Correction des exercices python Révision

Exercice 1 ||

- Ecrire un algorithme Python permettant d'échanger le premier élément avec le dernier élément d'une liste donnée.

- *Exemple ;* ***L = ["Python", "Java", "C ++", "Javascript"] en entrée***
  En sortie on doit avoir
  ["Javascript", "Java", "C ++", "Python"]

In [1]:
```python
#Révision python
#Exo 1
#coding: utf-8
def swapList(L):
    # obtenir le dernier élément de la liste
    swap = L[-1]

    # remplacer le dernier élément de la liste par le premier
    L[-1] = L[0]

    # remplacer  le premier élément de la liste par le dernier
    L[0] = swap
    return L


# Exemple
L = L = ["Python" , "Java" , "C++" , "Javascript"]
print(swapList(L))
# La sortie est  : ['Javascript', 'Java', 'C++', 'Python']
```

```
['Javascript', 'Java', 'C++', 'Python']
```

Exercice 2 ||

- Ecrire un algorithme python sous forme de fonction qui prend en paramètres une liste l et renvoie un tuple de deux listes **(l_even, l_odd)** où l_even est composé des éléments de l d'indexe pair et l_old est constitué par les éléments d'indexe impair

Exemple:

Si entrée on a :

- ***L = ["Python", "Java", "C ++", "C #", "VB.Net", "Javascript"]***

En sortie on doit avoir

- ***(['Python', 'C ++', 'VB.Net'], ['Java', 'C #', 'Javascript'])***

In [2]:
```python
#Révision python
#Exo 2
#coding: utf-8

def odd_event(L):
    # obtenir la longueur de la liste
    n = len(L)
    # initialisation des listes d'indices impair et d'indices pair
    l_odd = []
    l_even = []

    # construire les liste L_odd et L_even
    for i in range(0 , n):
        if( i%2 == 0):
            l_even.append(L[i])
        else:
            l_odd.append(L[i])

    return (l_even , l_odd)

# Exemple
L = ["Python" , "Java" , "C++" , "C#" , "VB.Net" , "Javascript"]
print(odd_event(L))
# La sortie est  : (['Python', 'C++', 'VB.Net'], ['Java', 'C#', 'Javascript'])
```

```
(['Python', 'C++', 'VB.Net'], ['Java', 'C#', 'Javascript'])
```