

---

## Licence 1 Informatique – 2022/2023 – S2

### Algorithmique et programmation Python

#### TD 4 : Les listes

---

##### Rappel des consignes :

- Créez un dossier pour la séance.
- Regroupez les fichiers Python « .py » dans ce dossier.
- 1 exercice = 1 fichier Python
- Le nom du fichier doit être en relation avec le numéro de l'exercice (« exercice\_1.py », etc.)
- Rédigez en parallèle les algorithmes en pseudo-code dans un fichier word unique.
- A la fin de l'heure, vous regroupez les fichiers du dossier dans une archive (zip, 7z, rar, autre...)
- A rendre (copier) sur votre espace Moodle.
- N'oubliez pas de commenter abondamment votre code source Python.

##### Exercice 1 – Liste de valeurs réelles

Ecrire une application qui effectue les tâches suivantes :

- Demande à l'utilisateur la taille « n » d'une liste de réels.
- Crée une liste vide.
- Effectue la saisie des « n » valeurs et les insère dans la liste
- Calcule et affiche la moyenne des valeurs de la liste
- Trie les valeurs de la liste de manière descendante selon la méthode des Tris à Bulles ([Wikipédia](https://fr.wikipedia.org/wiki/Tri_%C3%A0_bulles)).
- Attention, vous devez programmer explicitement le tri et non pas utiliser la méthode intégrée `sort()`.
- Affiche la liste des valeurs triées.
- Calcule et affiche la médiane de la liste de valeurs (médiane = valeur centrale dans la liste triée si « n » est impair, moyenne des 2 valeurs centrales si « n » est pair)

##### Exercice 2 – Liste de mots (chaînes de caractères)

Ecrire une application qui effectue les tâches suivantes :

- Demander à l'utilisateur d'entrer une liste de mots (de chaînes de caractères). Les doublons sont possibles.
- Tous les mots doivent être convertis en majuscule (voir [https://www.w3schools.com/python/ref\\_string\\_upper.asp](https://www.w3schools.com/python/ref_string_upper.asp)).
- La saisie de la liste doit continuer tant qu'il n'a pas saisi le mot « STOP ».

- d. Affiche la liste des mots après les avoir triés (utilisez le tri par insertion : [https://fr.wikipedia.org/wiki/Tri\\_par\\_insertion](https://fr.wikipedia.org/wiki/Tri_par_insertion)). Attention, (a) « STOP » ne doit pas faire partie de la liste ; (b) vous devez programmer explicitement la procédure de tri.
- e. Afficher le dictionnaire des termes c.-à-d. la liste des mots distincts (sans doublons) (cf. du côté de l'opérateur **in** pour savoir si un terme est déjà dans une liste)
- f. Afficher le dictionnaire et leur nombre d'apparition de chaque terme (par exemple, si l'utilisateur a saisi [« TOTO », « TATA », « TOTO », « TUTU », « TOTO »], votre programme doit afficher TATA = 1, TOTO = 3, TUTU = 1 (cf. du côté de la fonction **count** pour compter un item dans une liste - [https://www.w3schools.com/python/ref\\_list\\_count.asp](https://www.w3schools.com/python/ref_list_count.asp))

### Exercice 3 - Valeur scrabble d'un mot

**Entrée** : une chaîne de caractères représentant un mot

**Sortie** : entier correspondant à la valeur scrabble du mot

**Calcul** : Additionner les valeurs des lettres composant le mot, en utilisant les valeurs officielles des lettres en français ([https://fr.wikipedia.org/wiki/Lettres\\_du\\_Scrabble#Français](https://fr.wikipedia.org/wiki/Lettres_du_Scrabble#Français)) (ex. 'PAPAYE' = 19 points).

**Indications** :

- Il paraît judicieux de convertir en majuscule la chaîne initialement saisie avant de procéder au traitement.
- Il est possible de convertir une chaîne de caractères en liste de lettres avec l'instruction `list()`, (ex. `list('toto')` → `['t','o','t','o']`)
- L'opérateur « `in` » permet de vérifier si une lettre appartient à une chaîne de caractères (ex. `'a' in 'tata'` renvoie `True` ; `'a' in 'toto'` renvoie `False`).