

Exercice 1 : Reprise de l'exercice 3 (les livres) de la première feuille de TD PHP

Question 1.1 : Implémenter les questions 3.2 puis 3.3 de l'exercice de TD

Question 1.2 : Nous allons maintenant faire évoluer notre structure de données vers un modèle objet.

- Définir dans un fichier nommé `Livre.class.php` une classe `Livre` permettant de représenter un titre (chaîne), plusieurs auteurs (tableau de chaînes) et un éditeur (chaîne); Prévoir un constructeur à trois arguments : le premier est le titre, le deuxième est soit un simple auteur, soit un tableau d'auteurs. Le troisième argument est un éditeur et doit posséder pour valeur par défaut la chaîne vide. Dans le constructeur, vous testerez le type du deuxième argument par les méthodes `is_array` et `is_string`
- Ajouter à cette classe une méthode `imageTR()` dont le résultat est une chaîne représentant **une ligne** de tableau HTML à 2 cellules, sur le modèle de l'exercice vu en TD. Les noms des auteurs apparaissent dans une même cellule, séparés par des virgules. L'éditeur n'apparaît pas. Vous êtes invités à consulter la documentation de la fonction prédéfinie `implode` (rubrique chaînes de caractères) qui permet de constituer une chaîne à partir d'éléments de tableau (et sa réciproque `explode` qui vous sera utile par la suite)

Question 1.3 :

- Définir dans un fichier `bibLivre.php` une fonction `imageLivres(array $tab_livres)` dont le résultat est un table HTML composée de livres (utiliser la méthode `imageTR()`). L'argument est supposé être un tableau d'instances de la classe `Livre`.
- Créer un fichier `pagetest.php` dans lequel vous définirez un tableau de livres (par exemple ceux cités dans l'exercice de TD) puis génère une page HTML affichant le contenu de ce tableau.
- Tester dans un navigateur le fichier `pagetest.php` et mettez au point.

Lecture et écriture de fichiers en PHP**Ouverture d'un fichier**

Avant de réaliser des opérations de lecture ou d'écriture sur un fichier, il est nécessaire de l'ouvrir. On dispose pour cela de la fonction `fopen` dont la syntaxe (simplifiée) est :

```
resource fopen(string $nom, string mode)
```

Le premier paramètre est le nom du fichier que l'on souhaite utiliser (éventuellement avec son chemin). Le second paramètre détermine le mode d'accès au fichier :

- mode `"r"` : le fichier est ouvert en lecture seule. La lecture commence au début du fichier.
- mode `"r+"` : le fichier est ouvert en lecture et en écriture. Ces opérations commencent au début du fichier.
- mode `"w"` : le fichier est ouvert en écriture seule. L'écriture commence au début du fichier.
- mode `"w+"` : le fichier est ouvert en lecture et en écriture. Ces opérations commencent au début du fichier.

- Pour les modes "w" et "w+", si le fichier n'existe pas, il est créé. Si le fichier existe, son contenu est effacé.
- mode "a" : le fichier est ouvert en écriture seule, et les données sont écrites en fin de fichier, à la suite de celles qui existent déjà. Si le fichier n'existe pas, il est créé.
- mode "a+" : le fichier est ouvert en lecture et écriture. Les données sont écrites en fin de fichier, à la suite de celles qui existent déjà. Les lectures se font à partir du début du fichier. Si le fichier n'existe pas, il est créé.

Le résultat de la fonction `fopen` est un identifiant de fichier, qui doit être utilisé comme premier paramètre de la plupart des fonctions de manipulation des fichiers. Il faut donc impérativement récupérer cette valeur dans une variable.

Lecture à partir d'un fichier, ligne par ligne

```
string fgets(resource $id_file, int n)
```

Cette fonction retourne la chaîne lue jusqu'à la longueur $(n - 1)$ octets, ou bien la fin du fichier, ou encore un retour chariot (le premier des trois qui sera rencontré). Le paramètre n est devenu optionnel depuis PHP 4.2.0. Si il est omis, il prend la valeur par défaut de 1024. Si une erreur survient, `fgets` retourne FALSE.

Ecriture dans un fichier

```
integer fputs(resource $id_file, string chaine)
```

Ecrit la chaîne passée en second paramètre dans le fichier d'identifiant `$id_file`. L'entier renvoyé par cette fonction est le nombre d'octets écrits, ou FALSE en cas d'erreur.

Fermeture d'un fichier

Quand on a fini d'utiliser un fichier, il faut le fermer à l'aide de la fonction `fclose` :

```
boolean fclose($id_file)
```

Le paramètre est un identifiant de fichier et le résultat est TRUE si l'opération s'est bien déroulée, FALSE sinon.

Remarque : Pour les opérations de lecture et d'écriture, il faut que le serveur ait les droits nécessaires à la réalisation de ces opérations au niveau du système de fichiers.

Exercice 2 : On considère une liste de livres stockée dans un fichier texte dont le format est le suivant : TITRE\n AUTEUR(S)\n [EDITEUR\n] \n. Les auteurs sont supposés être séparés par des virgules. Un exemple de fichier respectant ce format est donné ci-dessous :

```
Perl précis et concis
J. Vromans
```

```
ADN, mots et modèles
Stephane Robin, Francois Rodolphe , Sophie Schbath
Editions Belin
```

```
Introduction a l'analyse genetique
Griffith, Miller, Suzuki, Lewontin, Gelbart
De Boeck-Universite
```

```
XML langage et applications
A. Michard
```

Question 2.1 : Écrire une fonction `lire_livres($nom_fichier)` qui retourne un tableau d'objets de type `Livre` représentant les livres du fichier.

Vous trouverez l'exemple dans `Lecture/TechnoWeb/TP7/livres.txt` Pour découper une chaîne en tableau de chaînes, on peut utiliser la fonction `explode`. La fonction `trim` peut également vous être utile (voir documentation)