

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14

{



}



# Machine Learning avec 'Scikit-Learn'

< Formateur : Onesime **MBULAYI** >

## Résumé 'la formation se déroulera en 9 séances'

{

**Sommaire ()** Installation et préparation de l'environnement du travail **Python** et **ANACONDA**.

**Séance 1()** Introduction en Machine Learning

**Séance 2()** Prétraitement et exploration des données

**Séance 3()** Apprentissage supervisé : Regression

}

## Résumé 'la formation se déroulera en 9 séances'

{

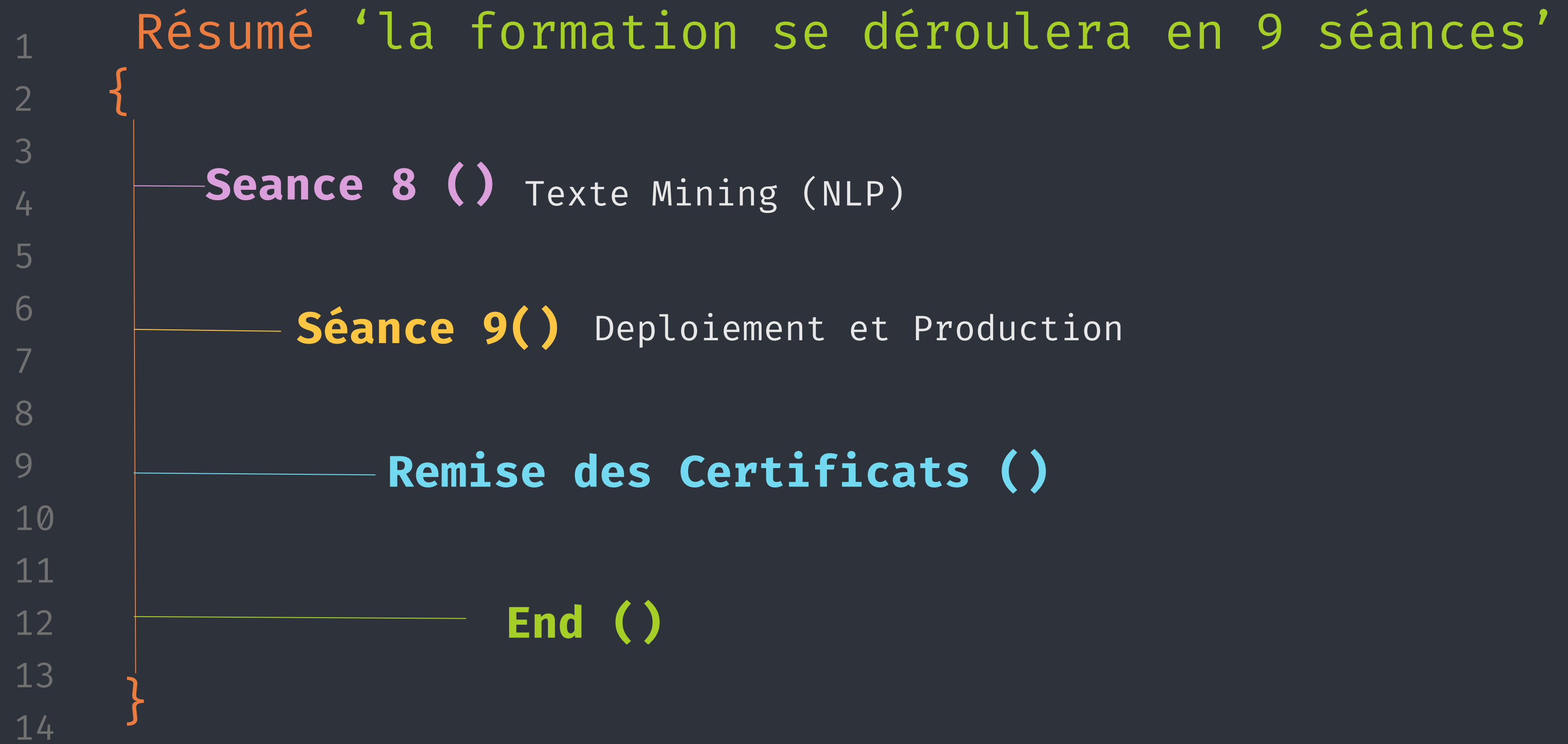
**Séance 4 ( )** Apprentissage supervisé : Classification

**Séance 5 ( )** Apprentissage non Supervisé : Clustering

**Séance 6 ( )** Evaluation et selection des modèles

**Séance 7 ( )** Réduction de la dimensionnalité

}

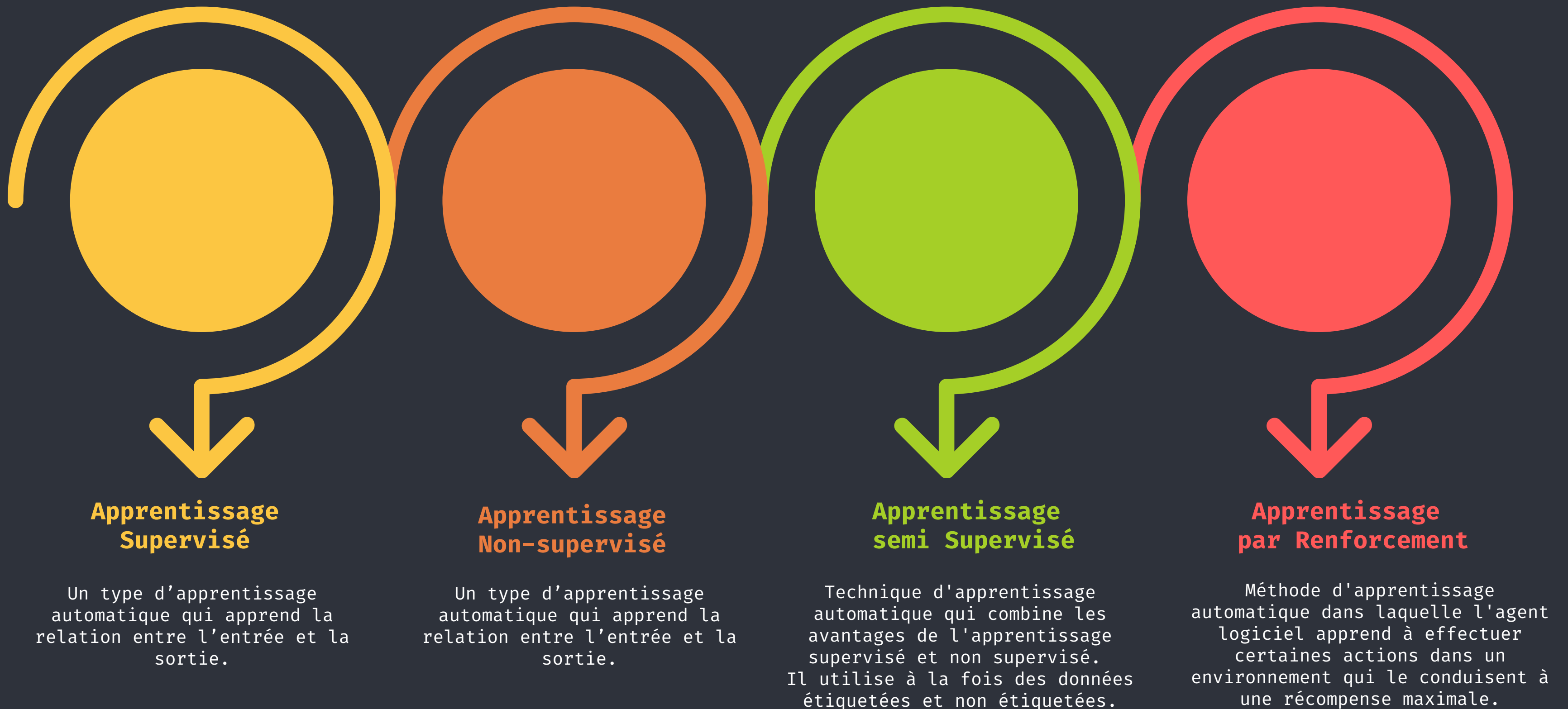


{ Séance 1()

‘Présentation de Machine Learning’

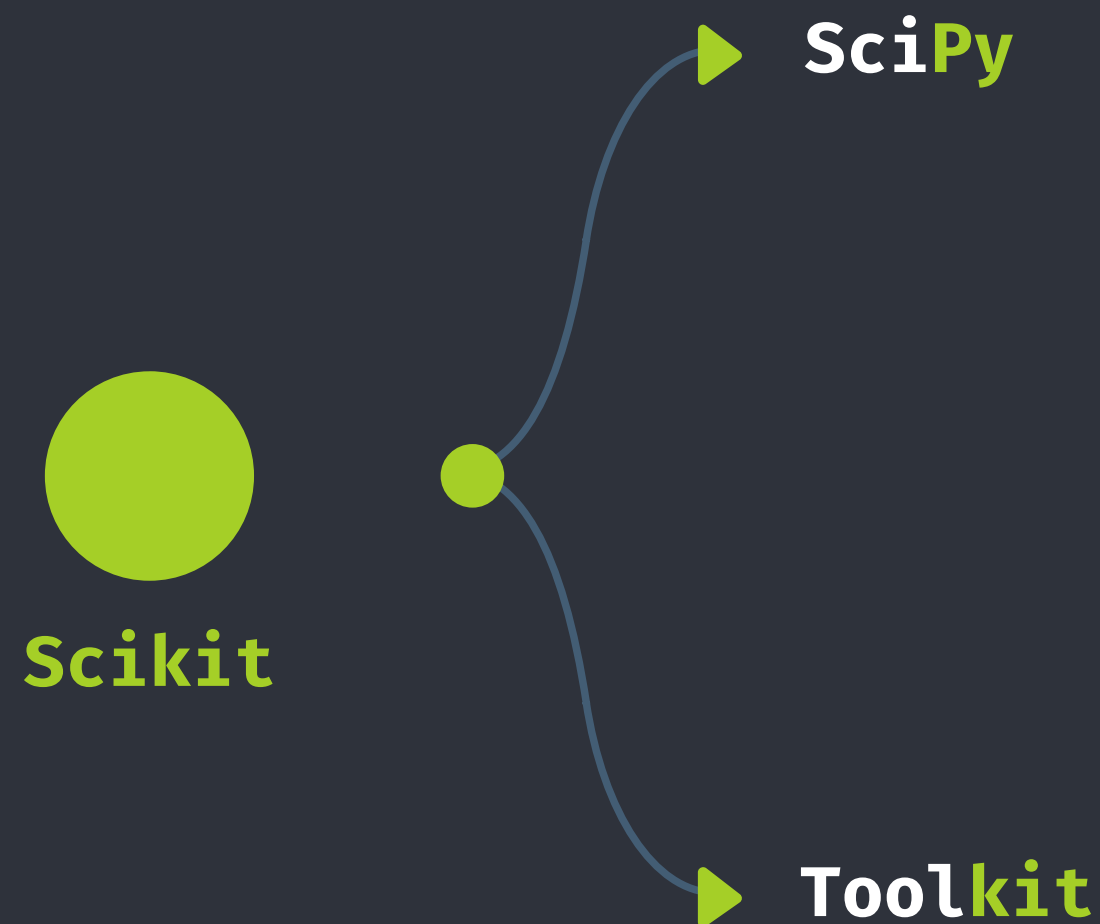
}

L'apprentissage automatique est un sous-domaine de **l'intelligence artificielle** consacré à la compréhension et à la création de méthodes permettant d'imiter la façon dont les humains apprennent. Ces méthodes incluent l'utilisation d'algorithmes et de données pour améliorer les performances sur un ensemble de tâches.



## 'C'est quoi Scikit-Learn ?'

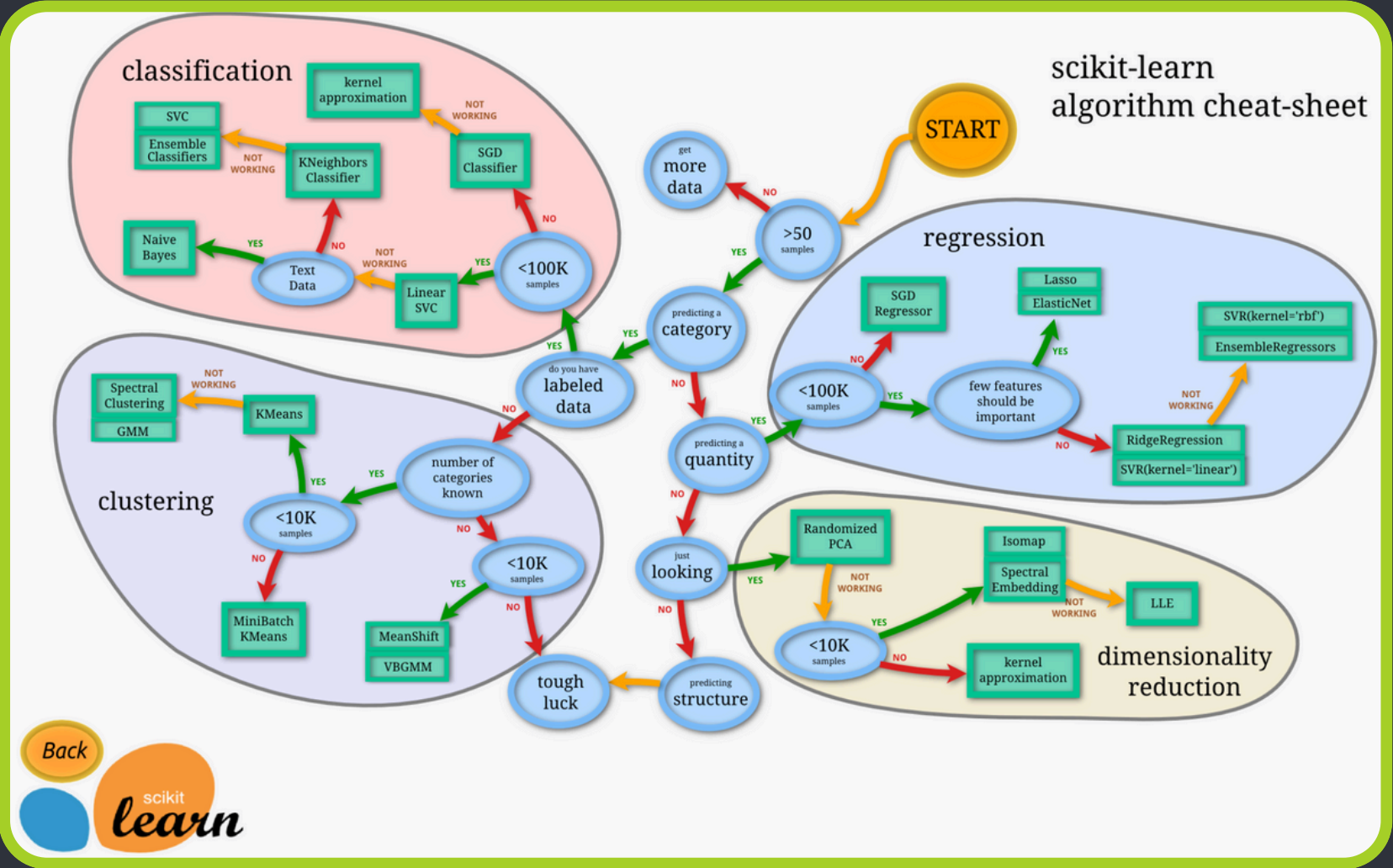
**Scikit-learn**, également connu sous le nom de `sklearn`, est une bibliothèque d'apprentissage automatique Python open source et robuste. Elle a été créée pour simplifier le processus de mise en œuvre de l'apprentissage automatique et des modèles statistiques en Python.



