R1.01 Initiation au développement
Feuille de TP n°8B
Listes, ensembles et tuples

Objectifs de la feuille

- Manipuler des dictionnaires, des ensembles et des listes
- Coder quelques fonctions simples en faisant attention au **bon** choix de boucle

Exercice 1 *Perrette et son troupeau*

Dans le monde de Perrette, on modélise les troupeaux sous la forme d'un dictionnaire dont les clés sont le nom des animaux et les valeurs leur nombre dans le troupeau. Par exemple, si Perrette possède dans son troupeau 14 veaux, 7 vaches et 42 poules, son troupeau sera modélisé de la façon suivante :

```
troupeau_de_perrette = {'veau':14, 'vache':7, 'poule':42}
```

1.1 Décrire les deux troupeaux suivants puis inventer votre propre troupeau :

```
troupeau_de_jean = {'vache':12, 'cochon':17, 'veau':3}
troupeau_vide = dict()
mon_troupeau = ...
```

1.2 La fonction `total_animaux` prend un troupeau en paramètre et renvoie le nombre total d'animaux dans le troupeau

a) Dans le fichier `test_troupeaux.py` , compléter le code de la fonction de test avec un troupeau de votre invention

a) Dans le fichier `troupeaux.py` , compléter le code de la fonction proposée.

1.3 Mêmes questions avec la fonction `tous_les_animaux` prend un troupeau en paramètre et renvoie l'ensemble des noms des animaux du troupeau.

1.4 Mêmes questions avec la fonction `specialise` prend un troupeau en paramètre et vérifie si le troupeau contient 30 individus ou plus d'un même type d'animal

1.5 Mêmes questions avec la fonction `le_plus_represente` prend un troupeau en paramètre et renvoie le nom du type d'animal qui a le plus d'individus dans le troupeau ou None si le troupeau est vide.

1.6 Mêmes questions avec la fonction `quantite_suffisante` vérifie si chaque type d'animal présent dans le troupeau passé en paramètre contient au moins 5 individus.

1.7 Mêmes questions avec la fonction `reunion_troupeaux` prend deux troupeaux en paramètre et renvoie le dictionnaire qui représente la réunification des deux troupeaux.

Exercice 2 *Des super-héros*

On dispose de dictionnaires dont les clefs sont les noms de personnage, et les valeurs des tuples contenant la force (int), l'intelligence (int) et la description (str) de ce personnage.

Voici un extrait d'un exemple d'un tel dictionnaire :

```
avengers = {
    'Spiderman': (5, 5, 'araignée a quatre pattes'),
    'Hulk': (7, 4, "Grand homme vert"),
    'Agent 13': (2, 3, 'agent 13'),
    'M Melin': (2, 6, 'expert en archi'), ...
    }
```

2.1 Sans utiliser d'ordinateur, compléter le tableau suivant :

description	expression	valeur
Force de Hulk	`avenger['Hulk'][0]`	`7`
		`'araignée a quatre pattes'`
	`avenger['Drax'][2]`	`'Benêt tatoué'`
Intelligence de Gamora		`4`
		`(2, 3, 'agent 13')`

Pour les questions suivantes, une bonne méthodologie et des bonnes pratiques sont de rigueur.

2.2 Dans un fichier `super_heros.py` , écrivez une fonction `intelligence_moyenne` qui prend en paramètre un tel dictionnaire et qui renvoie la valeur moyenne de l'intelligence de tous les personnages. Vous n'oublierez pas d'ajouter un fichier de tests.

Remarque et conseil

Pour écrire vos tests, utilisez un jeu de données limité dans lequel vous vous attardez uniquement sur ce qui est important à ce moment là.. Par exemple, ici, la description importe peu.

```
# dictionnaires bidons pour faire des tests :
exemple2 = {'a':(1, 1, 'a'), 'b':(3, 9, 'b'), 'c':(7, 2, 'c')}
assert intelligence_moyenne(exemple2) == 4
exemple3 = {'a':(1, 1, 'a'), 'b':(3, 9, 'b'), 'd':(4, 4, 'd')}
assert abs(intelligence_moyenne(exemple3)-14/3) <= 0.01
```

2.3 Écrivez une fonction `kikelplusfort` qui prend en paramètre un tel dictionnaire et qui renvoie le nom du personnage le plus fort.

2.4 Écrivez une fonction `combienDeCretins` qui prend en paramètre un tel dictionnaire et qui renvoie le nombre de personnages dont l'intelligence est strictement inférieure à la moyenne.