# TD_liste

March 18, 2021

## 1 Exercice 1 : niveau facile

[ ]:
```
liste = [4,7,3,4,6,9,13,2]
```

- Répondre aux questions suivantes sans utiliser l'ordinateur :

1. Quelle instruction python permet d'accéder à l'entier `9` ?
2. Quelle est la longueur de la liste ? Ecrire l'instruction python correspondante
3. Que renvoie l'instruction `liste[4]` ?
4. Que renvoie l'instruction `liste[-4]` ?
5. Que renvoie l'instruction `liste[-9]` ?
6. Que renvoie l'instruction `liste[1:7]` ?
7. Que vaut la liste après l'instruction `liste[3]=5` ?
8. Que vaut la liste après l'instruction `del(liste[5])` ?

- Vérifier vos réponses à l'aide de l'ordinateur

## 2 Exercice 2 : niveau facile

```
liste = ['Alice', 'Bob', 'Tom']
```

1. Ecrire l'instruction python permettant d'obtenir la documentation de la **méthode** de liste `insert`
2. Utiliser la méthode `insert` afin que `liste = ['Alice', 'Bob', 'Marc', 'Tom']`

## 3 Exercice 3 : niveau facile

On donne le code ci-dessous :

[4]:
```
t = []
for i in range(10):
    t.append(i**2)
```

1. Exécuter le code ci-dessous à la main (sans ordinateur !!) afin de donner la valeur de `t` après l'exécution de ce code.
2. Vérifier à l'aide de l'ordinateur
3. Modifier le programme précédent afin que t soit égal à `[0,4,16,36,64]`

## 4 Exercice 4 : niveau facile

On donne le code ci-dessous :

[ ]:
```
t = [0]*10

for i in range(len(t)):
    t[i] = 10 - i
```

1. Exécuter le code ci-dessous à la main (sans ordinateur !!) afin de donner la valeur de `t` après l'exécution de ce code.
2. Vérifier à l'aide de l'ordinateur

## 5 Exercice 5 : niveau facile

[ ]:
```
def longueurNomV1(liste_nom) :
    liste_longueur = []
    for nom in liste_nom:
        # A compléter
```

1. Compléter la **fonction** `longueurNomV1` qui :
   - prend une liste de noms en **paramètre**
   - **renvoie** la liste contenant le nombre de caractères de chacun des noms

Un exemple d'appel est donné ci-dessous

[10]:
```
liste_nom = ['Jean-Michel', 'Marc', 'Vanessa', 'Anne', 'Maximilien',␣
↪'Alexandre-Benoît', 'Louise']
```

[14]:
```
t = longueurNomV1(liste_nom)

print(t)
```

```
[11, 4, 7, 4, 10, 16, 6]
```

2. Compléter la **fonction** `longueurNomV2` qui :
   - prend une liste de noms en **paramètre**
   - **renvoie** la liste contenant le nombre de caractères de chacun des noms

[ ]:
```
def longueurNomv2(liste_nom) :
    liste_longueur = [0]*len(liste_nom)
    for i in range(len(liste_nom)):
        # A compléter
```

Un exemple d'appel est donné ci-dessous

[10]:
```
liste_nom = ['Jean-Michel', 'Marc', 'Vanessa', 'Anne', 'Maximilien',␣
↪'Alexandre-Benoît', 'Louise']
```

[22]:
```
t = longueurNomV2(liste_nom)

print(t)
```

```
[11, 4, 7, 4, 10, 16, 6]
```

## 6 Exercice 6 : niveau intermédiaire

Écrire une **fonction** `affiche_liste` qui :

- prend une liste en **paramètre**
- **affiche** « proprement » tous les éléments d'une liste. (voir appel ci-dessous)

*(Conseil : Après avoir consulté la documentation de la fonction* ***print****, on pourra utiliser son paramètre* ***end****)*

Un exemple d'appel est donné ci-dessous

[8]:
```
t=['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin', 'Juillet', 'Août',␣
↪'Septembre', 'Octobre', 'Novembre', 'Décembre']
```

[9]:
```
afficheListe(t)
```

```
Janvier Février Mars Avril Mai Juin Juillet Août Septembre Octobre Novembre
Décembre
```

## 7 Exercice 7 : niveau intermédiaire

Écrire une **fonction** `enleveDoublon` qui :

- prend une liste d'entiers en **paramètre**
- **renvoie** une liste contenant la liste de ces entiers mais en omettant les doublons.

