Python

Introduction à la programmation en Python

K. Boudjelaba
k.boudjelaba@carnus.fr

13 mars 2023

Table des matières

1 Notions fondamentales **2**
1.1 Les entrées et sorties . . . 2
1.2 Déclaration de variables . . . 3
1.3 Réalisation de calculs simples . . . 4
1.4 Déclaration de tableaux . . . 5
1.5 Les structures de contrôle . . . 6
1.6 La boucle for . . . 6
1.7 La boucle while . . . 6
1.8 Les courbes . . . 7
1.9 Les fonctions . . . 8

2 Exercices **9**

1 Notions fondamentales

1.1 Les entrées et sorties

Sortie : affichage à l'écran

```
print("Message")
print("La valeur de la variable x est :", x)
```

Entrée : saisie au clavier

```
a = int(input("Veuillez saisir la valeur de a "))
b = float(input("Veuillez saisir la valeur de b "))
c = input("Veuillez saisir c ")
```

Exemples

```
print("Phrase à afficher")
```

```
Phrase à afficher
```

```
y = 5;
print("y = ",y)
```

```
y =  5
```

```
c = input("Veuillez saisir une chaîne de caractères : ")
print(c)
```

```
Veuillez saisir une chaîne de caractères : Lycée Charles Carnus
Lycée Charles Carnus
```

```
a = int(input("Veuillez saisir la valeur de a "))
print("a =", a)
```

```
Veuillez saisir la valeur de a 12
a = 12
```

```
c = float(input("Veuillez saisir un nombre réel : "))
print("c =", c)
```

```
Veuillez saisir un nombre réel : 3.14
c = 3.14
```

1.2 Déclaration de variables

```
a = 10
b = 5.3
c = [1, 4, 3]
d = "Mot"
e = ["Mot1","Mot2","Mot3"]
f = int(input("Veuillez saisir la valeur de f "))
g = float(input("Veuillez saisir la valeur de g "))
```

Exemples

```
x = [1,4,3]
print("x = ", x)
print("x est de type :", type(x))
```

```
x =  [1, 4, 3]
x est de type : <class 'list'>
```

```
a, b, c, d = 23, 2.5, 't', '5.7'
print("La valeur de a est :", a)
print("a est de type : ", type(a))
print("La valeur de b est :", b)
print("b est de type : ", type(b))
print("La valeur de c est :", c)
print("c est de type : ", type(c))
print("La valeur de d est :", d)
print("d est de type : ", type(d))
```

```
La valeur de a est : 23
a est de type :  <class 'int'>
La valeur de b est : 2.5
b est de type :  <class 'float'>
La valeur de c est : t
c est de type :  <class 'str'>
La valeur de d est : 5.7
d est de type :  <class 'str'>
```

```
import numpy as np
y = np.array([1.2, 5.6, 8])
print("y = ", y)
print("y est de type :", type(y))
```

```
y =  [1.2 5.6 8. ]
y est de type : <class 'numpy.ndarray'>
```

```
z = 'Lycée Charles Carnus'
print("z =", z)
print("z est de type :", type(z))
```

```
z = Lycée Charles Carnus
z est de type : <class 'str'>
```

```
z1 = "Lycée Charles Carnus"
print("z1 =", z1)
print("z1 est de type :", type(z1))
```

```
z1 = Lycée Charles Carnus
z1 est de type : <class 'str'>
```

```
e = []
print("e =", e)
```

```
e = []
```

1.3 Réalisation de calculs simples

```
a = 1.5
b = 3
somme = a+b
soustraction = a-b
multiplication = a*b
division = a/b
puissance = a**b
division_entiere = a//b
reste_division_entiere = a%b
```

```
import numpy as np
c = np.sqrt(9)
d = 2*np.pi
```

