

Corrigé sujet **40** - Année : 2023

Sujet 40 - 2023 ⬇

Exercice 1

```
1 def nombre_de_mots(phrase):
2     nbr_espaces = 0
3     for caractere in phrase:
4         if caractere == " ":
5             nbr_espaces += 1
6     if phrase[-1] == "!" or phrase[-1]=="?":
7         nbr_mots = nbr_espaces
8     else:
9         nbr_mots = nbr_espaces + 1
10    return nbr_mots
```



Commentaires

Le nombre de mots est le nombre d'espace plus 1, sauf lorsque la phrase se termine par un point d'exclamation ou un point d'interrogation c'est alors le nombre d'espace.

Exercice 2

```
1 class Noeud:
2     def __init__(self, valeur):
3         '''Méthode constructeur pour la classe Noeud. Paramètre d'entrée
4         : valeur (int)'''
5         self.valeur = valeur
6         self.gauche = None
7         self.droit = None
8
9     def getValeur(self):
10        '''Méthode accesseur pour obtenir la valeur du noeud Aucun
11        paramètre en entrée'''
12        return self.valeur
13
14    def droitExiste(self):
15        '''Méthode renvoyant True si le sous-arbre droit est non vide
16        Aucun paramètre en entrée'''
17        return (self.droit is not None)
18
```

```

19     def gaucheExiste(self):
20         '''Méthode renvoyant True si le sous-arbre gauche est non vide
21         Aucun paramètre en entrée'''
22         return (self.gauche is not None)
23
24     def inserer(self, cle):
25         '''Méthode d'insertion de clé dans un arbre binaire de recherche
26         Paramètre d'entrée : cle (int)'''
27         if cle < self.getValeur() : # ❶
28             # on insère à gauche
29             if self.gaucheExiste() : # ❷
30                 # on descend à gauche et on recommence le test initial
31                 self.gauche.inserer(cle)
32             else:
33                 # on crée un fils gauche
34                 self.gauche = Noeud(cle) # ❸
35             elif cle > self.getValeur() : # ❹
36                 # on insère à droite
37                 if self.droitExiste() :
38                     # on descend à droite et on recommence le test initial
39                     self.droit.inserer(cle)
40                 else:
41                     # on crée un fils droit
42                     self.droit = Noeud(cle)

```

1. Insertion à gauche puisque la clé est inférieure à la valeur du noeud.
2. Le fils gauche n'est pas vide, on relance l'insertion sur ce fils
3. Le fils gauche est vide, il devient donc un nouveau noeud avec la valeur à insérer
4. Traitement similaire pour une insertion à droite