

Listes

Définition de la structure abstraite **Liste**.

Structures abstraites

On veut pouvoir raisonner sur un algorithme de façon **abstraite** sans avoir à se soucier de la façon dont cet algorithme sera réalisé - on dit **implémenté**.

On définit des structures en précisant les propriétés qu'elles doivent posséder sans se soucier de la façon dont elles seront implémentées dans tel ou tel langage.

Exemple

- On a besoin d'une structure TANT QUE ... RÉPÉTER On définit ce qu'on entend par une telle structure, comment elle fonctionne.
- On peut écrire un algorithme ou du pseudo-code avec cette structure.
- Ce pseudo-code pourra être traduit dans n'importe quel langage dans lequel la structure a été implémentée.

Définition des listes

C'est une structure permettant de regrouper des données et dont l'accès est **séquentiel**.

- Les données stockées dans la liste sont rangées dans un certain ordre.
- On appelle **indice** ou **rang** leur numéro d'ordre qui commence en général par 0.
- La liste a une **taille variable**. On peut lui ajouter ou lui retirer des éléments.
- Par **accès séquentiel** on veut dire que pour atteindre l'élément de rang 10, il faut parcourir avant les éléments de rang 0 à 9.

Quand on voudra parler de **façon abstraite** d'une liste contenant les éléments 4, 9, 17, 3 on pourra la noter 4:9:17:3.



le type `list` de Python implémente un tableau, pas une liste ! En première année on étudie la structure de **tableaux**. Cette structure est contraignante :

- contient un nombre d'items fixe, tous de même type,
- accès direct à un item quand on connaît son indice.

Python a un type `list` qui n'implémente pas une liste mais un tableau dynamique - peut changer de taille mais cela peut occasionner une perte de performances.

Interface d'une liste

Interface : moyens d'interagir avec la structure.

méthode	description
CREER_LIST_VIDE()	renvoie une liste vide
LONGUEUR(L)	renvoie le nombre d'éléments de L
RECHERCHER(L, e)	renvoie l'indice de la première occurrence de e dans L (1)
LIRE(L, i)	renvoie la valeur de l'élément d'indice i dans L <i>Précondition</i> : $0 \leq i < \text{LONGUEUR}(L)$
INSERER(L, e, i)	insère l'élément e à l'indice i dans L (2) <i>Précondition</i> : $0 \leq i \leq \text{LONGUEUR}(L)$
SUPPRIMER(L, i)	supprime l'élément d'indice i dans L (2) <i>Précondition</i> : $0 \leq i < \text{LONGUEUR}(L)$
MODIFIER(L, e, i)	remplace par e l'élément d'indice i dans L (2) <i>Précondition</i> : $0 \leq i < \text{LONGUEUR}(L)$

(1) Que faire si e n'est pas dans L ?

(2) Faut-il modifier L ou créer une nouvelle liste contenant la modification ?