

41 Exercices En Programmation Orientée Objet PYTHON Pour S'entraîner



Formateur En Applications Informatiques

Patrice REY



Pour une demande de cours personnalisé >> reypatrice@orange.fr



FORMATEUR DÉVELOPPEUR AUTEUR

Informaticien et formateur indépendant,
diplômé en informatique et certifié Microsoft MCTS.
Passionné par les technologies WPF, LINQ et DirectX, spécialiste du
langage C#, programmeur passionné des API 3D, je vous fais
partager ma passion pour le développement et la programmation
des applications informatiques au travers de mes ouvrages.

Patrice REY
33000 BORDEAUX



**41 Exercices
En
Programmation
Orientée Objet
PYTHON**

Patrice REY

© 2023 Patrice REY
email → patricerey33@orange.fr
site web → www.reypatrice.fr
Amazon auteur → amzn.to/3fXLGqa
Linkedin auteur → bit.ly/3fe5YhD
Facebook auteur → bit.ly/3WuEJPs
Demande de cours personnalisé → reypatrice@orange.fr
(C# - WPF - Python - Java - EXCEL)

Édition : BoD - [Books on Demand](https://www.bod.fr), info@bod.fr
Impression : BoD - [Books on Demand](https://www.bod.fr), In de Tarpen 42,
Norderstedt (Allemagne)
Impression à la demande,
ISBN : 978-2-3225-0852-5
Dépôt légal : MAI 2023

Tous les noms de produits ou marques cités dans ce livre
sont des marques déposées par leurs propriétaires respectifs

Ce pictogramme mérite une explication.
Son objet est d'alerter le lecteur sur
la menace que représente pour l'avenir
de l'écrit, particulièrement dans
le domaine de l'édition techni-
que et universitaire, le dévelop-
pement massif du **photo-
copillage**.

Le Code de la propriété
intellectuelle du 1^{er} juillet 1992
interdit en effet expressément la
photocopie à usage collectif
sans autorisation des ayants droit. Or,
cette pratique s'est généralisée dans les



établissements d'enseignement supérieur,
provoquant une baisse brutale des achats
de livres et de revues, au point que la
possibilité même pour les auteurs
de créer des œuvres nouvelles et
de les faire éditer correctement
est aujourd'hui menacée.

Nous rappelons donc que
toute reproduction, partielle ou
totale, de la présente publication
est interdite sans autorisation du
Centre français d'exploitation du
droit de copie (**CFC**, 20 rue des Grands-
Augustins, 75006 Paris).

Table des matières

Le code source de programmation
est téléchargeable gratuitement à l'adresse web
www.reypatrice.fr

Introduction	9
--------------------	---

Logiciels requis pour la programmation en PYTHON

F01 - Installer Anaconda et Jupyter Notebook	17
1 - La distribution Anaconda	17
2 - Prise en main de Jupyter Notebook	23
3 - Comprendre l'interface des notebooks	29
4 - Mon premier notebook	35
F02 - Les usages courants dans l'édition de script	39

Exercices corrigés

S01 - Première série d'exercices	45
Exercice n°1	45
Exercice n°2	47
Exercice n°3	48
Exercice n°4	49
Exercice n°5	53
Exercice n°6	54
Exercice n°7	56
Exercice n°8	58
S02 - Seconde série d'exercices	63
Exercice n°9	63
Exercice n°10	66

Exercice n°11	67
S03 - Troisième série d'exercices	69
Exercice n°12	69
Exercice n°13	71
Exercice n°14	73
Exercice n°15	82
Exercice n°16	84
S04 - Quatrième série d'exercices	87
Exercice n°17	87
Exercice n°18	89
Exercice n°19	92
Exercice n°20	94
Exercice n°21	98
S05 - Cinquième série d'exercices	103
Exercice n°22	106
Exercice n°23	110
Exercice n°24	115
Exercice n°25	120
Exercice n°26	125
Exercice n°27	130
Exercice n°28	135
S06 - Sixième série d'exercices	143
Exercice n°29	145
Exercice n°30	147
Exercice n°31	149
Exercice n°32	151
Exercice n°33	153

Exercice n°34	154
Exercice n°35	155
Exercice n°36	157
Exercice n°37	159
Exercice n°38	161
Exercice n°39	163
Exercice n°40	166
Exercice n°41	168
Index	177

Introduction

Python est un langage de programmation (langage de script) permettant de faire de la programmation impérative (écrire une séquence d'instructions), de la programmation fonctionnelle (résoudre des problèmes en fabriquant des fonctions) et de la programmation orientée objet (définir des objets que l'on fait interagir entre eux).