Exemples

```
x = 2.5
y = 2
r1 = x+y
print("La somme = ", r1)
r2 = x-y
print("La soustraction = ", r2)
r3 = x*y
print("Le produit = ", r3)
r4 = x/y
print("Le rapport = ", r4)
r5 = x//y
print("Le quotient de la division entière = ", r5)
r6 = x%y
print("Le reste de la division entière = ", r6)
r7 = x**y
print("x à la puissance y = ", r7)
```

```
La somme =  4.5
La soustraction =  0.5
Le produit =  5.0
Le rapport =  1.25
```

```
Le quotient de la division entière =  1.0
Le reste de la division entière =  0.5
x à la puissance y =  6.25
```

```
import numpy as np
print("La racine carrée de 25 est :", np.sqrt(25))
p = 2*np.pi*5
print("Le périmètre d'un cercle de rayon 5 est :", p)
print("\nLe résultat peut s'écrire sous la forme : {:.2f} m".format(p))
```

```
La racine carrée de 25 est : 5.0
Le périmètre d'un cercle de rayon 5 est : 31.41592653589793

Le résultat peut s'écrire sous la forme : 31.42 m
```

1.4 Déclaration de tableaux

```
import numpy as np
x = np.arange(fin)
y = np.arange(debut,fin)
z = np.arange(debut,fin,pas)
```

Syntaxe	Signification
np.arange(fin)	génère un tableau [0, 1, ..., fin-1]
np.arange(debut,fin)	génère un tableau [debut, debut+1, debut+2, ..., fin-1]
np.arange(debut,fin,pas)	génère un tableau [debut, debut+pas, debut+2*pas, ..., fin-pas]

```
import numpy as np
x = np.array([1, 4, 3])
m = np.array([1.1, 1.2],[2.1, 2.2])
y = np.arange(8)
z = np.arange(2,10,3)
```

Syntaxe	Signification
np.array([1, 4, 3])	tableau contenant les nombres 1, 4 et 3
np.arange(8)	tableau contenant les nombres 0, 1, ..., 7
np.arange(2,10,3)	tableau contenant les nombres 2, 5 et 8

Exemples

```
import numpy as np
x = np.array([1, 4, 3])
print("x =",x)
m = np.array([[11, 12],[21, 22]])
print("m =",m)
y = np.arange(8)
print("y =",y)
z = np.arange(2,10,3)
print("z =",z)
```

```
x = [1 4 3]
m = [[11 12]
 [21 22]]
y = [0 1 2 3 4 5 6 7]
z = [2 5 8]
```

1.5 Les structures de contrôle

```
a = 1.5
if a == 1.5:
    print('Fin du programme')
```

```
b = 3
if b > 0:
    print("b est un nombre positif non nul")
else:
    print("b est nul ou négatif")
```

```
delta = 2.6
if delta > 0:
    print("delta est positif")
elif delta == 0:
    print("delta est nul")
else:
    print("delta est négatif")
```

1.6 La boucle for

```
for i in range(5):
    print(i)
```

1.7 La boucle while

```
i = 0
while (i<5):
    print(i)
    i = i+1
```

1.8 Les courbes

```
# importation des bibliothèques
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

# génère un vecteur de -2pi à +2pi contenant 256 éléments linéairement espacés
X = np.linspace(    2np    .pi, 2  np  .pi, 256, endpoint=True)

# calcul du cosinus pour les différents éléments
Y = 2  np  .cos(X)

# courbe
plt.figure()
plt.plot(X,Y) # instruction pour tracer la courble
plt.title('Signal sinusoïdal') # titre de la figure
plt.xlabel('Abscisses') # titre (label) de l'axe des abscisses
plt.ylabel('Ordonnées') # titre (label) de l'axe des ordonnées
plt.grid() # activation de la grille
plt.show() # affichage de la figure
```

Exemples

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

X = np.linspace(-3, 3, 256, endpoint=True)
Y = X**2

plt.figure()
plt.plot(X,Y)
plt.title('Signal sinusoïdal')
plt.xlabel('Abscisses')
plt.ylabel('Ordonnées')
plt.grid()
plt.show()
```

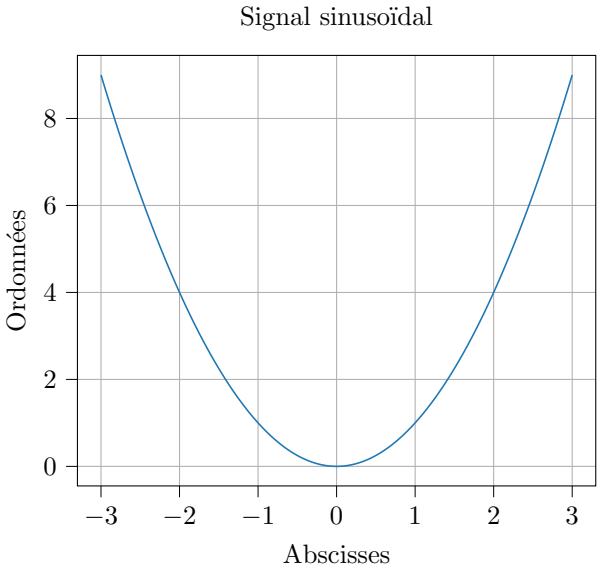

FIGURE 1 – Affichage de la courbe

1.9 Les fonctions

Exemples

