

Sujet 11

❑ Exercice 1 : *Nom de variables en python*

En Python, un nom de variable valide, :

- doit commencer par une lettre ou un *underscore* : `_`
- ne contient pas d'espace
- peut contenir des lettres ou des chiffres
- ne doit pas être un des mots réservés du langage. Les mots réservés sont : `and`, `as`, `assert`, `break`, `class`, `continue`, `def`, `del`, `elif`, `else`, `except`, `exec`, `finally`, `for`, `from`, `global`, `if`, `import`, `in`, `is`, `lambda`, `not`, `or`, `pass`, `print`, `raise`, `return`, `try`, `while`, `with`, `yield`.

Ecrire une fonction `nom_valide` qui prend en argument une chaîne de caractères et renvoie `True` si cette chaîne de caractère représente un nom de variable valide et `False` sinon.

🔗 Aide

On pourra créer une liste contenant les mots réservés et utiliser `in` pour savoir si une chaîne de caractères se trouve ou pas dans cette liste.

Exemples :

```

1 >>> nom_valide("ma_variable")
2 True
3 >>> nom_valide("la temperature")
4 False
5 >>> nom_valide("5 cases")
6 False
7 >>> nom_valide("for")
8 False

```

❑ Exercice 2 : *Insertion dans une liste déjà triée*

Pour insérer un élément dans une liste *déjà triée par ordre croissant*, on propose l'algorithme suivant :

1. Ajouter l'élément à la fin de la liste
2. Echanger l'élément avec son voisin de gauche tant qu'il est plus petit et que le début de liste n'est pas atteint

Compléter l'implémentation suivante de cette algorithme

```

1 def insertion(element, liste):
2     '''Insère element dans liste (qui est déjà triée)'''
3     liste ..... (element)
4     pos = .....
5     while pos > ... and liste[pos] < .....:
6         liste[pos], liste[pos-1] = ....., .....
7         pos = .....

```

Exemple

```

1 >>> liste = [1,4,7,13,17]
2 >>> insertion(8,liste)
3 >>> liste
4 [1,4,7,8,13,17]

```