



Pythonismes

Python comme un Framework

Dans ce TD, vous allez observer et apprendre quelques caractéristiques intrinsèques à Python, appelées *pythonismes*. Ces exercices s'inspirent du livre *Fluent Python* aux éditions O'Reilly.

Exercice 1. Un jeu de cartes

```
1 import collections
2
3 Carte = collections.namedtuple('Carte', ['rang', 'couleur'])
4
5 class JeuCartes:
6     rangs = [str(n) for n in range(2, 11)] + list('VDRA')
7     couleurs = ' pique coeur carreau trèfle'.split()
8
9     def __init__(self):
10         self._cartes = [Carte(rang, couleur) for couleur in
                           self.couleurs for rang in self.rangs]
```

1. Instanciez le Valet de pique
2. Quelle méthode ajouter pour pouvoir afficher la taille de votre jeu ?
3. Et pour traiter le jeu de cartes comme une liste ?
4. Afficher alors le premier et le dernier élément du jeu
5. Tirez au sort 3 cartes
6. Utilisez le slicing pour afficher tous les AS du jeu
7. Le jeu est-il itérable ?
8. Et dans l'ordre inverse ?

Exercice 2. Ordre des cartes différent

Au bridge, l'ordre des couleurs est trèfle, carreau, coeur, pique en croissant. Ecrivons cela dans un dictionnaire :

```
1 coul_val = dict(trèfle=0, carreau=1, coeur=2, pique=3)
```

1. Comment classer les cartes par rang et couleur avec cet ordre ?
2. Ecrire alors une boucle qui affiche le jeu en utilisant cet ordre.

Exercice 3. Vecteurs On voudrait implémenter en Python une classe Vecteur 2D qui permet de réaliser des opérations du style :

```
1 v1 = Vecteur(4,5)
2 v2 = Vecteur(3,2)
3 print(v1+v2)
4 print(5*v2)
5 print(abs(v1))
6 print(v1*v2)
```

Implémentez cette classe en utilisant les méthodes Python adaptées.