R208
Analyse et traitement de données structurées
Travaux dirigés

Sami Evangelista

IUT de Villetaneuse

Département Réseaux et Télécommunications

2023–2024

http://www.lipn.univ-paris13.fr/~evangelista/cours/R208

Table des matières

Introduction aux travaux pratiques

Le but des travaux pratiques de ce module est de développer un paquet python permettant de gérer une bibliothèque de livres. Le paquet sera livré avec un programme principal permettant de manipuler une bibliothèque.

La structure du répertoire de notre projet est visible sur la Figure 1. Cela signifie qu'une fois tous les exercices terminés, vous devrez avoir tous les fichiers présents sur la figure à l'emplacement indiqué. Vous pourrez aussi créer d'autres fichiers si besoin. Par la suite lorsqu'il vous sera demandé de créer un fichier ou répertoire, vérifiez bien sur la figure que vous le créer au bon emplacement. C'est important pour le fonctionnement du paquet.

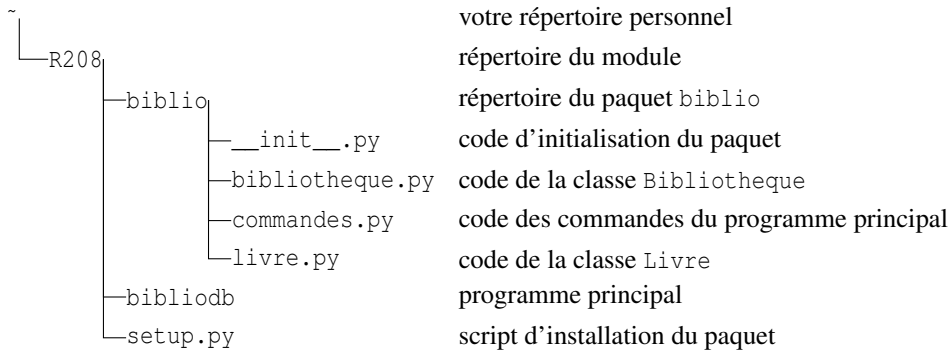

FIGURE 1 – Structure et contenu du répertoire de notre projet

Règles de codage

Les règles de codage suivantes devront être respectées :
- une docstring par module
- une docstring par fonction ou méthode
- lignes de 80 caractères maximum
- fonctions et méthodes de 30 lignes maximum

Votre code sera analysé avec l'outil pylint. Vous pouvez l'installer et le tester de la manière suivante :

```
$ pip3 install pylint  # on installe pylint (à faire une seule fois)
$ cd ~/R208            # on se déplace dans le répertoire du projet
$ pylint biblio        # on analyse le code du paquet biblio avec pylint
```

L'outil pylint peut faire de nombreuses analyses sur le code mais nous ne l'utiliserons que pour vérifier les règles énoncées plus haut. On peut spécifier les analyses qui seront effectuées en remplissant un fichier `pylintrc` avec les règles à vérifier. Pour vérifier nos règles de codage nous utiliserons le fichier à l'URL suivante :

`https://lipn.univ-paris13.fr/~evangelista/cours/R208/pylintrc`

Copiez ce fichier dans le répertoire `~/R208` pour qu'il soit pris en compte par la commande `pylint biblio`.

Consignes d'envoi du code

En fin de module, vous enverrez le contenu du répertoire R208. Le répertoire sera archivé à l'aide de la commande ci-dessous, exécutée depuis votre répertoire personnel :

```
$ tar cvzf R208-PRENOM_NOM.tgz R208
```

(En remplaçant bien entendu `PRENOM_NOM` par vos prénom et nom.) Exécutez ensuite la commande suivante pour vérifier le contenu de l'archive :

```
$ tar tzf R208-PRENOM_NOM.tgz
```

- La date limite est fixée au *Mercredi 15 mai*.
- Vous enverrez uniquement le fichier `R208-PRENOM_NOM.tgz` créée avec la commande `tar` donnée plus haut.
- Le fichier doit être envoyé à sami.evangelista@lipn.univ-paris13.fr.
- Il sera tenu compte du respect des règles de programmation énoncées plus haut dans la notation : jusqu'à 4 points pourront être retirés en cas de non respect des règles.
- La réception d'un fichier après la date limite entraînera une pénalité de 2 points par jour.

Exercice 1 — Création du paquet

Dans cet exercice nous allons créer la structure de base de notre paquet ainsi que notre programme principal.

