

"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 *structures* Python (listes, dictionnaires) 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 dictionnaires, 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 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 la gestion d'une bibliothèque.

Un **usager** est décrit par :

- Un identifiant (chaîne alphanumérique)
- 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, voir description des emprunts ci-dessous)

Exemple :

```
id:                "user1",
nom:               "Nonyme",
prénom:            "Alphonse",
date de naissance: "01/01/2003",
emprunts : "emprunt_1","emprunt_2","emprunt_4"
```

Un **livre** de la bibliothèque est décrit par :

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

Exemple :

```
id:      "1",
titre:   "Nana",
auteur:  "Zola Emile",
mots-clés: "Drame", "Classique", "Troisième Empire"
```

Un **emprunt** est décrit par :

- Un identifiant (chaîne alphanumérique)
- L'identifiant du livre emprunté (chaîne alphanumérique); cet identifiant doit se retrouver dans la liste des livres
- La date de début d'emprunt (une chaîne de caractères 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)

Exemple :

```
id:      "emprunt_4",
livre:   "3",
début:   "01/01/2015",
attendu: "01/02/2015",
retour:  "15/01/2015"
```

L'historique des emprunts est conservé sans nettoyage (pour pouvoir faire des statistiques) : en d'autres termes, 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'utilisateur, 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 contient le programme principal de votre application. C'est à partir de ce fichier que vous allez travailler.

## 4 Travail à faire

### Exercice 1. Conception du système d'information

Quelles structures de données proposez vous pour manipuler les différentes informations contenues dans le système d'information de la médiathèque (usagers, livres, emprunts) ? Pourquoi ?

### Exercice 2. Chargement des informations de la médiathèque

On suppose que les données du système d'information sont disponibles via les fichiers CSV suivants (le séparateur étant le caractère ";" :

- `usagers.csv` : contient une ligne par usager. Chaque ligne comprend 5 colonnes : identifiant, nom, prénom, date de naissance, historique des emprunts
- `livres.csv` : contient une ligne par livre. Chaque ligne comprend 4 colonnes : identifiant, titre, nom de l'auteur, liste de mots-clés.
- `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 encore été rendu, la dernière colonne contient Néant.

Dans ce qui suit, nous supposons que les informations du système sont représentées par des *structures de données* Python de type *dictionnaire de dictionnaires*.

Ainsi, les usagers sont représentés par un dictionnaire associant à un identifiant d'utilisateur un dictionnaire contenant le nom, prénom, date de naissance et historique des emprunts de l'utilisateur.

Les livres sont représentés par un dictionnaire associant à un identifiant de livre un dictionnaire contenant le titre, l'auteur, et les mots clés du livre.

Les emprunts sont représentés par un dictionnaire associant à un identifiant d'emprunt un dictionnaire contenant l'identifiant du livre emprunté, la date d'emprunt, la date attendue de retour, la date de retour effective de l'emprunt.

1. Définissez et affectez trois variables `dict_usagers`, `dict_livres` et `dict_emprunts` contenant les exemples de données mentionnées plus haut dans cet énoncé.
2. Écrivez trois fonctions appelées respectivement `charger_usagers`, `charger_livres` et `charger_emprunts` prenant en paramètre un chemin vers un fichier CSV et retournant un dictionnaire.

### Exercice 3. Fonctionnalités de base

Pour chacune des questions ci-dessous, prenez soin de bien commenter votre fonction (notamment en termes d'entrées / sortie), et de **l'accompagner de tests**.

1. Écrivez une fonction

`changer_nom_usager(dict_usagers, id_usager, nv_nom)`

qui modifie le nom de l'utilisateur d'identifiant `id_usager`. La liste des utilisateurs est donnée dans `dict_usagers`. Le nouveau nom est donné par `nv_nom`.

2. Écrivez une fonction

`ajouter_mot_cle(dict_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 `dict_livres`.

3. Écrivez une fonction

`lister_usagers_majeurs(dict_usagers)`

qui renvoie la liste des identifiants des utilisateurs qui sont majeurs (en fonction de la date au moment où la fonction est exécutée). `dict_usagers` donne le dictionnaire des utilisateurs. Pour déterminer si un utilisateur est majeur, vous disposez d'une fonction `majeur` dans le fichier `utils.py`.

