

Exercice 1 *2 points*

On donne la liste `nombres = [3, 8, -7, 1, 5, -6]`.

1. Par quelle instruction obtient-on la valeur -7 ?

```
nombres[2]
```

2. Créer en compréhension la liste `positifs` contenant uniquement les nombres positifs de la liste `nombres`.

```
positifs = [nb for nb in nombres if nb > 0]
```

Exercice 2 *2 points*

1. On considère la fonction mathématique $f : x \mapsto 2x + 3$. Écrire le code de la fonction `f`.

```
def f(x):  
    return 2*x + 3
```

2. Créer **en compréhension** une liste `valeurs` contenant l'image des entiers de 1 à 10 par la fonction `f`.

```
valeurs = [f(x) for x in range(1, 11)]
```

Exercice 3 *3 points*

On considère le tableau `tab = [[1, 9, 4], [4, 1, 8], [7, 10, 1]]`

1. Par quelle instruction obtient-on la valeur 10 ?

```
tab[2][1]
```

2. Écrire un code qui calcule dans une variable `somme` la somme de tous les éléments de `tab`.

```
tab = [[1, 9, 4], [4, 1, 8], [7, 10, 1]]  
somme = 0  
for ligne in tab:  
    for elt in ligne:  
        somme += elt  
  
# ou bien  
somme = 0  
for i in range(3):  
    for j in range(3):  
        somme += tab[i][j]
```

Exercice 4 *3 points*

On dispose d'une liste `tab` (non donnée dans l'énoncé), dont on sait qu'elle possède deux nombres consécutifs égaux.

Écrire un code permettant d'afficher ces deux nombres consécutifs égaux.

```
for i in range(len(tab)-1):  
    if tab[i] == tab[i+1]:  
        print(tab[i], tab[i+1])
```

Exercice 5 *3 points*

1. Créer une liste `lst` contenant 100 fois le nombre 0. (la méthode de construction est libre)

```
lst = [0]*100  
# ou bien  
lst = []  
for _ in range(100):  
    lst.append(0)
```

2. Remplacer chaque élément de `lst` par le mot 'pair' ou 'impair' suivant que l'indice de l'élément est pair ou impair.

```
for i in range(100):
    if i % 2 == 0:
        lst[i] = 'pair'
    else:
        lst[i] = 'impair'
```

Exercice 6 *4 points*

On considère la liste

```
tab = ["un", "deux", "trois", "quatre", "cinq", "six", "sept", "huit", "neuf"]
```

1. Écrire une fonction `nb_voyelles` qui prend en paramètres une chaîne de caractères `mot` et qui renvoie le nombre de voyelles de `mot`.

```
def nb_voyelles(mot):
    voyelles = "aeiouy"
    n = 0
    for l in mot:
        if l in voyelles:
            n += 1
    return n
```

2. Écrire un code permettant de construire une liste `new_tab` qui contient les éléments de `tab` ayant exactement 2 voyelles.

(Vous pouvez vous servir de la question précédente même si vous ne l'avez pas traitée.)

```
new_tab = []
for mot in tab:
    if nb_voyelles(mot) == 2:
        new_tab.append(mot)
```

ou mieux, en compréhension :

```
new_tab = [mot for mot in tab if nb_voyelles(mot) == 2]
```

Exercice 7 *3 points*

Coder une fonction `extremes` qui prend en paramètre une liste `lst` et qui renvoie la différence entre le dernier élément de `lst` et le premier élément de `lst`.

Si la liste `lst` est vide, la fonction doit afficher le message "liste vide !" et renvoyer `None`.

Exemple d'utilisation :

```
>>> extremes([2, 5, 3, 6])
4
>>> extremes([])
'liste vide !'
```

Code de votre fonction :

```
def extremes(lst):
    if lst == []:
        print("liste vide !")
        return None
    else:
        val = lst[-1] - lst[0]
        return val
```