I 1.1 Créez les répertoires `R208` et `biblio` et le fichier `__init__.py`. Reportez vous à la figure en introduction pour les créer au bon emplacement.

I 1.2 Éditez ce fichier pour qu'il y soit défini à l'aide d'une variable le numéro de version de notre paquet (0.0.0 pour l'instant) et qu'un message de bienvenue contenant le numéro de version soit affiché.

I 1.3 Pour tester, dans le répertoire `R208`, importez le paquet `biblio` depuis l'interpréteur python. On devrait avoir :

```
>>> import biblio
Bienvenue, vous utilisez le paquet biblio v0.0.0!
>>> print(biblio.version)
0.0.0
```

I 1.4 Créez le fichier `bibliodb` et attribuez lui le droit en exécution. C'est un script python qui contiendra le programme principal de notre paquet.

I 1.5 Éditez le en y ajoutant une unique instruction **import** `biblio`. Pour l'instant le programme ne fera rien de plus. Nous le complèterons dans les exercices suivants.

I 1.6 Lancez votre script dans le terminal et vérifiez qu'il affiche bien le message de bienvenue.

Une fois le TP terminé vous passerez le numéro de version à 1.0.0.

Exercice 2 — Script d'installation du paquet

Un paquet python est généralement livré avec un fichier `setup.py` qui est un script python permetant d'installer ou de supprimer le paquet sur le système ou dans votre répertoire personnel. Nous allons créer ce script dans cet exercice.

I 2.1 Créez le fichier `setup.py`.

I 2.2 Éditez le pour qu'il ait contenu ci-dessous :

```
1  #!/usr/bin/env python3
2
3  """Script d'installation du paquet biblio."""
4
5  from setuptools import setup
6  ____________________________
7
8  setup(
9      name="biblio",
10     version=________,
11     description="un paquet pour gérer des bibliothèques",
12     packages=["biblio"],  # répertoire dans lequel se trouve le paquet
13     scripts=["bibliodb"]  # script contenu dans le paquet
14 )
```

Vous complèterez les lignes 6 et 10 du fichier pour que le numéro de version défini dans le paquet `biblio` soit passé en argument de la fonction `setup`.

I 2.3 Lancez le script d'installation du paquet :

```
$ python setup.py install --user
```

L'option `--user` permet d'installer le paquet dans votre répertoire personnel. Si l'option n'est pas spécifiée, python tentera d'installer le paquet dans un répertoire système ce qui provoquera une erreur étant donné que vous n'avez pas les droits nécessaires.

Si l'installation s'est bien passée, le script `bibliodb` devrait être présent dans le répertoire `~/.local/bin/`. Si ce répertoire est bien présent dans votre variable d'environnement `PATH`, il peut maintenant être exécuté depuis n'importe quel répertoire, sans préciser le chemin complet de la commande.

I 2.4 Déplacez vous dans un répertoire quelconque, puis entrez la commande **`bibliodb`** pour tester si le shell trouve bien le fichier. Passez au point suivant si la commande n'est pas trouvée. Sinon la manipulation suivante n'est pas nécessaire.

I 2.5 Rajoutez la ligne suivante dans votre fichier `~/.bashrc` :

```
export PATH=$HOME/.local/bin:$PATH
```

(Attention à ne pas mettre d'espaces autour du =.) Cette ligne modifie votre variable d'environnement `PATH` pour y ajouter le répertoire `~/.local/bin/`. Refaites ensuite le test précédent dans un nouveau terminal (l'ouverture d'un nouveau terminal va lancer un nouveau shell bash, ce qui aura pour effet de recharger le fichier `~/.bashrc`).

Attention à ne pas confondre le fichier `~/.local/bin/bibliodb` qui est le programme principal de notre paquet obtenu après installation (avec la commande de l'instruction I 2.3) avec le fichier `~/R208/bibliodb` qui est celui sur lequel nous travaillerons mais qui n'est pas celui exécuté lorsqu'on lance **`bibliodb`** depuis le terminal.

Par la suite, à chaque fois que vous ferez une modification de votre paquet et que vous voudrez tester cette modification, vous devrez d'abord relancer le script d'installation (instruction I 2.3) avant de lancer la commande **`bibliodb`**.

Exercice 3 — Création des classes

Dans cet exercice nous allons créer les deux classes que nous utiliserons dans notre paquet : `Livre` et `Bibliotheque`.