4. Écrivez une fonction

`lister_prets_en_retard(dict_emprunts)`

qui renvoie la liste des identifiants des emprunts en retard. La liste des emprunts est donnée par `dict_emprunts`.

5. Écrivez une fonction

`lister_livres_sur_mot_cle(dict_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 `dict_livres`.

#### Exercice 4. Fonctionnalités avancées

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

1. Écrivez une fonction

`livres_empruntes(dict_emprunts, dict_livres)`

qui retourne la liste des titres des livres actuellement empruntés (croisement de 2 listes). Le dictionnaire des emprunts est donné par `dict_emprunts` et celui des livres est donné par `dict_livres`.

2. Écrivez une fonction

`livre_plus_recentement_rendu(dict_emprunts, dict_usagers, id_usager)`

qui renvoie l'identifiant du livre rendu le plus récemment par l'utilisateur `id_usager`. Si l'utilisateur n'a jamais rendu de livre, la fonction renvoie `None`. La liste des utilisateurs est donnée par `dict_usagers`. La liste des emprunts est donnée par `dict_emprunts`.

## 3. Écrivez une fonction

```
usagers_emprunt_id_livre(dict_usagers,dict_emprunts,id_livre)
```

qui retourne la liste des identifiants des usagers ayant emprunté un livre donné (croisement de 2 listes). Le livre ciblé est donné par son identifiant `id_livre`. Le dictionnaire des emprunts est donné par `dict_emprunts` et celui des usagers est donné par `dict_usagers`.

## 4. Écrivez une fonction

```
usagers_emprunt_titre_livre(dict_usagers,dict_emprunts,dict_livre,titre)
```

qui retourne la liste des identifiants d'usagers ayant emprunté le livre de titre `titre` (croisement de 3 listes). Le dictionnaire des emprunts est donné par `dict_emprunts`, celui des usagers est donné par `dict_usagers` et celui des livres par `dict_livres`.

## 5. Écrivez une fonction

```
liste_mots_cles(dict_usagers,dict_emprunts,dict_livres,id_usager)
```

qui retourne la liste des mots clés (sans doublon) des livres empruntés par un usager d'identifiant donné. Notez que le croisement de telles informations permettrait de guider l'usager pour de futures lectures, mais que cela peut être ressenti comme intrusif.

## 6. Écrivez une fonction

```
afficher(dict_usagers,dict_emprunts,dict_livres,id_usager)
```

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 : JJ/MM/AAAA
Livres actuellement empruntés :
    Les Misérables (4), retour attendu le JJ/MM/AAAA
    Tintin et les Picaros (18), retour attendu le JJ/MM/AAAA
    Poil de Carotte (54), retour attendu le JJ/MM/AAAA
```

**Exercice 5. Bonus**

## 1. Écrire une fonction

```
max_emprunts_actuellement(dict_usagers,dict_emprunts)
```

qui retourne les informations de l'usager qui a le plus d'emprunts actuellement.

## 2. Écrivez une fonction

```
max_emprunts(dict_emprunts)
```

qui retourne les informations de l'usager ayant eu le plus de livres empruntés en même temps.

## 5 Analyse critique du système d'information

Dans notre système d'information de la médiathèque on accède aux éléments d'information sur un usager / livre / emprunt via l'opérateur [ ].

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

- **Manque de lisibilité** : pour accéder au nom d'un usager donné d'identifiant *u*, il faut écrire

`dict_usagers[u]['nom']`

- **Évolutivité compliquée** : on peut vouloir modifier le système d'information (par exemple ajouter l'adresse et le numéro de téléphone d'un usager ). De tels changements imposent de revoir plusieurs fonctions du système d'information, potentiellement disséminées dans plusieurs modules.

Dans le prochain TP, nous allons introduire le concept d'objets et nous verrons comment *encapsuler* les informations du système d'information et les opérations autorisées sur ces informations.