```
# importation des bibliothèques
import numpy as np

# définition de la fonction
# (fonction qui calcule la résistance équivalente de 2 résistances associées en
    parallèle)
def ma_fonction(R1,R2):
    return 1/(1/R1+1/R2)

# utilisation (appel) de la fonction
Req = ma_fonction(10,20)
print("La résistance équivalente est :", Req, "Ohms")
```

```
La résistance équivalente est : 6.666666666666666 Ohms
```

2 Exercices

Exercice 1

— Affecter aux variables temps et distance les valeurs 12.348 s et 120.9 m.
— Calculer et afficher la valeur de la vitesse.

Exercice 2

— Saisir un nom et un âge en utilisant l'instruction input().
— Afficher les données saisies.

Exercice 3

— Ecrire un programme qui, à partir de la saisie d'un rayon et d'une hauteur, calcule le volume d'un cône droit : $V = \frac{1}{3}\pi r^2 h$
— Comparer la précision de calcul avec votre calculatrice ou celle de l'ordinateur.

Exercice 4

— Ecrire un programme pour saisir un nombre réel.
  — S'il est positif ou nul, afficher sa racine,
  — sinon afficher un message d'erreur.

Exercice 5

— Ecrire un programme pour saisir deux nombres entiers.
— Rechercher et afficher le plus petit des deux nombres saisis.

Exercice 6

— Ecrire un programme pour initialiser une liste de trois nombres.
— Rechercher et afficher le plus grand des trois nombres.

Exercice 7

— Ecrire un programme qui calcule la moyenne d'une liste en utilisant une boucle for puis une boucle while.
— Ecrire un programme qui calcule la moyenne d'un tableau (vecteur) en utilisant une boucle for puis une boucle while.

Exercice 8

— Ecrire un programme, qui affiche 10 fois "Lycée Charles Carnus" à l'aide de l'instruction for.
— Ecrire un programme, qui affiche 10 fois "Lycée Charles Carnus" à l'aide de l'instruction while.

Exercice 9

— Ecrire un programme qui affiche les nombres de 2 jusqu'à 20 avec un pas de 2 en utilisant une boucle for puis while.

Exercice 10

— Ecrire un programme qui affiche les tables de multiplications de 1 à 10. Aide : utiliser des boucles imbriquées.

Exercice 11

— Ecrire un programme pour tracer la courbe de la fonction $f(x) = x^2$ sur l'intervalle $[-3, 3]$.

Exercice 12

— Ecrire un programme qui utilise une fonction pour réaliser la conversion d'une durée en minutes en une durée en secondes.

Exercice 13

— Ecrire un programme qui utilise une fonction pour calculer le minimum, le maximum et la moyenne d'une liste d'entiers.

Exercice 14

— Lors de la saisie d'un nombre par cast (int (input()) : indiquer une chaîne de caractères en lieu et place d'un nombre, rechercher comment éviter ce bug (aide : commande try)

Exercice 15

Soit la liste : liste = [0, 1, 2, 3, 4, 5]

— Ecrire un programme pour calculer et afficher sous forme d'une liste, le carré de cette liste en utilisant deux méthodes

Exercice 16

Soit la liste : liste = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

— Ecrire un programme pour afficher sous forme d'une liste, les nombres pairs de cette liste en utilisant deux méthodes

Exercice 17

— Ecrire une fonction "puissance" qui a pour arguments x, n et qui retourne le résultat de x^n.

Exercice 18

— Tracer la fonction $y = \dfrac{\sin(x)}{x}$ sur l'intervalle $[0, 20]$

Exercice 19

— Tracer la fonction $y = \dfrac{\sin(x)}{x}$ sur l'intervalle $[-20, 20]$
— Conclure

Exercice 20

— Exécuter les 2 cellules suivantes et conclure

