

Exercice 1 : Ecrivez par compréhension la liste suivante : liste = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

```
1 liste = [i for i in range(10)]
```

Indiquez ce qu'affiche :

print(liste[0])	0
print(liste[-1])	9
print(liste[-3])	7
print(liste[3 :])	[3, 4, 5, 6, 7, 8, 9]
print(liste[:3])	[0, 1, 2]
print(liste[5 :])	[5, 6, 7, 8, 9]
print(liste[:5])	[0, 1, 2, 3, 4]
print(liste[1 :5])	[1, 2, 3, 4]

Exercice 2 : Faites afficher tout le contenu d'une liste avec une boucle while et les indices, en partant du premier jusqu'au dernier :

```
ind = 0
while ind < len(liste) :
    print(liste[ind] , end=" ")
    ind = ind + 1
```

Exercice 3 : Faites afficher tout le contenu d'une liste avec une boucle while et les indices, en partant du dernier jusqu'au premier :

```
ind = len(liste)-1
while ind >=0 :
    print(liste[ind] , end=" ")
    ind = ind - 1
```

Exercice 4 : Inverser la première et la dernière valeur d'une liste :

```
a = liste[0]
liste[0] = liste[-1]
liste[-1] = a
```

Exercice 5 : Faites une permutation circulaire de 1 vers la droite, décalage des éléments de 1 à droite : 2 solutions proposées :

```
fin = liste[-1]
i = len(liste) - 1
while i >= 1 :
    liste[i] = liste[i-1]
    i = i - 1
liste[0] = fin
```

```
fin = liste[-1]
i = len(liste) - 2
while i >= 0 :
    liste[i+1] = liste[i]
    i = i - 1
liste[0] = fin
```

Exercice 6 : Ecrire un algorithme qui renvoie la liste des doublons contenus dans la liste :

```
liste = [1, 1, 2, 3, 4, 5, 5, 5, 6, 7, 8, 8, 8, 10]
doublons=[]
for i in range(len(liste)) :
    for j in range(i+1, len(liste)):
        if liste[i] == liste[j] and liste[i] not in doublons :
            doublons.append(liste[i])
```

Exercice 7 : On donne ci-dessous un bout de code incomplet :

```
1 def ajoute_elt(l,elt) :
2     l.append(l[-1])
3     i = len(l)-2
4     while i >0 :
5         l[i] = l[i-1]
6         i -= 1
7     l[0] = elt
8     return l
9
10 liste = [1,9,7,1,3]
11 liste = ajoute_elt(liste,"ok")
12 print(liste)
```

Question : Compléter directement sur cette feuille ce code afin que son exécution donne :

```
>>> (executing file "ex1.py")
['ok', 1, 9, 7, 1, 3]
```

Exercice 8 : On donne ci-dessous un bout de code incomplet un peu différent du précédent :

```
1 def ajoute_elt(l,elt,indice) :
2     l.append(l[-1])
3     i = len(l)-2
4     while i > indice :
5         l[i] = l[i-1]
6         i -= 1
7     l[indice] = elt
8     return l
9
10 liste = [1,9,7,1,3]
11 liste = ajoute_elt(liste,["olivia","marion","lea"],3)
12 print(liste)
```

Question : Compléter directement sur cette feuille ce code afin que son exécution donne :

```
>>> (executing file "ex2.py")
[1, 9, 7, ['olivia', 'marion', 'lea'], 1, 3]
```

Exercice 9 : On donne ci-dessous un bout de code incomplet :

```
1 def bon_ordre(l) :
2     a = l[-1]
3     l[-1] = l[1]
4     l[1] = l[0]
5     l[0] = a
6     return l
7
8 term_nsi = ["we","can","Yes"]
9
10 ll = bon_ordre(term_nsi)
11 print(ll)
12
13 message = ""
14 for mot in ll :
15     message += mot+" "
16 print(message)
```

Question : Compléter directement sur cette feuille ce code afin que son exécution donne :

```
>>> (executing file "ex3.py")
['Yes', 'we', 'can']
Yes we can
```

Exercice 10 : On donne ci-dessous un script :

```
1 def scinde_liste(l, indice) :
2     l_g = l[:indice]
3     l_d = l[indice:]
4     return l_g , l_d
5
6 liste = [1,9,7,1,3]
7 a,b = scinde_liste(liste,2)
8 print(a*2)
9 print(b)
```

Question : Que donne son exécution ? :

```
>>> (executing file "ex10.py")
[1, 9, 1, 9]
[7, 1, 3]
```

Exercice 11 : On donne ci-dessous le script précédent un peu modifié :

```
1 def scinde_liste(l, indice) :
2     l_g = l[:indice]
3     l_d = l[indice:]
4     return l_g , l_d
5
6 liste = [1,9,["Baptiste","petit"],1,3]
7 a,b = scinde_liste(liste,2)
8 print(a)
9 print(b)
10 print(b[0][1])
11 b[0][1] = "grand"
12 print(liste)
```

Question : Que donne son exécution ? :

```
>>> (executing file "ex11.py")
[1, 9]
[['Baptiste', 'petit'], 1, 3]
petit
[1, 9, ['Baptiste', 'grand'], 1, 3]
```

Exercice 12 : On donne ci-dessous un bout de code incomplet :

```
1 def modif_ordre(l) :
2     i_g = 0
3     i_d = len(l) - 1
4     while i_g < i_d :
5         a = l[i_d]
6         l[i_d] = l[i_g]
7         l[i_g] = a
8         i_g += 1
9         i_d -= 1
10    return l
11
12 liste = [1 , 2 , 3 , 4 , 5 , 6 ,7]
13 print(modif_ordre(liste))
```

Question : Compléter directement sur cette feuille ce code afin que son exécution donne :

```
>>> (executing file "ex6.py")
[7, 6, 5, 4, 3, 2, 1]
```

Exercice 13 :

- Que donne l'exécution de ce script ? :

```
1 l = [0, ["N", "S", "I"], 1, 2, [3, 4]]
2 del l[1]
3 print(l)
```

```
>>> (executing file "ex13.py")
[0, 1, 2, [3, 4]]
```

- Que donne l'exécution de ce script ? :

```
1 l = [0, ["N", "S", "I"], 1, 2, [3, 4]]
2 del l[1:]
3 print(l)
```

```
>>> (executing file "ex13.py")
[0]
```

- Que donne l'exécution de ce script ? :

```
1 l = [0, ["N", "S", "I"], 1, 2, [3, 4]]
2 del l[1][1:]
3 print(l)
```

```
>>> (executing file "ex13.py")
[0, ['N'], 1, 2, [3, 4]]
```

- Que donne l'exécution de ce script ? :

```
1 l = [0, ["N", "S", "I"], 1, 2, [3, 4]]
2 del l[len(l)-1][-1]
3 print(l)
```

```
>>> (executing file "ex13.py")
[0, ['N', 'S', 'I'], 1, 2, [3]]
```