

Bases de la programmation impérative

Ensimag 1A — feuille 1

1. Exécution pas à pas

On considère le programme suivant :

```
1 a = 5
2 b = 3
3 c = a + b/2
4 a += c
5 b, d = 4, b + c
6 print("valeurs de a,b,c,d:", a, b, c, d)
```

1.1. Exécuter le programme en modifiant sur un dessin l'état des différentes variables.

2. Échange

On considère le code suivant, censé échanger les contenus des variables a et b .

```
1 a = 42
2 b = 24
3 # on échange
4 tmp = a
5 b = tmp
6 a = b
7 print("a vaut", a, "b vaut", b)
```

2.1. Exécuter le code pas à pas.

2.2. Changer l'ordre des instructions pour obtenir un code correct.

2.3. Réécrire le code d'échange en une seule ligne.

3. Types

Exécuter le code suivant :

```
1 a, b = "1", "3"
2 print(a+b)
3 a, b = 1, 3
4 print(a+b)
5 a, b = "1", 3
6 print(a+b)
```

4. Factorisation

On considère le code suivant :

```
1 a = 1
2 b = 12
3 c = 5
4 d = 7
5
6 e = (a+b)/2
7 f = (c+d)/2
8 g = (e+f)/2
```

- 4.1. On souhaite éviter le code redondant. Écrire une fonction *moyenne* renvoyant la moyenne de 2 nombres et modifier le code pour l'utiliser.
- 4.2. Quels sont les intérêts d'un tel changement ?

5. Énigme

```
1
2 def e1(a):
3     b = 2*a
4     return a, b
5
6
7 def m(aa, bb, cc, dd):
8     return (aa+bb)/2, (cc+dd)/2
9
10
11 def op(aa, bb, cc, dd):
12     return aa + 2*cc, bb + 2*dd
13
14
15 def enigme(a, b):
16     aa, ab = e1(a)
17     ba, bb = e1(b)
18     cc, dd = m(aa, ba, ab, bb)
19     cc, dd = cc - aa, dd - ab
20     aa, ab = op(aa, ab, cc, dd)
21     ba, bb = op(ba, bb, -cc, -dd)
22     return aa, ba
```

- 5.1. Renommer les variables pour éviter les confusions ?
- 5.2. Que renvoie la fonction enigme (en fonction de *a* et *b*) ?