**La bibliothèque** permet aux praticiens de mettre en œuvre rapidement une vaste gamme d'algorithmes d'apprentissage automatique supervisés et non supervisés via une interface cohérente.

**Sklearn** a été construit sur **SciPy** et fonctionne sur tous les types de données numériques stockées sous forme de tableaux NumPy, de matrices creuses SciPy et de tous les autres types de données pouvant être convertis en tableaux numériques tels que **Pandas DataFrames**.

'Ecosysteme de Scikit-Learn ?'





# 'Data?'



Le premier aspect de sklearn que nous allons explorer concerne les données ; Scikit-learn est livré avec des ensembles de données d'apprentissage automatique standard, ce qui signifie que vous n'êtes pas obligé de les télécharger à partir d'un site Web ou d'une base de données externe.

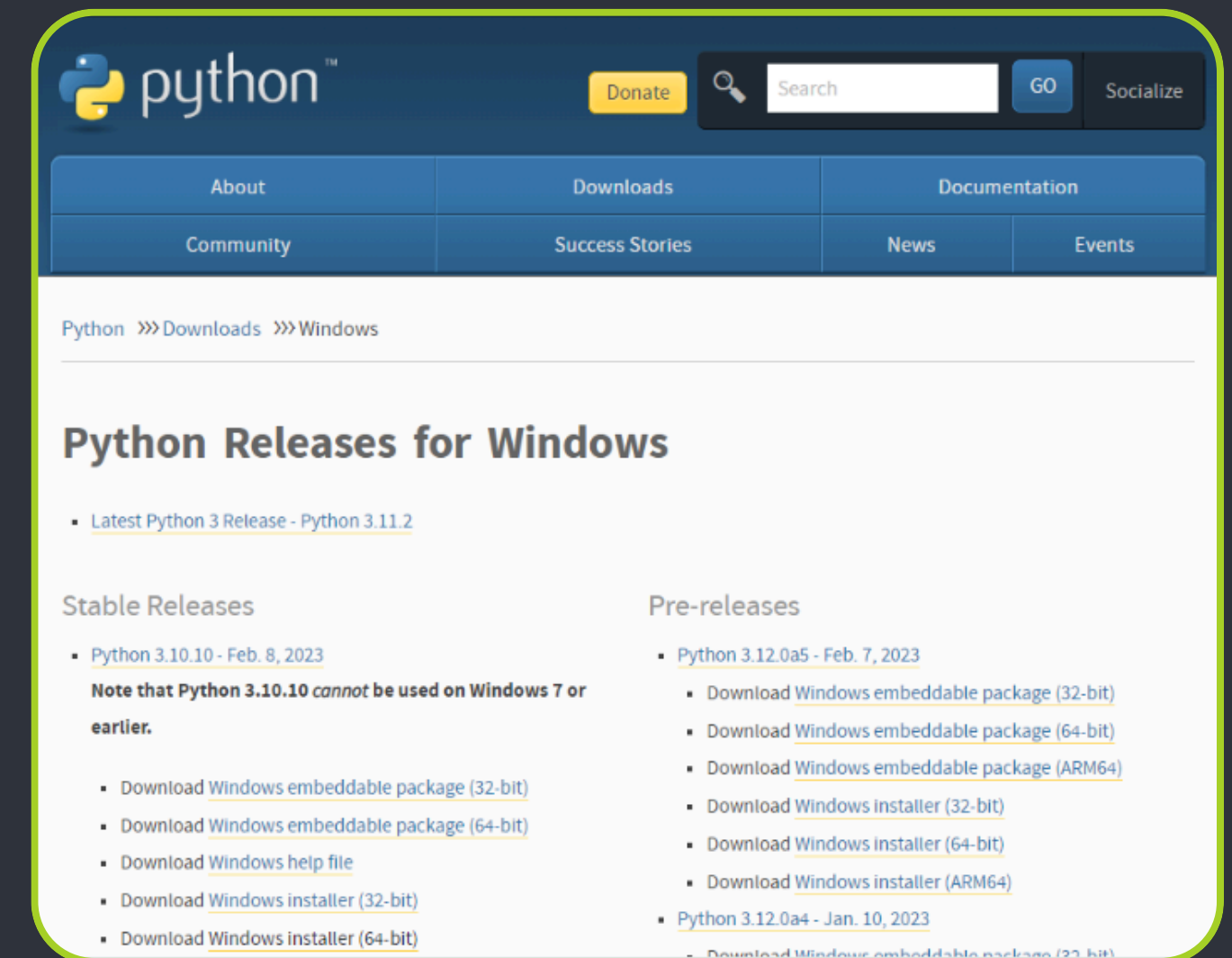
Parmi les exemples de **jeux de données(DATASET)** disponibles dans sklearn, on trouve **Boston, iris, diabetes, digits, linnerud, wine, breast\_cancer, sample\_images**. Par exemple, le data d'iris peut être utiliser pour la classification et le dataset diabète pour la régression.

Pour notre exemple, nous utiliserons **le jeu de données sur le iris.**

## Environnement 'Installation Python';

### Étape 1 : Téléchargement du programme d'installation Python

1. Accéder à l'officiel **Python download page for Windows**.
2. Trouvez une version stable de Python
3. Cliquez sur le lien approprié à votre système pour télécharger le fichier exécutable : programme d'installation Windows (64 bits) ou programme d'installation Windows (32 bits).



## Environnement 'Installation Python';

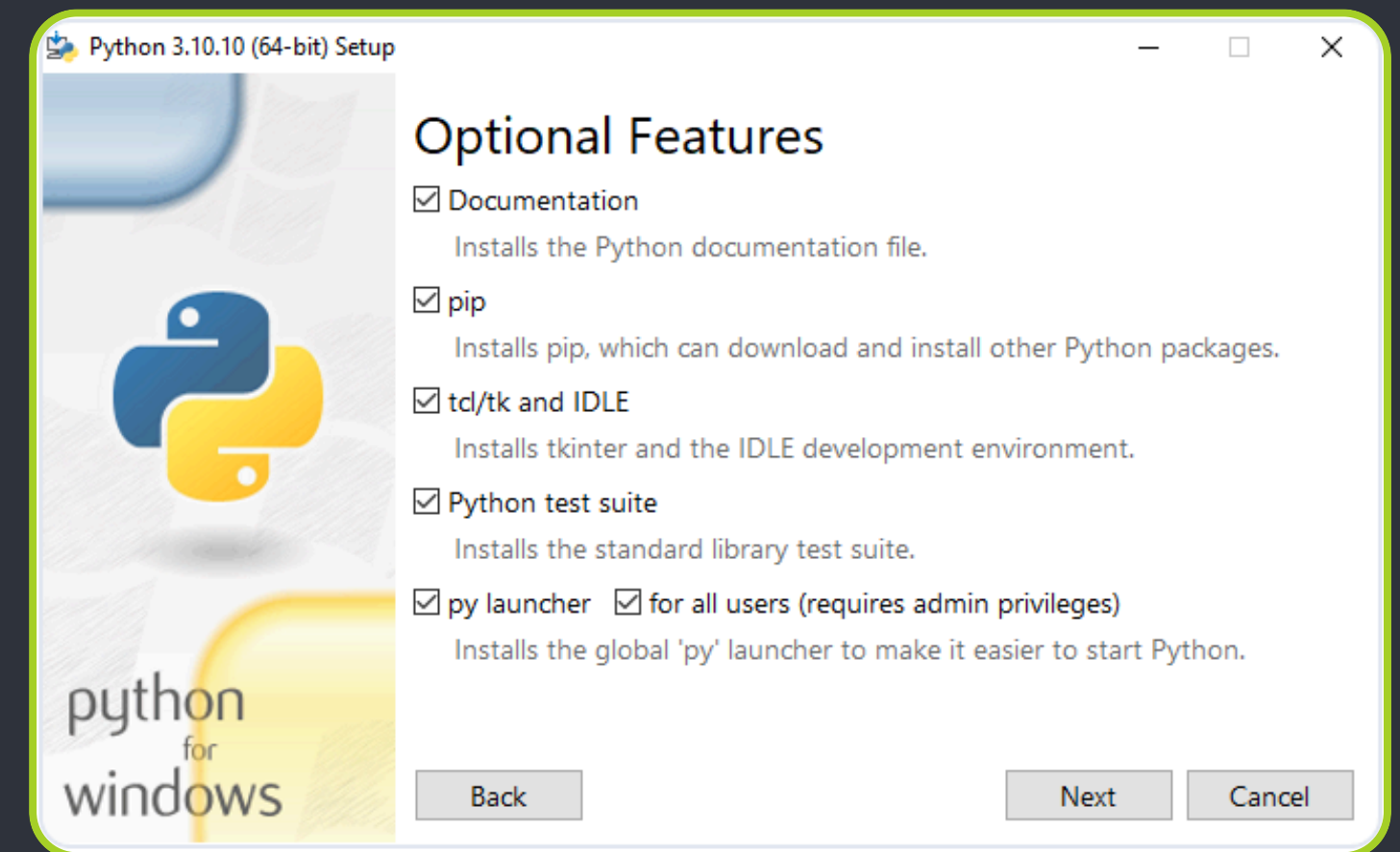
### Étape 2: Exécution du programme d'installation

1. Une fois le programme d'installation téléchargé, double-cliquez sur le fichier .exe, par exemple **python-3.10.10-amd64.exe**, pour exécuter le programme d'installation Python.
2. Cochez la case Installer le lanceur pour tous les utilisateurs, qui permet à tous les utilisateurs de l'ordinateur d'accéder à l'application de lancement Python.
3. Cochez la case Ajouter python.exe au PATH, qui permet aux utilisateurs de lancer Python à partir de la ligne de commande.



## Environnement 'Installation Python';

1. Si vous débutez avec Python et que vous souhaitez l'installer avec les fonctionnalités par défaut décrites dans la boîte de dialogue, cliquez sur Installer maintenant et passez à l'étape 4. Pour installer d'autres fonctionnalités facultatives et avancées, cliquez sur Personnaliser l'installation et continuez.
2. Les fonctionnalités facultatives incluent des outils et des ressources communs à Python et vous pouvez les installer tous, même si vous ne prévoyez pas de les utiliser.



# Environnement 'Installation Python';

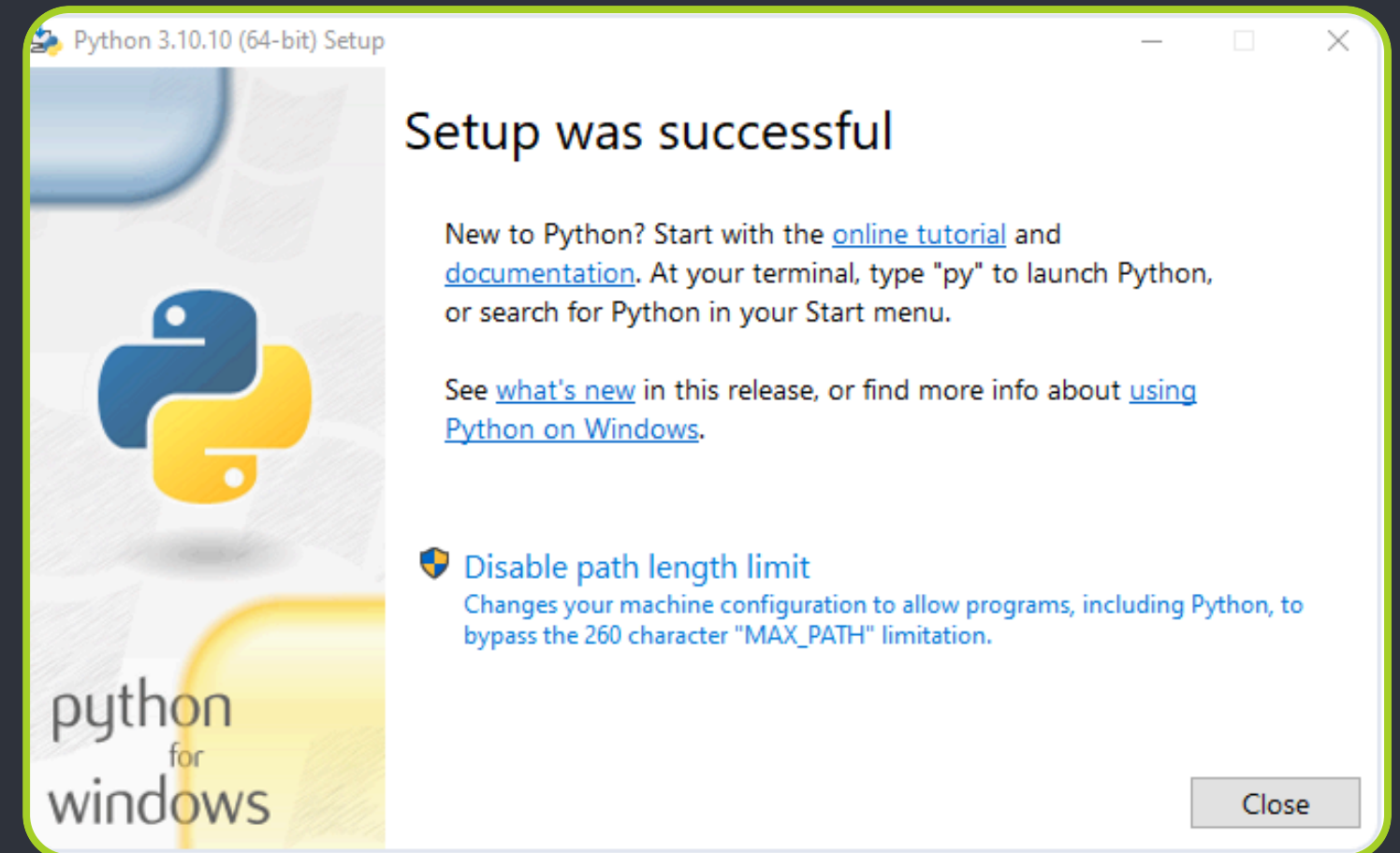
Sélectionnez les options qui correspondent à vos besoins :

- **Installer pour tous les utilisateurs** : recommandé si vous n'êtes pas le seul utilisateur sur cet ordinateur
- **Associer les fichiers à Python** : recommandé, car cette option associe tous les types de fichiers Python au lanceur ou à l'éditeur
- **Créer des raccourcis pour les applications installées** : recommandé pour activer les raccourcis pour les applications Python
- **Ajouter Python aux variables d'environnement** : recommandé pour activer le lancement de Python
- **Précompiler la bibliothèque standard** : non requis, cela pourrait ralentir l'installation
- **Télécharger les symboles de débogage et télécharger les binaires de débogage** : recommandé uniquement si vous prévoyez de créer des extensions C ou C++

Notez le répertoire d'installation de Python au cas où vous auriez besoin de vous y référer ultérieurement.