Ce langage fut inventé en 1989 par **Guido van Rossum** et il est publié en 1991 sous licence libre. On peut le télécharger gratuitement sur le site officiel à l'adresse web <http://www.python.org>, et l'installer sur n'importe quel ordinateur (Windows, Mac et Linux). Il est très utilisé dans le monde scientifique, les universités, les classes préparatoires et l'enseignement en général car il possède de nombreux avantages :

- une syntaxe claire et simple,
- une évolution permanente (la version 3.11 est la version actuelle),
- une certaine puissance de calcul,
- des milliers de bibliothèques dans tous les thèmes divers et variés.

Python est aussi utilisé dans le monde professionnel comme par exemple avec le site Youtube implémenté en Python. Un autre exemple est le *framework* **Django** utilisé par de nombreux professionnels pour faire du développement web en Python. Les milliers de bibliothèques accessibles gratuitement font de ce langage un outil particulièrement puissant. Nous citerons par exemple :

- le module **Pygame** dédié à la création des jeux en 2D,
- le module **Blender** pour faire de la modélisation en 3D,
- le module **PIL** (*Python Imaging Library*) pour réaliser du traitement d'images,
- le module **Scipy** pour les sciences,
- le module **Matplotlib** pour les graphiques,
- le module **Numpy** pour le calcul,
- etc.

Le contenu du livre

Face à une telle richesse et de si nombreuses possibilités, ce livre vous propose un ensemble de 41 exercices corrigés portant sur la programmation orientée objet (POO) en Python. J'utilise ces exercices sur la POO au quotidien pour l'enseignement aux étudiants des **classes préparatoires MPSI, MP2I et PCSI**, et aux étudiants de l'**ENSC** (*École Nationale Supérieure de Cognitique*) pour la partie concernant la programmation orientée objet en Python.

Les fiches **F01** et **F02** commencent par une introduction permettant de voir comment installer les **notebooks Jupyter** à partir d'une distribution **Anaconda 3** avec Python, et quels sont les usages courants dans l'édition des scripts. Les fiches **S01** à **S06** contiennent six séries d'exercices corrigés couvrant les divers concepts de la programmation orientée objet en Python.

exos_coriges_serie_01_08

$$d = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2}$$

expression LaTeX : <http://www.hostmath.com/>

```
In [14]: import math
class Point2d:
    # constructeur
    def __init__(self, x:float, y:float):
        if type(x) is not str:
```

exos_coriges_serie_09_11

Vous devez écrire une classe **Domino** qui formalise un domino à travers de ses deux faces. On personnalisera l'affichage texte pour la représentation de ses deux faces. On documentera le tout comme cela est nécessaire.

```
In [34]: class Domino:
    # classe qui formalise un domino à travers de ses deux faces A et B
    # constructeur avec paramètres
    def __init__(self, a:int, b:int):
```

S01 à S06

exos_coriges_serie_12_16

Ter cas : calculer la vitesse finale pour un objet qui accélère

Un objet commence au repos et accélère à 2 m/s² le long d'une ligne droite de 9 m. Quelle est la vitesse finale de l'objet ?

on calcule la vitesse finale v connaissant les trois autres composantes par la formule :

$$v = \sqrt{u^2 + 2as}$$

v en m/s
 u en m/s
 a en m/s²
 s en m

```
In [3]: import math
class AccelerationDistance:
    # classe pour gérer la formule de l'accélération constante sur une distance
    # constructeur avec paramètres
    def __init__(self, u_vit_fin:float=None, u_vit_ini:float=None, a_acc:float=None):
```

serie_17_21

$(4, 0)$ $(14, 2)$ $(12, 1)$

```
In [16]: class JeuDeCartes():
    # classe qui gère un jeu de 52 cartes
    # constructeur
    # valeur = ("pique", "coeur", "carreau", "trèfle")
    # valeur = (2,3,4,5,6,7,8,9,10,"valet", "dame", "roi", "as")
```

S01 à S06

exos_coriges_serie_17_21

cône de révolution

Le volume d'un cône de révolution est égal à un tiers de l'aire de sa base multipliée par la hauteur du cône. Ce volume est égal aussi au tiers du volume du cylindre droit correspondant.

