

~ Contrôle Flash #2 ~

Nom :	Prénom :	Classe :
Remarque :	Note :	

Mise en place :

Partie Théorique : Un stylo, un crayon à papier, une gomme, le sujet et une feuille de brouillon. **PC éteint !**

Partie pratique : PC allumés : Téléchargez le fichier `listes_flash.py` reçu par message NEO. Ensuite, créez un script pour le contrôle.

Vous rendrez le fichier par message NEO et le sujet pour la partie théorique.

Nom du script python à rendre : ***NOM_Prénom_controle_flash_2.py***

Partie Théorique

Répondez sur le sujet.

1. Quelle est la définition d'une interface ?

.....

.....

.....

.....

.....

.....

2. Quel est l'interface d'un tableau ?

.....

.....

.....

.....

.....

.....

3. [Bonus] Comment sait-on qu'une cellule est la dernière d'une liste chaînée ?

.....

.....

.....

Exercice pratique : Créer et afficher une liste chaînée

Voici quatres fonctions, *cellule*, *concatener*, *liste_vide*, *tete* et *queue*, présentées ci-dessous :

```
def cellule(tete, queue):
    """
    Construit une liste chaînée en rajoutant l'élément tête au
    *début* de la liste chaînée queue. La liste queue n'est pas
    modifiée par cette opération.
    """
    return [tete, queue]

def tete(liste):
    """
    Renvoie le premier élément de la liste chaînée passée en
    paramètre.
    """
    return liste[0]

def queue(liste):
    """
    Renvoie la liste chaînée liste privée de son premier élément
    (la liste passée en paramètre n'est cependant pas modifiée
    par cette opération).

    Déclenche une erreur si appelé sur une liste vide.
    """
    return liste[1]

def concatener(l1, l2):
    """
    Renvoie une nouvelle liste chaînée qui sera la concaténation
    de l1 et de l2.

    Les listes l1 et l2 ne sont pas modifiées par cette appel.
    """
    if est_vide(l1):
        return l2
    else:
        return cellule(tete(l1), concatener(queue(l1), l2))

def liste_vide():
    """
    Renvoie une liste vide.
    """
    return None # On décide de représenter la liste vide par None
```

Questions :

- Placez `listes_flash.py` dans le dossier que votre script.
- Au début de votre script, ajoutez la ligne suivante :

```
from listes_flash import *
```

Réalisez le code python pour effectuer les actions suivantes :

1. Créer une liste chaînée `ma_liste` contenant les valeurs 5, 4 et 1. Il faudra donc créer 3 cellules.
2. Afficher la première cellule de `ma_liste`.
3. Afficher la tête de la première cellule de `ma_liste`.
4. Afficher la tête de la seconde cellule de `ma_liste`.
5. Afficher la queue de la troisième cellule de `ma_liste`.