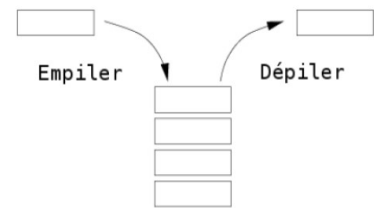


En informatique, une **pile** (en anglais stack) est une structure de données fondée sur le principe « **dernier arrivé, premier sorti** » (ou **LIFO** pour **Last In, First Out**), ce qui veut dire que les derniers éléments ajoutés à la pile seront les premiers à être récupérés.

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



Voici quelques exemples d'usage courant d'une pile:

- 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.
(classique : $(10 + 5) \times 3$; polonaise : $10\ 5\ +\ 3\ \times$)
- La fonction «Annuler la frappe» (en anglais «Undo») d'un traitement de texte mémorise les modifications apportées au texte dans une pile.
- La pile d'exécution d'un algorithme récursif

Opérations de bases :

- **CREER_PILE()** qui retourne un objet de type pile : *La pile existe et elle est vide*
- **EMPLER(P,e)** : L'élément e est inséré au sommet de la pile P.
- **DEPILER(P)** qui retourne l'élément situé au sommet de la pile et le supprime de la pile
- **EST_VIDE(P)** qui retourne un objet de type Booléen : *Retourne Vrai si la pile est vide et retourne Faux sinon.*

Exercice 1 : Donner l'état de la pile créée à la fin de l'exécution du programme:

```
P=CREER_PILE()
EMPLER(P,3)
EMPLER(P,2)
N=DEPILER(P)
EMPLER(P,5)
EMPLER(P,N)
EMPLER(P,9)
```



Exercice 2 : Donner le contenu de P1 et P2 à la fin

```
P1=CREER_PILE()
P2=CREER_PILE()
Pour i allant de 1 à 5 :
    EMLER(P1,i)
    EMLER(P2,i+1)
Si i est pair alors
    N=DEPILER(P1)
    EMLER(P2, N)
```



Exercice 3 : Écrire l'algorithme de la fonction sommet(P) qui renvoie le sommet de la pile (sans le supprimer) s'il existe.

Fonction sommet(P) :

Exercice 4 : a) Écrire l'algorithme de la fonction taille(P) qui retourne la taille de la pile P.

Fonction taille(P)

b) écrire une version où la pile n'est pas modifiée à la fin

.....

Exercice 5 : on se donne une pile P1 contenant des entiers positifs.

1. Ecrire un algorithme pour déplacer les entiers de P1 dans une pile P2 de façon à avoir dans P2 tous les nombres pairs en dessous des nombres impairs
2. Ecrire un algorithme pour copier dans P2 les nombres pairs contenus dans P1. Le contenu de P1 après exécution de l'algorithme doit être identique à celui avant exécution. Les nombres pairs dans P2 doivent apparaître dans l'ordre où ils sont dans P1

Implémentations : notebook : implementations_piles.ipynb