Définissez une classe **ConeRevolution** qui hérite de la classe précédente **CylindreDroit**. Le constructeur comportera lui aussi deux paramètres que sont le rayon de la base et la hauteur du cône. Cette nouvelle classe comportera sa propre méthode pour le calcul du volume du cône.

```
In [3]: class ConeRevolution(CylindreDroit):
```

Exercice n°23

La multiplication matricielle entre deux matrices ne peut se faire que selon la condition suivante : une matrice de dimension $n \times p$ peut être multipliée par une matrice de dimension $p \times q$ pour donner une matrice de dimension $n \times q$. Cela veut dire que pour une multiplication matricielle de matrices, le nombre de colonnes de la première matrice doit être égal au nombre de lignes de la seconde matrice. La figure ci-dessous est un exemple de multiplication matricielle dans laquelle la matrice 3×3 est multipliée par une matrice 3×1 pour donner une matrice résultante 3×1 .

matrice 3x3		matrice 3x1	=	matrice 3x1																				
<table border="1"> <tr><td>C1</td><td>C2</td><td>C3</td></tr> <tr><td>L1 11,0000</td><td>12,0000</td><td>13,0000</td></tr> <tr><td>L2 21,0000</td><td>22,0000</td><td>23,0000</td></tr> <tr><td>L3 31,0000</td><td>32,0000</td><td>33,0000</td></tr> </table>	C1	C2	C3	L1 11,0000	12,0000	13,0000	L2 21,0000	22,0000	23,0000	L3 31,0000	32,0000	33,0000	*	<table border="1"> <tr><td>C1</td></tr> <tr><td>L1 1,0000</td></tr> <tr><td>L2 2,0000</td></tr> <tr><td>L3 3,0000</td></tr> </table>	C1	L1 1,0000	L2 2,0000	L3 3,0000	=	<table border="1"> <tr><td>C1</td></tr> <tr><td>L1 74,0000</td></tr> <tr><td>L2 134,0000</td></tr> <tr><td>L3 194,0000</td></tr> </table>	C1	L1 74,0000	L2 134,0000	L3 194,0000
C1	C2	C3																						
L1 11,0000	12,0000	13,0000																						
L2 21,0000	22,0000	23,0000																						
L3 31,0000	32,0000	33,0000																						
C1																								
L1 1,0000																								
L2 2,0000																								
L3 3,0000																								
C1																								
L1 74,0000																								
L2 134,0000																								
L3 194,0000																								

Dans la figure ci-dessous, on a la schématisation de la multiplication matricielle $3 \times 3 \times 3 \times 1 = 3 \times 1$. La matrice résultante 3×1 avec ses coefficients $c1$, $c2$ et $c3$ sont obtenus de la façon suivante :

- $a11 \cdot b1 + a12 \cdot b2 + a13 \cdot b3 = c1$
- $a21 \cdot b1 + a22 \cdot b2 + a23 \cdot b3 = c2$
- $a31 \cdot b1 + a32 \cdot b2 + a33 \cdot b3 = c3$

La figure ci-dessous schématise géométriquement l'angle entre les vecteurs $u(+5,+3)$ et $v1(-5,-1)$ qui donne un angle de 160.34 degrés, et schématise géométriquement l'angle entre les vecteurs $u(+5,+3)$ et $v2(+2,+3)$ qui donne un angle de 25.34 degrés.

En reprenant la classe précédente **Vecteur2d**, ajoutez une méthode statique qui retourne l'angle en degrés entre deux vecteurs passés en paramètre.

```
In [1]: class Point2d :
...
# classe qui gère un point
...
# constructeur
def __init__(self, ch:str, pos_x:float, pos_y:float):
```

En reprenant la classe précédente **Vecteur2d**, ajoutez une méthode statique qui retourne les composantes du vecteur projeté résultantes de la projection orthogonale d'un vecteur sur un vecteur support.

```
In [2]: class Point2d :
...
# classe qui gère un point
...
# constructeur
def __init__(self, ch:str, pos_x:float, pos_y:float):
    if isinstance(ch, str) is False:
        raise TypeError("ch doit être de type str")
    if isinstance(pos_x, float) is False:
        raise TypeError("pos_x doit être de type float")
    if isinstance(pos_y, float) is False:
```

S01 à S06

Exercice n°28

Dans le domaine de l'algèbre linéaire, on applique à la figure schématisée ci-dessous deux transformations successives :

- en premier une translation de coefficients $T_x = +8$ et $T_y = +2$.
- en second une rotation de centre l'origine $O(0,0)$ et de valeur d'angle $\beta = -90$ degrés.

