

"Le système d'information (SI) est un ensemble organisé de ressources qui permet de collecter, stocker, traiter et distribuer de l'information" (source : wikipedia).

Cette activité de programmation va consister à programmer des fonctions de manipulation des données d'une bibliothèque. Le sujet propose une modélisation de l'information sous la forme de listes de listes qu'il faudra manipuler.

1 Objectifs pédagogiques

L'objectif de ce TP est triple :

- Se remémorer tous vos acquis en python via la manipulation de fonctions, de listes de listes, de fichiers texte.
- Manipuler et modéliser des entités comme des *auteurs*, des *livres*, des *emprunts* afin de vous préparer à la structuration d'un programme en objets.
- Développer le début d'un système d'information de médiathèque afin de vous préparer à la modélisation d'un tel système sous la forme d'une base de données

2 Le système d'information

Nous allons manipuler les informations nécessaires à une médiathèque :

Un **usager** est décrit par une liste Python contenant successivement :

- Un identifiant (numérique)
- Un nom (chaîne de caractères)
- Un prénom (chaîne de caractères)
- Une date de naissance (une chaîne de caractère de la forme "JJ/MM/AAAA")
- Les emprunts (une liste d'identifiants d'emprunts, un identifiant étant un entier)

Un exemple d'usager :

```
[1, "Nonyme", "Alphonse", "01/01/2003", [1, 2, 4]]
```

Les usagers sont stockés dans une liste.

Un **livre** de la bibliothèque est décrit par une liste Python contenant successivement :

- Un identifiant (numérique)

- Un titre (chaîne de caractères)
- Un auteur (chaîne de caractères)
- Des mots-clés (une liste de chaînes de caractères)

Un exemple de livre :

```
[1, "Nana", "Zola Emile", ["Drame", "Classique", "Troisième Empire"]]
```

Les livres de la bibliothèque sont stockés sous la forme d'une liste.

Un **emprunt** est décrit par une liste Python contenant successivement :

- Un identifiant (numérique)
- L'identifiant du livre emprunté (numérique); cet identifiant doit se retrouver dans la liste des livres
- La date de début d'emprunt (une chaîne de caractère de la forme "JJ/MM/AAAA")
- La date de retour attendue du livre (une chaîne de caractères de la forme "JJ/MM/AAAA")
- La date effective de retour du livre (une chaîne de caractères de la forme "JJ/MM/AAAA" si le livre a été rendu, None sinon)

Un exemple d'emprunt :

```
[4, 3, "01/01/2015", "01/02/2015", "15/01/2015"]
```

Les emprunts de la bibliothèque sont stockés sous la forme d'une liste. L'historique des emprunts est conservé sans nettoyage (certainement pour pouvoir faire des statistiques) : un livre rendu voit la date de retour effective mise à jour, mais l'emprunt ne disparaît pas de la liste. Le lien entre un emprunt et l'emprunteur se fait par la liste des usagers (rappel : chaque usager comprend une information sur la liste de ses emprunts)

Les divers identifiants (d'usager, de livre, d'emprunt) permettent d'identifier de manière unique ce à quoi ils se réfèrent, et permettent de faire le lien entre les trois listes.

3 Ce qui vous est donné

Vous avez à disposition deux fichiers Python :

- `utils.py` : définit deux fonctions sur les dates, qui vous seront utiles. Remarquez l'utilisation de la librairie puissante `datetime`
- `mediatheque_fichier0.py` : ce fichier définit une fonction qui renvoie le nombre d'emprunts d'un usager. C'est à partir de ce fichier que vous allez travailler. Le fichier définit aussi un petit système d'information constitué de 3 listes, une respectivement pour les auteurs, les livres, et les emprunts.

4 Travail à faire

4.1 Partie 1 : manipulation du système d'information

Ouvrez le fichier `mediatheque_fichier0.py`, exécutez-le pour constater l'affichage suite à l'appel de la fonction `nb_livres_empruntes`. Étudiez cette fonction pour en comprendre le fonctionnement. Cela vous permettra de réviser la définition de fonction, le passage d'arguments, le parcours de liste, les conditionnelles.

