

Piles et files

Un pan du numérique consiste à enregistrer des données puis à aller les chercher lorsqu'on en a besoin. L'enregistrement des données pourrait se concevoir à la base comme quelque chose de peu important, on obtiendrait alors un fouillis dans lequel les chances de retrouver telle ou telle donnée à un instant précis pourrait facilement tourner à la recherche d'une aiguille dans une botte de foin. Les piles et les files sont deux moyens très courants d'enregistrer des données ; ces deux moyens ont chacun leurs avantages, leurs contraintes. Le choix du programmeur d'utiliser pile, file ou tout autre type de rangement pour enregistrer des données relève de l'objectif à atteindre en termes de traitement des données considérées.

L'illustration simple que nous allons donner d'une pile et d'une file réside dans les deux photos avec à gauche une file de canards et à droite une pile de cailloux.



La différence qu'il y a entre une pile et une file réside dans l'observation du **premier sortie** (First Out) qui pour une pile à droite sera le dernier entré (Last In) et pour une file à gauche sera au contraire le premier entrée (First In).



Sources : Adresses internet des auteurs des deux photos libres de droit extraites du site Pixabay
(à gauche) https://pixabay.com/fr/users/couleur-1195798/?utm_source=link-attribution&utm_medium=referral&utm_campaign=image&utm_content=1401225
(à droite) https://pixabay.com/fr/users/mike_68-10359383/?utm_source=link-attribution&utm_medium=referral&utm_campaign=image&utm_content=6754295

La pile

Revenons à notre illustration de la pile, avec cet empilement de cailloux où une des premières choses que l'on perçoit est que l'accès à n'importe quel élément de la pile ne peut se faire qu'en manipulant ceux qui sont au-dessus. Si l'on veut par exemple prendre le caillou situé au niveau de la flèche bleue, il nous faudra déplacer successivement le premier caillou (situé tout en haut de la pile), puis le second, etc. jusqu'au cinquième pour accéder au caillou voulu. Le sommet de la pile situé au niveau de la flèche verte est à la fois le dernier à avoir été placé sur la pile et le premier à pouvoir en être extrait, on retient qu'une pile fonctionne en **mode LIFO** (Last In First Out).



Afin d'avoir un exemple informatique concret de la notion de Pile, nous allons travailler à partir d'une phrase de Kazuo Ishiguro prix Nobel de littérature 2017 :

« *Je songe à ma pile de vieux livres de poche de pages tremblotantes, comme si elles avaient autrefois fait partie de la mer.* ».

Source : Site Wikipédia CC BY-SA 4.0

Nous vous proposons de copier la phrase de Kasuo Ishiguro sur une page vierge et de suivre la démarche suivante en 6 étapes :

1^{ère} étape : Coller la phrase. Celle-ci est en Arial taille 11.

2^{ème} étape : Sélectionner la phrase puis modifier sa taille et prendre la taille 16.

3^{ème} étape : Sélectionner à nouveau la phrase puis modifier la police de caractère en prenant par exemple *Old English Text MT*.

4^{ème} étape : Couper la partie de phrase : «*comme si elles avaient autrefois fait partie de la mer*».

5^{ème} étape : Coller la partie de phrase précitée.

6^{ème} étape : Sélectionner la phrase et modifier en bleu la couleur des caractères.

Les résultats des opérations successives qui vous venez de réaliser au cours de ces 6 étapes sont enregistrées sous la forme d'une pile que nous pouvons représenter sous la forme du tableau à une seule colonne ci-dessous :

<i>« Je songe à ma pile de vieux livres de poche de pages tremblotantes, comme si elles avaient autrefois fait partie de la mer. »</i>
<i>« Je songe à ma pile de vieux livres de poche de pages tremblotantes, comme si elles avaient autrefois fait partie de la mer. »</i>
<i>« Je songe à ma pile de vieux livres de poche de pages tremblotantes, »</i>
<i>« Je songe à ma pile de vieux livres de poche de pages tremblotantes, comme si elles avaient autrefois fait partie de la mer. »</i>
<i>« Je songe à ma pile de vieux livres de poche de pages tremblotantes, comme si elles avaient autrefois fait partie de la mer. »</i>
<i>« Je songe à ma pile de vieux livres de pages tremblotantes comme si elles avaient autrefois fait partie de la mer. »</i>

Le seul moyen que vous avez à présent de parcourir la pile est de faire soit un (ou plusieurs) Ctrl+Z soit un (ou plusieurs) Ctrl + Y. Seul l'élément placé au sommet de la pile apparaît à l'écran. Vous pouvez accéder à un élément donné de la pile uniquement en dépilant (Ctrl+Z) successivement tous les éléments qui sont au-dessus de lui. Une fois un élément atteint, vous pouvez empiler (Ctrl+Y) les éléments dépilés précédemment car ils ont été conservés en mémoire.

