

Ce Tp reprend les exemples traités dans le poly de cours.

1- METHODE POP() DE LA CLASSE LIST DE PYTHON :

La méthode `pop()` prend en paramètre un entier `i`. Cette méthode permet de retirer de la liste l'élément dont l'index est `i` et de le renvoyer. Par défaut l'index `i` comme valeur `-1` ; il correspond donc au dernier élément de la liste.

⇒ Exécuter dans la console les lignes données ci-contre, et vérifier le bon fonctionnement de cette méthode `pop()`.

```
>>> l=[0,1,2,3,4]
>>> l.pop()
4
>>> l.pop(1)
1
>>> l
[0, 2, 3]
```

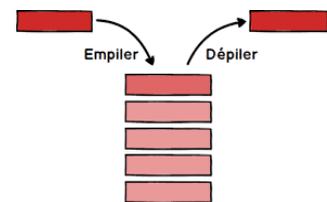
2- IMPLEMENTATION PYTHON D'UNE PILE :

⇒ Télécharger le fichier `pile.py` proposé sur nsibranly.fr.

Il contient le script incomplet de la classe `Pile` qui implémente une structure de donnée de type `Pile`. La méthode `__str__()` qui s'y trouve permet un affichage vertical du contenu de la pile. Le fichier contient aussi un programme principal test.

⇒ Compléter le script de cette classe `Pile` en définissant les méthodes `empiler()`, `estVide()`, `depiler()`, `taille()`, `sommet()`, `vider()` décrites dans le poly de cours.

⇒ Compléter le programme principal pour afficher dans la console les résultats des tests proposés dans le poly de cours. Coller un screenshot dans un fichier `.doc` à enregistrer sous le nom `pileFile.doc`



3- UTILISATION DE CETTE CLASSE POUR TRAITER LE SUJET TYPE BAC (2019) :

Les réponses attendues dans ce paragraphe sont des copies d'écran à coller, toujours dans le fichier `pileFile.doc`.

⇒ Compléter la classe `Pile` en y écrivant le code de la méthode `maxPile()` de la question 3, page 3 du poly de cours. Tester dans la console cette méthode avec la pile `P` donnée ci-contre, et avec comme argument le nombre 4. Copier les screenshots de l'exécution dans la console de `>>> print(P)` et de `>>> P.maxPile(4)`.

⇒ Compléter la classe en écrivant le code de la méthode `retourner()` de la question 4, page 3 du poly de cours. Tester dans la console cette méthode avec la pile donnée ci-contre, en ayant comme argument le nombre 3. Copier les screenshots de l'exécution dans la console de `>>> P.retourner(3)` et de `>>> print(P)`.

⇒ Compléter encore cette classe en écrivant le code de la méthode `triCrepes()` de la question 5, page 4 du poly de cours. Tester dans la console cette méthode, toujours avec la pile donnée ci-dessus. Copier les screenshots de l'exécution dans la console de `>>> P.triCrepes()` et de `>>> print(P)`.

6
3
9
5

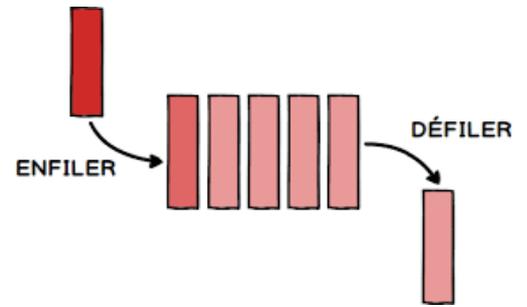
4- IMPLEMENTATION PYTHON D'UNE FILE :

⇒ Télécharger le fichier *file.py* proposé sur nsibranly.fr .

Il contient le script de la fonction *creer_file_vide()* qui renvoie une liste python vide et le script de la fonction *afficher()* qui permet un affichage horizontal du contenu d'une file. Le fichier contient aussi un début de programme principal test.

⇒ Compléter ce fichier en définissant les fonctions *enfiler()*, *estVide()*, *defiler()*, *taille()*, *tete()*, *vider()* décrites dans le poly de cours.

⇒ Compléter le programme principal pour afficher dans la console les résultats des tests proposés dans le poly de cours. Coller un screenshot dans le fichier *pileFile.doc* .



5- UTILISATION DE CETTE CLASSE POUR TRAITER LE SUJET TYPE BAC (2019) :

⇒ Compléter encore le fichier *file.py* en y écrivant le code de la fonction *ajout()* de la question 1 , page 7 du poly de cours. Tester cette fonction en exécutant dans la console 5 fois `>>> ajout(f)` et ensuite `>>> afficher(f)` . Coller un screenshot dans le fichier *pileFile.doc* .

⇒ Compléter le fichier *file.py* en y écrivant le code de la fonction *affich_seq()* de la question 2 , page 7 du poly de cours. Tester cette fonction en exécutant dans la console 5 fois `>>> affich_seq(f)` et ensuite `>>> afficher(f)` . Coller un screenshot dans le fichier *pileFile.doc* .

⇒ Compléter le fichier *file.py* en y écrivant le code de la fonction *tour_de_jeu()* de la question 3 , page 8 du poly de cours. Tester dans la console cette fonction en exécutant `>>> tour_de_jeu(f)` . Coller un screenshot dans le fichier *pileFile.doc* .

```
>>> tour_de_jeu(f)
ECOUTE ...
1 couleur(s) à saisir :
Couleur 1 à saisir : r

ECOUTE ...
2 couleur(s) à saisir :
Couleur 1 à saisir : r
Couleur 2 à saisir : b

ECOUTE ...
3 couleur(s) à saisir :
Couleur 1 à saisir : r
Couleur 2 à saisir : b
Couleur 3 à saisir : b

ECOUTE ...
4 couleur(s) à saisir :
Couleur 1 à saisir :
FIN
```

⇒ Télécharger sur nsibranly.fr le fichier *sons.zip* contenant 8 fichiers au format mp3. Décompresser et enregistrer les 8 fichiers dans votre répertoire de travail.

⇒ Modifier à présent le code de ce fichier *file.py* afin que le résultat de l'exécution de la fonction `>>> tour_de_jeu(f)` provoque à présent non plus un affichage de la séquence de couleur, mais une lecture à voix haute des couleurs.

Vous pourrez utiliser la bibliothèque python-vlc

```
import vlc # pip install python-vlc
```

Pour créer un objet *python-vlc* lié à un fichier, on exécute par exemple :

```
b = vlc.MediaPlayer("bleu.mp3")
```

Pour lire ce fichier, on exécute :

```
b.play()
```

Pour arrêter la lecture, on exécute :

```
b.stop()
```

⇒ Tester l'ensemble

· **UPLOADER** les fichiers *pile.py* , *file.py* et *filePile.doc* sur nsibranly.fr