

Bases de la programmation impérative

Ensimag 1A

1. Listes simplement chaînées

On se propose d'implémenter des listes simplement chaînées.

On utilise pour ce faire deux types différents.

Le type `Cellule` :

```
1 class Cellule:
2     """
3     une cellule d'une liste. Contient une valeur
4     et un pointeur vers la cellule suivante.
5     """
6     # pylint: disable=too-few-public-methods
7     def __init__(self, valeur, suivant=None):
8         self.valeur = valeur
9         self.suivant = suivant
```

Chaque `Cellule` contient un des éléments de la liste.

Enfin, un type `Liste` :

```
1 class Liste:
2     """
3     une liste simplement chainee.
4     Contient un pointeur sur la cellule en tete de liste
5     et un autre sur la queue de liste.
6     """
7     def __init__(self):
8         self.tete = None
9         self.queue = None
```

La figure 1 illustre une liste `l` contenant les éléments 5,3,2,4.

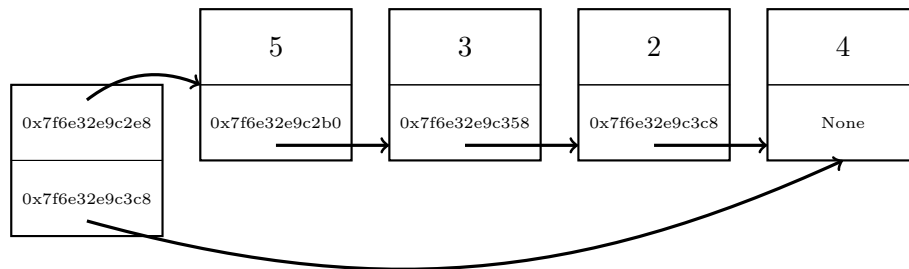


FIGURE 1 – liste `l`

1.1. D'après le dessin, quelle est l'adresse de chaque cellule ?

1.2. Où est `l` sur le dessin ? Comment accéder à la valeur 3 à partir de la variable `l` ?

- 1.3. Écrire une méthode `ajouter_en_tete` permettant l'ajout d'une nouvelle cellule contenant une valeur donnée en tête de liste.
- 1.4. Écrire un générateur (méthode `cellules`) permettant d'itérer sur toutes les cellules.
- 1.5. Implémentez la méthode suivante :

```
1     def recherche(self, valeur):
2         """
3         Renvoie - la premiere cellule contenant la valeur donnee,
4                 - None si aucune cellule ne contient la valeur
5                 donnee.
6         """
```

- 1.6. Implémentez `__str__` pour le type `Liste`.
- 1.7. Implémentez l'ajout d'un élément en queue de liste.
- 1.8. Implémentez la méthode suivante :

```
1     def supprimer(self, valeur):
2         """
3         enleve la premiere cellule contenant la valeur donnee.
4         ne fait rien si la valeur n'est pas dans la liste.
5         """
```

- 1.9. Qu'affiche le code suivant sur la liste (10, 9) :

```
1     for cellule in l.cellules():
2         print(cellule.valeur)
3         cellule.suivant = None
```

Que penser de l'état de la liste après son exécution ?