

Numérique et Sciences Informatiques
Chapitre II - Structures de données linéaires
Travaux Dirigés 02

1. Exécuter le logiciel *Thonny*.

2. Dans cette question, nous allons implémenter les Listes grâce à des p -uplets.

- (a) Créer un script intitulé `TD02\les_listes_01.py`.
- (b) Créer une fonction `creer_liste()` qui renvoie une Liste, vide s'il n'y a pas de paramètre (i.e. un tuple vide), composée d'une tête et d'un reste s'il y a deux paramètres dont le deuxième est une Liste et émet une erreur dans les autres cas.
- (c) Créer la liste `(3, (2, (1, ())))` qui sera stockée dans la variable `ma_liste`.
- (d) Créer une fonction `est_vide(liste)` qui teste la vacuité de `liste`.
- (e) Créer une fonction `tete(liste)` qui renvoie la tête de `liste`.
- (f) Créer une fonction `reste(liste)` qui renvoie le reste de `liste`.
- (g) Créer une fonction `longueur(liste)` qui renvoie la longueur de la Liste mise en paramètre, c'est-à-dire le nombre d'éléments qui la composent.

```
1 >>> longueur(ma_liste)
2 3
```

- (h) De quel type est `ma_liste` ?

3. Dans cette question, nous allons implémenter les Listes en programmation orientée objet.

- (a) Ouvrir le script `TD02\les_listes.py` avec *Thonny*.
- (b) Compléter les méthodes de la classe `Liste`.
- (c) Créer la liste `(3. (2. (1. ())))` qui sera stockée dans la variable `ma_liste`.
- (d) De quel type est `ma_liste` ?

4. Dans cette question, nous allons implémenter les Listes par des listes chaînées.

Une liste chaînée est composée de maillons.

Un maillon est composé d'une donnée (ou valeur) et d'un pointeur vers le maillon suivant.

On prendra pour maillon vide le maillon dont la donnée et le suivant valent `None`.

- (a) Ouvrir le fichier `TD02\liste_chaine.py` avec *Thonny*.
- (b) Compléter les classes `Maillon` et `ListeChaine`.
- (c) Créer la liste `3 -> 2 -> 1 -> None` qui sera stockée dans la variable `ma_liste`.
- (d) De quel type est `ma_liste` ?