

TP5 : Scripting

1. Scripts & Packages

Dans la première partie de ce TP, vous allez aborder - via l'écriture de fichier .py - le principe d'une application fonctionnant sur base de packages créés par vous-même. Lors des précédents TP, vous avez eu la possibilité de travailler avec, par exemple, la **librairie statistics** et une de ses fonctions, **mean**. Aujourd'hui, vous allez créer votre librairie **ephec** et en définir certaines fonctionnalités.

1.1 - Initialiser un projet, un venv et un fichier main.py

Commencez par **créer un projet** dans votre IDE Pycharm tout en créant un **venv**.

Ensuite, créez un fichier **main.py** à la racine du projet. Ce fichier sera le point d'entrée lorsque vous lancerez votre programme. (Si un fichier main.py a été généré automatiquement, supprimez les lignes de code s'y trouvant).

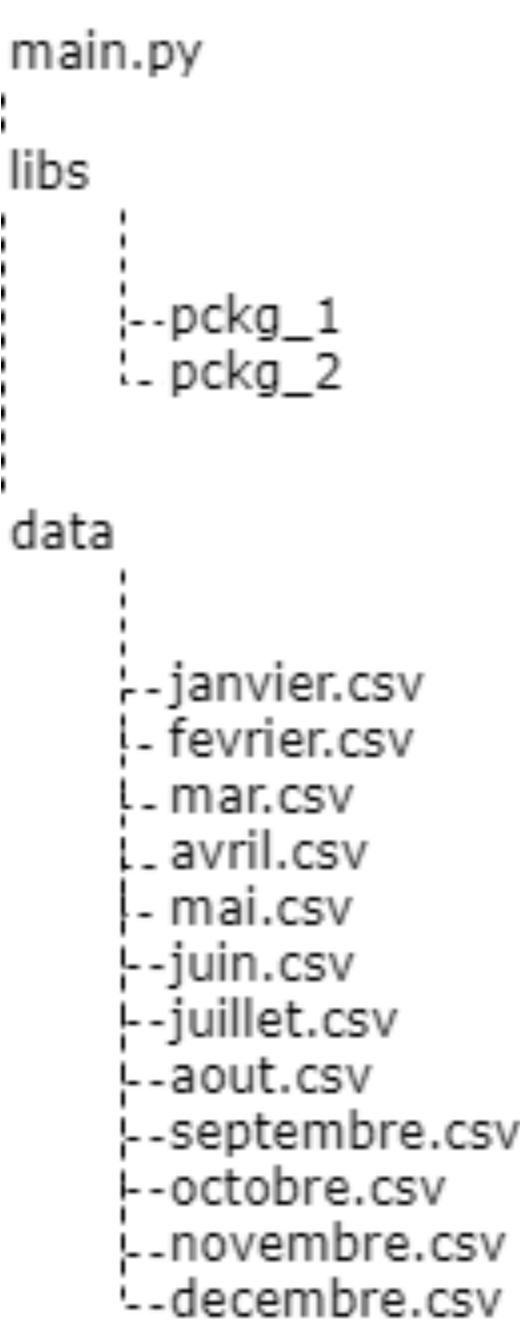
Vous avez désormais tous les outils nécessaire pour commencer le TP.

1.2 - Application de vente de BD

Le thème du programme est **l'analyse des ventes d'une petite échoppe de BD**. Le vendeur souhaite avoir une idée de son chiffre d'affaire par BD. Ses ventes sont rassemblées dans des fichiers **.csv mensuels**. De plus, il aimerait avoir une vue d'ensemble du nombre de ventes de chacune de ses BD afin de savoir lesquelles sont populaires. Malheureusement, il ne comprend rien en informatique et il a besoin de votre aide.

Nous souhaitons que votre projet ait une organisation de dossier telle que sur l'image ci-dessous.

- Le fichier **main.py** qui est votre fichier principal.
- Le dossier **data** contient des fichiers .csv contenant les données concernant les ventes mensuelles.
- Le dossier **libs** contiendra les packages que vous allez créer avec vos fonctionnalités.



1.2.1 Création du package bd

Après avoir organisé votre projet, veuillez créer un package nommé **bd**. Dans ce package se trouveront les fonctionnalités spécifiquement liées aux bandes dessinées. Se trouvera donc dans le package un fichier **__init__.py** (allez vous renseigner sur le sujet) et les fichiers de code source.

Le code source sera constitué d'un script définissant une classe BandeDessinee. Cette classe possède deux attributs : Le prix, qui est spécifié lors de la création puis ne peut plus être modifié, et la quantité vendue, qui peut être augmentée via une méthode **add_sales(num_books)**.

