

Exercice 1

Utiliser une fonction

1.1 `return P * X / 100`

1.2 Place des arguments

```
1 pourcentage(15,90)
2 pourcentage(50,2100)
```

Exercice 2

Valeur de retour

2.1 Lignes entre les `"""`

Non ces lignes ne sont pas exécutées. Il s'agit de commentaires, sur plusieurs lignes, jusqu'à la fermeture `"""`

2.2 Affiche 5

Exercice 3

Python turtle

3.1 Boucle fonction

```
1 def rectangle(a,b):
2     for i in range(2):
3         t.forward(a)
4         t.left(90)
5         t.forward(b)
6         t.left
```

3.2 Appel fonction

```
1 rectangle(100,200)
```

3.3 Rectangles emboîtés

```
1 a = 10
2 b = 20
3 while a <= 100:
4     rectangle(a,b)
```

```
5 a = a + 10
6 b = b + 20
```

Exercice 4

Utiliser structure conditionnelle

4.1 solde 30

```
1 def solde_30(prix):
2     return prix * 30 / 100
```

4.2 Structure conditionnelle

```
1 def solde_30(prix):
2     if prix < 30:
3         return prix
4     elif prix < 100:
5         return prix - prix * 30 / 100
6     else:
7         return prix - 30
```

4.3 Appel de la fonction

```
1 solde_30(120)
```

Exercice 5

Volume sphere

5.1 Cube

```
1 def cube(x):
2     return x ** 3
```

5.2 volumeSphere

```
1 import math
2 def volumeSphere(r):
3     return 4 / 3 * math.pi * cube(r)
```

5.3 calcul

```
1 >>> volumeSphere(10)
2 # resultat en cm3
```

```
3 4188.790204786391
```

Exercice 6

Paramètres d'une fonction

```
1 def get_name ():
2     ....
3 def calc_calories (mile, calories_pre_mile):
4     ....
```

Exercice 7

Fonctions et sequences

7.1 séquences

```
1 ['A', 'B', 'C', 'D']
2 D C ['A', 'B']
3 0
4 1
```

7.2 retournement de liste

```
1 def retourne_liste(L):
2     L_inverse = []
3     for i in range(len(L)):
4         elem = L.pop()
5         L_inverse.append(elem)
6     return L_inverse
```

Exercice 8

TP Force d'un mot de passe

```
1 def NbCMin (p):
2     min = 0
3     for c in p:
4         if c.islower():
5             min += 1
6     return min
7 NbCMin("abcDEfg")
8 # affiche 5
```

```
1 def NbCMaj(p):
```

```
2     maj = 0
3     for c in p:
4         if c.isupper():
5             maj += 1
6     return maj
7 NbCMaj("abcDEfg")
8 # affiche 2
```

```
1 def NbCAlpha(p):
2     n_symbol = 0
3     for c in p:
4         if c.isalpha() != True:
5             n_symbol += 1
6     return n_symbol
7 NbCAlpha("abc12345D!!")
8 # affiche 7
```