Un exemple d'appel est donné ci-dessous

[23]:
```
liste_entier = [23, 45, 23, 43, 7, 66, 21, 45, 23, 7, 200, 200]
```

[31]:
```
liste = enleveDoublon(liste_entier)
print(liste)
```

```
[23, 45, 43, 7, 66, 21, 200]
```

## 8 Exercice 8 : niveau intermediaire

Écrire une **fonction** `alterne` qui :

- prend en **paramètres** 2 listes de **même longueur**
- **renvoie** une nouvelle liste qui devra contenir tous les éléments des deux listes passées en paramètres en les alternant

Un exemple d'appel est donné ci-dessous :

[32]:
```
t1 = ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin', 'Juillet', 'Août',␣
 ↪'Septembre', 'Octobre', 'Novembre', 'Décembre']
t2 = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
```

[37]:
```
liste = alterne(t1,t2)

print(liste)
```

```
['Janvier', 31, 'Février', 28, 'Mars', 31, 'Avril', 30, 'Mai', 31, 'Juin', 30,
'Juillet', 31, 'Août', 31, 'Septembre', 30, 'Octobre', 31, 'Novembre', 30,
'Décembre', 31]
```

## 9 Exercice 9 : niveau intermédiaire

[38]:
```
a = [8468, 4560, 3941, 3328, 7, 9910, 9208, 8400, 6502, 1076, 5921, 6720, 948,␣
 ↪9561, 7391, 7745, 9007, 9707, 4370, 9636,
     5265, 2638, 8919, 7814, 5142, 1060, 6971, 4065, 4629, 4490, 2480, 9180,␣
 ↪5623, 6600, 1764, 9846, 7605, 8271, 4681,
     2818, 832, 5280, 3170, 8965, 4332, 3198, 9454, 2025, 2373, 4067]

b = [9093, 2559, 9664, 8075, 4525, 5847, 67, 8932, 5049, 5241, 5886, 1393,␣
 ↪9413, 8872, 2560, 4636, 9004, 7586, 1461, 350,
     2627, 2187, 7778, 8933, 351, 7097, 356, 4110, 1393, 4864, 1088, 3904,␣
 ↪5623, 8040, 7273, 1114, 4394, 4108, 7123, 8001,
     5715, 7215, 7460, 5829, 9513, 1256, 4052, 1585, 1608, 3941]
```

Il y a un nombre qui est exactement à la même place dans la liste `a` et dans la liste `b`

Ecrire le programme qui permet de trouver ce nombre et son indice

## 10 Exercice 10 : niveau intermédiaire

[ ]:
```
def permute(liste,i,j):
    """
    description: échange dans la liste les éléments d'indice i et j
    liste (list)
    i, j (int): indices des éléments à permuter
    ATTENTION : MODIFICATION DE LA LISTE EN PLACE !!
    """
    #A compléter
```

1. Ecrire le code de la fonction `permute`

Un exemple d'appel est donné ci-dessous :

[45]:
```
liste = ["Bob", "Alice", "Marc", "Tom", "John"]
```

[46]:
```
permute(liste,0,3)

print(liste)
```

```
['Tom', 'Alice', 'Marc', 'Tom', 'John']
```

[ ]:
```
def reverse(liste):
    """
    description: inverse l'ordre des éléménts de la liste
    liste (list)
    ATTENTION : MODIFICATION DE LA LISTE EN PLACE !!
    """
    #A compléter
```

2. Ecrire le code de la fonction `reverse` ci-dessus. On pourra utiliser la fonction `permute`

Un exemple d'appel est donné ci-dessous :

[52]:
```
liste = ["Bob", "Alice", "Marc", "Tom", "John"]

reverse(liste)

print(liste)
```

```
['John', 'Tom', 'Marc', 'Tom', 'John']
```