Dans le système des coordonnées homogènes, on compose les deux matrices 3×3 relatives aux deux transformations :

Exercice n°27

Dans le domaine de l'algèbre linéaire, on applique une rotation de centre l'origine $O(0,0)$ et de valeur d'angle $\beta = +90$ degrés à la figure schématisée ci-dessous.

Dans le système des coordonnées homogènes, pour définir une matrice de rotation de valeur d'angle β , on part d'une matrice 3×3 dans laquelle on remplace les coefficients suivants :

S01 à S06

Le cercle est une figure géométrique basique. En géométrie euclidienne, un cercle est une courbe plane fermée constituée de points situés à égale distance d'un point nommé centre. Cette distance est appelée rayon du cercle. La figure ci-dessous schématise un cercle de centre $O(0,0)$ et de rayon $r = 5$.

Définissez une classe **Cercle**. Les objets construits à partir de cette classe seront des cercles de tailles variées. En plus du constructeur qui reçoit la valeur d'un rayon, vous définirez une méthode qui devra renvoyer la surface du cercle et une autre méthode qui devra renvoyer la circonférence du cercle. On rappelle que la circonférence d'un cercle correspond à la mesure de son contour, donc de son périmètre.

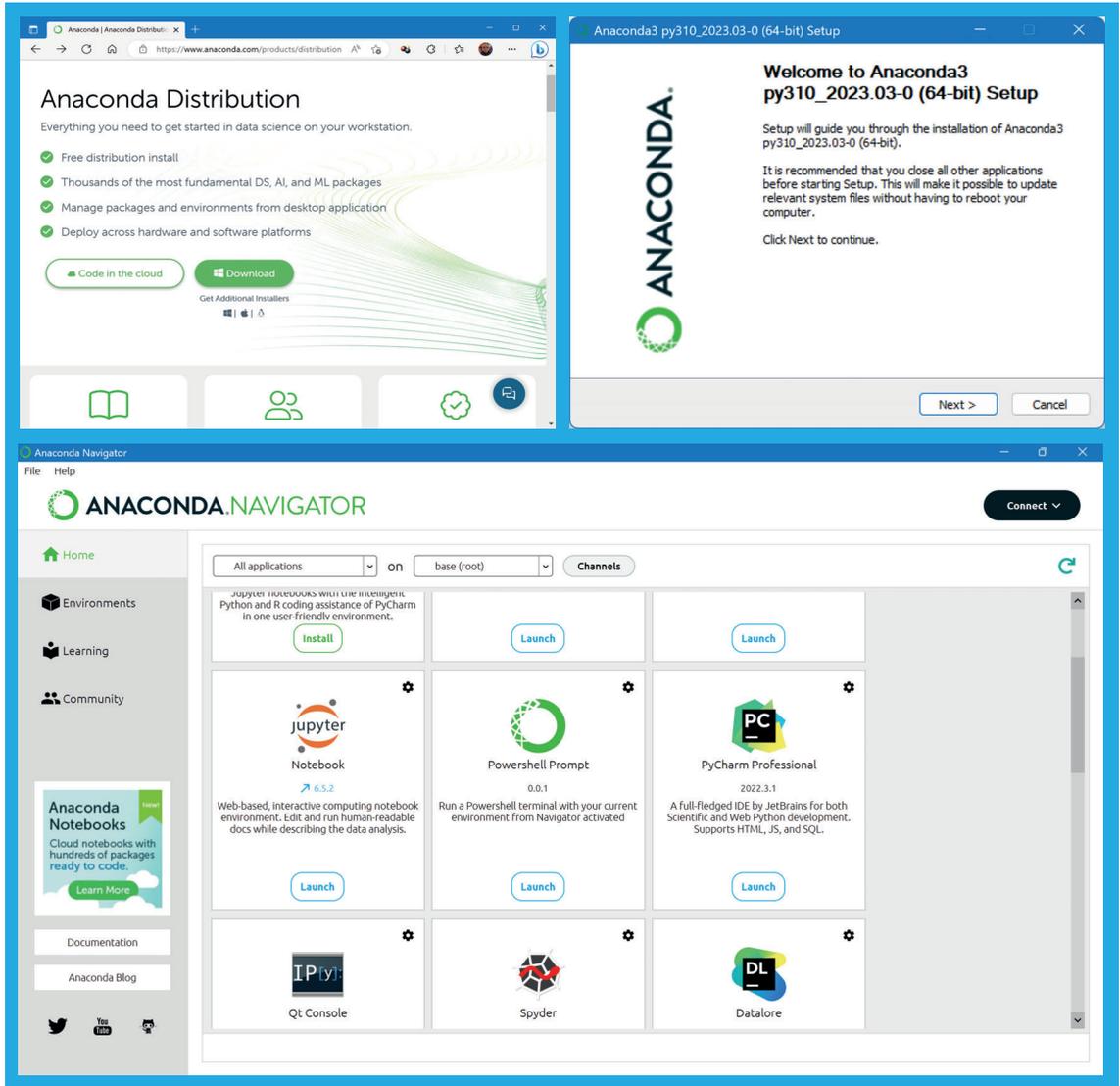
La figure ci-dessous montre le détail du compte bancaire de Mr VERMOULLU, compte référencé par le numéro ds456789. Ce compte dispose d'un solde initial de 1000€. Il y a des retraits spéciaux par des montants négatifs (retrait DAB, etc) et des versements spéciaux par des montants positifs (versement au guichet, virement du salaire, etc). On vous demande de définir une classe **CompteBancaire** qui gère cela au travers de l'identification du compte et des écritures comptables.