I 3.1 Créez les fichiers `livre.py` et `bibliotheque.py`.

Un objet `Livre` est caractérisé par un titre, une année et un genre (p.ex, roman, histoire, manuel, ...). Le titre et le genre sont des chaînes de caractères. L'année est un entier.

I 3.2 Définissez dans le fichier `livre.py` une classe `Livre` contenant une méthode `__init__` permettant d'initialiser un livre avec un titre, une année et un genre passés en paramètres :

```
def __init__(self, titre, annee, genre):
    ...  # à compléter
```

Un objet `Bibliotheque` a pour attributs une liste de livres (des objets de la classe Livre) et une liste de genres (des chaînes de caractères).

I 3.3 Définissez dans le fichier `bibliotheque.py` une classe `Bibliotheque` avec une méthode `__init__` permettant d'initialiser une bibliothèque :

```
def __init__(self):
    ...  # à compléter
```

La méthode initialisera la bibliothèque avec une liste de livres et une liste de genres vides.

Exercice 4 — Fonction d'affichage du menu du programme

Dans cet exercice, nous allons programmer la fonction la plus simple de notre programme principal : celle qui permet d'afficher le menu de notre programme.

Vous trouverez en annexe un exemple d'exécution de notre programme. On s'y réfèrera dans cet exercice et dans les suivants pour expliquer le fonctionnement des différentes fonctionnalités du programme.

I 4.1 Ajoutez dans le fichier `commandes.py` une fonction `affiche_menu` qui se contentera d'afficher le menu du programme principal (voir les lignes 2 à 13 de l'exemple en annexe).

I 4.2 Ajoutez dans le fichier `bibliodb` une fonction `main` qui contiendra le code de notre programme principal. Le programme principal créera un objet `Bibliotheque` (celle sur laquelle le programme travaillera) puis traitera des commandes saisies par l'utilisatrice (avec la fonction **input**). Chaque commande correspondra à une fonction du fichier `commandes.py`, comme c'est le cas avec la commande d'affichage du menu : à la commande `M` correspond la fonction `affiche_menu`. Le programme s'arrêtera à la saisie de la commande `Q`.

I 4.3 Faites en sorte que la fonction `main` soit appelée à l'exécution de `bibliodb` depuis le terminal.

I 4.4 Réinstallez votre paquet (instruction I 2.3 de l'exercice 2) puis testez votre programme principal.

Exercice 5 — Fonction d'ajout de genres

I 5.1 Ajoutez dans la classe `Bibliotheque` :
- une methode `ajoute_genre` qui ajoute un genre passé en paramètre à la liste des genres de la bibliothèque ;
- une méthode `genre_existe` qui teste (en renvoyant `True` ou `False`) si un genre passé en paramètre a déjà été ajouté à la bibliothèque.

I 5.2 Ajoutez dans le fichier `commandes.py` une fonction `nouveau_genre` permettant de saisir un nouveau genre et de l'ajouter à une bibliothèque passée en paramètre. La fonction devra détecter si le genre a déjà été saisi (voir les lignes 14 à 25 de l'exemple en annexe). Vous utiliserez dans cette fonction les méthodes définies au point précédent.

I 5.3 Modifiez votre programme principal pour qu'il appelle la fonction `nouveau_genre` si la commande `NG` est saisie.

I 5.4 Testez.

Dans les exercices suivants vous suivrez les mêmes étapes pour ajouter de nouvelles fonctionnalités à votre programme : ajout d'une (de) méthode(s) dans la (les) classe(s) puis programmation de la commande dans `commandes.py` et enfin modification du programme principal pour appeler cette commande.

Cette façon de procéder est assez classique en programmation : on sépare le code de manipulation des données (placé dans les classes) de la partie interface utilisateur (placée ici dans le fichier `commandes.py`).

Exercice 6 — Fonction d'ajout de livres

Programmez la fonction d'ajout d'un livre à la bibliothèque (voir les lignes 26 à 57). On supposera que l'année saisie par l'utilisatrice est correcte (i.e., c'est un entier inférieur ou égal à l'année courante) : il n'y a donc aucune vérification à programmer sur l'année saisie. Dans un premier temps, le nouveau livre saisi sera ajouté en fin de liste (i.e., on ne se souciera pas d'un quelconque ordre dans la liste). C'est dans l'exercice 10 que nous programmerons une méthode de tri des livres.

