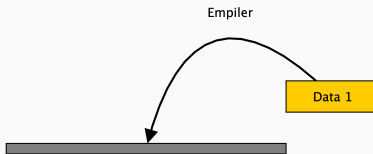


Structures de données : Pile

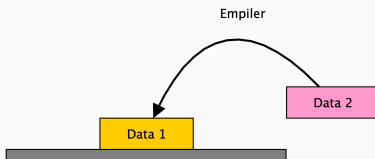
Chapitre 12

NSI2

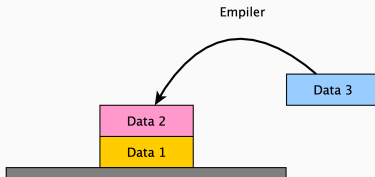
9 novembre 2021



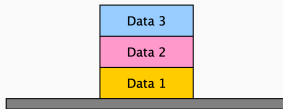
On part d'une structure vide...



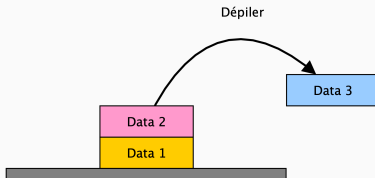
Sur laquelle on peut empiler...



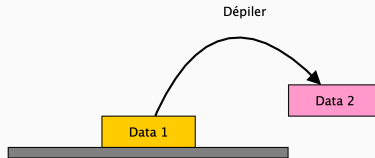
Au fur et à mesure.



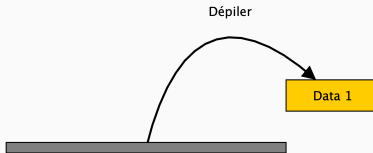
Seul le dernier élément est accessible.



On peut dépiler.



Les dernières valeurs empilées sont les premières dépilées.



On parle de LIFO (*Last In First Out*).

- Lors de la navigation sur le web, on peut considérer que les liens sont sauvegardés sur une pile par le navigateur.
- De même pour les logiciels qui utilisent la fonction **annuler** (le fameux **CTRL+Z**).

- Lors de la navigation sur le web, on peut considérer que les liens sont sauvegardés sur une pile par le navigateur.
- De même pour les logiciels qui utilisent la fonction **annuler** (le fameux **CTRL+Z**).
- Lors d'appels de fonctions, les états-mémoires sont sauvegardés sur une pile (notamment lors d'appels récursifs, on l'a déjà vu).

- Lors de la navigation sur le web, on peut considérer que les liens sont sauvegardés sur une pile par le navigateur.
- De même pour les logiciels qui utilisent la fonction **annuler** (le fameux **CTRL+Z**).
- Lors d'appels de fonctions, les états-mémoires sont sauvegardés sur une pile (notamment lors d'appels récursifs, on l'a déjà vu).
- Des piles sont utilisées dans divers algorithmes, notamment pour parcourir un arbre « en profondeur ».

- Lors de la navigation sur le web, on peut considérer que les liens sont sauvegardés sur une pile par le navigateur.
- De même pour les logiciels qui utilisent la fonction **annuler** (le fameux **CTRL+Z**).
- Lors d'appels de fonctions, les états-mémoires sont sauvegardés sur une pile (notamment lors d'appels récursifs, on l'a déjà vu).
- Des piles sont utilisées dans divers algorithmes, notamment pour parcourir un arbre « en profondeur ».
- D'autres applications sont données en exercice.

Interface de la structure de données pile

Elle est très simple!

Interface de la structure de données pile

Elle est très simple!

- *pile_vide()* crée une pile vide

Interface de la structure de données pile

Elle est très simple!

- *pile_vide()* crée une pile vide
- *empiler(pile,valeur)* empile la valeur sur la pile

Interface de la structure de données pile

Elle est très simple!

- *pile_vide()* crée une pile vide
- *empiler(pile,valeur)* empile la valeur sur la pile
- *depiler(pile)* renvoie la valeur sur la pile et l'enlève de la pile

Interface de la structure de données pile

Elle est très simple!

- *pile_vide()* crée une pile vide
- *empiler(pile,valeur)* empile la valeur sur la pile
- *depiler(pile)* renvoie la valeur sur la pile et l'enlève de la pile
- *est_vide()* indique si la pile est vide ou non

- Simple liste python..

- Simple liste python..
- Liste encapsulée dans un objet.

- Simple liste python..
- Liste encapsulée dans un objet.
- Listes imbriquées :

```
[ ]  
[1] # on a empilé 1  
[2, [1]] # puis 2  
[3, [2, [1]]] # puis 3  
...
```

Implémentation objet

```
class Stack:
    def __init__(self):
        """ Creates an empty stack """
        self.content = []

    def is_empty(self) -> bool:
        """ Indicates whether the stack's empty or not """
        return self.content == []
```

Implémentation objet (suite et fin)

```
def push(self, value):  
    """ Pushes the value on the top of the stack """  
    self.content.append(value)  
  
def pop(self):  
    """ Retrieves the value from the top of the stack """  
    if self.is_empty():  
        raise IndexError('Stack Empty')  
    return self.content.pop()
```