

Piles et files

Lycée Pierre Corneille – MP

2016-2017

Notion de pile (*stack*)

- Structure de donnée variable

Notion de pile (*stack*)

- Structure de donnée variable
- Dernière donnée reçue = première donnée sortie (*Last In, First Out*)

Notion de pile (*stack*)

- Structure de donnée variable
- Dernière donnée reçue = première donnée sortie (*Last In, First Out*)
- Type abstrait

Notion de pile (*stack*)

- Structure de donnée variable
- Dernière donnée reçue = première donnée sortie (*Last In, First Out*)
- Type abstrait
- Implémentation en Python avec des listes

- Traitements de texte : Historique des modifications (Ctrl-Z)

Utilisation de piles

- Traitements de texte : Historique des modifications (Ctrl-Z)
- Navigateurs : Historique de navigation

Utilisation de piles

- Traitements de texte : Historique des modifications (Ctrl-Z)
- Navigateurs : Historique de navigation
- Calculatrice HP 15C : Notation polonaise inversée

Utilisation de piles

- Traitements de texte : Historique des modifications (Ctrl-Z)
- Navigateurs : Historique de navigation
- Calculatrice HP 15C : Notation polonaise inversée
- Algorithmes récursifs : Pile d'appels

Opérations sur les piles

Créer une pile (vide)

```
def creer_pile():  
    return []
```

Opérations sur les piles

Ajouter un élément au sommet de la pile : *empiler* - *push*

```
def empiler(x, pile):  
    pile.append(x)
```

Opérations sur les piles

Dénombrer la pile

```
def hauteur(pile):  
    return len(pile)
```

Opérations sur les piles

Dénombrer la pile

```
def hauteur(pile):  
    return len(pile)
```

La pile est-elle vide ?

```
def estVide(pile):  
    return hauteur(pile)==0
```

Opérations sur les piles

Retirer le dernier élément ajouté : *dépiler* - *pop*

```
def depiler(pile):  
    return pile.pop()
```

Opérations sur les piles

Retirer le dernier élément ajouté : *dépiler* - *pop*

```
def depiler(pile):  
    return pile.pop()
```

Erreur si la pile est vide !

Opérations sur les piles

Accéder au dernier élément ajouté : *peek*

```
def sommet(pile):  
    return pile[-1]
```

Opérations sur les piles

Accéder au dernier élément ajouté : *peek*

```
def sommet(pile):  
    return pile[-1]
```

Erreur si la pile est vide !

Notion de file (*queue*)

- Structure de donnée variable

Notion de file (*queue*)

- Structure de donnée variable
- Premier arrivé = premier servi (*First In, First Out*)

Notion de file (*queue*)

- Structure de donnée variable
- Premier arrivé = premier servi (*First In, First Out*)
- Type abstrait

Notion de file (*queue*)

- Structure de donnée variable
- Premier arrivé = premier servi (*First In, First Out*)
- Type abstrait
- Implémentation en Python avec des listes

- Accès aux ressources d'un réseau

Utilisation de files

- Accès aux ressources d'un réseau
- Calcul de moyenne mobile

Utilisation de files

- Accès aux ressources d'un réseau
- Calcul de moyenne mobile
- Modélisation probabiliste des files d'attente

Opérations sur les files

Créer une file (vide)

```
def creer_file():  
    return []
```

Opérations sur les files

Placer un élément en file d'attente (*enqueue*)

```
def enfiler(x, file):  
    file.append(x)
```

Opérations sur les files

Dénombrer la file

```
def longueur(file):  
    return len(file)
```

Opérations sur les files

Dénombrer la file

```
def longueur(file):  
    return len(file)
```

La file est-elle vide ?

```
def estVide(file):  
    return len(file)==0
```

Opérations sur les files

Faire sortir le premier élément de la file (*dequeue*)

```
def defiler(file):  
    return file.pop(0)
```

Opérations sur les files

Faire sortir le premier élément de la file (*dequeue*)

```
def defiler(file):  
    return file.pop(0)
```

Erreur si la pile est vide!