

TP 1 : Listes, et Dictionnaires en Langage Python

Consignes :



- Le compte-rendu doit être rendu **sous forme numérique** de préférence en format pdf.
- Le compte-rendu doit être soumis sur Moodle dans la section TP 1 puis dans évaluation.
- Le projet zippé contenant tous vos fichiers doit être rendu également.

Exercice 1 : Utilisation d'un Dictionnaire

Nous avons à notre disposition un dictionnaire `SOC1ROANNE_VENTES` associant à des noms de commerciaux d'une société `SOC1ROANNE` le nombre de ventes réalisées par ces derniers.

Par exemple :

```
SOC1ROANNE_VENTES={"Martin":38, "Dupont":52, "Durand":26, "Perrin":64,"Blanc":42}
```

- 1-1 Écrire une fonction `Ex1Q1()` en Python prenant en entrée un dictionnaire de ce type et retournant le nombre total de ventes dans la société.
- 1-2 Écrire une fonction `Ex1Q2()` en Python prenant en entrée un dictionnaire de ce type et retournant le nom du vendeur ayant réalisé le plus de ventes. Si plusieurs vendeurs ont réalisé la même vente en quantité, cette fonction devra retourner le nom de tous ces vendeurs dans une liste.
- 1-3 Vous testerez le bon fonctionnement de ces deux fonctions en les appelant judicieusement dans un autre programme.

Exercice 2 : Manipulation d'un Dictionnaire

- 2-1 Écrire une fonction `Ex2()` en Python prenant en entrée une chaîne de caractères contenant, sur chaque ligne, trois champs séparés par des caractères ';' (un numéro d'étudiant, un nom et un prénom) et retournant un dictionnaire dont les clés sont les numéros d'étudiants et les valeurs sont, pour chaque clé, une chaîne correspondant à la concaténation des prénom et nom de la personne.
- 2-2 Vous pourrez tester votre fonction `Ex2()` avec la chaîne suivante :

```
BaseEtudiants = """2021420025;BLANC;Ludovic
2021421143;DUPOND;Claire
2021420247;DURAND;Juliette
2021420009;MARTIN;Alexandre
2021422278;PERRIN;Laurent"""
```

Exercice 3 : Bataille Navale

Le but de cet exercice est de simuler une partie de bataille navale en utilisant des dictionnaires notamment pour la grille et le position des différents bateaux.

Le joueur aura gagné lorsqu'il aura coulé tous les navires de la grille.

Le joueur disposera d'une grille carrée de côté 10, dont les lignes sont numérotées de 1 à 10 et les colonnes de A à J, où il devra identifier une flotte composée de cinq navires.

Cette flotte est composée d'un porte-avions (5 cases), un croiseur (4 cases), 2 contre-torpilleurs (3 cases) et un torpilleur (2 cases).

Il faudra programmer en Python un jeu de bataille navale pour lequel la position des navires sera lue dans un autre fichier (`Bateaux.txt`) dont voici un exemple contenant deux navires nommés respectivement Porte-avions et Contre-torpilleur 1 :

```
Porte-avions,D1,D2,D3,D4,D5
Contre-torpilleur 1,E4,F4,G4
```

Le programme demandera au joueur de saisir une case de la grille dans laquelle jouer jusqu'à ce qu'il ait coulé tous les bateaux.

Vous pouvez obtenir comme grille de jeu :

```
Quelle est la case ? A1
À l'eau
- - - - -
  A B C D E F G H I J
- - - - -
01 X
02
03
04
05
06
07
08
09
10
- - - - -
```

```
Quelle est la case ? D4
Porte-avions coulé
- - - - -
  A B C D E F G H I J
- - - - -
01 X 0
02 X 0
03 0
04 X X 0
05 0
06
07
08
09
10
- - - - -
```

Remarque : Pour les questions suivantes **3-1** à **3-7**, à vous de choisir le(s) bon(s) argument(s) d'entrée et le(s) paramètre(s) de sortie.

- 3-1** Pour cela, créer tout d'abord une fonction `PositionsNavires()` permettant d'obtenir un dictionnaire à partir du fichier `Bateaux.txt`.
Par exemple, `b={"Porte-avions": ["D1", "D2", "D3","D4","D5"],...}`
- 3-2** Créer une autre fonction `GrilleNavires()` permettant d'initialiser la grille (qui conservera l'historique des choix effectués) sous forme d'un autre dictionnaire.
Par exemple, `g={"1": [" ", " ", " ", " ", " ", " ", " ", " ", " ", " "],...}`
- 3-3** Créer une fonction `AfficherGrille()` permettant d'afficher le dictionnaire associé à la grille comme dans l'illustration ci-dessus.
- 3-4** Créer une fonction `Position()` permettant de saisir une proposition et de vérifier si cette dernière est bien comprise dans la grille.
- 3-5** Créer une fonction `Coule()` permettant de vérifier si un navire de la flotte est coulé.
- 3-6** Créer une fonction `Gagne()` permettant de savoir si vous avez gagné en vérifiant que tous les navires sont coulés.
- 3-7** Créer une fonction `Grille()` permettant de mettre à jour le dictionnaire représentant la grille suite aux différentes propositions du joueur.
- 3-8** Appeler ces fonctions de manière judicieuse de façon à pouvoir réaliser une partie.
- 3-9** Reprendre la question **3-1** en lisant cette fois-ci un fichier au format YAML.
- 3-10** Enregistrer à la toute fin de votre partie la grille (le dictionnaire associé) dans un fichier YAML.