

T.P. I - Lecture de code (I)

Code Capytale : e602-764562

I - Modification du contenu des variables

L'instruction `x = 2` permet de **stocker** l'entier 2 dans la variable qui porte le **nom** `x`. Dans la suite du code, les calculs seront effectués en remplaçant la lettre `x` par la valeur 2.

```
x = 2
y = 3 * x + 1
print("x", x)
print("y", y)
```

qui affiche

```
x 2
y 7
```

Remarquons que la quantité à **droite** du signe `=` est **calculée** puis stockée dans la variable qui porte le nom indiqué à **gauche** du signe `=`. Ainsi, la modification d'une variable **après** un calcul n'affecte **que** la variable qui est modifiée.

```
x = 2
y = 3 * x + 1
x = 4
print("x", x)
print("y", y)
```

qui affiche

```
x 4
y 7
```

Exercice 1. (Variables) Pour chacune des questions suivantes, indiquer les variables créées ainsi que leur contenu à l'issue de la suite d'instructions suivantes. Effectuez le raisonnement sur une **feuille de papier puis vérifiez** en évaluant le code.

1.

```
a = 12
b = 10
c = a
a = b
b = c
print("a", a)
print("b", b)
print("c", c)
```

2.

```
a = 12
b = 10
a = b
b = a
print("a", a)
print("b", b)
```

3.

```
a = 2
b = 4
c = a
b = a
c = b - a
print("a", a)
print("b", b)
print("c", c)
```

4.

```
a = 4
b = 2
c = b
b = a
print(..., ...)
print(..., ...)
print(..., ...)
```

5.

```
a = 4
b = 2
c = b
b = a
a = c
print(..., ...)
print(..., ...)
print(..., ...)
```

6.

```
x = 5
y = 2 * x + 3
x = 100
print(..., ...)
print(..., ...)
```

7.

```

a = 3
b = 4
x = a
a = b
b = x
print (...)
print (...)
print (...)

```

II - Définition de fonctions

Le mot-clef **def** permet de définir une fonction dont on précise les paramètres entre parenthèses. L'indentation (c'est-à-dire le décalage par rapport au début de la ligne) permet de délimiter le début et la fin de la définition. L'instruction **return** permet de préciser la valeur renvoyée lors de l'appel de la fonction. Par exemple, la fonction suivante correspond à la fonction mathématique $f : x \mapsto 2x + 3$

```

def f(x):
    return 2 * x + 3

```

On peut ensuite évaluer cette fonction en choisissant différents paramètres :

```

print("f(4)", f(4))

x = 12
print("f(12)", f(x))

y = 20
print("f(20)", f(y))

```

qui affiche

```

f(4) 11
f(12) 27
f(20) 43

```

Exercice 2. (Fonctions) Déterminer le contenu de la variable **x** à l'issue de la suite d'instructions suivante. Effectuez le raisonnement sur une **feuille de papier puis vérifiez** en évaluant le code ci-dessous.

1.

```

x = 5
def fct1(y) :
    return 3 * y + 25

x = fct1(x)

```

2.

```

x = 5
def fct1(x) :
    return 3 * x + 25

x = fct1(x)

```