Examinez comment on accède aux éléments d'information sur un usager via l'opérateur `[]`.

Cette manière de gérer l'accès aux données peut poser plusieurs problèmes :

- **Manque de lisibilité** : Il faut faire un effort de mémoire pour se souvenir que, par exemple, pour un usager `u`, `u[1]` permet d'accéder au nom de l'usager.
- **Evolutivité compliquée** : on peut vouloir modifier le système d'information. Par exemple, on peut vouloir inverser des informations (pour un usager mettre le prénom avant le nom), ou l'enrichir (ajouter l'adresse et le numéro de téléphone pour un usager juste après les noms et prénoms). De tels changements imposent de revoir tout le code (pour le moment, vous n'avez qu'une seule fonction, mais cela va augmenter), et de changer les indices un peu partout, et on va sans doute en oublier, ce qui va provoquer des bugs.

Pour contourner ces problèmes, nous allons créer pour chaque information deux fonctions, une pour la lire (fonction `get`), une autre pour la modifier (fonction `set`). Par exemple :

```
def usager_get_id(u):  
    return u[0]  
  
def usager_set_id(u, nvlle_val):  
    u[0] = nvlle_val
```

En procédant ainsi pour chaque information, il devient alors facile de modifier le système d'information en modifiant seulement les fonctions `get` et `set`. Par ailleurs, le nom des fonctions de type `get` et `set` est facile à mémoriser.

Pour le moment, ne créez pas toutes les fonctions de type `get` et `set`, mais juste celles dont vous avez besoin pour modifier la fonction `nb_livres_empruntes`. Reportez ces fonctions dans un fichier nommé `systeme_information_mediathèque.py` et importez ce fichier dans `mediatheque_fichier0.py` (que vous pourrez renommer en `mediatheque.py`). Vous enrichirez au fur et à mesure le fichier `systeme_information_mediathèque.py`.

4.2 Partie 2 : révisions sur les listes, listes de listes, etc.

Rédigez et *testez* les fonctions décrites ci-dessous.

4.2.1 Lecture de fichiers texte

Une première chose à faire est de pouvoir lire les données du système d'information à partir du disque. Voici comment est stocké sur disque le système d'information. Le stockage utilise 4 fichiers de types csv (le séparateur est le point-virgule) :

- `usagers.csv` : contient une ligne par usager. Chaque ligne comprend 4 colonnes : identifiant, nom, prénom, date de naissance
- `livres.csv` : contient une ligne par livre. Chaque ligne comprend 6 colonnes : identifiant, titre, nom de l'auteur, mot-clé 1, mot-clé 2, mot-clé 3. Si le livre est associé à moins de 3 mots-clés, le champ correspondant est remplacé par Néant.
- `emprunts.csv` : contient une ligne par emprunt. Chaque ligne comprend 5 colonnes : identifiant de l'emprunt, identifiant du livre emprunté, date d'emprunt, date attendue de retour, date de retour effective. Si le livre n'a pas été rendu, la dernière colonne contient Néant.
- `emprunts_usagers.csv` : c'est ce fichier qui fait le lien entre les usagers et les emprunts. Le fichier contient une ligne par association usager-emprunt. Chaque ligne comprend 2 colonnes : identifiant de l'utilisateur, identifiant de l'emprunt.

Créez les fonctions :

