

TD 19 : Vol au-dessus d'un nid de coucou

On dispose d'un fichier CSV qui indique les vols disponibles pour des destinations de rêves. On souhaite gérer ces vols à l'aide d'un dictionnaire de dictionnaires comme indiqué par l'exemple ci-dessous :

```
vols = {'Lisbonne': {'heure': 21:10,
                    'num': 'EJU7674',
                    'compagnie': 'EASYJET'},
        'Vienne': {'heure': 21:25,
                   'num': 'OS430',
                   'compagnie': 'AUSTRIAN AIRLINES'},
        'Londres': {'heure': 21:55,
                    'num', 'BA357'
                    'compagnie': 'BRITISH AIRWAYS'}
        ...}
```

1. Allez sur python, et créez TD19-avions.
2. Écrire une fonction qui permet d'obtenir, à partir d'un fichier CSV (le fichier planes.csv qui se trouve dans l'espace classe), la liste des destinations sous la forme d'un dictionnaire dont le format est indiqué ci-dessus.

On donne le prototype de la fonction :

```
get_flights(filename : str) -> dict
```

- filename -- fichier csv (ville,compagnie,départ,n° de vol)
- la fonction retourne un dictionnaire de dictionnaires {ville destination: {heure départ, n° de vol, compagnie aérienne}}, ou None si erreur.

NB : Un fichier CSV est un fichier texte, par opposition aux formats dits « binaires ». Chaque ligne du texte correspond à une ligne du tableau et les virgules correspondent aux séparations entre les colonnes. Les portions de texte séparées par une virgule correspondent ainsi aux contenus des cellules du tableau.

Source : [wikipedia](https://fr.wikipedia.org/wiki/Comma-separated_values)

NB : on pourra utiliser le module csv de Python.

3. Créer une fonction test dans lequel vous vérifierez les tests unitaires suivants :
- 'Berlin' n'est pas une destination
 - 'Londres' est une destination
 - La compagnie aérienne qui assure le vol à destination de 'Lisbonne' est 'EASYJET'
 - Le n° du vol 'OSS117' ne dessert pas 'Lisbonne'
 - Le vol 'BA357' existe

TD 19 : Vol au-dessus d'un nid de coucou

NB : on utilisera une compréhension de liste quand cela s'avérera nécessaire.

4. La destination 'ville' fait-elle partie des destinations ? Conclure et modifier le programme en conséquence.

NB : si vous avez utilisé la fonction "reader()" du module csv, vous pouvez transformer l'objet renvoyé par cette fonction en une liste à l'aide de la fonction "list()".

5. Revalider les tests unitaires.
6. Documenter la fonction (docstring).

On souhaite améliorer le programme en attribuant automatiquement comme clef de dictionnaire les champs de l'en-tête du fichier CSV. Puis en filtrant les destinations sur le nom d'une compagnie aérienne particulière.

7. Copier et modifier la fonction `get_flights()` en `get_flights2()` pour gérer automatiquement les clefs de dictionnaire.
8. Ajouter un paramètre `company` à la fonction `get_flights2()` pour filtrer les destinations (la fonction renvoie uniquement les destinations desservies par cette compagnie). Par défaut ce paramètre est initialisé à chaîne vide (si aucun vol n'est filtré, c'est à dire si il n'y a pas de compagnie spécifiée lors de l'appel de la fonction alors la fonction renvoie tous les vols).

On obtient donc : `get_flights(filename : str, company="") -> dict`

NB : on utilisera une compréhension de liste pour la génération du dictionnaire retourné par la fonction.

9. Revalider les tests unitaires précédents.
10. Documenter la fonction (docstring).
11. Dans le `main.py`, écrire un programme qui, à partir du nom d'une compagnie aérienne saisi interactivement par l'utilisateur, donne, sous forme d'une liste créée par extension, toutes les destinations desservies par cette compagnie.

NB : on fera attention à la casse de la saisie.