compte bancaire	
numéro de compte	ds456789
titulaire	VERMOULLU
solde initial	1 000,00 €
écritures comptables	
retrait DAB	-150,00 €
versement guichet	225,00 €
retrait DAB	-200,00 €
virement salaire	1 250,00 €
solde	875,00 €
	2 125,00 €

```
In [1]: class CompteBancaire:
...
# classe qui gère un compte bancaire
...
# constructeur avec paramètre
def __init__(self, ch_num:str, ch_nom:str, solde_ini:float):
    self_numero = ch_num
    self_titulaire = ch_nom
    self_solde_ini = solde_ini
    self_solde_en_cours = self_solde_ini
# méthode(s)
def __repr__(self):
    ch = "numéro de compte : " + self_numero + "\n"
    ch = "titulaire : " + self_titulaire + "\n"
    ch = "solde en cours : " + str(self_solde_en_cours)
    return ch
# versement sur le compte
def verser(self, montant:float):
```

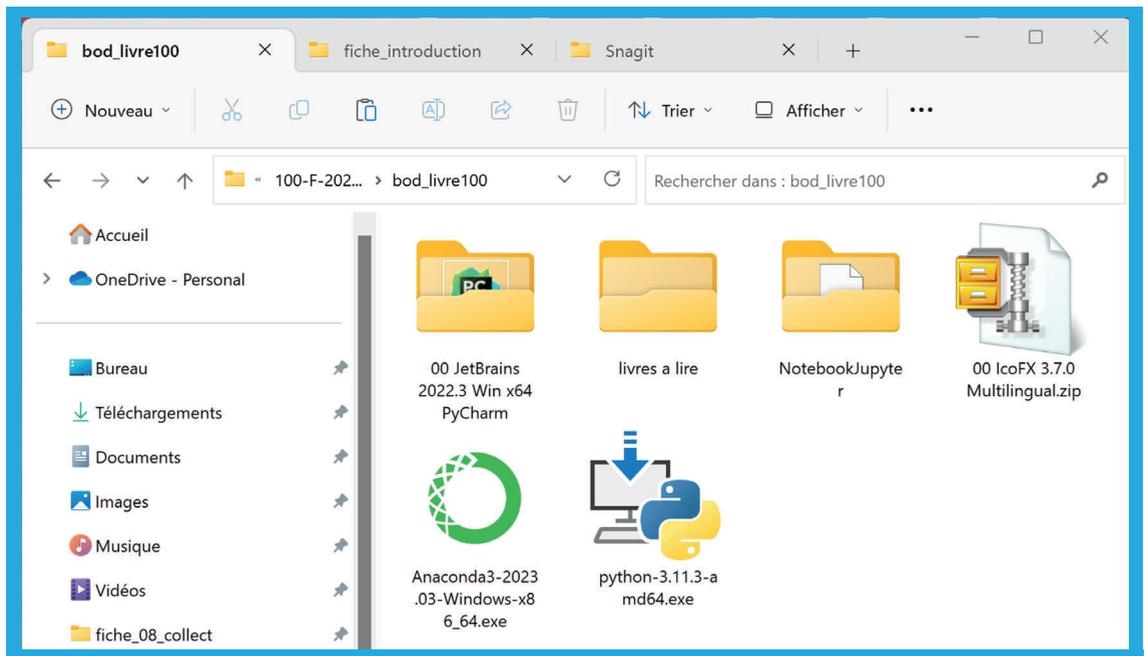
Les logiciels requis pour le développement