- `charger_usagers(nom_fichier)` qui crée et renvoie une liste d'utilisateurs à partir de la lecture du fichier nommé `nom_fichier`. Le fichier est de type csv (cf descriptif du fichier `usagers.csv` ci-dessus). Une fois chargé, chaque utilisateur a une liste d'emprunts vide (cette liste sera mise à jour quand on lira le fichier `emprunts_usagers.csv`).
- `charger_livres(nom_fichier)` qui crée et renvoie une liste de livres à partir de la lecture du fichier nommé `nom_fichier`. Le fichier est de type csv (cf descriptif du fichier `livres.csv` ci-dessus).
- `charger_emprunts(nom_fichier)` qui crée et renvoie une liste d'emprunts à partir de la lecture du fichier nommé `nom_fichier`. Le fichier est de type csv (cf descriptif du fichier `emprunts.csv` ci-dessus).
- `charger_emprunts_usagers(usagers, nom_fichier)` qui met à jour la liste des emprunts des utilisateurs listés dans `usagers` à partir des informations lues dans le fichier `nom_fichier`. Le fichier est de type csv (cf descriptif du fichier `emprunts_usagers.csv` ci-dessus).
- `charger_mediathèque(nom_fichier1, nom_fichier2, nom_fichier3, nom_fichier4)` qui charge l'ensemble des informations (appel aux trois fonctions ci-dessus) à partir des quatre fichiers qui représentent respectivement `usagers.csv`, `livres.csv`, `emprunts.csv` et `emprunts_usagers.csv` et renvoie les trois listes créées.

4.2.2 Manipulation d'une seule liste

- la fonction `changer_nom_utilisateur(liste_utilisateurs, id_utilisateur, nv_nom)` qui modifie le nom de l'utilisateur d'identifiant `id_utilisateur`. La liste des utilisateurs est donnée dans `liste_utilisateurs`. Le nouveau nom est donné par `nv_nom`.

- la fonction `ajouter_mot_cle(liste_livres, id_livre, nv_mot_cle)` qui ajoute le mot clé `nv_mot_cle` au livre d'identifiant `id_livre`. La liste des livres est donnée par `liste_livres`.
- la fonction `lister_usagers_majeurs(usagers)` qui renvoie la liste des identifiants des usagers qui sont majeurs (en fonction de la date au moment où la fonction est exécutée). `usagers` donne la liste des usagers. Pour déterminer si un usager est majeur, vous disposez d'une fonction `majeur` dans le fichier `utils.py`.
- la fonction `lister_pret_en_retard()`.
- la fonction `lister_livres_sur_mot_cle(livres, mot_cle)` qui renvoie la liste des identifiants de tous les livres associés au mot clé `mot_cle`. La liste des livres est donnée par `livres`.

4.2.3 Manipulation de deux listes

Les fonctions ci-dessous vont requérir de croiser l'information de deux listes.

- titres des livres actuellement empruntés (2 listes)
- la fonction `livre_plus_recemment_rendu(emprunts, liste_usagers, id_usager)` qui renvoie l'identifiant du livre le plus récemment rendu par l'utilisateur d'identifiant `id_usager`. Si l'utilisateur n'a jamais rendu de livre, la fonction renvoie `None`. La liste des usagers est donnée par `liste_usagers`. La liste des emprunts est donnée par `emprunts`.
- liste des personnes ayant emprunté le livre d'identifiant X (2 listes).

4.2.4 Manipulation de trois listes

Les fonctions ci-dessous vont requérir de croiser l'information de trois listes.

- liste des personnes ayant emprunté le livre de titre X (3 listes).
- la fonction `liste_usagers_ayant_emprunte_livre_titre_X(usagers, livres, emprunts, titre)` qui renvoie la liste des usagers ayant emprunté le livre de titre `titre`. Les listes des usagers, des livres et des emprunts sont donnés respectivement par `usagers`, `livres` et `emprunts`.
- liste des mots clés (sans doublon) emprunté par X → enjeux sociétaux.
- la fonction `XX` qui affiche la fiche d'identité d'un usager sous la forme suivante (le chiffre entre parenthèses indique l'identifiant du livre) :

```

Nonyme Albert
Naissance : XX/XX/XXXX
Livres actuellement empruntés :
    Les Misérables (4), retour attendu le XX/XX/XXXX
    Tintin et les Picaros (18), retour attendu le XX/XX/XXXX
    Poil de Carotte (54), retour attendu le XX/XX/XXXX

```

4.2.5 Manipulation de trois listes

- (bonus) le nombre max de livres empruntés en même temps par quelqu'un.