

NSI Révisions listes

```
from copy import deepcopy
from lycee import *

listeVide = []

liste01 = [0]*5
#par compréhension
liste02 = [0 for x in range(5) ]

for i in range (1,4) :
    liste01.append(i)
    print(liste01)

print(liste01[0:5])
print(liste01[5:])
print(liste01[-1])

liste0123 = [ i for i in range(21)]
print(liste0123)
listePair = [i for i in range(21) if i%2 == 0]
print(listePair)

#copie en profondeur
listeCopieProfonde = deepcopy(liste01)
print( liste01, listeCopieProfonde)
liste01.append(36)
print( liste01, listeCopieProfonde)

parcours_par_index(liste01)
print(parcours_par_valeur.__doc__)
parcours_par_valeur(liste01)

#parcours

def parcours_par_index(l):
    """
    affiche les éléments de la liste l
    parcours par index
    """
    for i in range (len(l)):
        print(f"index {i} , valeur {l[i]}")

def parcours_par_valeur(l):
    """
    affiche les éléments de la liste l
    parcours par valeur
    """
    i = 0
    for elt in l:
        print(f"index {i} , valeur {elt}")
        i+=1

def recherche_une_valeur(l,val):
    trouve = False
    for elt in l :
        if elt == val :
            trouve = True
    return trouve
```

NSI Révisions listes

```
def recherche_nb_valeur(l,val):
    compteur = 0
    for elt in l :
        if elt == val :
            compteur += 1
    return compteur

def min.ma_liste):
    """
    Renvoie le minimum
    """
    min = ma_liste[0]
    ind_min = 0
    for i in range(1, len(ma_liste)):
        if ma_liste[i] < min :
            min = ma_liste[i]
            ind_min = i
    return min,i

def max.ma_liste):
    """
    Renvoie le minimum
    """
    max = ma_liste[0]
    ind_max = 0
    for i in range(1, len(ma_liste)):
        if ma_liste[i] > max :
            max = ma_liste[i]
            ind_max = i
    return max,i
```

```
X55 = [[0 for x in range(5)] for i in range(5)]
```

```
X0123 = [[x + 5*i for x in range(5)]
```

```
def matriceDouble(m):
    m1 = deepcopy(m)
    for i in range(len(m1)) :
        for j in range(len(m1)):
            m1[i][j] = 2* m1[i][j]
    return m1

def matriceDiagonale(m) :
    diag = []
    for i in range(len(m)):
        for j in range(len(m)):
            if i == j :
                diag.append(m[i][j])

    return diag
```