Exercice 7 — Fonction d'affichage des livres

Programmez la fonction d'affichage des livres de la bibliothèque (voir les lignes 58 à 65).

Indications. Programmez la méthode `__str__` de la classe `Livre` pour que python convertisse automatiquement des objets de cette classe en objet **str** au besoin. Par exemple :

```
>>> print(Livre("Les misérables", 1862, "roman"))
1862 [roman] Les misérables
>>> print(Livre("Un bouquin sans genre", 1820, ""))
1862 Un bouquin sans genre
```

Exercice 8 — Autres fonctions

I 8.1 Programmez la fonction d'affichage des genres de la bibliothèque (voir les lignes 66 à 70).

I 8.2 Programmez la fonction de suppression d'un livre à partir de son titre (voir les lignes 71 à 88).

I 8.3 Programmez la fonction de suppression d'un genre (voir les lignes 89 à 101). Dans le cas où le genre supprimé est associé à un (ou plusieurs) livre(s), le genre du (des) livre(s) concerné(s) sera initialisé à une chaîne vide (`""`).

Exercice 9 — Sauvegarde et chargement de la bibliothèque

On souhaite maintenant que le contenu de notre bibliothèque puisse être sauvegardé dans un fichier pour pouvoir le retrouver à chaque exécution du programme. On choisit pour cela le format JSON.

L'utilisation de fichiers sera transparente dans le programme, c'est-à-dire que l'utilisatrice n'aura à aucun moment besoin de préciser qu'elle veut sauvegarder sa bibliothèque dans un fichier ou encore de préciser l'emplacement du ficher dans lequel sa bibliothèque sera sauvegardée. Ceci sera fait automatiquement.

Modifiez votre paquet et le programme principal, afin

— qu'au démarrage du programme, le contenu de la bibliothèque soit rechargé depuis le fichier `~/.biblio.json`;
— et qu'après chaque modification de la bibliothèque (p.ex., après l'ajout d'un livre) le contenu de la bibliothèque soit sauvegardé dans ce même fichier.

De plus votre programme devra traiter les cas exceptionnels suivants :

— Si au lancement du programme, le fichier n'existe pas, la bibliothèque sera initialisée avec un contenu vide.
— De même, si le fichier existe mais qu'il est mal formé (erreur de syntaxe JSON), la bibliothèque sera initialement vide.
— Enfin, s'il est impossible d'écrire dans le fichier `~/.biblio.json`, la bibliothèque ne sera pas sauvegardée.

Dans ces trois situations, le programme ne devra pas s'arrêter en levant une exception mais afficher un message d'avertissement.

Indications. Un objet `Bibliotheque` ou `Livre` ne peut pas être directement être transformé dans le format JSON. Pour l'écriture vous pourrez donc définir des méthodes qui permettront de traduire vos objets en données pouvant être écrites dans un fichier JSON (p.ex. un dictionnaire contenant les valeurs des attributs de l'objet). De même, pour la lecture, vous pourrez définir des méthodes prenant en paramètre une données JSON et initialisant les attributs de l'objet à partir de cette donnée.

Exercice 10 — Tri des livres

Faites en sorte, à la création d'un nouveau livre (fonction `NL`) que la liste des livres reste triée. Les livres seront triés sur l'année, puis à année égale sur le genre, et enfin à année et à genre égaux sur le titre (voir les lignes 60 à 65 de l'exemple). Vous pourrez programmer la méthode `__lt__` de la classe `Livre` pour comparer plus facilement les livres.

Exercice 11 — Traitement des exceptions

Rajoutez les traitements d'exceptions suivants à votre programme :

— En cas d'interruption du programme (Ctrl+C tappé par l'utilisatrice), le programme devra s'arrêter "proprement" (sans afficher une exception).

— De même, si l'utilisatrice tappe Ctrl+D (ce qui correspond au caractère de fin de fichier), le programme se terminera de la même manière.

— Si une autre exception est levée lors de l'exécution, un message demandant à l'utilisatrice d'envoyer un rapport de bug sera affiché (p.ex., *Une erreur est survenue. Merci d'envoyer un rapport de bug à* `support@bibliodb.edu`) et le programme se terminera. L'adresse électronique apparaissant dans le message devra être stockée dans une variable du fichier `__init__.py`.