```
def spam(divideBy):
    try:
        return 42 / divideBy
    except ZeroDivisionError as e:
        print('Error: Invalid argument: {}'.format(e))

print(spam(2))
print(spam(12))
print(spam(0))
print(spam(1))
```

```
def spam(divideBy):
    try:
        return 42 / divideBy
    except ZeroDivisionError as e:
        print('Error: Invalid argument: {}'.format(e))
    finally:
        print("-- division finished --")

print(spam(12))
print(spam(0))
```

Exercice 21

— Exécuter les 5 cellules suivantes et conclure

```
spam = [2.1, 5.6, 3.14, 1.05, -7]
spam.index(3.14)
```

```
spam = [2.1, 5.6, 3.14, 1.05, -7]
spam.sort()
spam
```

```
spam.sort(reverse=True)
spam
```

```
age = 21
print('Mineur' if age < 18 else 'Adulte')
```

```
age = 15
print('Enfant' if age < 13 else 'Adolescent' if age < 18 else 'Adulte')
```

Exercice 22

— Corriger les erreurs des cellules suivantes

```
# 1
import numpy as np
y = cos(pi)
print(y)
```

```
# 2
x = 5
if x == 0
    print(x)
```

```
# 3
n = 0
if n > 0 :
    somme = n
carre = somme ** 2
```

```
# 4
4 + 3*ab
```

```
# 5
abs("42.7")
```

```
# 6
print("Début du programme")
 x = 5
```

```
# 7
n = 5
if n=5:
prin(n)
```

```
# 8
x = 2
y = 2x**2 + 2x + 8
z = 2(x**2 + x + 4)
```

```
# 9
liste0 = [0, 1, 2, 3]
liste0[4]
```

```
# 10
def ma_fonction(n) :
    nbc = floor(log10(n)
    return nbc + 1
```

```
# 11
def f1(x):
    return x
def f0():
    return 0
f1(2)
f0(2)
```

```
# 12
n = input("un nombre ? ")
carre = n**2
```

```
# 13
varialbe = 12
carre = variable**2
```

Exercice 23

— Ecrire un programme pour calculer le minimum, le maximum et la moyenne du vecteur a = [1, -2, 3, 0, 1.8, 1], en utilisant des boucles ‘for‘et ‘if‘.

```
a = [1, -2, 3, 0, 1.8, 1]
b = np.array(a) # Conversion en vecteur
```

Exercice 24

Soit le vecteur (ou liste) suivant : a = 2.1, 4.2, 7.3, 1.4, 8.5, 3.6
Ecrire un programme pour afficher les éléments du vecteur :

— 1.4
— 3.6
— 4.2, 7.3
— 2.1, 4.2, 7.3
— 1.4, 8.5, 3.6
— L’indice (rang) du maximum
— L’indice (rang) du minimum

Exercice 25

Soit les vecteurs suivants :
a = 2.1, 4.2, 7.3, 1.4, 8.5, 3.6
b = 10.1, 13.2, 19.3, 16.4, 17.5, 11.6

— Ecrire un programme pour obtenir ce tableau :

2.1	4.2	7.3	1.4	8.5	3.6
10.1	13.2	19.3	16.4	17.5	11.6

— Ecrire un programme pour obtenir ce tableau :

2.1	10.1
4.2	13.2
7.3	19.3
1.4	16.4
8.5	17.5
3.6	11.6

Exercice 26

Soit la matrice (tableau) :

2.1	4.2	7.3	1.4	8.5
10.1	13.2	19.3	16.4	17.5
22.1	28.2	26.3	25.4	29.5
39.1	35.2	37.3	36.4	34.5

Ecrire un programme pour afficher :
— Cette matrice (tableau)
— 26.3
— 29.5
— 35.2
— La 2ème ligne : 10.1, 13.2, 19.3, 16.4, 17.5
— La dernière ligne : 39.1, 35.2, 37.3, 36.4, 34.5
— La 2ème et la 3ème ligne
— La 3ème colonne : 7.3, 19.3, 26.3, 37.3
— La dernière colonne : 8.5, 17.5, 29.5, 34.5
— La 2ème, la 3ème et la 4ème colonne
—

19.3	16.4
26.3	25.4