## Environnement 'Installation Python';

1. Cliquez sur **Installer** pour démarrer l'installation.
2. Une fois l'installation terminée, un **message indiquant que l'installation a réussi s'affiche.**





# Environnement 'Installation Python';

## Étape 3 Ajout de Python aux variables d'environnement (facultatif)

Ignorez cette étape si vous avez sélectionné Ajouter Python aux variables d'environnement lors de l'installation.

Si vous souhaitez accéder à Python via la ligne de commande mais que vous n'avez pas ajouté Python à vos variables d'environnement lors de l'installation, vous pouvez toujours le faire manuellement.

Avant de commencer, recherchez le répertoire d'installation de Python sur votre système. Les répertoires suivants sont des exemples de chemins de répertoire par défaut :

- **C:\Program Files\Python310** : si vous avez sélectionné Installer pour tous les utilisateurs lors de l'installation, le répertoire sera alors à l'échelle du système
- **C:\Users\Onesime\AppData\Local\Programs\Python\Python310** : si vous n'avez pas sélectionné Installer pour tous les utilisateurs lors de l'installation, le répertoire sera alors dans le chemin d'accès de l'utilisateur Windows.

Notez que le nom du dossier sera différent si vous avez installé une version différente, mais démarrera toujours avec Python.

1. Allez dans **Démarrer** et saisissez les **paramètres système avancés** dans la barre de recherche.
2. Cliquez sur **Afficher les paramètres système avancés**.
3. Dans **la boîte de dialogue Propriétés système**, cliquez sur **l'onglet Avancé**, puis sur **Variables d'environnement**.

# Environnement 'Installation Python';

1. En fonction de votre installation :

- Si vous avez sélectionné **Installer pour tous les utilisateurs** lors de l'installation, sélectionnez **Chemin** dans la liste des **variables système** et cliquez sur **Modifier**.
- Si vous n'avez pas sélectionné **Installer pour tous les utilisateurs** lors de l'installation, sélectionnez **Chemin** dans la liste des variables utilisateur et cliquez sur **Modifier**.
- Cliquez sur **Nouveau** et entrez le chemin du répertoire Python, puis cliquez sur **OK** jusqu'à ce que toutes les boîtes de dialogue soient fermées.

## Étape 4 Vérifier l'installation de Python

- Vous pouvez vérifier si l'installation de Python a réussi soit via la ligne de commande, soit via l'application Environnement de développement intégré (IDLE), si vous avez choisi de l'installer.
- Allez dans **Démarrer** et saisissez cmd dans la barre de recherche. Cliquez sur **Invite de commandes**.
- Saisissez la commande suivante dans l'invite de commandes :

```
python --version
```

Output

```
Python 3.10.10
```



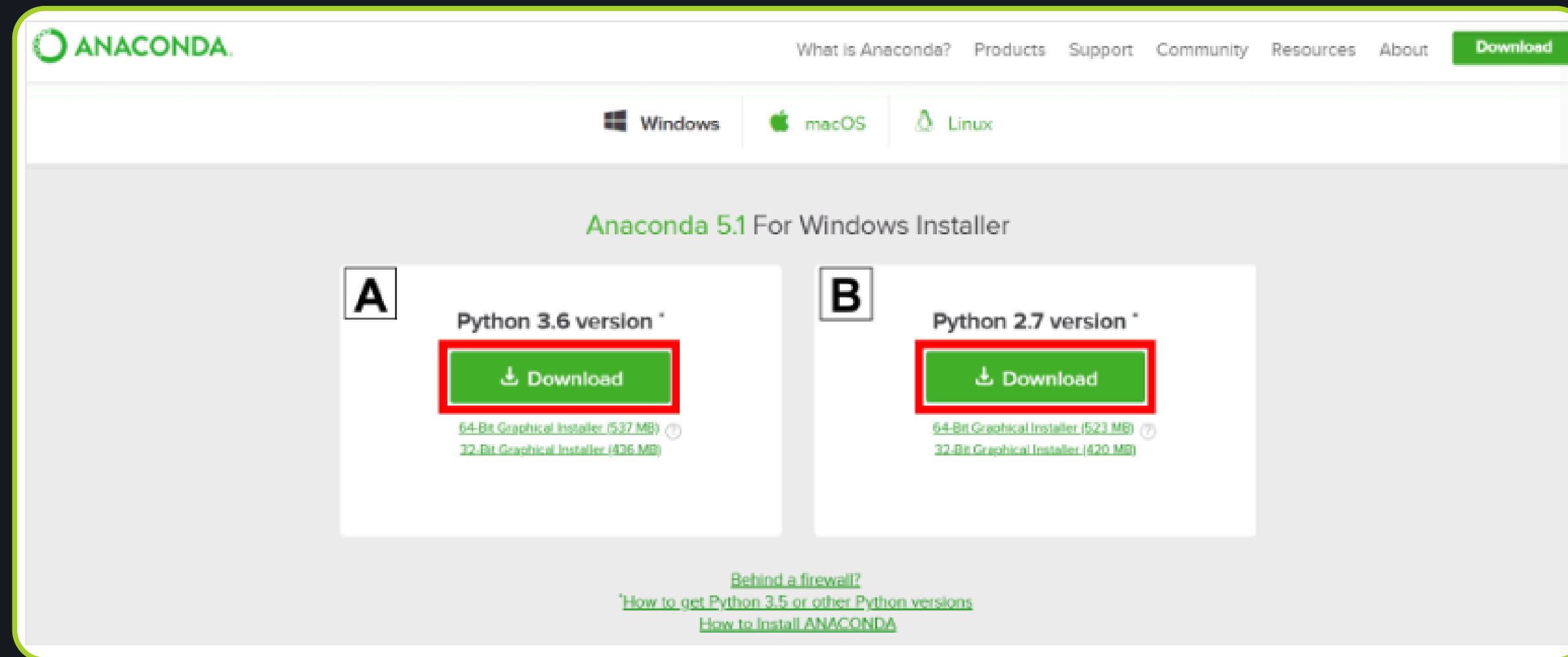
## Environnement 'Installation d'anaconda';

**Anaconda** est un gestionnaire de paquets, un gestionnaire d'environnement et une distribution Python qui contient une collection de paquets open source. C'est un avantage car lorsque vous travaillez sur un projet de Data Science, vous constaterez que vous avez besoin de nombreux paquets différents (numpy, scikit-learn, scipy, pandas pour n'en citer que quelques-uns), avec lesquels une installation d'Anaconda est préinstallée.

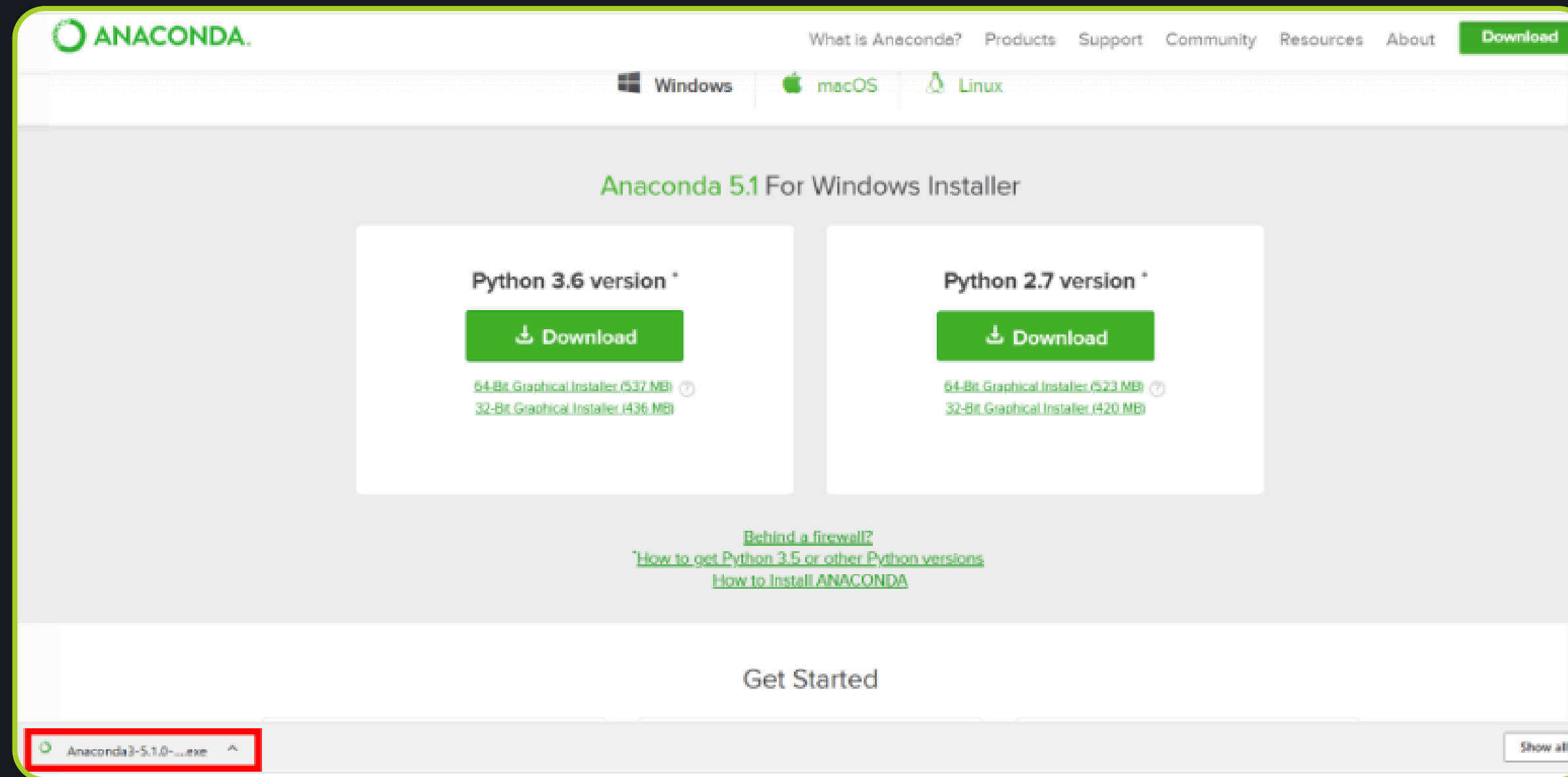
Si vous avez besoin de paquets supplémentaires après l'installation d'Anaconda, vous pouvez utiliser le gestionnaire de paquets d'Anaconda, **conda** ou **pip** pour installer ces paquets. C'est très avantageux car vous n'avez pas à gérer vous-même les dépendances entre plusieurs paquets. Conda facilite même le basculement entre Python 2 et 3. En fait, l'installation d'Anaconda est également la méthode recommandée pour installer **Jupyter Notebooks**.

## Comment télécharger et installer Anaconda

1. Accédez au site Web d'Anaconda et choisissez un programme d'installation graphique Python 3.x (A) ou un programme d'installation graphique Python 2.x (B).



## 2. Localisez votre téléchargement et double-cliquez dessus.



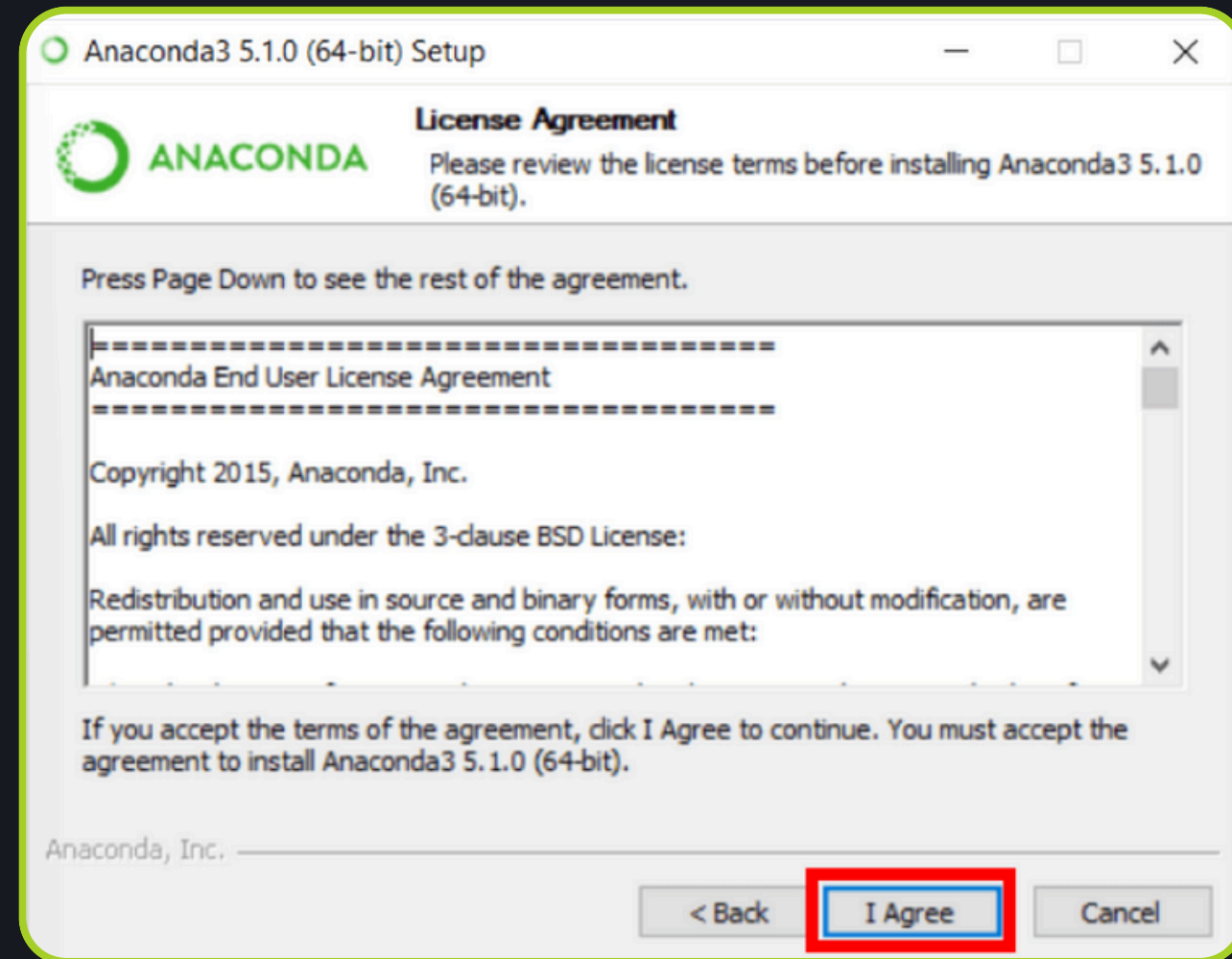
## Environnement 'Installation d'anaconda';

Lorsque l'écran ci-dessous apparaît, cliquez sur Suivant.