Faites également en sorte que chaque BandeDessinee puisse produire une représentation textuelle semblable à celle-ci :

```
La BD Tintin a été vendue 63 fois au prix de 9,95 € ==> 626.85 €
```

1.2.2 Création du package file_mngmt

A présent, créez un second package, appelé file_mngt. Dans ce package, créez un module/script Python nommé **csv_opener.py**. Ce module contiendra quatre fonctions :

- **get_data_files_names(folder)**
 - Cette fonction ira récupérer tous les noms de fichiers se trouvant dans le dossier **data**.
 - Renseignez vous sur la librairie **os**
 - Attention aux fichier cachés, nous ne souhaitons pas qu'ils soient présents dans notre liste de noms de fichiers.
- **get_data_in_file(filename)**
 - Cette fonction ouvre le fichier qui porte le nom passé en paramètre et affecte les données dans un dictionnaire. De plus, un pré-traitement des données doit être effectué. Dans le dictionnaire final retourné, chaque BD doit avoir une quantité de vente totale sur le mois et un prix unique. Exemple de dictionnaire :

```
>>> print(result["Lucky Luke"])
>>> {"price":14,"quantity": 8}
```
 - Lors du cours précédent, vous aviez dû travailler sur l'ouverture de fichiers .csv.
 - Attention à l'encodage lors de l'ouverture, assurez-vous qu'il soit en **utf-8**.
 - Attention à la ligne de titres des fichiers .csv. Elle ne nous intéresse pas.
- **concatenate_all_files_data(whole_data)**
 - Cette fonction est presque similaire à la précédente sauf que le but est de rassembler la totalité des ventes mensuelles en un seul dictionnaire. Elle prend en paramètre une liste de dictionnaires qui ont été générés par les appels à get_data_in_file().
- **get_data_from_folder_files(folder)**
 - Cette fonction est la fonction qui sera appelée lorsque l'on souhaitera travailler avec cette librairie. Elle permet de réaliser toutes les étapes précédentes et retourne le dictionnaire final traité.

1.2.3. Fichier main.py

Ce fichier représente le point d'entrée de votre programme. Il sert à désigner la logique d'exécution que votre programme doit suivre. Il ne comportera donc que les import nécessaire au bon fonctionnement du programme ainsi que la condition if __name__ "main" contenant la suite d'instruction devant être exécutée.

Il aura bien entendu une interface en ligne de commande claire avec notamment un affichage du mode d'emploi via l'option "help".

Nous attendons de vous que ce script aille appeler la fonction **get_data_from_folder_files** de votre librairie **file_mngt** afin de recevoir les données des fichiers .csv prétraitées par votre librairie. A partir du dictionnaire obtenu, il crée une liste d'objets BandeDessinee, puis affiche le résultat suivant en console :

```
-----
-----
La BD Tintin a été vendue 63 fois au prix de 9,95 € ==> 626.85 €
-----
-----
La BD Asterix a été vendue 67 fois au prix de 12,5 € ==> 837.5 €
-----
-----
La BD XIII a été vendue 11 fois au prix de 5 € ==> 55.0 €
-----
-----
La BD Thorgal a été vendue 62 fois au prix de 8,49 € ==> 526.38 €
-----
-----
La BD Gaston a été vendue 16 fois au prix de 6,78 € ==> 108.48 €
-----
-----
La BD Spirou et Fantasio a été vendue 37 fois au prix de 7,23 € ==>
267.51 €
-----
-----
La BD Largo Winch a été vendue 53 fois au prix de 9,99 € ==> 529.47
€
-----
-----
La BD Okko a été vendue 37 fois au prix de 8,25 € ==> 305.25 €
-----
-----
La BD Murena a été vendue 13 fois au prix de 4,23 € ==> 54.99 €
-----
-----
-----
LE GRAND GAGNANT EST : Asterix AVEC 837.5 € DE CHIFFRE D'AFFAIRES
-----
-----
```

1.3 - Bonus : Utilisation de subprocess (spoiler TP06)

Nous avons vu qu'il était possible de lister les fichiers présents dans un certain dossier grâce à la librairie **os** et sa méthode **listdir**.

En Python, il faut savoir que nous pouvons exécuter des commandes DOS/unix via la librairie **subprocess**. Il est donc possible d'afficher le contenu d'un dossier, d'un fichier, écrire du texte en console, etc. Pour résumer, toute commande utilisable en terminal.

Pour cet exercice, nous vous demandons de modifier votre projet précédent.

Premièrement, vous devrez créer un nouveau module **csv_opener_subprocess.py** dans votre package **ephec**.

Ensuite, lorsque vous pensez que certaines lignes de votre code peuvent être remplacées par des appels à des commandes shell via **subprocess**, nous vous invitons à créer une fonction associée dans votre nouvelle librairie.