1 Listes

Exercice 1.1 (A la découverte des listes)

1. Tapez les instructions suivantes dans la console puis `type(L)` et notez le contenu de la variable `L` dans chacun des cas suivants :

   (a) `L = []`

   (b) `L = ['s', "14", '12.5', "texte"]`

   (c) `L = list(range(6))`

   (d) `L = [k**2 for k in range(10)]`

2. Comment peut-on reconnaître la syntaxe d'une liste en `Python` ?

3. (a) Saisir les instructions ci-contre et notez le contenu de `L` à chaque étape.

```
L = []
L.append("abc") ; L
L.append('d') ; L
```

   (b) Que fait la *méthode* `append` ?

4. (a) Saisir les instructions ci-contre et notez le contenu de `L` à chaque étape.

```
L = []
L.extend([32,765] ; L
L.extend([list(range(3))])
```

   (b) Que fait la *méthode* `extend` ?

Exercice 1.2 (Listes par compréhension)

1. Quelle liste est produite par l'instruction `L = [k**2 for k in range(10)]` ?

2. Et par l'instruction `L = [k**3 for k in range(6) if k%2 == 0]` ?

Exercice 1.3 (Indexation et extraction de données) On considère la liste suivante `L = list(range(5))`.
Testez chacune des instructions proposées, notez le résultat.

1. `L[2]`
2. `L[1:3]`
3. `L[0:3:2]`
4. `L[:2]`
5. `L[1:]`
6. `len(L)`

Exercice 1.4 (Indices négatifs) On considère la liste `L = list(range(5))`.
Testez chacune des instructions proposées, notez le résultat.

1. `L[-1]`
2. `L[-2]`
3. `L[1:-1]`
4. `L[-2:]`

Exercice 1.5 (♦Slicing) On considère la liste `L = list("Bienvenue en spé NSI")`.

1. Donnez deux instructions permettant de renvoyer `'I'` à partir de la liste `L`.
2. Quelle instruction faut-il saisir pour obtenir la liste `['B','i','e','n']` à partir de la liste `L` ?
3. Quelle instruction faut-il saisir pour obtenir la liste `['s','p','é',' ','N','S','I']` à partir de la liste `L` ?
4. Quelle instruction faut-il saisir pour obtenir la liste `['e','v','n','e','e',' ','p','','S']` à partir de la liste `L` ?
5. Quelle instruction faut-il saisir pour obtenir la liste `['I','S','N',' ','é','p','s']` à partir de la liste `L` ?
6. Quelle instruction faut-il saisir pour obtenir la liste `['é','s','n',' ','u','e']` ?

Exercice 1.6 (Tests) On considère la liste `L = list(range(5))`.
Testez chacune des instructions proposées, notez le résultat.

1. `2 in L`
2. `3 not in L`
3. `L == [1,0,2,4,3]`
4. `L[1:3] == [1,2]`

Exercice 1.7 (Opérations)

1. Que contient `L` après l'instruction `L = [1,2,3] + [7,8]` ?
2. Et après l'instruction `L = [1,2,3]*3` ?

Exercice 1.8 (Des méthodes agissant sur les listes)
On donne la liste suivante `liste = [1, 2, 3, 2, 4, 5, 3, 7, 8, 10, 11, 3]`.
Testez les instructions suivantes et décrire l'effet de chacune des fonctions proposées :
Vous pourrez vous aider de l'instruction `help(list)`.

1. `liste.index(3)`

2. `liste.remove(3) ; liste`

3. `liste.pop(9) ; liste`

4. `liste.count(3))`

5. `liste.reverse() ; liste`

Exercice 1.9 (Modifier et copier une liste, insérer et supprimer un élément)
On donne la liste suivante `liste = [1, 2, 3, 2, 4, 5, 3, 7, 8, 10, 11, 3]`.
Testez les instructions suivantes et décrire l'effet des instructions proposées :

1. `del(liste[2]) ; liste`

2. `liste.insert(2, 91) ; liste`

3. `liste[1] = 7 ; liste`

4. `L = liste ; L ; liste[3] = 25 ; liste ; L`

5. `L = [elem for elem in liste] ; liste[4] = 0 ; liste, L`

Exercice 1.10 (Fonctions « maison » améliorées)

1. Ecrire une fonction `ListAleaInt(n, min, max)` qui fabrique une liste de `n` nombres entiers naturels compris entre `min` et `max` de façon aléatoire (*utiliser la fonction `randint` du module `random`*).
2. Définir une liste numérique `L` de 20 éléments. Quel test permet de savoir si un nombre entier `n` appartient à la liste `L` ?
3. Ecrire une fonction `RemoveAll(n,L)` supprimant toutes les occurrences du nombre `n` dans la liste `L`,
4. Ecrire une fonction `MultiPop(Li,L)` renvoyant la liste des éléments d'index donnés dans la liste `Li` et la liste modifiée en supprimant les éléments correspondants dans la liste `L` (si `Li` est vide, renvoyer la liste vide et la liste `L`),
5. Ecrire une fonction `AllIndex(e,L)` renvoyant la liste des index de l'élément `e` dans la liste `L` (renvoie la liste vide si l'élément n'est pas présent) .