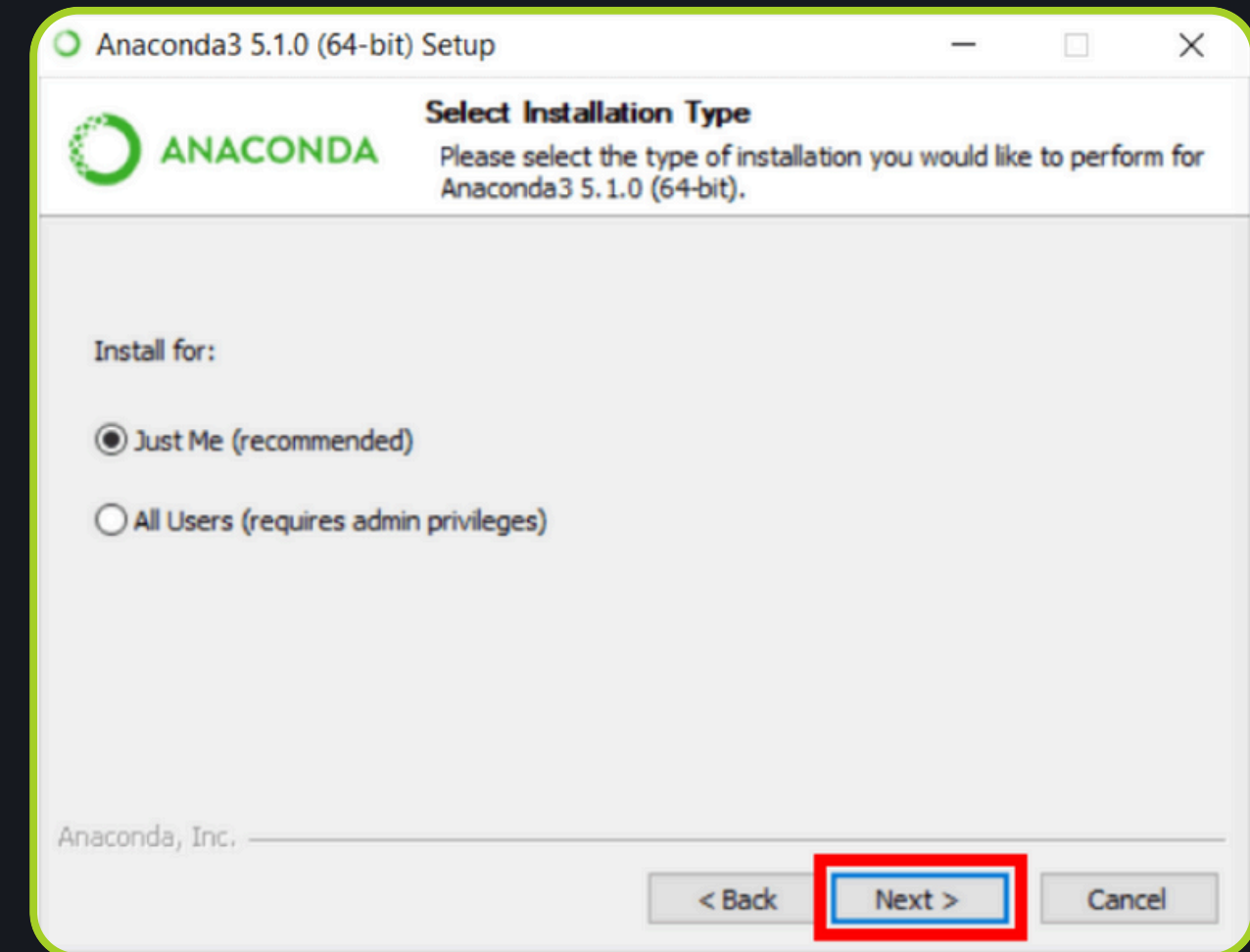


## Environnement 'Installation d'anaconda';

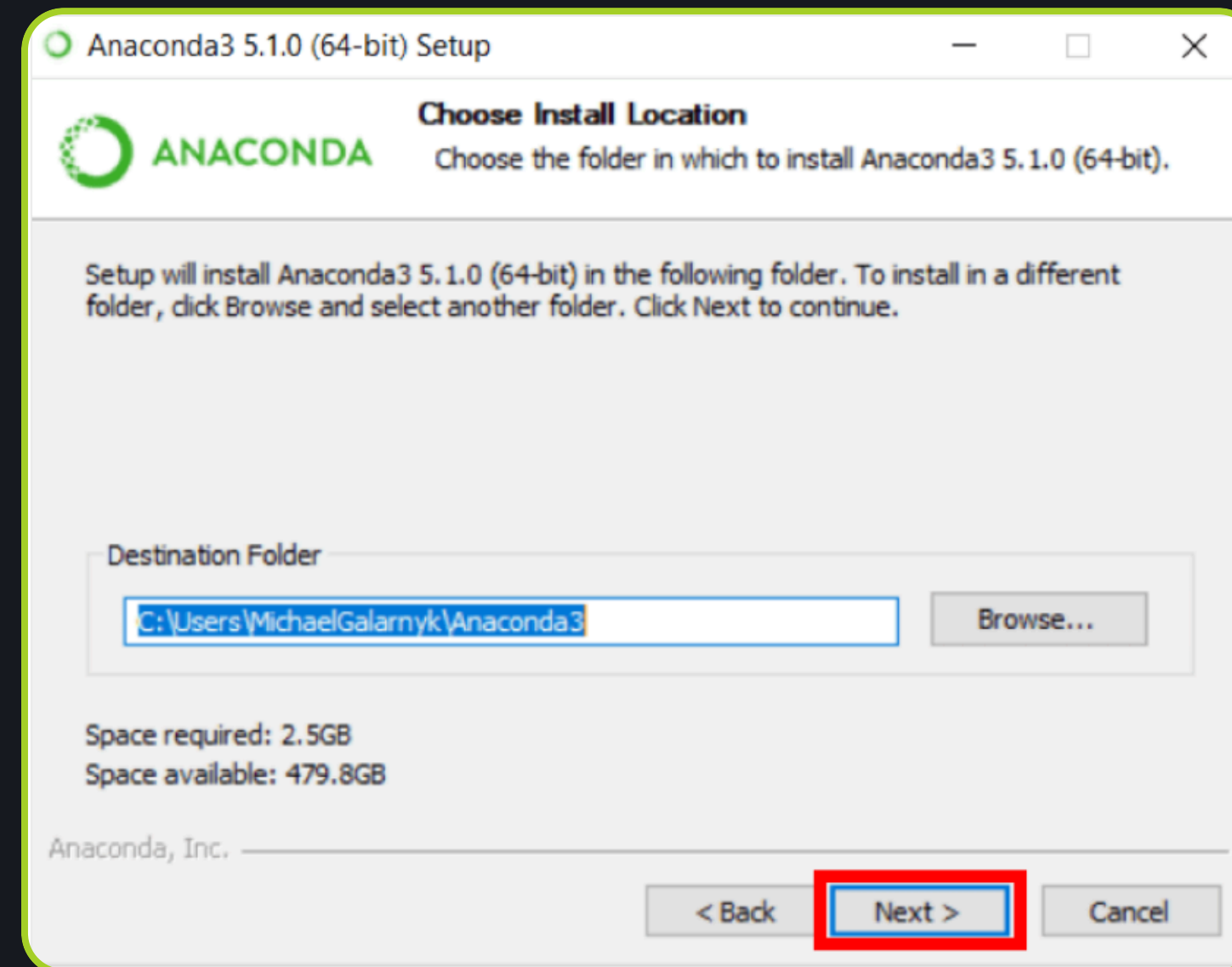
3. Lisez le contrat de licence et cliquez sur J'accepte.



4. Cliquez sur Suivant.

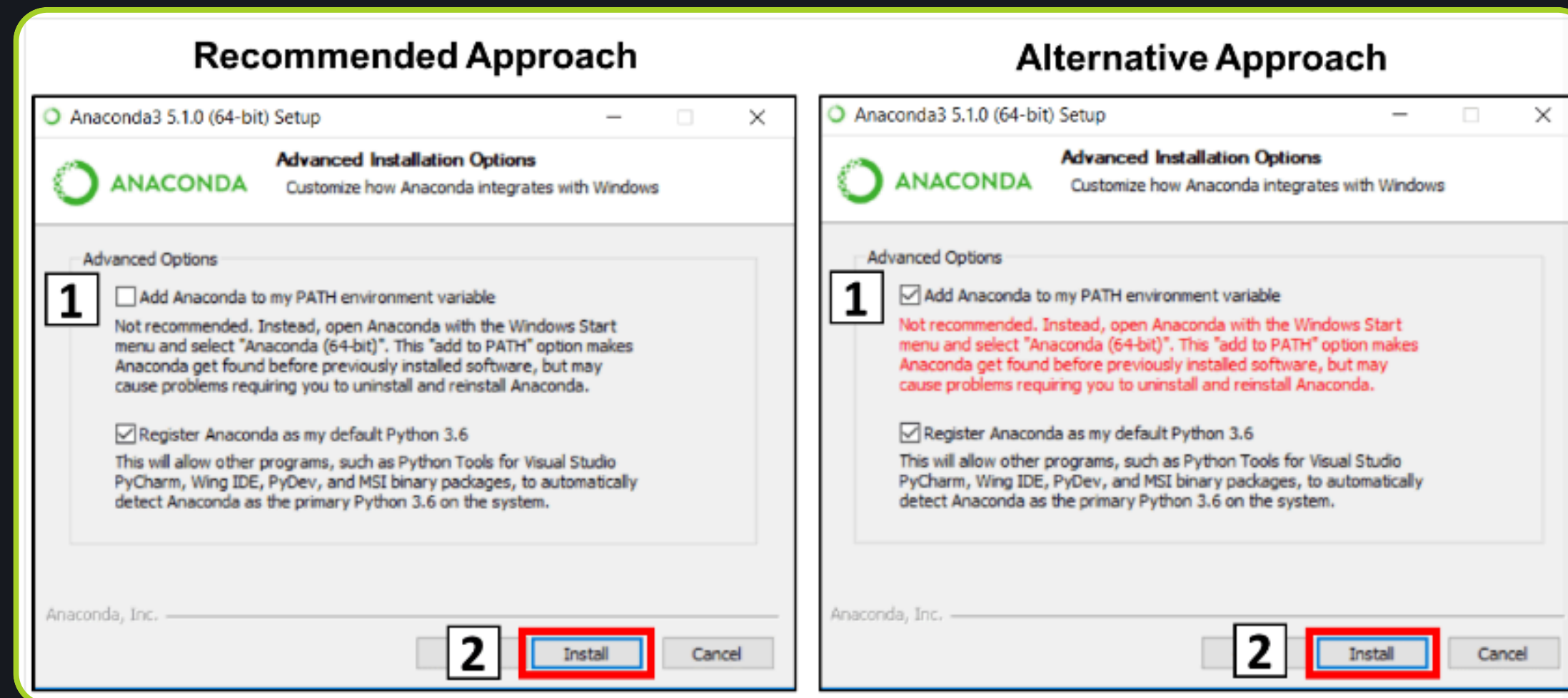


5. Notez votre emplacement d'installation, puis cliquez sur Suivant.



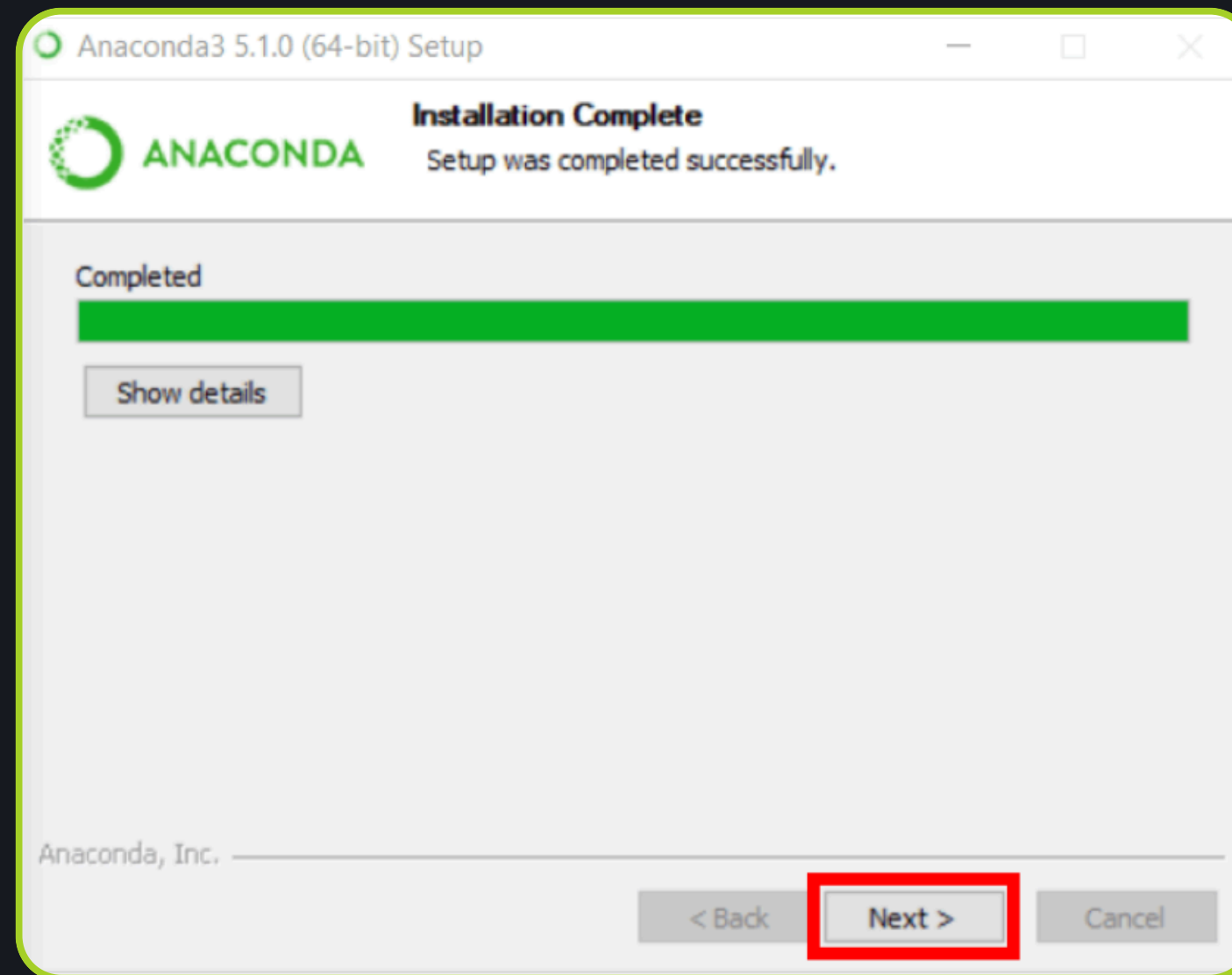
6. Il s'agit d'une partie importante du processus d'installation. L'approche recommandée consiste à ne pas cocher la case pour ajouter Anaconda à votre chemin. Cela signifie que vous devrez utiliser Anaconda Navigator ou l'invite de commande Anaconda (située dans le menu Démarrer sous « Anaconda »). lorsque vous souhaitez utiliser Anaconda (vous pouvez toujours ajouter Anaconda à votre PATH plus tard si vous ne cochez pas la case).

