

Épisode V : Premiers pas en Python

EXERCICE 1

Que contiennent les variables x , y , z après les instructions suivantes ?

```
x = 6
y = x + 3
x = 3
z = 2 * y - 7
```

	étape 1	étape 2	étape 3	étape 4
x				
y				
z				

EXERCICE 2

L'instruction $i = i + 1$ a-t-elle un sens ; si oui lequel ? Et $i + 1 = i$?

EXERCICE 3

Que contiennent les variables x , y , z après les instructions suivantes ?

```
x = 6
y = 7
z = x
z = z + y
```

	étape 1	étape 2	étape 3	étape 4
x				
y				
z				

EXERCICE 4

Quel est le résultat de l'instruction $x = 2 * x$? Si la valeur initiale de x est 1, donner les valeurs successives de x après une, deux, trois, etc. exécutions de cette instruction.

	étape 1	étape 2	étape 3	étape 4	étape 5	étape 6	étape 7	...	étape n
x	1								

EXERCICE 5

Parmi les codes suivants, quels sont les programmes Python qui s'exécutent sans causer d'erreur ? Indiquer les erreurs dans les codes erronés.

```
x = 1
x = y + 1
y = 2
```

```
y = 2
x = 1
x = y +- 2
```

```
y = 2
x = 1
x = y + 3
```

```
x = 1
y = 0
x + y = 1
```

```
y = 2
x = 1
x = y +/- 1
```

```
y = 2
x = 1
y = x --+ 1
```

EXERCICE 6

Écrire une suite d'instructions permettant d'échanger le contenu de deux variables a et b .

EXERCICE 7

Taper les instructions suivantes et prenez le temps d'expliquer précisément l'évolution des variables pendant l'exécution pas à pas de ces instructions. Les parties à droite des dièses # ne sont que des commentaires pour vous, l'ordinateur ne les interprétera pas même si vous les tapez.

```
x = 11 * 34
x = 13.4 - 6
y = 13 / 4 # division avec virgule
y = 13 // 4 # division entiere, notez la difference
z = 13 % 2 # pourquoi cela vaut-il 1 ?
x = 14 % 10 # quel sens donner au reste d'une division par 10 ?
y = 14 // 10
i = x + y
```

x															
y															
z															
i															

EXERCICE 8

Après avoir défini la fonction f comme ci-dessous,

```
# definition de la fonction f
def f(x):
    a = x+1
    return a*a + x + 1
```

taper les instructions suivantes qui appellent cette fonction en lui passant différents arguments et prenez le temps d'expliquer en détail les résultats obtenus.

```
x = 0
y = f(2)
t = 4
y = f(t) # on passe la valeur d'une variable
y = f(1) + f(2) # on effectue deux appels
z = x+1
y = f(z)
y = f(x+1) # on passe directement la valeur d'une expression
z = f(x-t)
t = f(t) # on peut meme passer la variable qui servira a
# stocker le resultat
x = f(f(1)) # on peut combiner deux appels, le resultat de
# l'un est passe en parametre a l'autre
```

EXERCICE 9

Parmi les codes suivants, quels sont les programmes Python qui ne comportent pas d'erreur? Rayer les codes erronés.

```
Def fun(x):
    return x + 1

y=fun(3)
```

```
def fun(y):
    return x + 1

y=fun(3)
```

```
def fun(x):
    return x + 1

y=fun(3)
```

```
def fun(x):
    return x + 1

y=fun(3)
```

```
def fun(x)
    return x + 1

y=fun(3)
```

```
def fun():
    return 2

y=fun(3)
```



EXERCICE 10

Écrire une fonction `discriminant(a, b, c)` qui retourne le discriminant du trinôme du second degré $ax^2 + bx + c$.



EXERCICE 11

1. Écrire une fonction `moyenne(a, b)` qui retourne la moyenne des deux nombres `a` et `b`. Testez-la avec les arguments 42 et 23.
2. Écrire une fonction `moyennePonderee(a, coef_a, b, coef_b)` qui retourne la moyennne pondérée par le coefficient `coef_a` pour la note `a` et par le coefficient `coef_b` pour la note `b`. Testez-la en appelant `moyennePonderee(5, 2, 12, 3)`.