Formations en Informatique de Lille – 2016/2017

Semaine 3 - Technologies du Web

janvier 2017

PHP

Exercice 1 : Comparaisons, ordres et tris

La fonction `sort()` permet de trier des tableaux PHP indexés par des entiers successifs à partir de 0. (Rappelons qu'il ne faut pas l'utiliser sur des tables associatives).
Cette fonction range les valeurs selon leur ordre implicite (ordre «naturel» sur les nombres, ordre lexicographique pour les chaînes, ...)
Mais on a parfois besoin de trier selon un ordre différent. De plus si le tableau contient des valeurs plus élaborées (objets, tables associatives) la relation d'ordre à utiliser n'est pas prédéfinie.
Pour ce faire, il convient d'abord de définir sur les valeurs une **fonction de comparaison** selon le principe suivant :

- comparaison($1,$2) < 0 si $1 est strictement plus petit que $2.
- comparaison($1,$2) $== 0$ si $1 est équivalent à $2.
- comparaison($1,$2) > 0 si $1 est strictement plus grand que $2.

L'implémentation de la fonction devra donc respecter quelques règles :

- comparaison($1,$1) vaut 0
- comparaison($1,$2) et comparaison($2,$1) sont de signes opposés
- si comparaison($1,$2)<0 et comparaison($2,$3)<0 alors comparaison($1,$3)<0

Voici par exemple une fonction qui compare des nombres selon leur valeur absolue (un nombre est considéré comme «plus petit» qu'un autre si sa valeur absolue est plus petite que celle de l'autre.

```
function compareAbs($i,$j){
    return abs($i)-abs($j);
}
```

La fonction PHP `usort(...)` (consultez la doc) permet de réaliser un tri selon une relation d'ordre définie par l'utilisateur . Le nom de la fonction de comparaison à utiliser est passé en paramètre de `usort` sous forme de chaîne, par exemple :

```
usort($tableau,"compareAbs")
```

Vous créerez sur le serveur un dossier spécifique pour cet exercice. Son contenu aura une structure analogue à celle de la séance précédente (sous-dossier`lib` ...). Les fonctions figureront dans un fichier du répertoire `lib`. Un script de test vous est fourni.
Question 1.1 : (illustration) Écrivez la fonction `compareAbs` (cf ci-dessus) et testez la avec la première partie du script de test. Essayez avec d'autres exemples de tableaux.
Question 1.2 : (illustration) NB : pour cette question, comme pour les suivantes, on considère que les chaînes ne comportent pas de lettres accentuées.
La fonction prédéfinie `strcmp()` est une fonction de comparaison de chaînes de caractères, selon l'ordre lexicographique (c'est à dire l'ordre usuel sur les chaînes).
Triez le tableau de chaînes fourni selon l'ordre défini par `strcmp()` puis affichez sa valeur.
Constatez qu'on obtient le même résultat si on utilise la fonction `sort()`.
Question 1.3 : (application directe)

1. Définissez une fonction `comparerChainesParLongueur()` qui compare 2 chaînes. Une chaîne sera considérée comme plus petite si sa longueur est plus petite.

   Définissez un tableau de chaînes, triez-le selon cet ordre et affichez le tableau trié.

2. La fonction de comparaison précédente renvoie 0 quand on l'applique à deux chaînes de même longueur (ce qui était demandé). Quand on l'utilise pour trier, le rangement obtenu entre chaînes de même longueur est donc indéterminé.
   On souhaite assurer que, après tri, les chaînes de même longueur apparaitront par ordre lexicographique **inverse**.
   Définissez pour cela une fonction `comparerChainesParLongueurPlus()` qui convient. Testez l'effet du tri avec cette nouvelle fonction de comparaison.

Exercice 2 : Vous allez reprendre l'exercice de la feuille précédente, concernant les livres.
Question 2.1 : Il s'agira cette fois de représenter en mémoire la totalité des ouvrages décrits dans le fichier texte, sans les afficher au fur et à mesure de leur lecture.

1. Écrire une fonction `loadBiblio($file)` dont l'argument est un fichier ouvert et le résultat un tableau PHP.
   Le fichier est supposé contenir des descriptions de livres.
   Le résultat est un tableau indexé par des entiers successifs. Chaque élément du tableau est lui-même un tableau représentant un livre (tel que renvoyé par `readBook()`) .
2. Écrivez une fonction `biblioToHTML($liste)` dont l'argument est un tableau de livres (similaire à celui renvoyé par la fonction précédente)
   Le résultat de`biblioToHTML($liste)` est une chaîne contenant la représentation HTML des livres de la liste, dans l'ordre de la liste fournie.
   Attention : comme précédemment, cette fonction ne doit produire aucune entrée/sortie (aucun "echo").
3. Écrivez une nouvelle version du script `bibliotheque.php` qui utilisera ces nouvelles fonctions. Vous l'appellerez `bibliotheque2.php`

Question 2.2 :

1. Définissez une fonction de comparaison de 2 livres. Cette fonction de comparaison, employée comme critère de tri, doit faire apparaître les livres selon l'ordre lexicographique des titres.
2. Ajoutez à la fonction `biblioToHTML` un argument **optionnel** `$sort`. Si `$sort` vaut `"titles"` alors les livres sont présentés par ordre croissant des titres. Si `$sort` vaut `"none"` (valeur par défaut) les livres sont présentés dans l'ordre de la liste reçue en argument. Dans tous les autres cas une exception est déclenchée.
3. Modifiez le script `bibliotheque2.php` pour qu'il affiche les livres par ordre alphabétique des titres.

Question 2.3 : *Paramétrage du script*
Vous allez maintenant concevoir un script `bibliotheque3.php`, amélioration du précédent.
S'il est appelé avec un argument (passé en mode GET) `ordre` valant `titres` , alors les livres sont affichés par ordre croissant des titres. S'il est appelé sans l'argument `ordre` ou avec `ordre` valant `aucun` , alors les livres sont affichés selon l'ordre du fichier de données. Dans tous les autres cas les livres ne sont pas affichés.
Vous testerez le script dans les différentes situations possibles, en essayant différentes URL, par exemple

```
..../bibliotheque3.php?ordre=titres
..../bibliotheque3.php?ordre=aucun
..../bibliotheque3.php
```

Question 2.4 : On souhaite maintenant pouvoir afficher les livres par ordre de catégorie et, au sein d'un même catégorie, par années de parution croissantes, et au sein d'une même année par ordre alphabétique des titres.
Cet ordre sera obtenu en appelant le script :

```
..../bibliotheque3.php?ordre=categories
```

(note : cette option de tri s'ajoute à la précédente mais ne la remplace pas)

1. Complétez la bibliothèque de fonctions pour implémenter cette nouvelle relation d'ordre.
2. Adaptez `bibliotheque3.php` pour prendre en compte la nouvelle option.