Si vous souhaitez pouvoir utiliser Anaconda dans votre invite de commande (ou git bash, cmd, powershell, etc.), veuillez utiliser l'approche alternative et cocher la case.



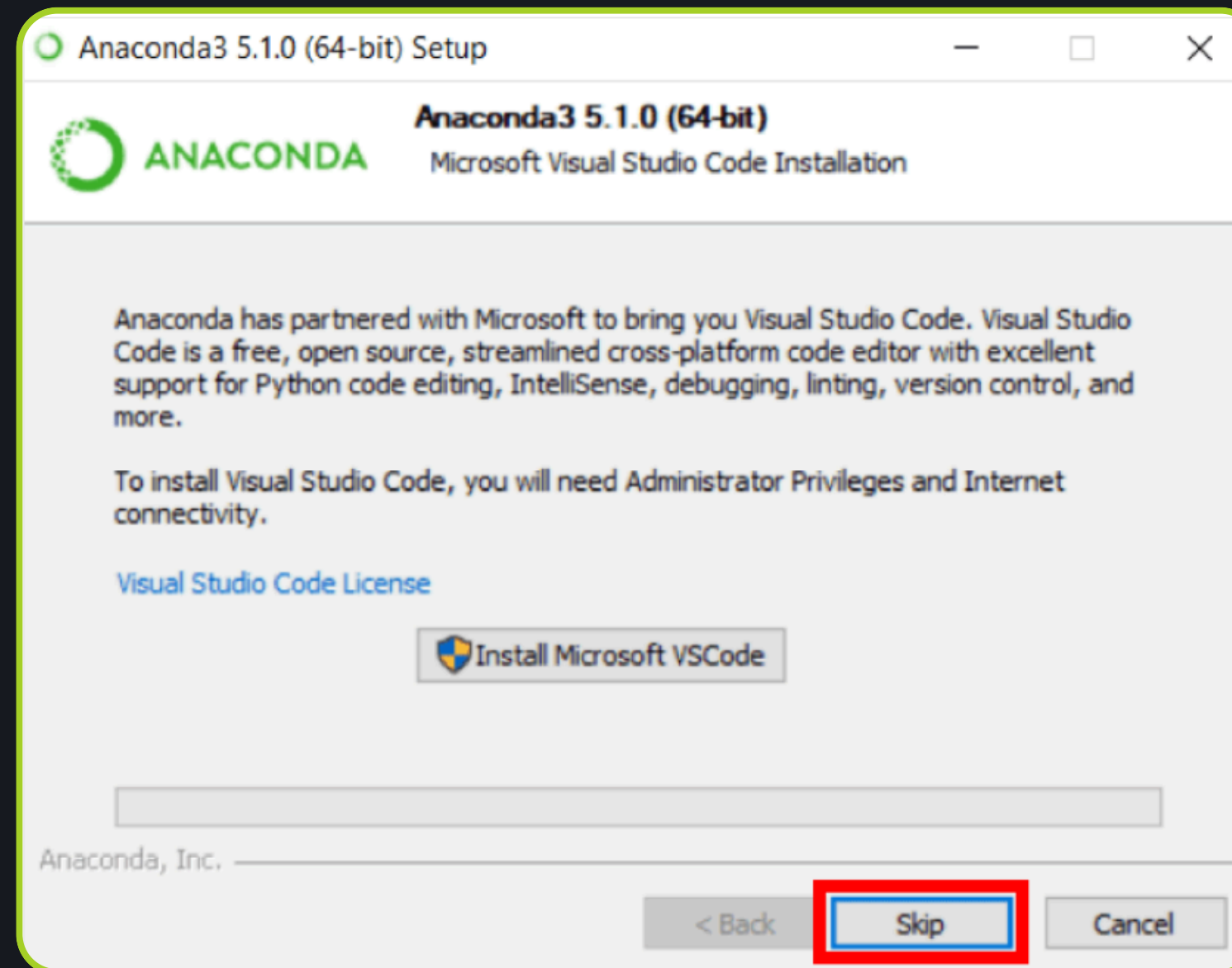


7. Cliquez sur Suivant.

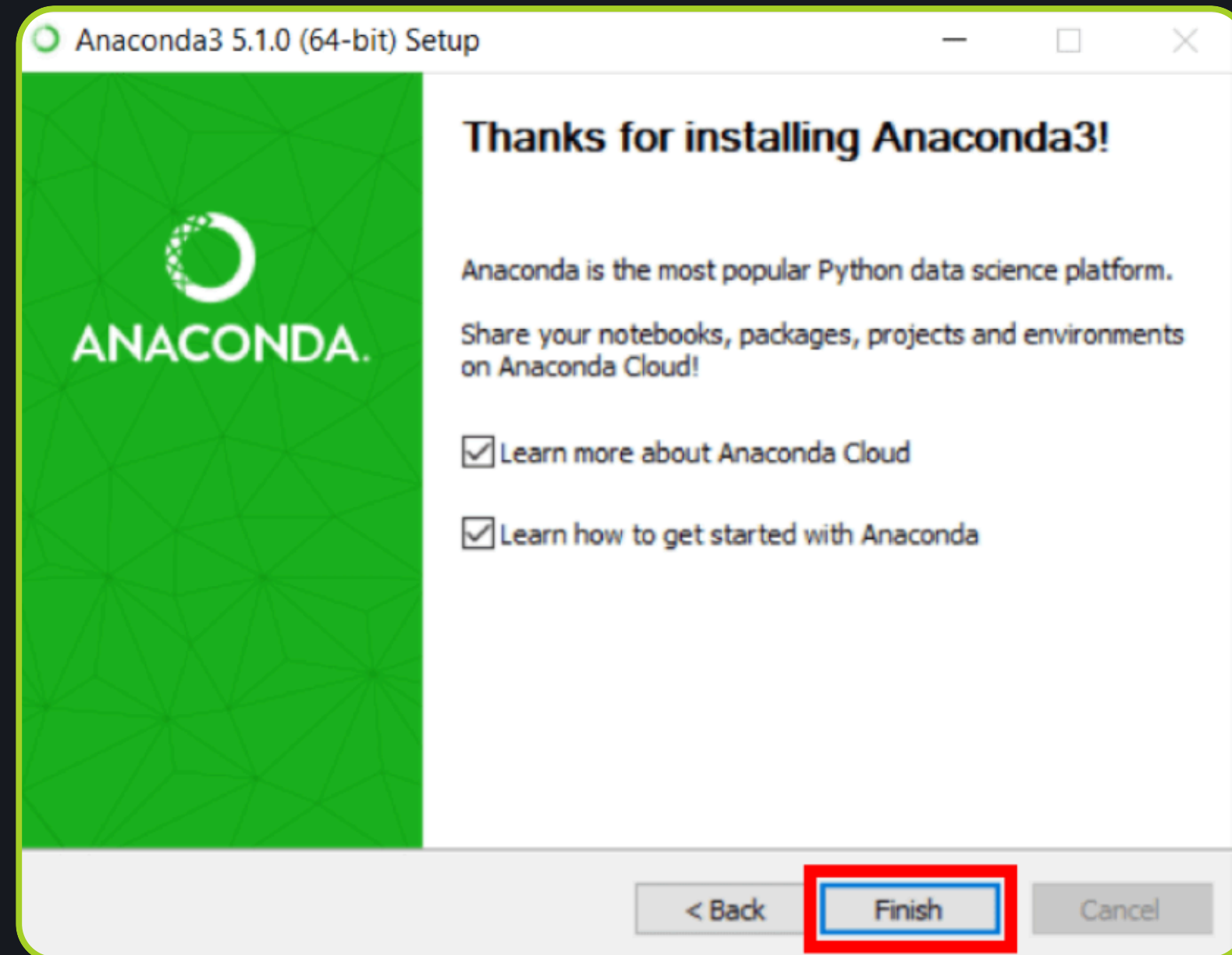




**8. Vous pouvez installer Microsoft VSCode si vous le souhaitez, mais cela est facultatif.**



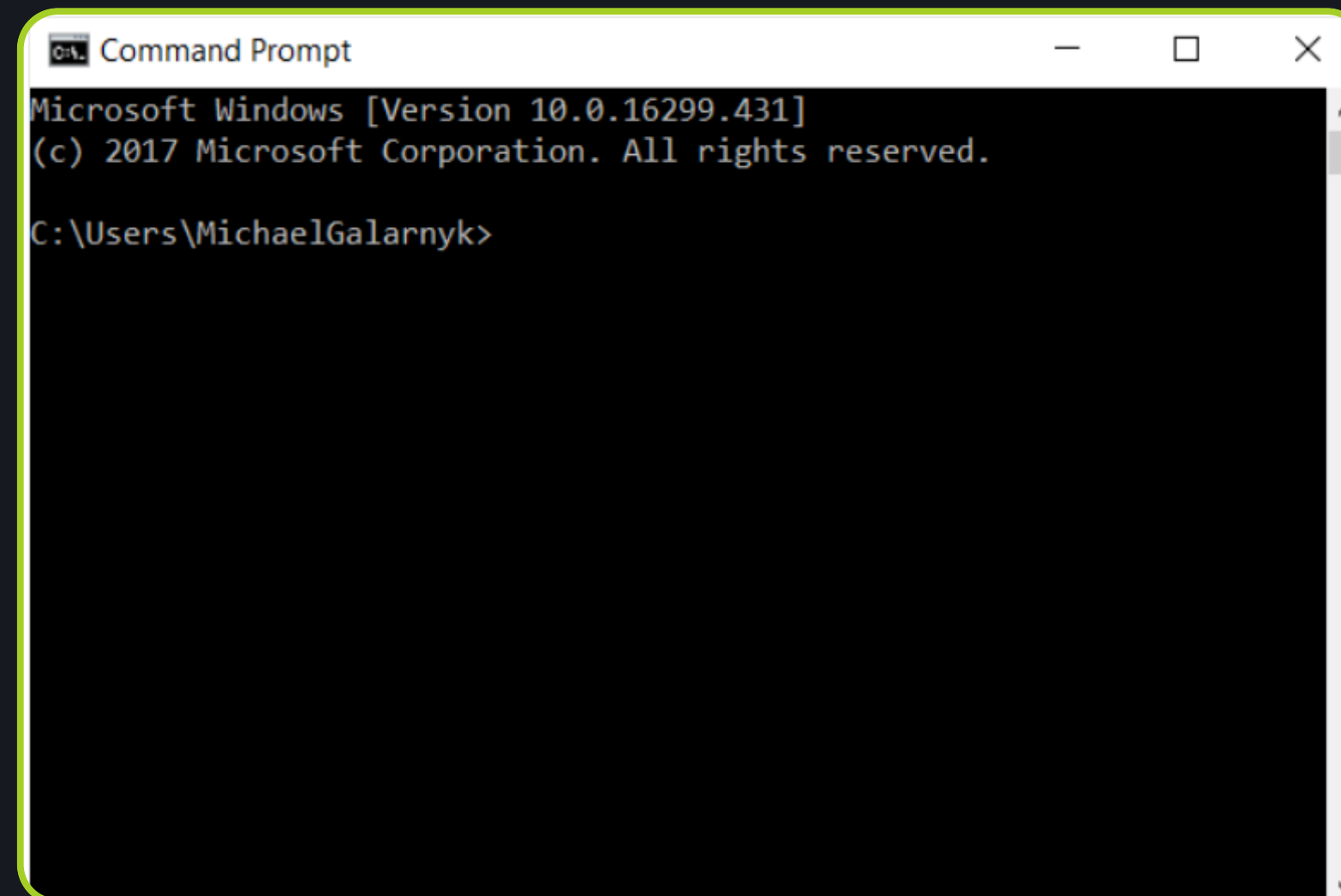
9. Cliquez sur Terminer.



## Comment ajouter Anaconda au chemin (facultatif)

Il s'agit d'une étape facultative. Elle concerne le cas où vous n'avez pas coché la case à l'étape 6 et souhaitez maintenant ajouter Anaconda à votre chemin. L'avantage de cette opération est que vous pourrez utiliser Anaconda dans votre invite de commande, Git Bash, cmdr, etc.

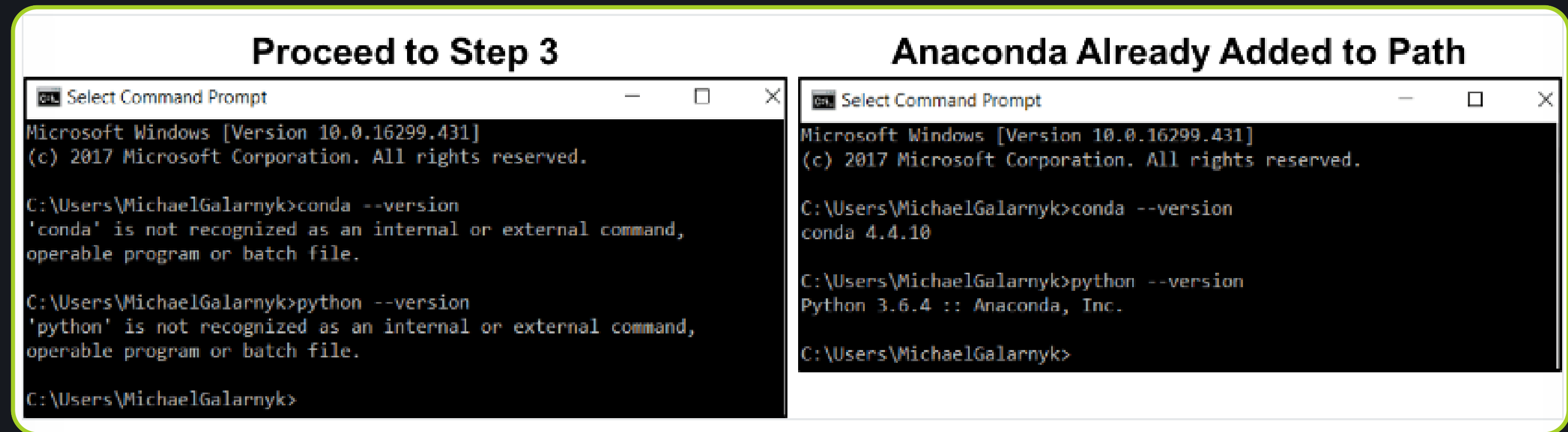
### 1. Ouvrez une invite de commande.



2. Vérifiez si Anaconda est déjà ajouté à votre chemin. Entrez les commandes ci-dessous dans votre invite de commande. Cela permet de vérifier si Anaconda est déjà ajouté à votre chemin. Si vous obtenez une erreur de commande non reconnue comme dans le côté gauche de l'image ci-dessous, passez à l'étape 3. Si vous obtenez un résultat similaire au côté droit de l'image ci-dessous, vous avez déjà ajouté Anaconda à votre chemin.

