

OBJECTIFS : L'objectif de ce TP est de réaliser des codes pythons utilisant la structure de listes

DOCUMENT A RENDRE : Pour l'exercice 1, un tableau est à compléter par écrit sur la feuille distribuée. Pour les autres exercices, vous rédigerez un compte-rendu numérique en utilisant un logiciel de traitement de texte (*Word ou Libre Office*). Le fichier constitué sera appelé *tp9.doc* ou *tp9.odt* et devra être transféré en fin d'activité **par l'intermédiaire** du site *nsibrantly.fr* : se loguer et transférer en utilisant le code **tp9** . Ce compte-rendu contiendra :

- les réponses aux différentes questions posées,
- les captures d'écran **des morceaux de codes** écrits **et** celles **des résultats des exécutions** données dans le shell. Pour faire ces captures, utiliser *l'Outil Capture d'écran* de Windows.

1. Comment créer et parcourir les listes :

⇒ Exécuter les scripts ci-dessous et donner le résultat de leur exécution :

<i>Scripts</i>	<i>Résultat de l'exécution dans le shell</i>
<pre>l = [25 , 12 , 2020 , "noel"] print(l)</pre>	<pre>>>> (executing file "exercice 1.py") [25, 12, 2020, 'noel']</pre>
<pre>l = [25 , 12 , 2020 , "noel"] n = len(l) print(n)</pre>	<pre>>>> (executing file "exercice 1.py") 4</pre>
<pre>l = [25 , 12 , 2020 , "noel"] a = l[3] print(a)</pre>	<pre>>>> (executing file "exercice 1.py") noel</pre>
<pre>l = [25 , 12 , 2020 , "noel"] l[3] = 2080 print(l)</pre>	<pre>>>> (executing file "exercice 1.py") [25, 12, 2020, 2080]</pre>
<pre>l = [25 , 12 , 2020 , "noel"] l[2] = 2080 print(l)</pre>	<pre>[25, 12, 2080, 'noel']</pre>

<pre>l = [25 , 12 , 2020 , "noel"] l[4] = 2080 print(l)</pre>	<pre>IndexError: list assignment index out of range</pre>
<pre>l = [25 , 12 , 2020 , "noel"] l.append(2080) print(l)</pre>	<pre>>>> (executing file "exercice 1.py") [25, 12, 2020, 'noel', 2080]</pre>
<pre>l = [25 , 12 , 2020 , "noel"] del l[1] print(l)</pre>	<pre>>>> (executing file "exercice 1.py") [25, 2020, 'noel']</pre>
<pre>l = [0]*4 print(l)</pre>	<pre>>>> (executing file "exercice 1.py") [0, 0, 0, 0]</pre>
<pre>l = ["coucou"]*4 print(l)</pre>	<pre>>>> (executing file "exercice 1.py") ['coucou', 'coucou', 'coucou', 'coucou']</pre>
<pre>l = ["t'es sérieux ?"]*4 l[0] = 25 l[1] = 12 l[2] = 2020 l[3] = "noel" print(l)</pre>	<pre>>>> (executing file "exercice 1.py") [25, 12, 2020, 'noel']</pre>
<pre>l = [i for i in range(8)] l.append("noel") print(l)</pre>	<pre>>>> (executing file "exercice 1.py") [0, 1, 2, 3, 4, 5, 6, 7, 'noel']</pre>
<pre>liste_1 = [i for i in range(8)] liste_2 = ["père" , "noel" , "lutin"] l = liste_1 + liste_2 print(l) print(len(l))</pre>	<pre>>>> (executing file "exercice 1.py") [0, 1, 2, 3, 4, 5, 6, 7, 'père', 'noel', 'lutin'] 11</pre>
<pre>l = [] l.append("nsi") print(l)</pre>	<pre>>>> (executing file "exercice 1.py") ['nsi']</pre>

