

# ◆ Révisions – Questions élémentaires sur les listes et les chaînes de caractères

**Exercice 1.** On considère la liste  $L = [ 1, 2, 'a', True, 5, 'b' ]$ .

1. Que vaut `len(L)` ?

**Solution.** `len(L)` vaut 6.

2. Que valent `L[1]`, `L[2]` et `L[-1]` ?

**Solution.** `L[1]` vaut 2, `L[2]` vaut 'a' et `L[-1]` vaut 'b'.

3. Quels sont les types de `L[1]`, `L[2]` et `L[3]` ?

**Solution.** `L[1]` est de type `int` (entier), `L[2]` est de type `str` (chaîne de caractères) et `L[3]` est de type `bool` (booléen).

4. Que renvoie `L[2]+L[5]` ?

**Solution.** `L[2]+L[5]` renvoie la chaîne de caractères 'ab'.

5. Que renvoie `L[5]*L[1]` ?

**Solution.** `L[5]*L[1]` renvoie la chaîne de caractères 'bb'.

6. Que renvoie `L[4]*L[1]` ?

**Solution.** `L[4]*L[1]` renvoie l'entier 10.

7. Que renvoient `L[:3]`, `L[3:]` et `L[1:4]` ?

**Solution.** `L[:3]` renvoie la liste `[ 1, 2, 'a' ]`, `L[3:]` renvoie la liste `[ True, 5, 'b' ]` et `L[1:4]` renvoie la liste `[ 2, 'a', True ]`.

8. Quel est l'effet sur `L` de l'instruction `L[1] = L[1] * 2` ?

**Solution.** L'instruction `L[1] = L[1] * 2` multiplie l'entier `L[1]` par 2 donc transforme `L` en la liste `[ 1, 4, 'a', True, 5, 'b' ]`.

9. Quel est l'effet sur `L` de l'instruction `L[1] = L[2]` ?

**Solution.** L'instruction `L[1] = L[2]` remplace le deuxième élément de `L` par le troisième (sans modifier ce dernier) donc elle transforme `L` en la liste `[ 1, 'a', 'a', True, 5, 'b' ]`.

10. Quel est l'effet sur `L` de l'instruction `L[1],L[2] = L[2],L[1]` ?

**Solution.** L'instruction `L[1],L[2] = L[2],L[1]` échange le deuxième et le troisième terme de la liste donc transforme `L` en la liste `[ 1, 'a', 2, True, 5, 'b' ]`.

**Exercice 2.** On considère la chaîne de caractères `ch = "2468"`.

1. Que vaut `len(ch)` ?

**Solution.** `len(ch)` vaut 4.

2. Quel est le type de `ch[1]` ? Quelle est sa valeur ?

**Solution.** `ch[1]` est de type `str` (chaîne de caractères) et vaut "4".

3. Quel est le type de `int(ch[1])` ? Quelle est sa valeur ?

**Solution.** `int(ch[1])` est de type `int` (entier) et vaut 4.

4. Que renvoient `ch[1]+ch[3]` et `int(ch[1])+int(ch[3])` ?

**Solution.** `ch[1]+ch[3]` renvoie la chaîne de caractères "48" et `int(ch[1])+int(ch[3])` renvoie l'entier 12.

5. Que renvoie `ch + "2"` ?

**Solution.** `ch + "2"` renvoie la chaîne de caractères "24682".

6. Que renvoie `ch * 2` ?

**Solution.** `ch * 2` renvoie la chaîne de caractères "24682468".

7. Que renvoient `ch[:2]`, `ch[2:]` et `ch[1:3]` ?

**Solution.** `ch[:2]` renvoie la chaîne de caractères "24", `ch[2:]` renvoie la chaîne de caractères "68" et `ch[1:3]` renvoie la chaîne de caractères "46".

8. Que renvoie `ch[:2]+ch[3:]` ?

**Solution.** `ch[:2]+ch[3:]` renvoie la chaîne de caractères "248".

**Exercice 3.** On considère la liste de listes

`L = [ [1,2,4,3], [5,6,7,8], [9,10,11,12] ]`.

1. Que vaut `len(L)` ?

**Solution.** `len(L)` vaut 3.

2. Quel est le type de `L[1]` ? Quelle est sa valeur ?

**Solution.** `L[1]` est de type `list` (liste) et vaut `[5,6,7,8]`.

3. Que vaut `len(L[1])` ?

**Solution.** `len(L[1])` vaut 4.

4. Quel est le type de `L[1][2]` ? Quelle est sa valeur ?

**Solution.** `len(L[1][2])` est de type `int` (entier) et vaut 7.

5. Que renvoie `L * 2` ?

**Solution.** `L * 2` renvoie la liste `[ [1,2,4,3], [5,6,7,8], [9,10,11,12], [1,2,4,3], [5,6,7,8], [9,10,11,12] ]`

6. Que renvoie `L[1] + [3]` ?

**Solution.** `L[1] + [3]` renvoie la liste `[5,6,7,8,3]`.

7. Que renvoie `L[1] * 3` ?

**Solution.** `L[1] * 3` renvoie la liste `[5,6,7,8,5,6,7,8,5,6,7,8]`.

8. Que renvoie `L[1][2] + 3` ?

**Solution.** `L[1][2] + 3` renvoie l'entier 10.

9. Que renvoie `L[1][2] * 3` ?

**Solution.** `L[1][2] * 3` renvoie l'entier 21.

10. Que renvoie `L + [13,14,15,16]` ?

**Solution.** `L + [13,14,15,16]` renvoie la liste `[ [1,2,3,4], [5,6,7,8], [9,10,11,12], 13, 14, 15, 16 ]`.

11. Quel est l'effet sur `L` de l'instruction `L.append([13,14,15,16])` ?

**Solution.** `L.append([13,14,15,16])` ajoute la liste `[13,14,15,16]` à la fin de `L` donc transforme `L` en la liste `[ [1,2,3,4], [5,6,7,8], [9,10,11,12], [13,14,15,16] ]`.

12. Que renvoie `[-3,-2,-1,0] + L` ?

**Solution.** `[-3,-2,-1,0] + L` renvoie la liste `[-3, -2, -1, 0, [1,2,3,4], [5,6,7,8], [9,10,11,12] ]`.

13. Quel est l'effet sur `L` de l'instruction `L.insert(0, [-3,-2,-1,0])` ?

**Solution.** `L.insert(0, [-3,-2,-1,0])` ajoute la liste `[-3,-2,-1,0]` au début de la liste `L` donc transforme `L` en la liste `[-3, -2, -1, 0, [1,2,3,4], [5,6,7,8], [9,10,11,12] ]`.

14. Quel est l'effet sur `L` de l'instruction `L.pop(1)` ?

**Solution.** L'instruction `L.pop(1)` supprime le deuxième élément de `L` dont transforme `L` en la liste `[ [1,2,3,4], [9,10,11,12] ]`.