

# T.P. III - Variables & Fonctions

Code Capytale : e602-764562

## I - Ce qu'il faut savoir

## II - Modification du contenu des variables

### II.1 - Les variables

#### Solution de l'exercice 1.

1. Les valeurs affichées sont

```
a 10  
b 12  
c 12
```

2. Les valeurs affichées sont

```
a 10  
b 10
```

3. Les valeurs affichées sont

```
a 2  
b 2  
c 0
```

4. En complétant les lignes

```
print("a", a)  
print("b", b)  
print("c", c)
```

les valeurs affichées sont

```
a 4  
b 4  
c 2
```

5. En complétant les lignes

```
print("a", a)  
print("b", b)  
print("c", c)
```

les valeurs affichées sont

```
a 2  
b 4  
c 2
```

6. En complétant les lignes

```
print("x", x)  
print("y", y)
```

les valeurs affichées sont

```
x 100  
y 13
```

7. En complétant les lignes

```
print("a", a)  
print("b", b)  
print("x", x)
```

les valeurs affichées sont

```
a 4  
b 4  
c 2
```

□

## III - Fonctions

### III.1 - Définition

#### Solution de l'exercice 2.

1. Comme `x` contient la valeur 5, l'appel `fct1(x)` renvoie la valeur 40 qui est stockée dans `x`.

Finalement, `x` contient la valeur 40.

2. Comme `x` contient la valeur 5, l'appel `fct1(x)` renvoie la valeur 40 qui est stockée dans `x`.

Finalement, `x` contient la valeur 40. □

### III.2 - Fonctions de plusieurs variables

### III.3 - Tracé de courbes

#### Solution de l'exercice 3.

1. On utilise les fonctions usuelles de Python :

```
def u(n):
    return 4**n / (3 * n**3 - 2 * n + np.exp(n))
```

2. On utilise la fonction définie précédemment :

```
print("u_10", u(10))
print("u_35", u(35))
```

3. Comme les abscisses sont entières, on utilise la fonction `range`.

```
import matplotlib.pyplot as plt

# La variable X stocke la liste des entiers de 0 à 25
X = range(0, 26)

# La variable Y stocke la liste des valeurs de u(n)
# pour n variant dans les entiers de 0 à 25
Y = [u(n) for n in X]

# Creation d une nouvelle figure
```

```
plt.clf()
# Trace les points avec des losanges (d) orange
plt.plot(X, Y, 'd', color="orange")
# Affiche le graphique
plt.show()
```

□