`conda --version`

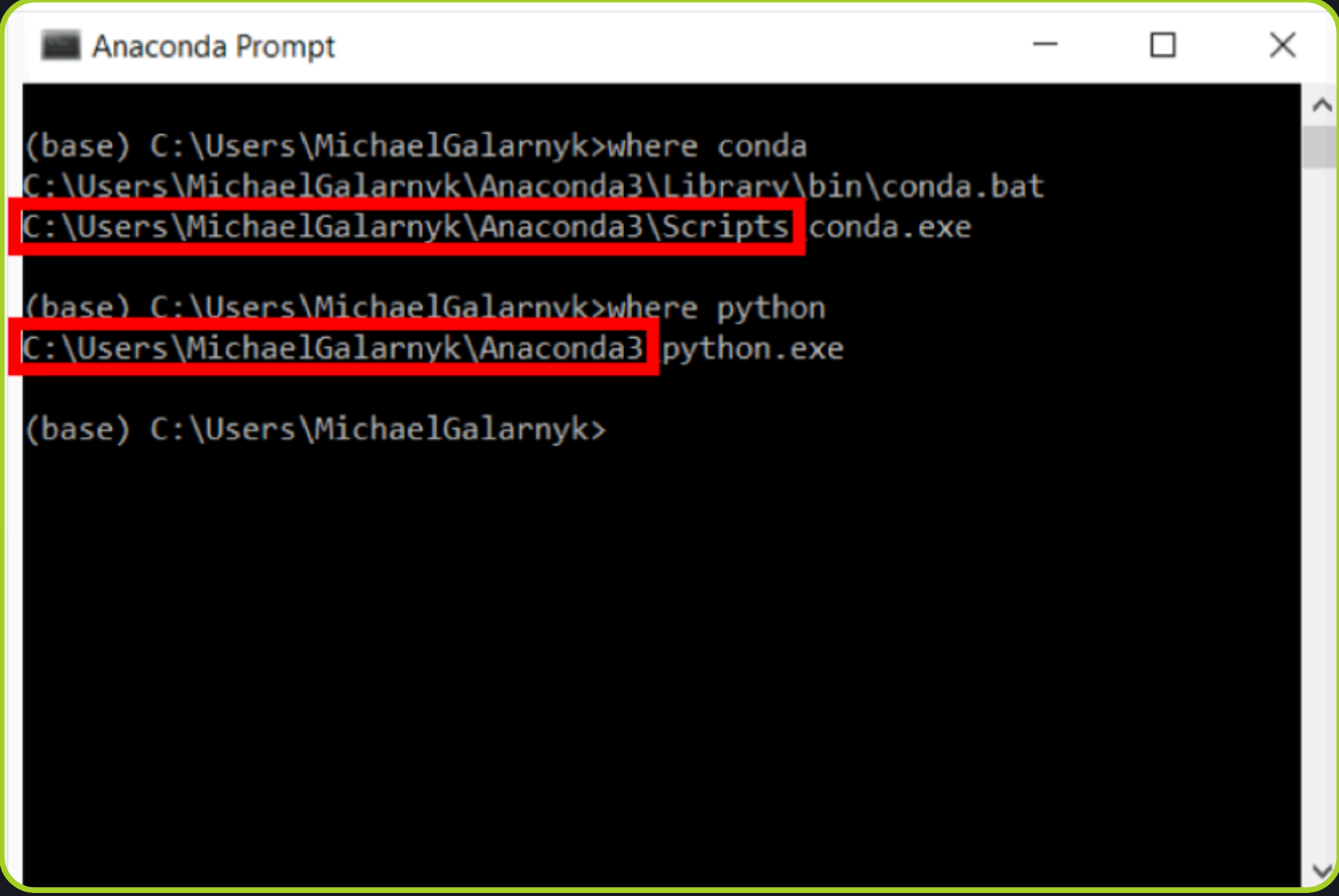
`python --version`



3. Si vous ne savez pas où se trouvent votre conda et/ou python, ouvrez une invite Anaconda et saisissez les commandes suivantes. Cela vous indique où se trouvent conda et python sur votre ordinateur.

`where conda`

`where python`



```
Anaconda Prompt

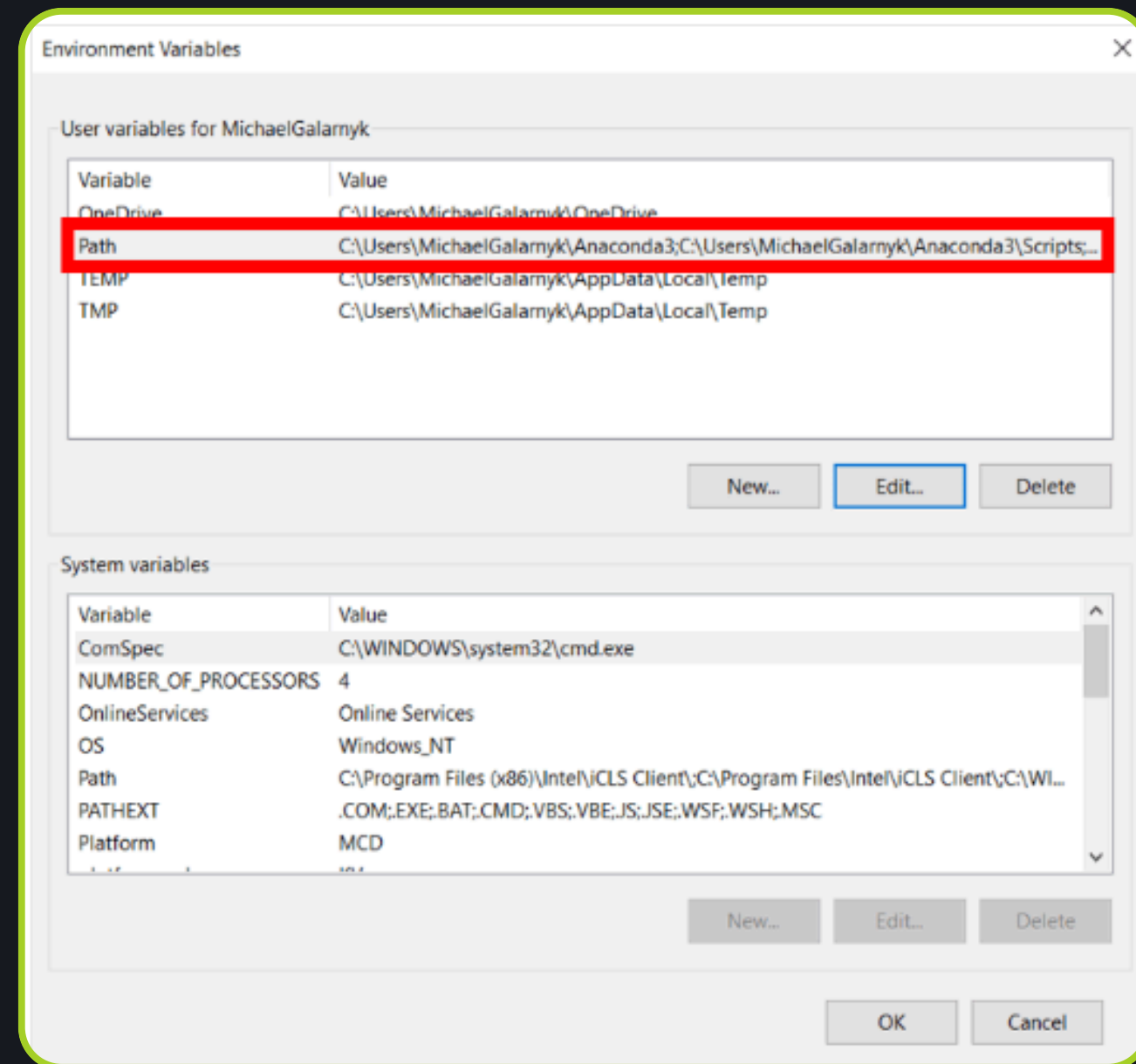
(base) C:\Users\MichaelGalarnyk>where conda
C:\Users\MichaelGalarnyk\Anaconda3\Library\bin\conda.bat
C:\Users\MichaelGalarnyk\Anaconda3\Scripts\conda.exe

(base) C:\Users\MichaelGalarnyk>where python
C:\Users\MichaelGalarnyk\Anaconda3\python.exe

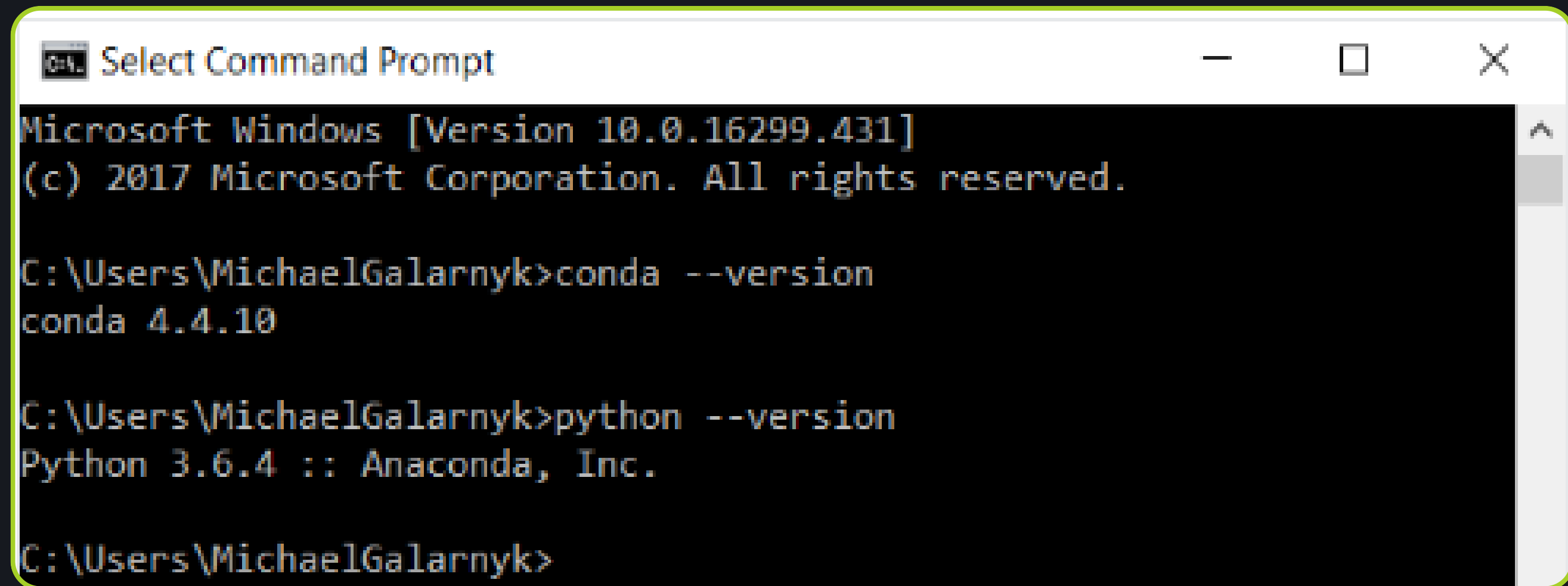
(base) C:\Users\MichaelGalarnyk>
```

## Environnement 'Installation d'anaconda';

4. Ajoutez conda et python à votre PATH. Vous pouvez le faire en accédant à vos variables d'environnement et en ajoutant la sortie de l'étape 3 (enfermée dans le rectangle rouge) à votre chemin.



5. Ouvrez une nouvelle invite de commande. Essayez de saisir `conda --version` et `python --version` dans l'invite de commande pour vérifier si tout s'est bien passé.



```
Microsoft Windows [Version 10.0.16299.431]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\MichaelGalarnyk>conda --version
conda 4.4.10

C:\Users\MichaelGalarnyk>python --version
Python 3.6.4 :: Anaconda, Inc.

C:\Users\MichaelGalarnyk>
```

# Interface du Jupyter Notebooks 'Comment utiliser';

**Le Jupyter Notebook** (anciennement appelé IPython Notebook) est une application Web qui vous permet de créer et de partager des documents informatiques contenant :

- 1.Des codes dynamiques
- 2.Des équations
- 3.Des visualisations
- 4.Des textes narratifs.

Le Jupyter Notebook peut être utile pour diverses fonctions. Il s'agit notamment de l'apprentissage et de l'enseignement d'un langage de programmation tel que Python, ainsi que du stockage et du partage de données.



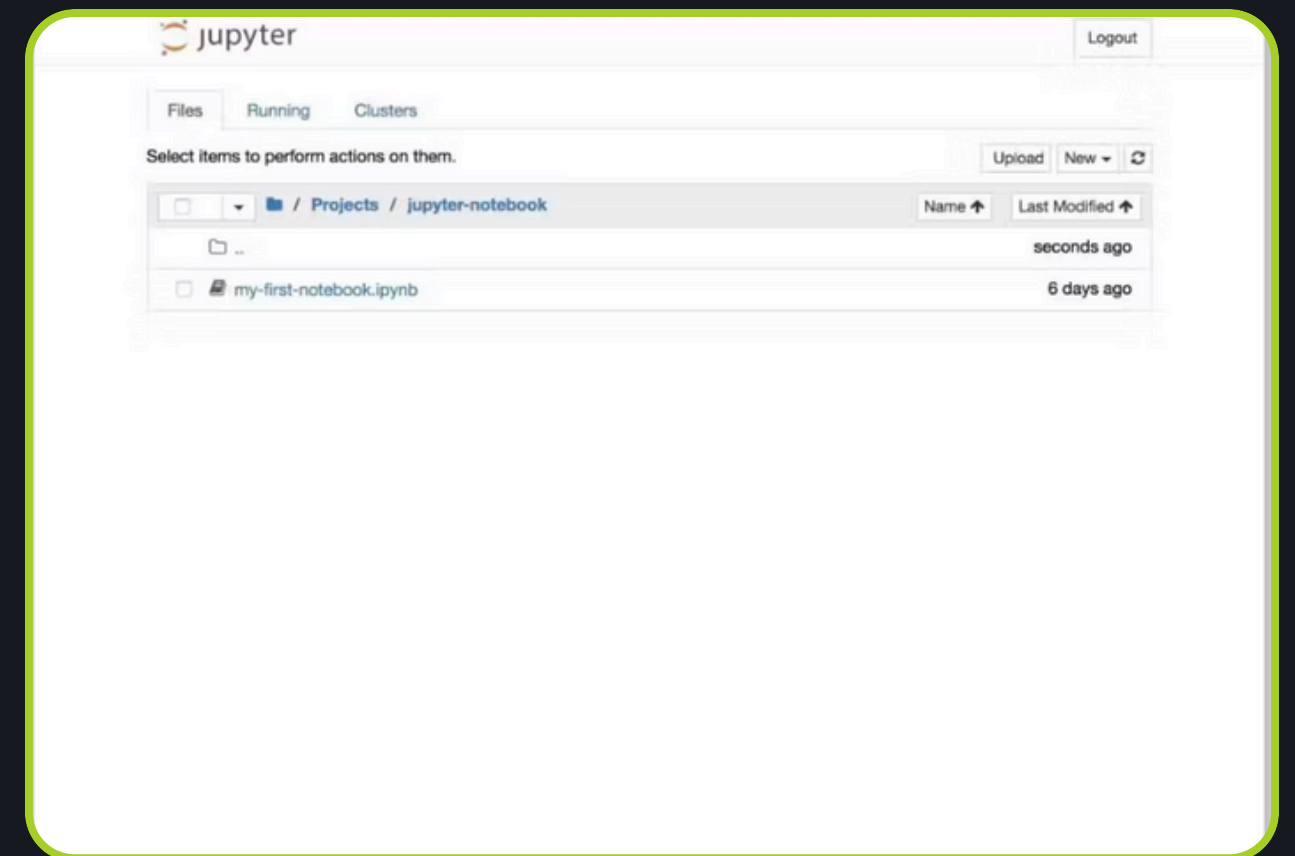
## Étape 1 : démarrage du serveur Jupyter Notebook

Maintenant que vous avez installé la distribution Anaconda, commençons à apprendre les bases.

