Université de Lille 1 SCIENCES ET TECHNOLOGIES

Semaine 3 - Technologies du Web

FORMATIONS EN INFORMATIQUE DE LILLE 1

février 2016

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);
}
```

Une fonction de comparaison peut ensuite être passée en paramètre de la fonction prédéfinie

```
usort($tableau,fonctionDeComparaison)
```

qui va trier le tableau selon l'ordre indiqué (NB : dans le mot `usort`, u signifie "user"). Par exemple

```
usort($tab,compareAbs);
```

Notez bien que c'est l'entité fonction qui est passée comme paramètre de `usort()`, donc le nom de la fonction, sans les parenthèses.

Question 1.1 : (illustration)

Définissez un tableau contenant des entiers (en choisir des positifs et des négatifs), triez-le selon les valeurs absolues et affichez le tableau. S'agissant d'un simple test et non d'une page web de production, vous pouvez utiliser la fonction `print_r(...)` (à placer de préférence dans un élément `<pre>`) pour visualiser le tableau.

Question 1.2 : (illustration)

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).

Définissez un tableau de chaînes, puis triez-le en utilisant `usort()` avec, comme argument, la fonction `strcmp`.

Affichez le résultat et constatez que vous obtenez le même résultat qu'avec la fonction `sort()`.

Question 1.3 : (application directe)

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.
Question 1.4 : (application directe)
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.
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.
Il sera ainsi possible de modifier ou classer ces données avant d'afficher la bibliothèque
Question 2.1 : Écrire une fonction `loadBiblio($file)` dont l'argument est un fichier ouvert et le résultat un tableau PHP.
Le fichier est supposé contenir des descriptions de livres.
Chaque valeur du tableau fabriqué représente un livre : chaque valeur du tableau est donc une table associative comme on les a définies à la question 1.8 de la feuille précédente.
Le tableau est indexé par des entiers successifs.
Question 2.2 : Écrivez une fonction `biblioToHTML($liste)` dont l'argument est un tableau de livres.
Le résultat de la fonction est une chaîne contenant la représentation HTML des livres de la liste, dans l'ordre de la liste fournie.
Attention : comme pour la fonction `livreToHTML()`, cette fonction ne doit produire aucune entrée/sortie (aucun "echo").
Écrivez une nouvelle version du script `bibliotheque2.php` (feuille précédente) en utilisant ces 2 nouvelles fonctions.
Remarque : votre script `bibliotheque2.php` ne doit plus comporter que très peu de lignes de code PHP (4 ou 5 environ)
Question 2.3 : 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 alphabétique de leur titre.
Question 2.4 : Écrivez le script `bibliotheque3.php` qui affiche les livres par ordre alphabétique des titres.
Question 2.5 : Définissez une nouvelle fonction de comparaison de 2 livres : on souhaite maintenant afficher les livres par ordre de catégorie, puis au sein d'un même catégorie par années de parution croissantes, et au sein d'une même année par ordre alphabétique des titres.