

Exercice 1 : Savoir manipuler une liste simple

On exécute le fichier `ds_1.py` suivant :

```
ds_1.py
1 l = ["nsi" , 20 , 9 , 3 , "sun"]
```

On exécute ensuite dans le *shell* les lignes suivantes. Pour chacune d'elles, donner le résultat retourné :

Ligne	Résultat retourné
<code>>>> l[1]</code>	
<code>>>> len(l)</code>	
<code>>>> l[-1]</code>	
<code>>>> l[-1][0]</code>	

Exercice 2 : Savoir lire et écrire un code avec fonctions et listes

1- Qu'obtient-on en exécutant ce code ?

```
# Fonctions
def tri(liste) :
    l = []
    for i in range(len(liste)) :
        if liste[i] > 10 :
            l.append(liste[i])
    return l

# Main
l = [18 , 4 , 9 , 14 , 1 , 13]
l_retour = tri(l)
print(l_retour)
```

⇒ Contenu de l'affichage dans le *shell* :

2- Réécrire ci-contre le code de la fonction :

- en mettant le nombre 10 en argument dans la fonction `tri()` afin de pouvoir modifier cette valeur plus facilement lorsque cette fonction est appelée,
- en utilisant à présent un parcours de liste du type `for ... in liste`

```
# Fonctions
def tri
    l = []

# Main
l = [18 , 4 , 9 , 14 , 1 , 13]
l_retour = tri(l , 10)
print(l_retour)
```

Exercice 3: Ecrire un code avec fonctions et listes simples

Soit le code python incomplet donné ci-dessous d'une fonction `compter()` qui prend en argument une liste de string et une autre variable de type string. Elle renvoie le nombre d'éléments de la liste qui sont identiques à cette variable.

```
# Fonctions
def compter(

# Main
liste = ["je", "moi", "il", "ce", "je", "ce", "matin", "pas", "je"]
nb = compter(liste, "matin")
print(nb)
print(compter(liste, "je"))
print(compter(liste, "ds"))
```

En l'exécutant, cela donne dans le shell :

⇒ Compléter ce code.

```
>>> (executing file "ds_4.py")
1
3
0
```

Exercice 4 : *Ecrire un code avec fonctions et liste de listes*

Soit le code python incomplet donné d'une fonction *somme()* qui prend en argument une variable de type string et une liste de listes composées chacune d'un string et d'un nombre. Cette fonction renvoie la somme de ces nombres pour les sous-listes dont le string est identique à la variable mise en argument :

Le code est donné juste après. Son exécution dans le shell :

```
>>> (executing file "ds_5.py") donne  
12  
2001  
7  
0
```

⇒ Compléter ce code.

Exercice 5 : *Calcul cumulatif*

Compléter le programme qui suit. Le programme retourne le nombre d'années pour que la consommation d'internet atteigne 6000 TWh. Celle-ci est actuellement de 1500 TWh. Cette consommation est multipliée par 1,2 tous les ans.

```
18 consommation = 1500  
19 annees = 0  
20  
21 while          :  
22  
23  
24  
25  
26  
27  
28 print("Nombre d'années pour atteindre 6000 Wh : ", annees)
```

Exercice 6 Calcul d'une moyenne

Compléter le code de la fonction moyenne qui calcule la moyenne d'une liste de notes aléatoires entre 0 et 20 (fonction randint(a,b) renvoie une valeur entière aléatoire entre a et b)

```
1 from random import randint  
2  
3 notes = [ randint(0,20) for i in range(10)]  
4  
5 def moyenne(l) :  
6     taille = len(l)  
7     compteur = 0  
8     somme = 0  
9     while  
10  
11  
12  
13  
14 print(moyenne(notes))
```