Les différents éléments importants d'une pile sont les suivants : **son sommet – dépiler l'élément au sommet – empiler un élément en le plaçant au sommet**. Le dernier élément placé sur la pile sera le premier à en sortir. On retient qu'une pile fonctionne en **mode LIFO** (Last In First Out).

La file

Nous allons observer notre illustration des petits canards en file indienne. Elle est intéressante à plusieurs titres d'une part les canetons ont naturellement l'habitude d'évoluer en file indienne derrière leur mère d'autre part – et cela va nous intéresser davantage ici – les canards de cette image ne peuvent pas se doubler.



On a ainsi un canard en tête de file (signalé par la grosse flèche jaune) et un canard en bout de file (signalé par la petite flèche jaune). Le canard en tête de file est le premier entré et il sera aussi le premier sorti. **Une file fonctionne en mode FIFO** (First In First Out).

Si l'on souhaite atteindre le 3^{ème} petit canard celui qui a une casquette bleue et qui est situé à hauteur de la flèche bleue, il nous faudra faire sortir/défiler le premier canard, faire sortir/défiler le second canard. Notre 3^{ème} canard deviendra alors le 1^{er} canard et pourra sortir de la file. Autrement dit, pour accéder à un élément d'une file – à moins qu'il ne soit le premier de la file – il nous faut défiler dans l'ordre ceux qui le précèdent.



Nous allons fournir un programme en Python qui permet d'obtenir l'élément de rang donné d'une file. Pour ce faire, vous pouvez télécharger l'environnement Spyder disponible à l'adresse suivante : <https://www.spyder-ide.org/> .

Le logiciel Python permet la construction de listes qui ne correspondent ni à des piles, ni à des listes pour la simple et bonne raison que l'on peut directement avoir accès à un terme de rang donné d'une liste alors que pour une pile ou une liste nous avons vu qu'accéder à un terme de rang donné nécessite une manipulation.

Reprenons nos canards que nous nommons C1 , C2 , ..., C9 , C10 où C1 est le canard en tête de file et C10 est le canard en bout de file.

Construisons la liste *canards* :

```
canards = [ "C1", "C2", "C3", "C4", "C5", "C6", "C7", "C8", "C9", "C10"].
```

Les rangs de nos canards vont ainsi du numéro 1 au numéro 10.

Si l'on souhaite avoir le canard de rang 3 - comme nous l'avons vu précédemment - , il nous faut défiler C1 et C2.

Plus généralement si l'on souhaite avoir le canard de rang *i*, soit le rang est 1 et dans ce cas, le canard cherché est C1, soit le rang cherché est différent de 1 et il nous faut défiler C1, ..., C(i-1).

Voici in extenso les lignes d'instructions en Python, lignes que vous pouvez recopier dans la ligne d'édition du logiciel Spyder.

```
# On définit la file de canards
canards=["C1","C2","C3","C4","C5","C6","C7","C8","C9","C10"]
def canardderang(i):
    file=canards
    # On fait le test pour savoir si le rang est = à 1
    if i==1 :
        resultat=file.pop(0)
    # Si le rang est différent de 1
    else :
        # On définit une variable affectée de la valeur de la variable i
        j=i
        while j!=1:
            # On défile la tête de la liste tant que j est différent de 1
            file.pop(0)
            j=j-1
        resultat=file.pop(0)
    print (resultat)
# On se propose de recherche le canard de rang 3
canardderang(3)
```

Les différents éléments importants d'une file sont les suivants : **la tête de la file – défiler la tête de file – le bout de la file - enfiler un élément en le plaçant en bout de file.** On retient qu'une file fonctionne en **mode FIFO**.

Conclusion

Dans cet article, nous avons mis en évidence – de façon sous-jacente - la nécessité de développer une sorte de pensée numérique afin de permettre d'évoluer dans des champs relativement abstraits. Nous espérons que les petits exemples concrets développés pour présenter la notion de pile puis celle de file vous auront aidé à percevoir ces deux notions bien différentes, notions qu'il convient de ne pas confondre.

Jean-Alain Roddier
IA-IPR de mathématiques