

TD 3 : Les listes, tuples et chaînes de caractères

Exercice N° 1 : Les listes

1. Quel 'est la liste affichée après exécuter ce programme :

```
L = [ ]  
L.append(5)  
L.insert(0, 10)  
print(L)
```

Solution :

La liste affichée est : [10, 5]

2. Quel 'est la liste affichée après exécuter ce programme :

```
L = [10, 20, 30, 40, 30]  
L.remove(30)  
L.pop(1)  
print(L)
```

Solution :

La liste affichée est : [10, 40, 30]

3. Quel 'est la liste affichée après exécuter ce programme :

```
A = [3, 5, 7, 9, 11]  
print(A[1:4])
```

Solution :

La liste affichée est : [4, 7, 9]

4. Qu'est le résultat affiché par le code suivant :

```
L = [1, 2, 3, 4, 3]  
a = len(L)  
s = sum(L)  
print(s / a)
```

Solution :

Le résultat affiché est : 13/5

5. Compléter le code suivant pour fusionner les deux listes A et B dans une liste C qui devra être triée :

```
A = [1, 3, 5, 7]  
B = [2, 4, 6, 8]  
# Compléter ici  
print(C) # Affiche [1, 2, 3, 4, 5, 6, 7, 8]
```

Solution :

```
A = [1, 3, 5, 7]  
B = [2, 4, 6, 8]  
C = sorted(A+B)  
print(C) # Affiche [1, 2, 3, 4, 5, 6, 7, 8]
```

Exercice N° 2 : Les chaînes de caractères

1. Quel 'est la chaîne affichée après exécuter ce programme :

```
chaîne = "Python"  
chaîne = chaîne[::-1]  
print(chaîne)
```

Solution :

La chaîne affichée est : nohtyP

2. Compléter le code suivant pour compter le nombre de mots dans une chaîne de caractères :

```
ch = "La pluie tombe"  
# Compléter ici  
print(n)
```

Solution :

```
ch = "La pluie tombe"  
L = ch.split()  
n = len(L)  
print(n)
```

3. Donner une fonction qui vérifie si une chaîne de caractères passée en paramètre est palindrome ou non.

Exemple :

« radar » est un palindrome

« Un radar nu » est un palindrome

Solution :

```
def palaindrome(ch) :  
    ch = ch.lower()  
    ch = ch.replace(" ", "")  
    R = ch[::-1]  
    return R==ch
```