

TP 2 : chaînes, tableaux et fonctions

Pour l'ensemble des questions (et toutes les autres à venir), vous devrez spécifier les types pour chaque paramètre et le type de retour pour toutes les fonctions.

I. Manipulation de tableaux

Nous souhaitons pouvoir convertir des tableaux PHP à 2 dimensions en tableaux HTML. Par exemple :

```
$t = [[1,2], [3, 4]]
```

Tableau PHP

1	2
3	4

Rendu HTML

1. Écrivez une fonction qui convertit un tableau PHP à deux dimensions en un tableau HTML.
2. Écrivez un script qui teste cette fonction, le tableau étant défini dans une constante.
3. Que se passe-t-il si les lignes n'ont pas toutes la même dimension ? Si votre fonction ne gère pas ce cas, modifiez-la.
4. Cette fois-ci, la fonction prend en paramètre deux tableaux PHP. Le premier correspond aux titres des colonnes. Le second contient les lignes du tableau. Chaque élément d'une ligne est indexé par le titre de la colonne correspondante. Par exemple :

```
$titres = ["nom", "prenom", "age"];  
$t = [ ["nom"=>"Eastwood", "prenom" =>"Clint", "age"=>62],  
      [ "nom"=>"Gims", "prenom"=>"Maître", "age"=>34],  
      [ "nom"=>"O'Neil", "prénom"=>"Ed", "age"=>74] ];
```

Tableau PHP

nom	prenom	age
Eastwood	Clint	62
Gims	Maître	34
O'Neil	Ed	74

Rendu HTML

Écrivez une nouvelle fonction et testez-la.

5. Que se passe-t-il si une ligne ne contient pas tous les titres ? Si ce n'est pas déjà le cas, modifiez votre script pour en tenir compte.

II. De la chaîne au tableau et inversement

1. Écrivez la fonction PHP `tableau2String` qui prend en paramètre un tableau à 1 dimension et un séparateur (une chaîne de caractères). Elle retourne une chaîne de caractères contenant les valeurs du tableau séparées par le séparateur spécifié (le séparateur ne sera pas ajouté à la fin de la chaîne). Par défaut, le séparateur est `;`. Vous testerez votre fonction dans votre script.
2. Comparez avec la fonction de l'API `implode`.
3. Écrivez la fonction PHP `string2Tableau` qui réalise l'opération inverse de la question précédente.
4. Comparez avec la fonction de l'API `explode`.
5. Vérifiez que vos fonctions gèrent le cas de la chaîne vide et/ou du tableau vide.
6. À l'aide de `var_dump`, vérifiez le résultat de `tableau2String([0, 1, 2])` puis de `string2Tableau(tableau2String([0, 1, 2]))`. Qu'observe-t-on ?

III. Plus loin avec les tableaux

Pour cet exercice, l'objectif est de tester les fonctions présentes dans l'API PHP pour manipuler des tableaux.

1. Créez une fonction permettant de générer un tableau contenant des valeurs entières aléatoires. La taille du tableau et les bornes de l'intervalle sont passées en paramètres.
2. Si vous ne l'avez pas déjà fait au TP précédent, écrivez une fonction `estPremier` qui retourne TRUE si l'entier passé en paramètre est premier ou FALSE sinon.
3. À l'aide de `array_filter`, gardez uniquement les nombres premiers du tableau.
4. Nous souhaitons maintenant que les entiers soient triés par ordre croissant.

5. Finalement, seuls les 10 premiers nombres premiers trouvés nous intéressent.
6. Supprimez tous les nombres premiers qui sont en doublon.

IV. Manipulation de tableaux et CSV

Nous supposons que des données sont stockées sous la forme d'un tableau de lignes, chaque ligne étant un tableau associatif.

1. Faites une fonction prenant en paramètre les données et un tableau contenant des clés. Elle retourne TRUE si toutes les lignes du tableau de données contiennent bien toutes les clés.
2. Ecrivez une fonction prenant en paramètre les données et une clé. Les données sont triées en fonction de la clé spécifiée. Nous supposons que toutes les lignes contiennent bien cette clé.

En PHP, il est possible de générer des documents autres que du HTML. Pour cela, il est nécessaire de modifier l'en-tête http à l'aide de la méthode `header` avant l'envoi de toute donnée. Par exemple, pour générer du CSV, nous pouvons utiliser les lignes suivantes :

```
header("Pragma: public");  
header("Expires: 0");  
header("Cache-Control: must-revalidate, post-check=0, pre-check=0");  
header("Cache-Control: private", false);  
header("Content-Type: text/csv; charset=UTF-8");  
header("Content-Disposition: attachment; filename=\"exemple.csv\"");  
flush();
```

Ensuite, il suffit d'écrire à l'aide de `echo` les différentes lignes, chaque valeur étant séparée par une virgule (ou un point-virgule si vous souhaitez une compatibilité avec *Excel*) et un `'\n'` à la fin de la ligne.

3. Ecrivez un script qui permet de sauvegarder un tableau de données dans un fichier CSV.