

# Interface et Implémentation

10 novembre 2021

Les quatre méthodes qui ont été définies dans la classe `list` en Python : `len`, `pop`, `insert` sont ce que l'on appelle une implémentation de la structure de donnée tableau.

## Définition .1.

L'implémentation d'une structure de données ou d'un algorithme est une mise en œuvre pratique dans un langage de programmation.

Il existe de nombreux langages de programmation qui implémentent le type abstrait *tableau*, nous avons vu l'année dernière les différences d'implémentation entre les `list` de Python et les `Array` de javascript.

Cependant, on retrouve des méthodes similaires qui sont ce que l'on appelle l'interface de la structure de données *tableau* :

1. "Insérer" : ajoute un élément dans le tableau à l'index souhaité. `ajout(index,élément)`
2. "Retirer" : retire un élément dans le tableau à l'index souhaité. `suppr(index)`
3. "Le tableau est-il vide" : Renvoie *Vrai* si le tableau est vide et faux sinon. `est_vide()`
- 4.
5. "Nombre d'éléments dans le tableau" : Renvoie le nombre d'éléments dans le tableau. `longueur()`

## Définition .2.

L'interface d'une structure de données est la *spécification* des méthodes pouvant être appliquées sur cette structure de données.

Cela permet d'expliquer l'intérêt du pseudo-langage (ou pseudo-code). L'intérêt de définir des structures de données avec une interface commune est de pouvoir écrire des algorithmes sur le papier en utilisant l'interface définie.

On utilise alors un pseudo-langage plus ou moins proche de la langue naturelle qui pourra être implémenté dans tous les langages de programmation ayant défini la structure de données.

## ■ Exercice .1.

Créer une classe *Tableau* qui implémente les quatre méthodes ci-dessus en stockant les données du tableau dans un attribut appelé *data* de type liste. ■