Tous les programmes composés dans ce livre utilisent **Jupyter Notebook** au sein d'une distribution **Anaconda 3** avec **Python 3.10** sur la plateforme **Windows 11**.



Les liens de téléchargement

Tout le code source de programmation de cet ouvrage pourra être téléchargé gratuitement à l'adresse web suivante <http://www.reypatrice.fr> en allant sur la fiche du livre. Dans l'archive téléchargée, vous trouverez dans les dossiers tout le nécessaire de travail correspondant aux différentes fiches du livre. De cette façon, vous serez capable de suivre et de programmer assez rapidement et assez facilement. C'est une démarche volontaire, dans un but pédagogique, pour progresser rapidement. Bonne lecture à tous.



**Logiciels requis pour
la programmation
en PYTHON**

Installer Anaconda et Jupyter Notebook

F01

Python est un langage de programmation facile à apprendre car il possède une syntaxe très simple et une documentation très fournie. Les classeurs (*notebooks*) **Jupyter** sont des documents qui permettent d'écrire du texte et du code Python. De plus, Python possède des bibliothèques et fonctions intégrées qui permettent, en quelques lignes de code, d'effectuer des traitements complexes sur les données. Enfin, avec une utilisation dans de nombreux domaines, Python est continuellement maintenu et amélioré, il possède une très grande communauté et une documentation très complète, en anglais comme en français. Toutes ces raisons font de Python un langage de choix pour tout projet concernant l'analyse de données en général.

1 - La distribution Anaconda

Anaconda est simplement une distribution Python et R la plus connue et la plus utilisée notamment en **Data Science** (science des données) car cette plateforme est dédiée à ce domaine précisément. Cette distribution libre Anaconda contient non seulement les environnements Python et R mais aussi toutes les bibliothèques essentielles pour exécuter du code Python. De plus, la distribution Anaconda inclut Jupyter Notebook, RStudio et beaucoup d'autres applications essentielles dédiées à la programmation Python. Dans la première partie de cet ouvrage, nous utiliserons uniquement Jupyter Notebook et le kernel Python sous Anaconda pour composer des notebooks exécutant du code Python.

Cette distribution est très simple à installer, elle est supportée sur Windows, MacOS et Linux, et elle possède une large communauté d'utilisateurs ainsi qu'une documentation très complète (en anglais). Encore mieux, une interface graphique a été développée pour cette distribution qui s'appelle **Anaconda Navigator** et qui permet en quelques clics d'installer de nouveaux packages (ou bibliothèques), de créer de nouveaux environnements, d'accéder aux applications, etc. En vous espérant convaincus, nous allons maintenant installer cette distribution Anaconda sur la plateforme **Windows 11** ainsi que l'interface graphique qui l'accompagne avec **Anaconda Navigator**.

On commence par se rendre à l'adresse <https://www.anaconda.com/products/individual> (figure 1 au repère 1). Puis en bas de page, on clique sur le bouton *Download* pour télécharger la version pour Windows concernant les processeurs à l'architecture *64 bit* (figure 1 au repère 2). On récupère ainsi l'exécutable téléchargé *Anaconda3-2023.03-Windows-x86_64.exe*. On démarre l'exécutable qui affiche la fenêtre d'installation d'Anaconda comme sur la figure 2.

On clique sur *Next* (figure 2), puis sur *I Agree* pour accepter la licence. On coche *Just me* (figure 3 au repère 1). Dans les options avancées, on laisse la case cochée *Register Anaconda3 as the system Python 3.10* par défaut (figure 3 au repère 2) puis on clique sur le bouton *Install* (figure 3 au repère 3). L'installation s'effectue puis on termine en cliquant sur le bouton *Finish* (figure 4).