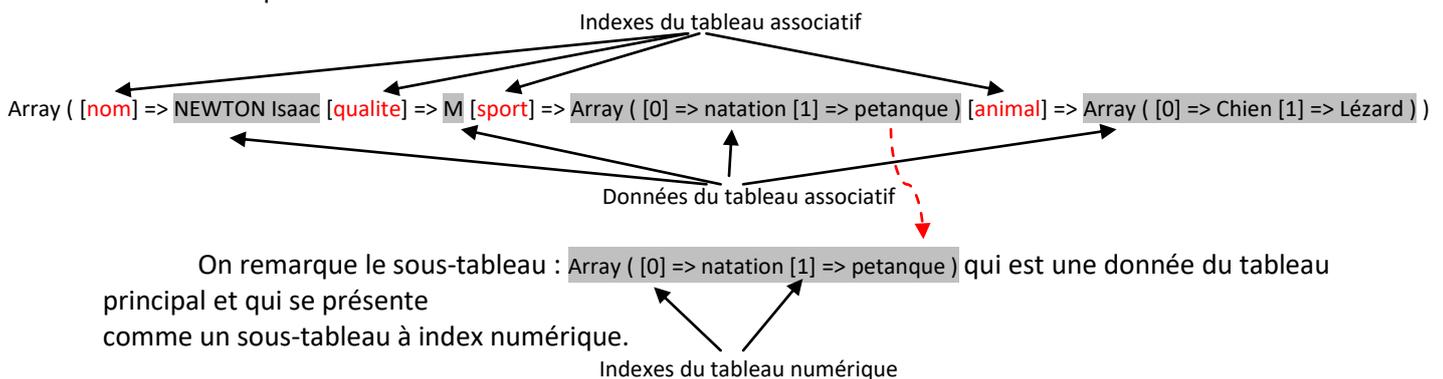
```
<?php print_r( $_POST); ?>
```

Votre analyse : Cet *array* est-il de type « associatif » ou de type « à index numérique » ?

Réponse : associatif : Les colonnes sont nommées, et pas numérotées.

MAIS il contient aussi des données de type « tableau à index numérique »

Exemple :





Deuxième essai : Modifiez le fichier `tp_array.php` fourni.

4. Créez un tableau HTML avec du PHP et incorporez une donnée du tableau PHP :

Exemple pour les 2 premières lignes du tableau :

```
<?php
echo "<table><tr><th>Clé</th><th>Valeur</th></tr>";
echo "<tr><td>NOM</td><td>" . $_POST['nom'] . "</td>";
echo "</table>";
?>
```

nb : Le `php` est en bleu, le `html` en vert. Notez le point « . » de concaténation.
nb : Le CSS va créer les encadrements (voir le fichier `tp_array.css`).

Pour accéder à une donnée du tableau secondaire, la ligne sera :

```
echo "<tr><td>SPORT</td><td>" . $_POST['sport'][0] . "</td>";
```

5. **Le vrai travail** : Comme on ne connaît pas à l'avance la taille des sous tableaux, l'exploitation du tableau `$_POST` doit se faire avec une boucle qui va parcourir le tableau du début à la fin. Le PHP propose une fonction très pratique : **foreach**.

```
foreach ( $_POST as $index => $valeur )
{
    echo "<tr><td> $index </td><td> $valeur </td>";
}
```

`$_POST` : Le tableau

`$index` : Contendra le prochain index (numérique ou associatif)

`$valeur` : Contendra la donnée correspondant à l'index.

A FAIRE : Afficher le tableau PHP complet dans un tableau HTML, en utilisant le « `foreach` ». Le résultat doit ressembler au modèle proposé au début de ce document.

Résultat du sondage

Clé	Valeur
nom	ANTOINE
qualite	M
sport	cyclisme
sport	ski
animal	Chien
animal	Fagot



Comme vous avez remarqué au « Premier essai », la valeur stockée est soit du texte, soit un autre tableau. Si c'est un tableau, la commande « `echo` » ne va pas fonctionner.

A FAIRE : Tester la `$valeur` : Si c'est un tableau, faire un autre « `foreach` », sinon, afficher la valeur directement ? Pour détecter si `$valeur` est tableau, utilisez la fonction `is_array($valeur)` (Voir tuto [is_array](#))

Faites valider votre travail avant de passer à la partie 2.

Partie 2 :

Vous allez utiliser un script PHP (non étudié aujourd'hui) qui récupère une liste de mesures météo dans une base de données. Voici un aperçu des données extraites :

idm	WS_ID	date_aprs	DIR	SPD	G	T	R	P	PM	H	B
1	F3G6T7	110815	280	43.4	55.8	41.6	0	0	0	70	1010
2	F3G6T7	110830	280	43.4	55.8	42.6	3	3	3	80	1010
3	F3G6T7	110845	280	49.6	62	43.6	3	3	3	80	1010

Du point de vue du PHP, le résultat est un *array* contenant 1 *ArrayObject* pour chaque ligne du tableau. C'est donc un tableau doublément mixte :

- index numérique et associatif
- tableau classique et tableaux objets.

TRAVAIL :

6. Copiez le fichier **tp_array_bdd.php** dans le sous-dossier tpPhp2 du serveur PHP. C'est un script PHP qui récupère les données dans la base de données. Il fournit un tableau. **Vérifiez que le nom du serveur SQL, le nom d'utilisateur et le mot de passe sont renseignés. Demandez au professeur en cas de doute**

7. Au même endroit, créez un script PHP (nommé **array1.php**).

Vous devrez tout d'abord ajouter les lignes suivantes au début de votre fichier **array1.php** :

```
include "tp_array_bdd.php";
$bdd = new accesBdd();
$data = $bdd->dataTParray();
```

Le tableau contenant les données de la BDD est maintenant dans la variable **\$data**.

8. Analysez la structure PHP en utilisant la fonction : **print_r(\$data)** ;
NB : Vous verrez le type principal de \$data en utilisant : **echo gettype(\$data)** ;
9. Affichez la température (T) de la première ligne de \$data : **echo \$data[0]['T']** ;

10. Le code que vous devez écrire :

On veut afficher en HTML la synthèse suivante à partir des données fournies par le tableau :

