

TP Informatique 3

Exercice 1

Expérimenter les saisies suivantes :

1. `[1,2,3]`, `[1, "0", True, 2.5]`, `[[1,2],3]`
2. `type([])`, `type([1,2,3])`, `type([1,True,"Vrai"])`
3. `[]`, `[]*3`, `[[] ]`, `[[] ]*3`
4. `[1]*5`, `[1,0]*5`
5. `[0,1,2]+[3,4,5]`

Exercice 2

Expérimenter les saisies suivantes :

1. `a=[1,3,5,7,9,11]`, `type(a)`, `len(a)`
2. `a[0]`, `a[-1]`, `a[:3]`, `a[3:]`, `a[1:4]`, `a[::2]`, `a[1::2]`, `a[::-1]`
3. `2 in a`, `3 in a`, `[3] in a`

Exercice 3

Expérimenter les saisies suivantes :

1. `list("bonjour")`, `list((1,2,3))`
2. `list(range(10))`, `list(range(1,20,2))`, `list(range(0,20,2))`
3. `list(range(10,0,-1))`, `list(range(11,0,-2))`

Exercice 4

1. Dans l'éditeur, saisir `n= ...` un entier de son choix.
2. Construire et afficher la liste constituée des entiers de 0 à n puis qui redescend à 0.

Exercice 5

Expérimenter les saisies suivantes :

1. `a=[1]`, `id(a)`, `a.append(2)`, `a`, `id(a)`, `a.pop()`, `a`, `id(a)`
2. `a=[1]`, `id(a)`, `a+=[2]`, `a`, `id(a)`, `a.pop()`, `a`, `a.append([2])`, `a`, `id(a)`
3. `a=[]`, `a.append(1)`, `a.append([2,3])`, `a`, `1 in a`, `2 in a`, `[2] in a`, `[2,3] in a`
4. `a=[1,3,5,7,9]`, `a[0]=0`, `a`, `a[-1]==9`
5. `a="bonjour"`, `b=list(a)`, `b`, `b.sort()`

Exercice 6

1. Dans l'éditeur saisir `a=...` un nombre à 10 chiffres de son choix.
2. Programmer l'affichage de `True` si le nombre `a` contient tous les chiffres de 0 à 9 et `False` sinon.

Exercice 7

Expérimenter les saisies suivantes :

1. `a=[1,2,3], b=a, b[0]=0, b, a, id(a), id(b)`
2. `a=[1,2,3], b=a[:], c=list(a), id(a), id(b), id(c), a[0]=0, b[0]=4, a, b, c`

Exercice 8

Expérimenter les saisies suivantes :

1. `a=[[1],2], b=list(a), a.append(0), a, b, a[0].append(3), a, b`
2. `from copy import deepcopy, a=[[1],2], b=deepcopy(a), a[0].append(3), a, b`

Exercice 9

Expérimenter les saisies suivantes :

1. `a=[]*3, a, a[0].append(1), a`
2. `a=[[],[],[]], a, a[0].append(1), a`
3. `a=[[]*3]*2, a, a[0].append(1), a`
4. `a=[[]*3,[0]*3], a, a[0].append(1), a`

Exercice 10

Construire efficacement les listes suivantes :

1. `[0,0,0,0,0], [0,1,0,1,0,1,0,1], [1,0,1,0,1,0,1,0,1]`
2. `[0,1,2,3,4,5], [1,3,5,7,9,11], [10,8,6,4,2,0]`
3. `[[1,1,1,1,1,1,1,1,1,1], [1,1,1,1,1,1,1,1,1,1]]`
4. `[[0],1,[0],1,[0],1,[0],1]`

Pour les listes contenant des sous-listes, on s'assurera que celles-ci sont indépendantes entre elles et non des étiquettes sur un même objet. On pourra utiliser la copie en profondeur pour la dernière question.

Exercice 11

1. Dans l'éditeur, saisir `a=...` une liste de son choix mais sans sous-liste.
2. Construire et afficher la liste `b` qui contient les éléments de `a` dans le même ordre mais dédoublés. Par exemple, si `a=[1,3,2]`, la liste `b` doit contenir `[1,1,3,3,2,2]`.