- Ouvrez l'application de terminal et accédez au dossier de votre choix.
- Choisissez un emplacement (le dossier ou le sous-dossier, si vous en avez créé un à cet effet) dans votre terminal et exécutez la commande suivante : **\$ jupyter notebook**

Cela devrait démarrer Jupyter. Votre navigateur vous redirigera ensuite vers l'URL suivante : <http://localhost:8888/tree>.

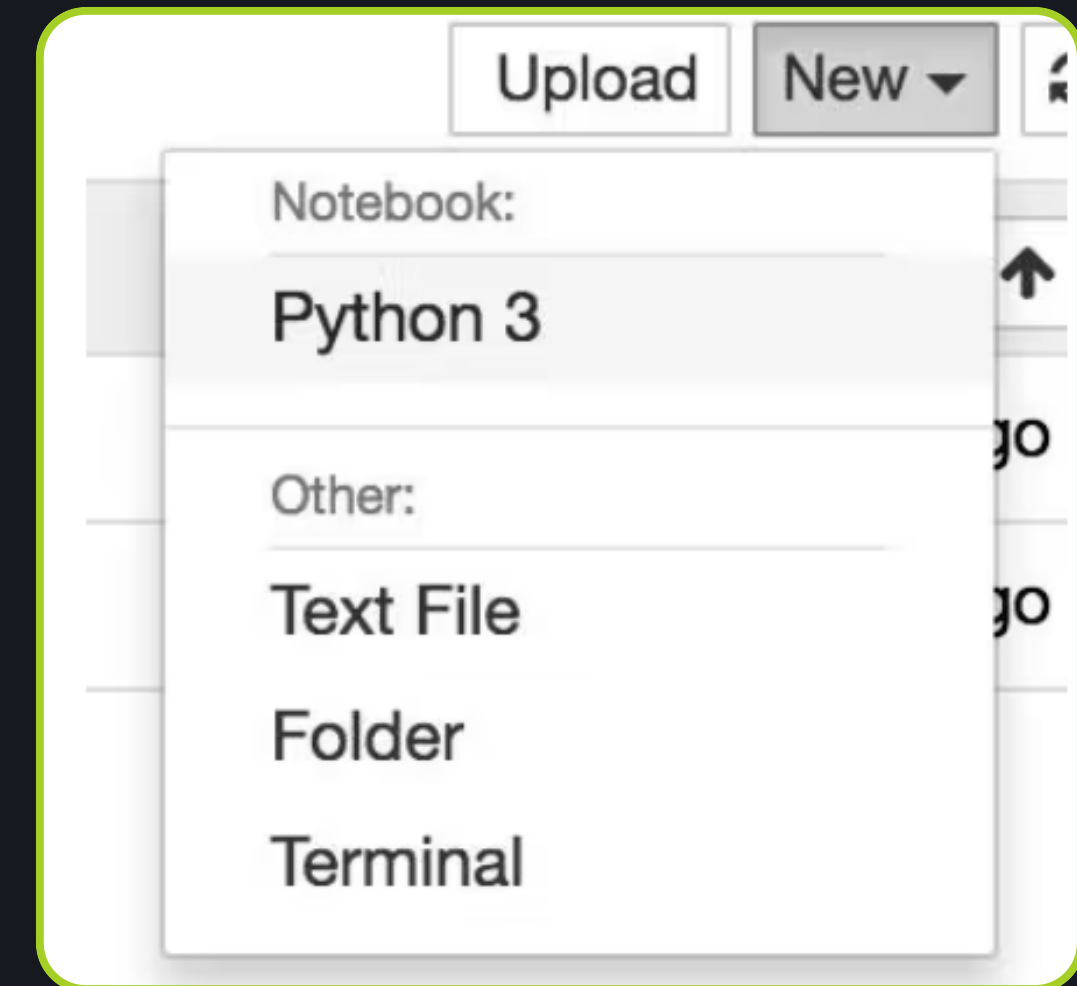
Votre écran devrait maintenant ressembler à ceci :



## Étape 2 : création du Notebook

Maintenant que vous avez appris à démarrer le serveur de blocs-notes, il est temps d'explorer comment créer un véritable document Jupyter Notebook.

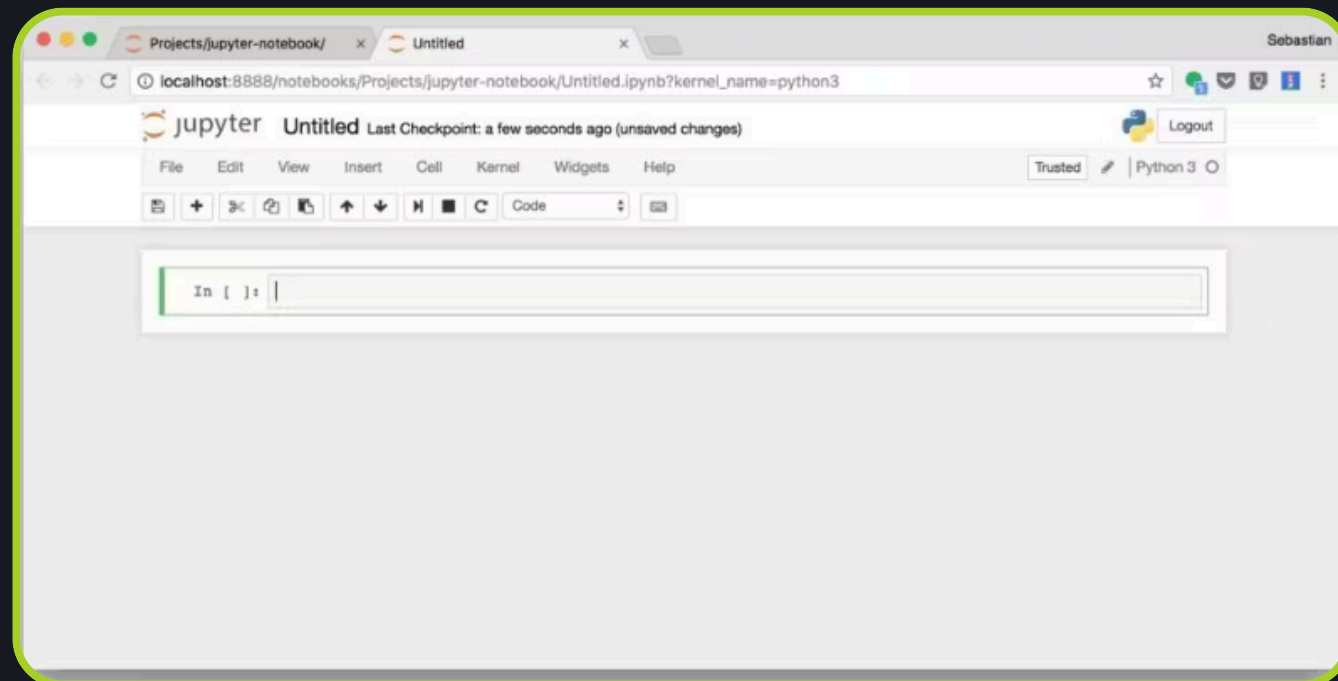
- Cliquez sur le bouton déroulant « Nouveau »
- Sélectionnez « Python 3 » dans la liste des options
- Cela créera un nouveau bloc-notes pour Python qui ressemblera à l'image ci-dessous.



# Jupyter Notebooks 'Comment utiliser';

## Étape 3 : Nommer du Notebook

- Vous pouvez modifier le nom du document en cliquant sur le texte « sans titre ».
- Après avoir nommé le fichier, enregistrez-le avec l'extension .ipynb.



Running en haut indique que le fichier est actuellement ouvert. Vous pouvez choisir de l'arrêter en cliquant sur le bouton Arrêter.

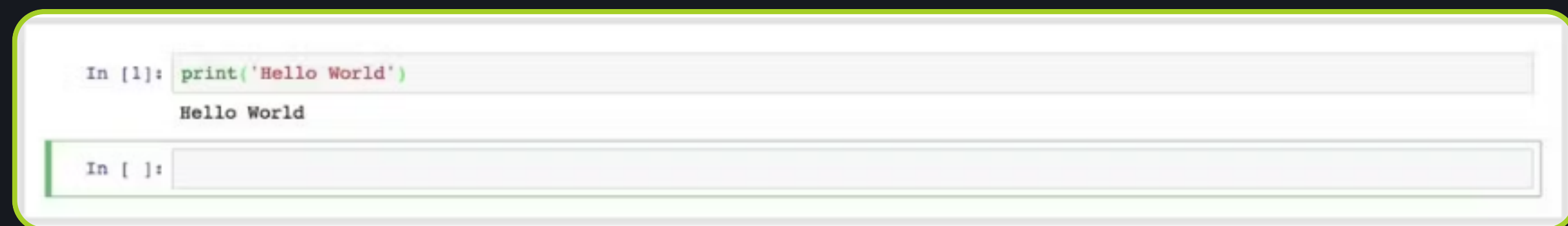
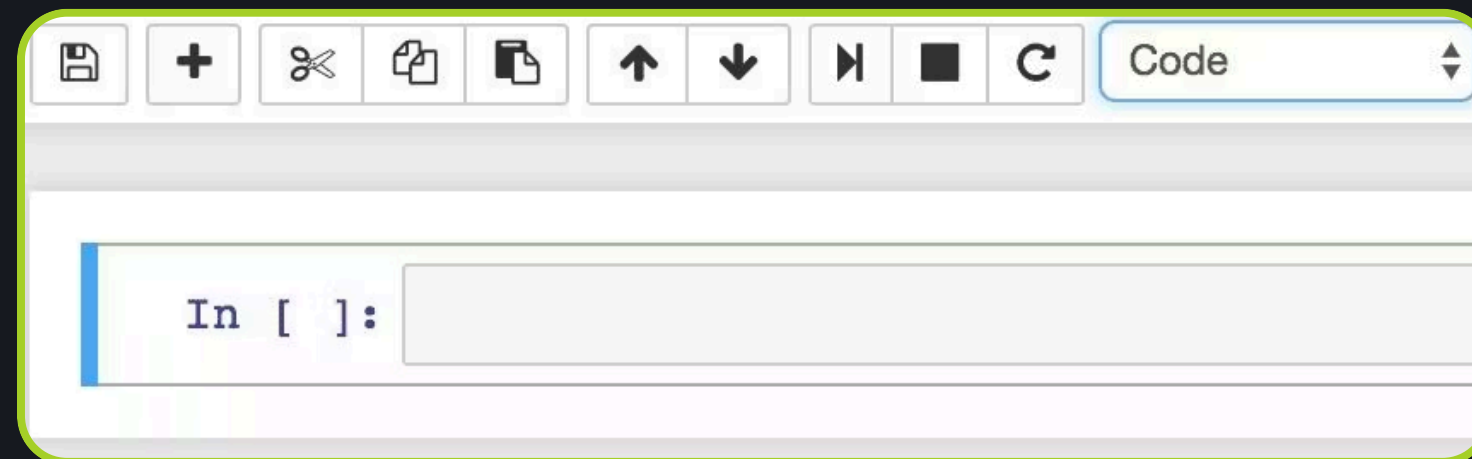


## Étape 4 : Exécuter les cellules Notebook

Le bloc-notes est composé de cellules. La première cellule vide devient disponible juste après la création d'un nouveau bloc-notes.

