

# Les files

## 1 Les files

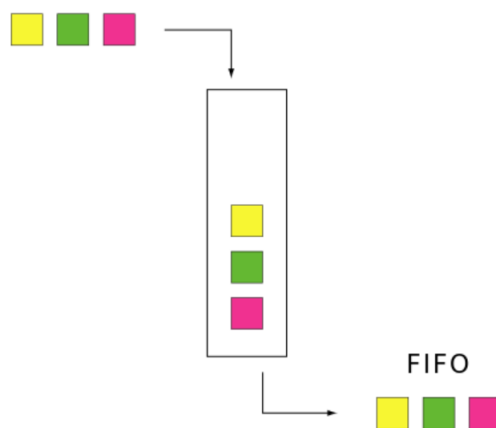


### 1.1 Définition

Dans une file, les éléments sont placés les uns à cotés des autres comme dans une pile, à la différence que seul l'on sort les éléments du plus ancien vers le plus récent.

Cela correspond à ce qui se passe dans une file d'attente

Les files sont basées sur le principe FIFO (First In First Out : le premier qui est rentré sera le premier à sortir. Ici aussi, on retrouve souvent ce principe FIFO en informatique.



## 2 interface des files

On note  $\text{File}[T]$  le type des files contenant des éléments de type  $T$ .

fonction	description
<code>creer_file() -&gt; File[T]</code>	On peut créer une file vide. ( <code>creer_file</code> )
<code>est_vide(p :File[T])-&gt; bool</code>	on peut savoir si une file est vide ( <code>file_vide</code> )
<code>enfiler(p :File[T], e :T) -&gt; None</code>	on peut enfiler un nouvel élément à l'arrière de la file.
<code>defiler(p :File[T]) -&gt; T</code>	on peut récupérer l'élément à l'avant de la pile.

### Exemple

Soit une file  $F$  composée des éléments suivants : 1, 12, 24, 17, 21 et 72 (le premier élément rentré dans la file est 72 ; le dernier élément rentré dans la file est 1). Quel est l'effet de l'instruction `ajout(F,25)`

## 3 Utilisation des files

Dans le domaine informatique, on retrouve par exemple les files dans les files d'impression où le premier document envoyé à l'imprimante sera le premier document à être imprimé.

## 4 Les différentes structures pour implémenter les files

On va ici chercher à "traduire" ces algorithmes dans un langage compréhensible pour un ordinateur (Python, Java, C,...) : on dit alors que l'on implémente un algorithme.

L'implémentation d'un type de données dépend du langage de programmation. Il faut, quel que soit le langage utilisé, que le programmeur retrouve les fonctions qui ont été définies pour le type abstrait (pour les listes, les piles et les files cela correspond aux fonctions définies ci-dessus). Certains types abstraits ne sont pas forcément implémentés dans un langage donné, si le programmeur veut utiliser ce type abstrait, il faudra qu'il le programme par lui-même en utilisant les "outils" fournis par son langage de programmation.

Pour implémenter les listes (ou les piles et les files), beaucoup de langages de programmation utilisent 2 structures : les tableaux et les listes chaînées.



## A vous de jouer ! .1

Construction d'une file avec deux piles On considère ici que l'on dispose de l'implémentation d'une pile : `Pile(T)`, et des fonctions `creer_pile()`, `est_vide(P)`, `empiler(P,e)`, `depiler(P)`.

On considère deux piles P1 et P2

1. Proposer une fonction , en pseudo langage, `enfiler(e)` :

```
Fonction enfiler(e):  
    '''  
    e est un element. La fonction utilise P1 et P2 pour enfiler  
    '''
```

2. Proposer une fonction `defiler()` en pseudo-langage :

```
Fonction defiler(e):  
    '''  
    e est un element. La fonction utilise P1 et P2 pour défiler.  
    '''
```