On veillera à coder ces fonctions en une ligne en utilisant des définitions par compréhension.

Exercice 1.11 (♦ Nombres palindromes)

L'objectif de cet exercice est de déterminer, à l'aide de listes, si un nombre entier naturel quelconque est un *nombre palindrome*, c'est-à-dire un nombre dont l'écriture en base 10 est symétrique. Ces nombres se lisent alors de la même façon dans les deux sens. Exemples : 11, 242,12321,...

1. Écrire une fonction `Decomposition(n)` renvoyant une liste donnant la décomposition en base 10 de n. Les index correspondront aux puissances de 10 de la décomposition.
2. Ecrire une fonction `Is_Palindrone(n)` renvoyant `True` si le nombre est un nombre palindrome et `False` sinon.

Exercice 1.12 (Avec `matplotlib`)

1. Construire aléatoirement une liste `Val` de 10 000 entiers compris entre 0 et 10.
2. Construire une liste `Eff` de 11 éléments constituée du nombre d'occurrences des entiers variant de 0 à 10 dans la `Val`. Les effectifs seront rangés à l'index correspondront à la valeur concernée.
3. Tester le code `Python` suivant :

```
import matplotlib.pyplot as plt

Freq = [elem/100 for elem in Eff]
plt.title("Fréquences simulées")
plt.ylabel("Fréquence en pourcentage")
plt.xticks(list(range(0,11)), list(range(0,11)))
plt.yticks(list(range(0,101,10)))
plt.bar(list(range(0,11)), Freq)
plt.show()
```

2 Tuples

Exercice 2.1 (A la découverte des tuples)

1. Tapez les instructions suivantes dans la console puis `type(T)` et notez le contenu de la variable `T` dans chacun des cas suivants :

   (a) `T = ()`

   (b) `T = (True,3.14)`

   (c) `T = 5, False, 2.7`

   (d) `T = tuple([1,2,3])`

   (e) `T = tuple("NSI")`

   (f) `T = tuple((3*k)**2 for k in range(5))`

2. Comment peut-on reconnaître la syntaxe d'un tuple en `Python` ?

3. Soit `T = (True,3.14)`.
   (a) Que renvoie l'instruction `T.append("NSI")` ?

   (b) Que renvoie l'instruction `T.extend((1,True))` ?

   (c) Que renvoie l'instruction `T[1]` ? Que renvoie l'instruction `T[1] = 7.1` ?

   (d) Que conclure à propos du type `tuple` ?

Exercice 2.2 (Tests sur les tuples) Les tests sur les tuples sont les mêmes que ceux des listes.
On donne le tuple `T = ('abc', 3, True, "NSI")`. Proposez un test utilisant ce tuple, utilisant l'opérateur proposé et renvoyant ce qui est précisé dans chacun des cas suivants :

1. `in` et `False`
2. `!=` et `False`
3. `not, in` et `True`
4. `==` et `True`

Exercice 2.3 (♦Accès aux données d'un tuple) De même que pour les listes, on accède à une donnée d'un tuple en indiquant entre crochets l'indice de l'élément souhaité. On peut également pratiquer du *slicing* avec les syntaxes de la forme `T[i:j:step]`.
On considère le tuple `T = (1,True,'a',47,7.52,False)`.

1. Quelle instruction faut-il saisir pour obtenir le tuple `(True,47,False)` à partir du tuple `T` ?

2. Quelle instruction faut-il saisir pour obtenir le tuple `(7.52,47,'a')` à partir du tuple `T` ?

Exercice 2.4 (Tuples par compréhension)

1. Quel tuple est produit par l'instruction `tuple(k for k in range(2,100) if not(k%2))` ? (on pourra le décrire par une phrase)

2. Quel tuple est produit par l'instruction `tuple(3*k for k in range(1,34))` ? (on pourra le décrire par une phrase)

3. Quelle instruction saisir pour produire un tuple contenant tous les entiers multiples de 7 compris entre 0 et 99 ?

4. Écrire une fonction `Table(n)` renvoyant un tuple composé des 10 premiers multiples de n.

Exercice 2.5 (Les fonctions renvoient souvent des tuples) Écrivez une fonction `DivisionEuclidienne(a,b)` renvoyant le quotient et le reste de la division euclidienne de l'entier positif `a` par l'entier strictement positif `b`.