

Une structure de données est une organisation logique des données permettant de simplifier ou d'accélérer leur traitement.

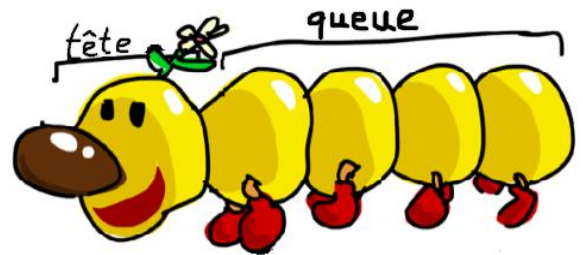
Sources : Laurent Godefroy, Stéphane Grandcolas, Benjamin Monmège, Didier Müller, David Roche, Dominique Laporte, wikipedia

## 1. Liste chaînée

### 1.1. Introduction et rappel historique

Citons cet extrait de Wikipedia.

Le langage Lisp fut inventé par **John McCarthy** en 1958 alors qu'il était au Massachusetts Institute of Technology (MIT). Il publia un article intitulé « Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I » (soit « Fonctions Récursives d'expressions symboliques et leur évaluation par une Machine, partie I ») dans la revue CACM en 1960 ; la partie II ne fut jamais publiée.



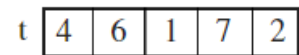
Le premier interpréteur fonctionnait sur un ordinateur IBM 704 et deux instructions de cette machine devinrent les deux opérations primitives de Lisp pour décomposer les listes :

- `car` (contents of address register) : le premier élément de la liste (la tête);
- `cdr` (contents of decrement register) : le reste de la liste (la queue).
- L'opération qui consiste à fabriquer une liste à partir d'un premier élément et d'une liste est notée `cons`.

Lisp, qui est toujours vivant aujourd'hui, s'appuie donc fortement sur les listes, structure de données que nous décrivons maintenant.

La structure de tableau permet de stocker des séquences d'éléments mais n'est pas adaptée à toutes les opérations que l'on pourrait vouloir effectuer sur des séquences. Les tableaux Python permettent par exemple d'insérer ou de supprimer efficacement des éléments à la fin d'un tableau, avec les méthodes `append` et `pop`, mais se prêtent mal à l'insertion ou la suppression d'un élément à une autre position.

Si par exemple, on veut insérer la valeur 3 à la première position du tableau :



On peut facilement le faire en utilisant la méthode `insert` de la classe `list` des tableaux en Python :

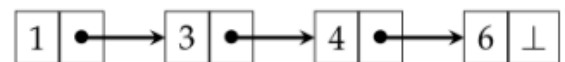
```
insert(0, 3)
```

Pour obtenir le nouveau tableau `t` :



Cette opération est cependant très coûteuse, car elle déplace *tous* les éléments du tableau d'une case vers la droite après avoir agrandi le tableau.

On peut représenter les listes chaînées de la manière suivante :



### 1.2. Primitives

Voici les primitives communément utilisées pour manipuler des listes :

- « `est_vide(lst)` » : renvoie vrai si la pile est vide, faux sinon.
- « `valeur_tête(lst)` » : renvoie la valeur de l'élément de tête
- « `ajoute_tête(lst, valeur)` » : ajoute valeur en tête de la liste (correspond à la fonction `cons` de Lisp) .
- « `enlève_tête(lst)` » : enlève le maillon en tête de la liste et retourne le reste de la liste (la queue).

### 1.3. Applications

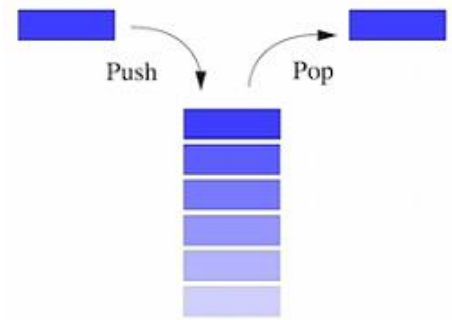
La liste sera adaptée à l'insertion d'éléments dans une liste triée. Par exemple :

- un répertoire de numéros de téléphone d'amis.
- Liste de mots (et définition) d'un dictionnaire.

## 2. Pile

En informatique, une pile (en anglais stack) est une structure de données fondée sur le principe « dernier arrivé, premier sorti » (ou **LIFO** pour Last In, First Out), ce qui veut dire que les derniers éléments ajoutés à la pile seront les premiers à être récupérés.

Le fonctionnement est donc celui d'une pile d'assiettes : on ajoute des assiettes sur la pile, et on les récupère dans l'ordre inverse, en commençant par la dernière ajoutée.



### 2.1. Primitives

Voici les primitives communément utilisées pour manipuler des piles :

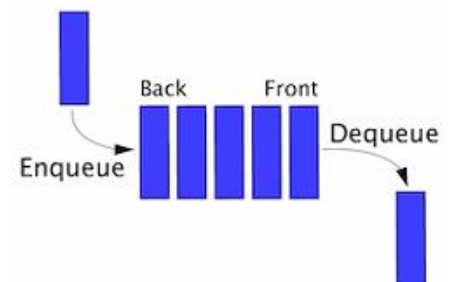
- « empiler » : ajoute un élément sur la pile (en anglais : « Push »).
- « dépiler » : enlève un élément de la pile et le renvoie (en anglais : « Pop »).
- « vide » : renvoie vrai si la pile est vide, faux sinon.
- « taille » : renvoie le nombre d'éléments dans la pile.

### 2.2. Applications

- Dans un navigateur web, une pile sert à mémoriser les pages Web visitées. L'adresse de chaque nouvelle page visitée est empilée et l'utilisateur dépile l'adresse de la page précédente en cliquant le bouton « Afficher la page précédente ».
- L'évaluation des expressions mathématiques en notation post-fixée (ou polonaise inverse) utilise une pile.
- La fonction « Annuler la frappe » (en anglais « Undo ») d'un traitement de texte mémorise les modifications apportées au texte dans une pile.
- Un algorithme de recherche en profondeur utilise une pile pour mémoriser les nœuds visités.
- Les algorithmes récursifs admis par certains langages (LISP, Algol, Pascal, C, etc.) utilisent implicitement une pile d'appel. Dans un langage non récursif (FORTRAN par exemple), on peut donc toujours simuler la récursion en créant les primitives de gestion d'une pile.

## 3. File

Une file (en anglais queue) est une structure de données basée sur le principe « Premier entré, premier sorti », en anglais **FIFO** (First In, First Out), ce qui veut dire que les premiers éléments ajoutés à la file seront les premiers à être récupérés. Le fonctionnement ressemble à une file d'attente : les premières personnes à arriver sont les premières personnes à sortir de la file.



### 3.1. Primitives

Voici les primitives communément utilisées pour manipuler des files :

- « enfile » : ajoute un élément dans la file (en anglais : « enqueue »).
- « défile » : renvoie le prochain élément de la file, et le retire de la file (en anglais : « dequeue »).
- « vide » : renvoie « vrai » si la file est vide, « faux » sinon.
- « taille » : renvoie le nombre d'éléments dans la file.

### 3.2. Applications

- En général, on utilise des files pour mémoriser temporairement des transactions qui doivent attendre pour être traitées.
- Les serveurs d'impression, qui doivent traiter les requêtes dans l'ordre dans lequel elles arrivent, et les insèrent dans une file d'attente (ou une queue).
- Certains moteurs multitâches, dans un système d'exploitation, qui doivent accorder du temps-machine à chaque tâche, sans n'en privilégier aucune.
- Un algorithme de parcours en largeur utilise une file pour mémoriser les nœuds visités.