Annexe 1 — Exemple d'exécution du programme

```
Bienvenue, vous utilisez le paquet biblio v0.0.0!
> Entrez votre commande (M pour afficher le menu) : M
****************************************
Menu principal
   [M]  Menu principal
   [LG] Liste des Genres
   [LL] Liste des Livres
   [NG] Nouveau Genre
   [NL] Nouveau Livre
   [SG] Suppression d'un Genre
   [SL] Suppression d'un Livre
   [Q]  Quitter le programme
****************************************
> Entrez votre commande (M pour afficher le menu) : NG
> Entrez le nom du genre : histoire
Le genre histoire a été créé.
> Entrez votre commande (M pour afficher le menu) : NG
> Entrez le nom du genre : roman
Le genre roman a été créé.
> Entrez votre commande (M pour afficher le menu) : NG
> Entrez le nom du genre : histoire
Ce genre existe déjà!
> Entrez votre commande (M pour afficher le menu) : NG
> Entrez le nom du genre : bd
Le genre bd a été créé.
> Entrez votre commande (M pour afficher le menu) : NL
> Entrez le titre du livre : Histoire de la révolution francaise
> Entrez l'année de parution du livre : 1960
> Entrez le genre du livre : histoir
Ce genre n'existe pas!
> Entrez le genre du livre : histoire
Le livre "Histoire de la révolution francaise" a été créé.
> Entrez votre commande (M pour afficher le menu) : NL
> Entrez le titre du livre : Le rouge et le noir
> Entrez l'année de parution du livre : 1830
> Entrez le genre du livre : roman
Le livre "Le rouge et le noir" a été créé.
> Entrez votre commande (M pour afficher le menu) : NL
> Entrez le titre du livre : L'incal
> Entrez l'année de parution du livre : 1972
> Entrez le genre du livre : bd
Le livre "L'incal" a été créé.
> Entrez votre commande (M pour afficher le menu) : NL
> Entrez le titre du livre : Les misérables
> Entrez l'année de parution du livre : 1862
> Entrez le genre du livre : roman
Le livre "Les misérables" a été créé.
> Entrez votre commande (M pour afficher le menu) : NL
> Entrez le titre du livre : Astérix et Cléopatre
> Entrez l'année de parution du livre : 1960
> Entrez le genre du livre : bd
Le livre "Astérix et Cléopatre" a été créé.
> Entrez votre commande (M pour afficher le menu) : NL
> Entrez le titre du livre : Un bouquin sans genre
> Entrez l'année de parution du livre : 2002
> Entrez le genre du livre :
Le livre "Un bouquin sans genre" a été créé.
> Entrez votre commande (M pour afficher le menu) : LL
```

```
6 livre(s) trouvé(s)
1830 [roman] Le rouge et le noir
1862 [roman] Les misérables
1960 [bd] Astérix et Cléopatre
1960 [histoire] Histoire de la révolution francaise
1972 [bd] L'incal
2002 Un bouquin sans genre
> Entrez votre commande (M pour afficher le menu) : LG
roman: 2 livre(s)
bd: 2 livre(s)
histoire: 1 livre(s)
sans genre: 1 livre(s)
> Entrez votre commande (M pour afficher le menu) : SL
> Entrez le titre du livre : Astérix et Cléopatr
Ce livre n'existe pas!
> Entrez votre commande (M pour afficher le menu) : SL
> Entrez le titre du livre : Astérix et Cléopatre
Le livre "Astérix et Cléopatre" a été supprimé.
> Entrez votre commande (M pour afficher le menu) : LL
5 livre(s) trouvé(s)
1830 [roman] Le rouge et le noir
1862 [roman] Les misérables
1960 [histoire] Histoire de la révolution francaise
1972 [bd] L'incal
2002 Un bouquin sans genre
> Entrez votre commande (M pour afficher le menu) : LG
roman: 2 livre(s)
bd: 1 livre(s)
histoire: 1 livre(s)
sans genre: 1 livre(s)
> Entrez votre commande (M pour afficher le menu) : SG
> Entrez le nom du genre : romane
Ce genre n'existe pas!
> Entrez votre commande (M pour afficher le menu) : SG
> Entrez le nom du genre : roman
Le genre roman a été supprimé.
> Entrez votre commande (M pour afficher le menu) : LL
5 livre(s) trouvé(s)
1830 Le rouge et le noir
1862 Les misérables
1960 [histoire] Histoire de la révolution francaise
1972 [bd] L'incal
2002 Un bouquin sans genre
> Entrez votre commande (M pour afficher le menu) : Q
```