

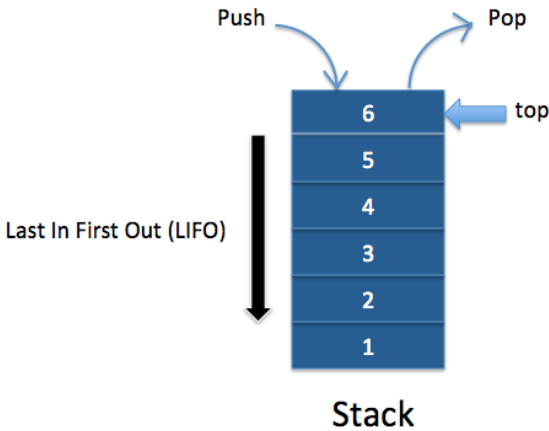
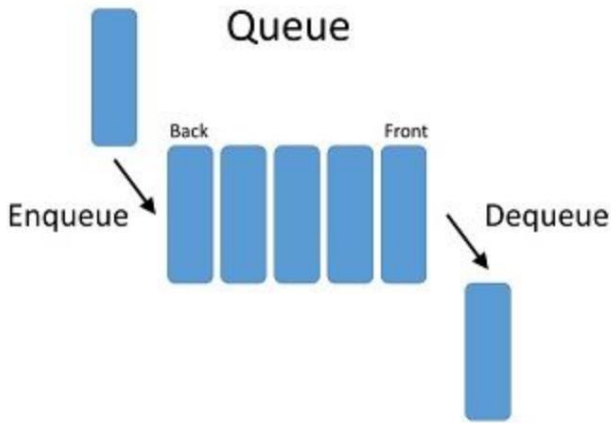


T NSI Structures de données: Piles & Files

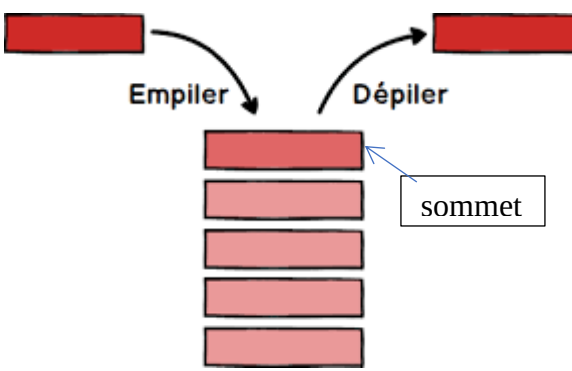
1. Définitions

Piles (en anglais <i>stack</i>)	Files (en anglais <i>Queue</i>)
	
LIFO : Last In First Out	FIFO : First In First Out
	
<ul style="list-style-type: none">• Sert à mémoriser les pages Web visitées par un navigateur• Évaluer des expressions mathématiques en notation post-fixée (polonaise inverse)• Gestion de la mémoire par les langage de programmation (pile et tas)	<ul style="list-style-type: none">• Mémoriser temporairement des transactions qui doivent attendre pour être traitées• Serveurs d'impression : traiter les requêtes dans l'ordre dans lequel elles arrivent• Moteurs multitâches, dans un système d'exploitation, qui doivent accorder du temps-machine à chaque tâche, sans en privilégier aucune

T NSI Structures de données: Piles & Files

2. Les Piles Mode LIFO (Last in First out)

2.1. Fonctions/méthodes

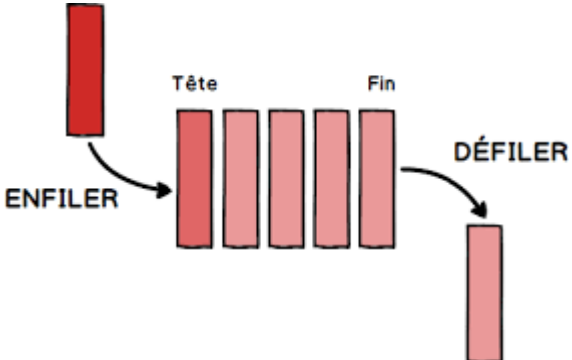
	«empiler» : ajoute un élément sur la pile. Terme anglais correspondant : « Push ».
	«dépiler» : enlève un élément de la pile et le renvoie. En anglais : « Pop».
	«est_Vide» : renvoie vrai si la pile est vide, faux sinon
	«taille» : renvoie le nombre d'éléments dans la pile

2.2. Une implémentation utilisant les listes « pythonesques »

<pre>class Pile: def __init__(self): self.valeurs = [] def empiler(self, valeur): self.valeurs.append(valeur) def depiler(self): if self.valeurs: return self.valeurs.pop()</pre> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>liste.pop() : enlève et renvoie le dernier élément de la liste</p> </div> <pre> def estVide(self): return self.valeurs == [] def taille(self): return len(self.valeurs) def __str__(self): ch = '' for x in self.valeurs: ch = " \\t" + str(x) + "\\t " + "\\n" + ch ch = "\\nEtat de la pile:\\n" + ch return ch</pre>	<pre>p = Pile() p.empiler(9) p.empiler(2) p.empiler(5) print(p) print("taille pile : ", p. taille()) p.depiler() p.empiler(7) print(p.estVide()) print(p)</pre>
--	---

3. Les Files Mode FIFO (First In First Out)

3.1. Fonctions/méthodes

	«enfiler» : ajoute un élément sur la file. Terme anglais correspondant : « enqueue ».
	«défiler» : renvoie le prochain élément de la file, et le retire de la file. Terme anglais : « dequeue »
	«est_Vide» : renvoie vrai si la file est vide, faux sinon
	«taille» : renvoie le nombre d'éléments dans la pile

3.2. Une implémentation utilisant les listes « pythonesques »

<pre> class File: def __init__(self): self.valeurs = [] def enfiler(self, valeur): self.valeurs.append(valeur) def defiler(self): if self.valeurs: return self.valeurs.pop(0) def est_Vide(self): return self.valeurs == [] def taille(self): return len(self.valeurs) def __str__(self): ch = '' for x in self.valeurs: ch += str(x) + " " ch = "\nEtat de la file:\n"+ch return ch </pre>	<pre> q = File() q.enfiler(9) q.enfiler(2) q.enfiler(5) print(q) q.defiler() q.enfiler(7) print("La file est-elle vide ? : ", q.est_Vide()) print(q) print("Longueur de la file:", q.taille()) </pre>
---	---