

# TP2 - Collections

---

A partir du travail d'Eric Thomas, mise à jour par Nicolas 'Lomens' Resin

Cette semaine, nous allons travailler sur différentes collections. Des cours sont données en supplément et le cours de Pierre Giraud est toujours utile

Pour vous aider, voici la doc des Structures de données de python : [Documentation](#)

## Exo 1

Plutôt un travail d'observation pour celui-ci. Dans un interpréteur entrez chaque ligne indépendamment.

```
import copy

a = ['A']
b = a
b[0] = 1
a
b
```

Qu'affiche l'interpreteur pour la valeur a ? et pour la valeur b ?

En faisant `b = a`, nous ne copions pas la valeur de a dans b mais une référence. Donc modifier b revient à modifier a. Pour solutionner ce problème, des solutions existent. Visitez la doc sur les copy [ICI](#) et essayer les fonction copy, deepcopy et [:] dans l'interpréteur. Une fois satisfait, continuez le TP

## Exo 2

Considérant ce dictionnaire

```
classDict = {
    "class": {
        "student": {
            "name": "Mike",
            "marks": {
                "physics": 70,
                "history": 80
            }
        }
    }
}
```

Faite dans l'ordre sans modifier le dictionnaire de base.

1. Donner le code qui permet d'avoir le nom de l'étudiant

2. Modifier la note de Mike en physics pour qu'il ai 89
3. Ajoutez une clé average qui possède la moyenne des notes de Mike
4. La clé student possède maintenant une liste avec comme étudiant Mike
5. Ajoutez un étudiant nommé Ted avec comme note 34 et 99
6. Calculez la moyenne de Ted
7. Ajouter une clé au même niveau que student nommé average\_grade qui est la moyenne de la classe (utilisez len)
8. Affichez le dictionnaire complet

## Exo 3

Un tableau contient n entiers (  $2 < n < 100$  ) aléatoires tous compris entre 0 et 500. Créer une fonction qui retourne s'il sont tous différents ou non.

## Exo 4

Vous tenez à jour les scores d'un match de baseball aux règles étranges. On vous donne une liste d'opérations permettant de calculer

- Un nombre entier x : Enregistre un nouveau score de x.
- '+' : Enregistre un nouveau score qui est la somme des deux scores précédents.
- 'D' : Enregistrez un nouveau score qui est le double du score précédent.
- 'C' : Invalide le score précédent, en le supprimant de l'enregistrement.

Retourner la somme de tous les scores de l'enregistrement après avoir appliqué toutes les opérations. Réaliser la fonction calculScore() permettant à partir d'une liste contenant les éléments de match de générer une pile de score permettant de calculer la somme des points de l'équipe.Ce résultat sera retourné par la fonction.

Exemple de traitement d'une liste

```
p = ["10", "2", "C", "D", "+"]
"10" --Ajoute 10 à la pile pile=[10].
"2" -- Ajoute 2 à la pile pile=[10 2].
"C" -- invalide le score précédent, pile=[10].
"D" -- Ajoute 2 * 10 = 20 pile=[10, 20].
"+" -- Ajoute 10 + 20 = 30 pile=[10, 20, 30].
La somme des points est de 10 + 20 + 30 = 60
```

## Exo 5

On représente un polynôme par une liste chaînée (deque). Chaque cellule de la liste correspond à un coefficient. Par exemple, ci-dessous, on représente la liste chaînée correspondant au polynôme  $10x^2 + 5x + 3$  par 10| 5| 3

1. Écrire une fonction qui prend en paramètre un polynôme et rajoute un coefficient à la liste. On supposera que le coefficient est de degré supérieur au degré du polynôme passé en paramètre. Le polynôme modifié sera retourné.
2. Écrire une fonction de saisie au clavier d'un polynôme et qui le retourne

3. Écrire une fonction d'affichage d'un polynôme. ´
4. Écrire une fonction de destruction d'un polynôme (fourni en paramètre). ´
5. Écrire une fonction d'addition de deux polynômes( fourni en paramètres avec retour du résultat).
6. Écrire une fonction de multiplication d'un polynôme par un monôme (un monôme est un polynôme dont un seul coefficient est non nul)( fourni en paramètres avec retour du résultat).

## Exo 6

A partir du fichier de base, étudier le moyen de récupérer les données depuis une API externe. Si le script ne trouve pas requests, il vous faudra installer la dépendance depuis le terminal :

```
pip install requests
```

Une fois compris, il vous sera demandé :

- De filtrer les Universités n'ayant pas d'info sur l'Etat ou elles se trouvent
- De trier les Universités par Etats
- D'afficher les informations d'une manière humainement lisible

Vous pouvez essayer de changer le pays et le programme devrait toujours fonctionner (si les universités déclare leur état/région)