

TP n° 3 : dictionnaires et ensembles

Objectifs

1. Manipulation des dictionnaires
2. Manipulation des ensembles

Exercice 1

Soit la liste de tuples suivante :

```
L=[('nom', 'Duprès'), ('prenom', 'Jaques'), ('poste', 'ingénieur'), ('date', date
time.date(2014, 10, 9))]
```

- 1- A partir de L, créer le dictionnaire **dic** de la forme {'nom' : 'Duprès', 'prenom' : 'Jaques', ...}
- 2- Exécuter et interpréter les instructions suivantes :
 - a. dic.keys()
 - b. dic.values()
 - c. dic.items()
 - d. dict(dic.items())
 - e. print({k:dic[k] for k in dic.keys()})
 - f. print({k:v for k,v in dic.items()})
- 3- Ecrire la phrase Jaques Duprès était un ingénieur en 2014-10-09
- 4- Créer une copie de **dic** nommée **dic1** et y changer le poste à 'chef de projet' et la date à (2016,10,9)
- 5- Ajouter la clé **age** à **dic** et la mettre à 30
- 6- Mettre à jour dynamiquement l'age de Jaques Duprès quand il a occupé le poste de chef de projet

NB. datetime.date(2014, 10, 9).year renvoie 2014, l'année correspondante à cette date

Exercice 2

Soit le tableau suivant représentant des informations physico-chimiques sur des éléments simples (température d'ébullition (T_e) et de fusion (T_f), numéro (Z) et masse atomique (A) :

| | | | |
|----|-----------|------|---------|
| Au | T_e/T_f | 2970 | 1063 |
| | Z/A | 79 | 196.167 |
| Ga | T_e/T_f | 2237 | 89.8 |
| | Z/A | 31 | 69.72 |

- 1- Affectez les données de ce tableau à un dictionnaire **dico** de façon à pouvoir écrire par exemple :
 >>> print dico["Au"]["Z/A"][0] # affiche : 79
- 2- Supprimer la ligne

| | | |
|-----|----|---------|
| Z/A | 79 | 196.167 |
|-----|----|---------|

de façon à ce que dico sera

| | | | |
|----|-----------|------|------|
| Au | T_e/T_f | 2970 | 1063 |
|----|-----------|------|------|

| | | | |
|----|--------------------------------|------|-------|
| Ga | T _e /T _f | 2237 | 89.8 |
| | Z/A | 31 | 69.72 |

3- Modifier dico pour qu'il sera

| | | | |
|----|--------------------------------|------|---|
| Au | T _e /T _f | 2970 | 0 |
| Ga | T _e /T _f | 2237 | 0 |
| | Z/A | 31 | 0 |

Exercice 3

Supposons qu'une liste **birthdays** contient des chaînes de caractères qui représentent des dates de naissance tels que 'JAN 2 2020', composé des 3 premiers lettres pour désigner le mois, le jour sur 1 ou 2 caractères et l'année sur 4 caractères. Toutes les informations sont séparées par des espaces.

- 1- A partir de la liste **birthdays**, construire la liste de dictionnaires **list_birth** qui représente chaque élément de birthdays comme un dictionnaire de la forme {'month': ..., 'day': ..., 'year': ..}
- 2- Afficher **list_birth** triée dans l'ordre chronologique

Par exemple Si `list_birth = [{'month': 'JAN', 'day': 7, 'year': 2020}, {'month': 'FIB', 'day': 12, 'year': 2018}, {'month': 'FIB', 'day': 12, 'year': 2018}]`
 Alors le résultat affiché sera

```
[ {'month': 'FIB', 'day': 12, 'year': 2018}, {'month': 'FIB', 'day': 12, 'year': 2018}, {'month': 'JAN', 'day': 7, 'year': 2020}]
```

- 3- Afficher la liste des dates de **list_birth** sous la forme d'une chaîne de caractères 'yy/dd/YYY'

Le résultat d'affichage : ['12/02/2018', '12/02/2018', '07/01/2020']

- 4- Construire le dictionnaire **occ_birth** qui associe à chaque date de naissance son nombre d'occurrence.
 Selon l'exemple précédent, **occ_birth** sera {'12/02/2018': 2, '07/01/2020': 1}
- 5- Lire à partir du clavier le nombre minimal d'occurrence d'une date de naissance **min_count** et afficher parmi les dates existantes celles qui ocurrent au moins **min_count** fois.

Si **min_count** = 2, alors le résultat s'affiche comme suit : ['12/02/2018']

Exercice 4

Soient les chaînes de caractères suivantes : 'aabbc' et 'ddeffa'

- 1- Construire les ensembles A et B à partir des éléments qui composent chaque chaîne de caractères
- 2- Vérifier si A et B sont disjoints (leur intersection est l'ensemble vide)
En utilisant l'ensemble en compréhension :
- 3- Construire l'ensemble {'ad', 'ae', 'af', 'ba', 'bd', 'be', 'bf', 'ca', 'cd', 'ce', 'cf'}
- 4- Ecrire le code python qui retrouve le résultat de A&B (ou A.intersection(B)) sans utiliser l'opérateur & ou la méthode intersection
- 5- Ecrire le code python qui retrouve le résultat de A-B (ou A.difference(B)) sans utiliser l'opérateur - ou la méthode difference

Bon travail