

Le type abstrait « liste »

« Le vocabulaire de python est très trompeur ! »

Le type « list » en python

- Il s'agit en fait du type abstrait « tableau » :

1	1	2	3	5	8	13
---	---	---	---	---	---	----

- Ce type s'appuie sur un ensemble de cases mémoire contiguës et ordonnées.
- Conséquence : il est facile d'ajouter ou supprimer un élément à la fin du tableau.

Ajouter un élément au milieu ou au début d'un tableau

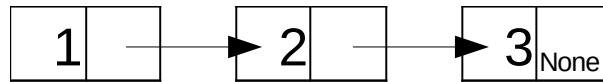
- Que faut-il faire pour ajouter un élément au début d'un tableau (= « List » de python) ?
 - Il faut décaler tous les éléments vers la droite pour faire la place au premier.
- Quel est son coût ?
 - Coût : longueur de la liste.

Il existe donc un autre type abstrait : la liste chaînée.

Le type abstrait de données (TAD)

« liste chaînée »

- L'objectif est de stocker une séquence de valeurs, par exemple des entiers.



- La structure de base est la cellule :



- Le premier élément est la valeur à stocker,
- Le second est l'adresse de la cellule suivante.

Opérations de base sur les listes chaînées

- Créer la liste : CreerListe()
- Insérer un élément x comme $i^{\text{ème}}$ élément de la liste : Insérer (x,i)
- Supprimer le $i^{\text{ème}}$ élément de la liste : Supprimer(i)
- Donner la longueur de la liste : LongueurListe()
- Renvoyer le $i^{\text{ème}}$ élément de la liste : Element(i)

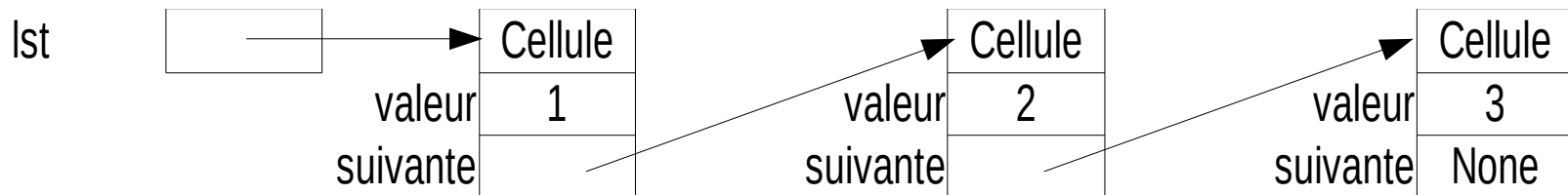
Représentation d'une liste

- Si le code pour créer une classe cellule est :

```
class Cellule:  
    """ une cellule d'une liste chainee """  
    def __init__(self, v, s):  
        self.valeur=v  
        self.suivante=s
```

- Le code suivant permet de créer la liste (1,2,3) :

```
MaListe=Cellule(1,Cellule(2,Cellule(3,None)))
```

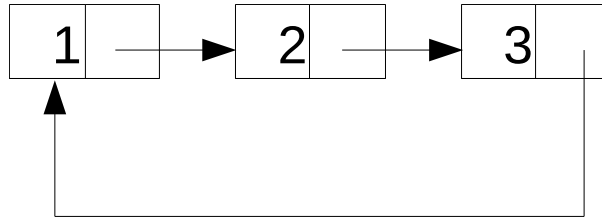


Autres types de listes

- Listes doublement chaînées :



- Listes cycliques :



TP11 – Différentes implantations et interfaces pour le TAD Liste