

Variables – corrigés des exercices

Exercice 1

1. 5

8

2. Il fallait faire attention à passer à la ligne dans la réponse à 1. Il fallait aussi comprendre le système d'affectation : dans `x + 3`, on ajoute 3 à la valeur précédente de `x` (5), puis qu'on affecte le tout à `x`.

Exercice 2

La fonction `type` permet d'identifier le type d'une variable.

Quels sont les affichages produits par le script ci-dessous ?

```
1 x = 2020*2
2 print(x, type(x)) # affiche : 4040, <class 'int'>
3
4 x = 2020.
5 print(x, type(x)) # affiche : 2020.0 <class 'float'>
6 # 2020. == 2020.0 et "en français" : 2020,0
7
8 x = 1+1<=2
9 print(x, type(x)) # affiche : True <class 'bool'>
10
11 x = 2019/2
12 print(x, type(x)) # affiche : 1009.5 <class 'float'>
13 # / est l'opérateur de la division décimale
14
15 x = 2019//2
16 print(x, type(x)) # affiche : 1009 <class 'int'>
17 # // est l'opérateur de la division euclidienne (entière)
18
19 x = 10 % 3
20 print(x, type(x)) # affiche : 1 <class 'int'>
21 # % se dit "modulo" et est l'opérateur donnant le reste de la division euclidienne
22
23 x = "nsi"
24 print(x, type(x)) # affiche : nsi <class 'str'>
25
26 x = [2, 4]
27 print(x, type(x)) # affiche : [2, 4] <class 'list'>
28
29 x = {"nsi1": 24, "nsi2": 24}
30 print(x, type(x))
31
32 x = (2022, 2023)
33 print(x, type(x)) # affiche : {'nsi1': 24, 'nsi2': 24} <class 'dict'>
```

Exercice 3

```
1 x = int(1.999)
2 print(x) # affiche 1
3 s = str(3.14)
4 t = s+s
5 print(t) # affiche '3.143.14'
```

Exercice 4

1. $5 - 3 - 2$ affiche 0 et $1 / 2 / 2$ affiche 0.25. On peut en déduire qu'en Python, la priorité des opérateurs est la même qu'en mathématiques.
2. (a) $1 + (2 * 3) - 4$
(b) $1+(2 * 4*3)$
(c) $1-a+(a*(a/2))-(a*a*(a/6))+(a*a*a*(a/24))$
3. (a) $1+2*(3-4)$
(b) $1+2+5*3+4$
(c) $1-(2-3+4)+5-6+(7-8)/2$

Exercice 5

1. À la fin de ce programme, **a** et **b** ont la même valeur, 6.

```
1 a = 5 # affecte la valeur 5 à la variable a
2 b = 6 # affecte la valeur 6 à la variable b
3 a = b # affecte la valeur de b, soit 6, à la variable a
4 b = a # affecte la (dernière) valeur de a, soit 6, à la variable b
```

2. À la fin de ce programme, **a** a pris la valeur de **b** et réciproquement en utilisant une variable dite « tampon ».

```
1 a = 7 # affecte la valeur 7 à la variable a
2 b = 8 # affecte la valeur 8 à la variable b
3 tmp = a # affecte la valeur de a (7) à la variable tmp
4 a = b # affecte la valeur de b (8) à la variable a
5 b = tmp # affecte la valeur de tmp (7) à la variable b
```

Exercice 6

1. (a) `//`
(b) `%`
(c) `/`
2. (a) Ce programme n'est correct que lorsque **nb_oeufs** est multiple de 6.
(b) Ce programme n'est correct que lorsque **nb_oeufs** n'est pas multiple de 6.
(c) **Une** solution correcte serait d'écrire : `nb_boites = (nb_oeufs + 5) // 6`. Par la suite, nous verrons d'autres façons plus claires d'un point de vue programmation. Peut-être que ces solutions vous appellent déjà de manière intuitive.