<pre>l = ["bonjour", "good morning", "guten morgen", "buongiorno"] for i in range(len(l)) : print(l[i])</pre>	<pre>bonjour good morning guten morgen buongiorno</pre>
<pre>l = ["bonjour", "good morning", "guten morgen", "buongiorno"] for mot in l : print(mot)</pre>	<pre>bonjour good morning guten morgen buongiorno</pre>
<pre>l = ["bonjour", "good morning", "guten morgen", "buongiorno"] for i in range(len(l)) : l[i] = "mot"+str(i) print(l)</pre>	<pre>['mot0', 'mot1', 'mot2', 'mot3']</pre>
<pre>l = ["bonjour", "good morning", "guten morgen", "buongiorno"] i = 0 for m in l : m = "mot"+str(i) i = i + 1 print(l)</pre>	<pre>['bonjour', 'good morning', 'guten morgen', 'buongiorno']</pre>

2. Plusieurs façons de créer une liste :

a) Le script ci-dessous permet de créer une liste ℓ contenant les 25 premiers nombres pairs :

Script :

```
l = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48]
print(l)
```

Résultat à l'exécution :

```
>>> (executing file "exercice_2.py")
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20,
22, 24, 26, 28, 30, 32, 34, 36, 38, 40,
42, 44, 46, 48]
```

- b) Le script ci-dessous, incomplet permet de créer cette même liste ℓ : \Rightarrow compléter ce script

Script :

```
l = [0] * 25
print(l)
```

```
for i in range(25) :
    l[i] = 2*i
```

```
print(l)
```

Résultat à l'exécution :

```
>>> (executing file "exercice_2.py")
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48]
```

- c) Le script ci-dessous, incomplet permet aussi de créer cette même liste ℓ : \Rightarrow compléter ce script

Script :

```
l = [ 2*i for i in range(25)]
print(l)
```

Résultat à l'exécution :

```
>>> (executing file "exercice_2.py")
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48]
```

- d) Enfin ce dernier script ci-contre, incomplet, permet de créer encore d'une autre manière cette liste ℓ :

\Rightarrow compléter ce script

Résultat à l'exécution :

```
>>> (executing file "exercice_2.py")
[]
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48]
```

```
l = []
print(l)
```

```
for i in range(25) :
    l.append(2*i)
```

```
print(l)
```

On ajoute les éléments les uns après les autres, en fin de liste

3. Parcourir une liste en utilisant les index :

Soit le code python incomplet ci-dessous :

```
# Définition des fonctions
def compte(mot) :
    """
    Fonction qui prend en argument une chaine de caractère appelée mot.
    Elle retourne le nombre de caractères qui composent cette chaine.
    """
    n = 0
    for i in range(len(mot)) :
        n = n + 1
    return n

# Programme principal
nsi=["Hector", "Paul", "Hans", "Martin", "Mathéo", "Thomas", "Guillaume", "Yahel", "Raiden",
"Yann", "Agatha", "Elias", "Malik", "Antoine", "Quentin"]
longueur = len(nsi)
nb_caracteres = 0
for i in range(longueur) :
    nb_caracteres = nb_caracteres + compte(nsi[i])

print("Le nombre total de caractères utilisés dans la liste est : ",nb_caracteres)
```

A l'exécution, on obtient dans le shell :

```
>>> (executing file "exercice_3.py")
Le nombre total de caractères utilisés dans la liste est : 31
```

⇒ Compléter ce code en utilisant **un parcours de liste par index** dans la fonction *compte()*.

4. Parcourir une liste par éléments :

Modifier ici le code précédent en utilisant à présent un **parcours de liste par élément** dans la fonction `compte()` et dans le programme principal :

```
# Définition des fonctions
def compte(mot) :
    """
    Fonction qui prend en argument une chaîne de caractère appelée mot.
    Elle retourne le nombre de caractères qui composent cette chaîne.
    """
    n = 0
    for c in mot :
        n = n + 1
    return n

# Programme principal
nsi=["Hector", "Paul", "Hans", "Martin", "Mathéo", "Thomas", "Guillaume", "Yahel", "Raiden",
    "Yann", "Agatha", "Elias", "Malik", "Antoine", "Quentin"]

nb_caracteres = 0
for mot in nsi :
    nb_caracteres = nb_caracteres + compte(mot)

print("Le nombre total de caractères utilisés dans la liste est : ",nb_caracteres)
```

Questions : Les codes utilisant un parcours de liste par index sont-ils plus simples que ceux utilisant un parcours par éléments ?