


Corrigé sujet 05 - Année : 2022

[Sujet 05 - 20222 ↓](#)

1. Exercice 1

Bug

La fonction à écrire s'appelle `RechercheMinMax` dans l'énoncé (avec le `R` majuscule) mais devient `rechercheMinMax` avec un `r` miniscule dans les appels.


 Script Python

```
1 def RechercheMinMax(tab):
2     if tab==[]:
3         return {'min':None, 'max':None}
4     mini,maxi = tab[0],tab[0]
5     for elt in tab:
6         if elt<mini: mini=elt
7         if elt>maxi: maxi=elt
8     return {'min' : mini, 'max' : maxi}
```

Commentaires

Encore une recherche classique de minimum et de maximum, le résultat est renvoyé sous la forme d'un dictionnaire.

2. Exercice 2

 Script Python

```
1 class Carte:
2     """Initialise Couleur (entre 1 a 4), et Valeur (entre 1 a 13)"""
3     def __init__(self, c, v):
4         assert 1<=c<=4, "La couleur est entre 1 et 4"
5         assert 1<=v<=13, "La valeur est entre 1 et 13" #(1)
6         self.Couleur = c
7         self.Valeur = v
8
9     """Renvoie le nom de la Carte As, 2, ... 10,
10     Valet, Dame, Roi"""
```

```

11     def getNom(self):
12         if ( self.Valeur > 1 and self.Valeur < 11):
13             return str( self.Valeur)
14         elif self.Valeur == 11:
15             return "Valet"
16         elif self.Valeur == 12:
17             return "Dame"
18         elif self.Valeur == 13:
19             return "Roi"
20         else:
21             return "As"
22
23     """Renvoie la couleur de la Carte (parmi pique, coeur, carreau, trefle)"""
24     def getCouleur(self):
25         return ['pique', 'coeur', 'carreau', 'trefle' ][self.Couleur - 1]
26
27     class PaquetDeCarte:
28         def __init__(self):
29             self.contenu = []
30
31         """Remplit le paquet de cartes"""
32         def remplir(self):
33             self.contenu = [Carte(couleur,valeur) for couleur in range(1, 5) for valeur in
34             range(1, 14)] #(2)
35
36         """Renvoie la Carte qui se trouve à la position donnee"""
37         def getCarteAt(self, pos):
38             assert 0<=pos<52, "Le numéro de la carte doit être entre 0 et 51"
39             if 0 <= pos < 52 :
40                 return self.contenu[pos]

```

1. Ce sont les instructions `assert` permettant de vérifier que la couleur est entre 1 et 4 inclus et la valeur entre 1 et 13 inclus.
2. On utilise une définition de liste par compréhension pour parcourir les couleurs et valeurs possibles.
3. Un jeu de cartes contient 52 cartes, la position d'une carte est entre 0 et 51 inclus.

⚠ Attention

- Le sujet demande d'ajouter des instructions `assert` dans la méthode `getCarteAt` mais cette méthode teste déjà que le numéro de la carte est entre les limites imposés. Le `assert` pourrait vérifier ici que `pos` est bien une variable de type `int`.
- Les *docstring* devraient être placées juste après la ligne `def` de définition des méthodes et pas avant.
- Il serait pertinent d'utiliser un dictionnaire afin d'associer valeur et nom de d'une carte : `{1 : 'As', 2: '2', ..., 12: 'Dame', 13 : 'Roi'}`