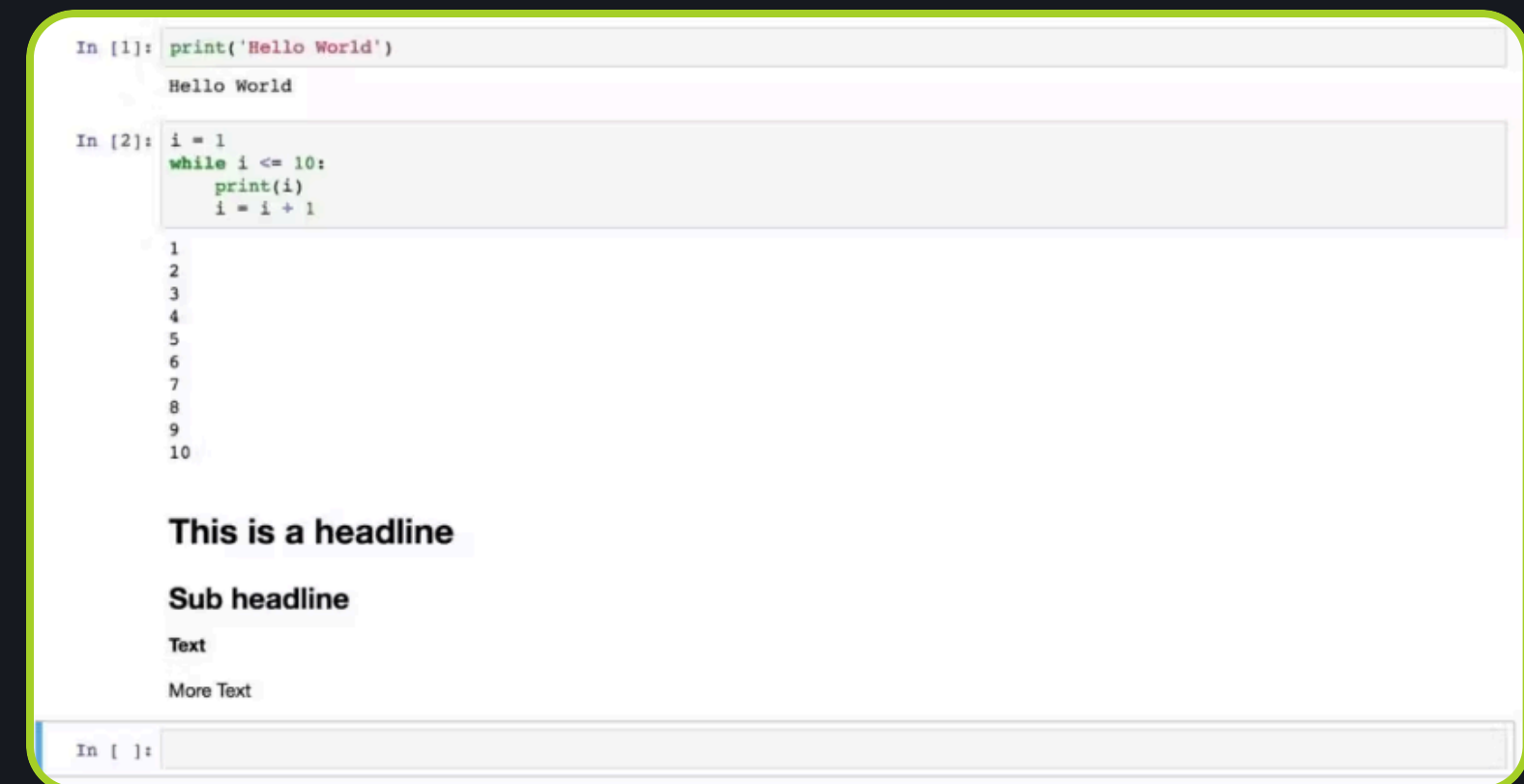
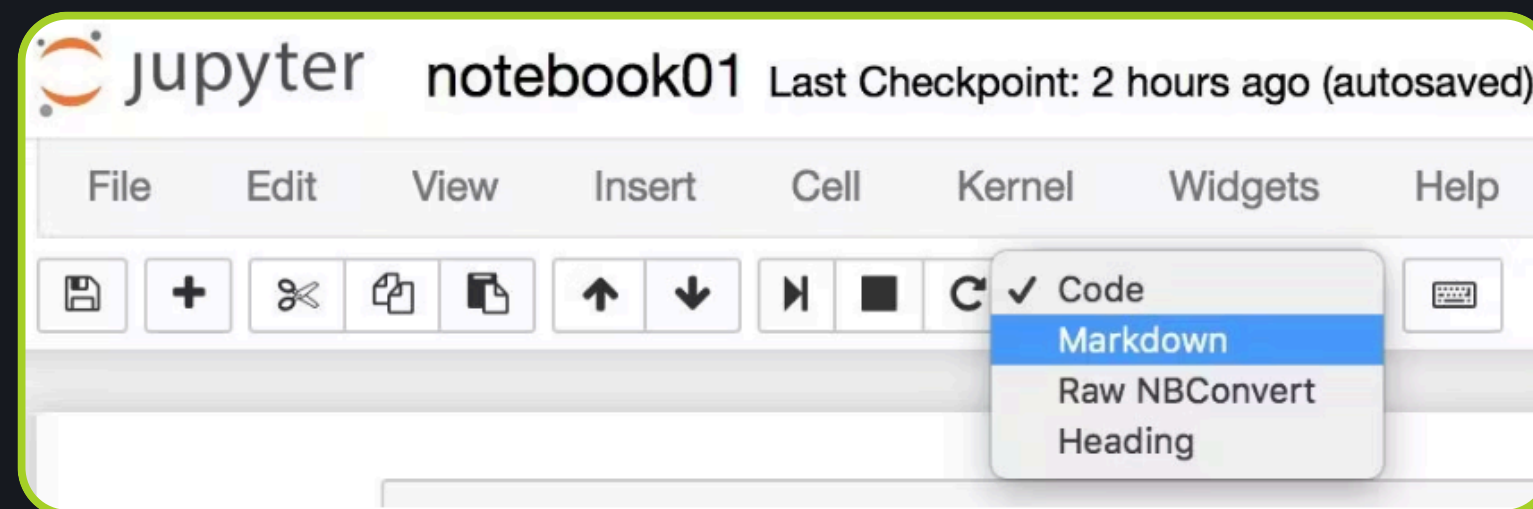
Elle vous permet de saisir directement le code Python.

Vous pouvez exécuter le code dans cette cellule en cliquant sur le bouton Exécuter la cellule ou en appuyant sur **Shift + Return**.



## Étape 5 : Ajout d'un texte explicatif

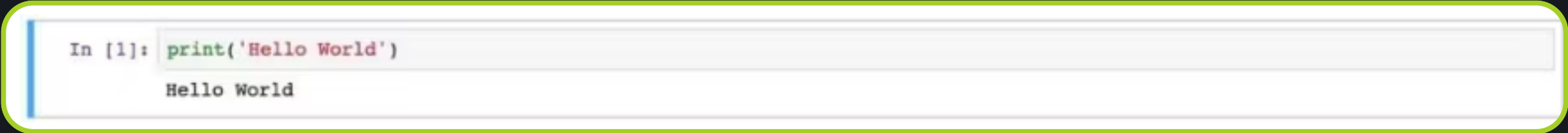
- Pour inclure un texte explicatif dans votre bloc-notes, vous devrez modifier le type de cellule de Code à Markdown comme indiqué dans l'image ci-dessous.
- Une fois le type de cellule modifié, vous pouvez saisir le code Markdown et compiler la cellule en appuyant sur les touches Maj + Retour.
- La cellule de l'éditeur Markdown sera désormais remplacée par le résultat obtenu.



## Étape 6 : Modes Édition et Commande

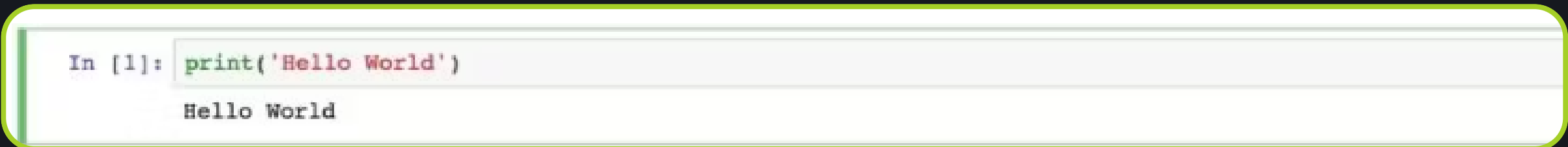
Si une cellule est active, elle est généralement distinguée par deux modes :  
**le mode Commande et le mode Édition.**

Si vous cliquez sur une cellule, elle s'ouvre généralement en mode Commande.  
Ceci est indiqué par une marge de couleur bleue à gauche.

A screenshot of a Jupyter Notebook cell in Command mode. The cell has a light blue vertical bar on the left side. The code 'In [1]: print('Hello World')' is in the top bar, and the output 'Hello World' is in the bottom bar.

```
In [1]: print('Hello World')
Hello World
```

Si vous cliquez sur la zone de code de la cellule, vous accédez au mode Édition. Ceci est  
indiqué par une marge de couleur verte à gauche.

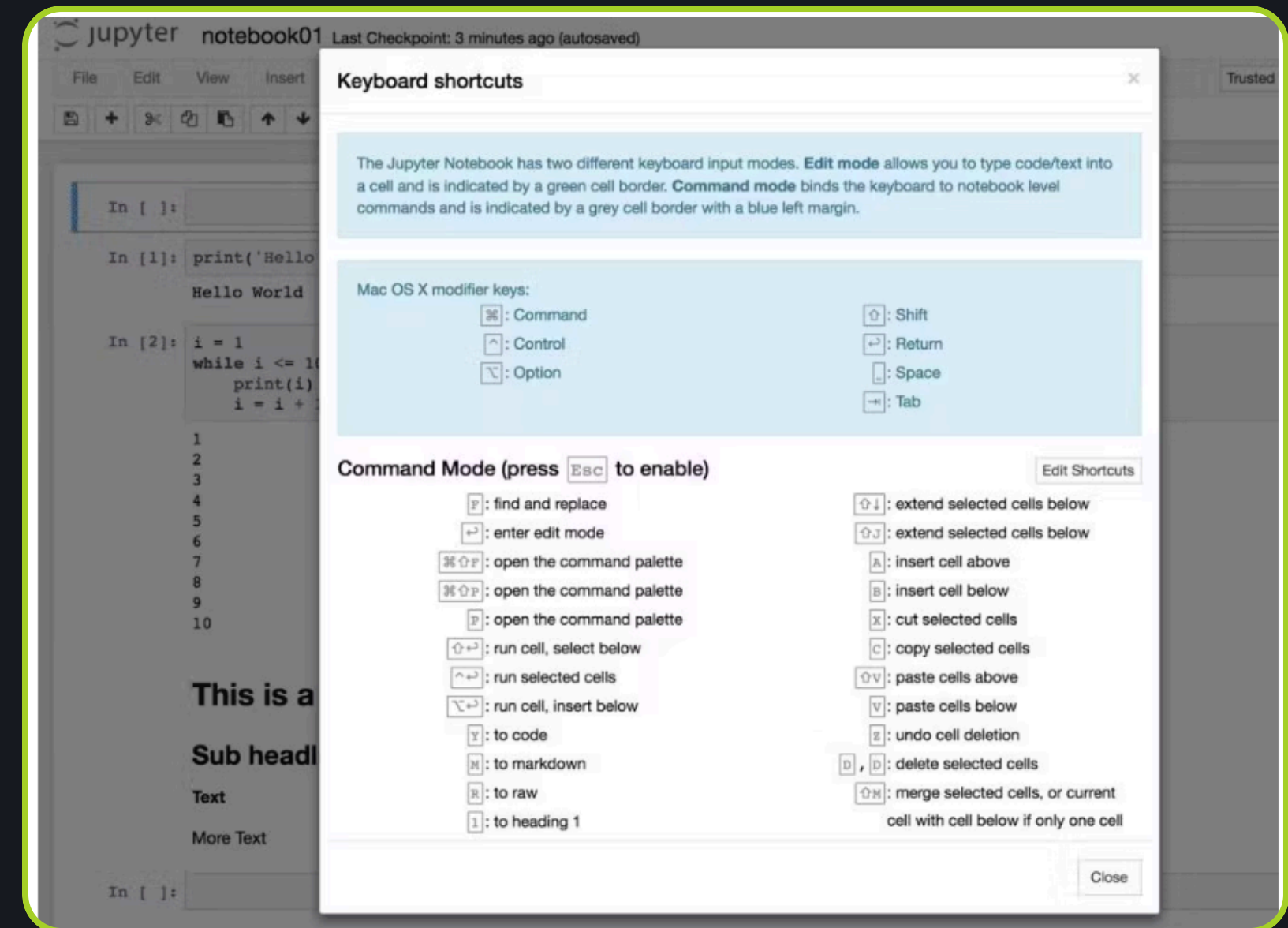
A screenshot of a Jupyter Notebook cell in Edit mode. The cell has a light green vertical bar on the left side. The code 'In [1]: print('Hello World')' is in the top bar, and the output 'Hello World' is in the bottom bar.

```
In [1]: print('Hello World')
Hello World
```

# Jupyter Notebooks 'Comment utiliser';

Appuyez sur ÉCHAP pour revenir au mode Commande.

Utilisez Aide → Raccourcis clavier pour obtenir un aperçu des fonctions disponibles dans les deux modes.



## Étape 7 : points de contrôle

Une fonction utile du bloc-notes Jupyter est sa capacité à créer des points de contrôle. Un point de contrôle enregistre l'état actuel du bloc-notes, ce qui permet de le rétablir ultérieurement si des modifications ont été apportées au bloc-notes depuis la création du point de contrôle.

Pour créer un nouveau point de contrôle, vous pouvez utiliser l'option Enregistrer et point de contrôle du menu Fichier (Fichier → Enregistrer et point de contrôle). Cela crée un point de contrôle et enregistre le fichier du bloc-notes.

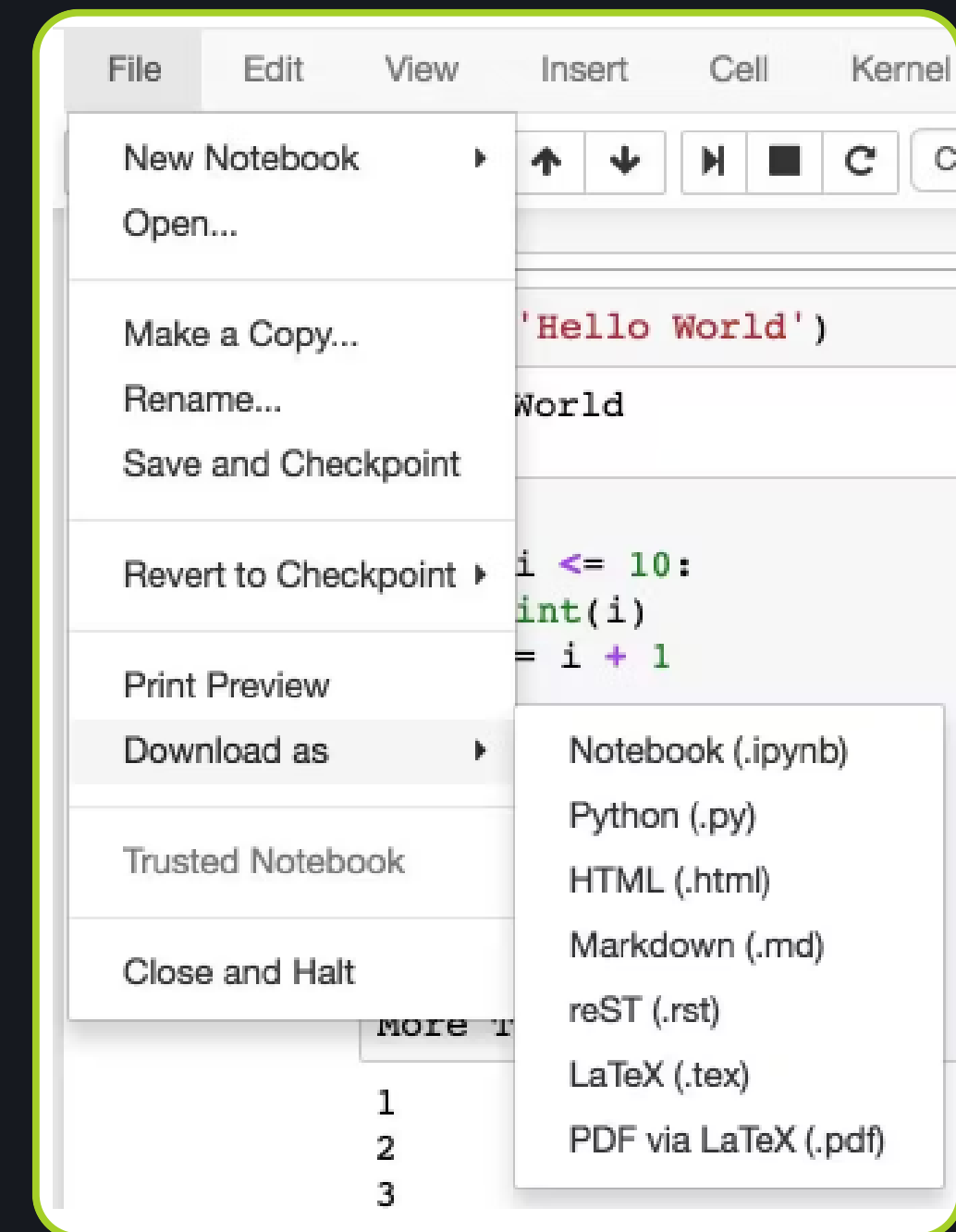
Si vous souhaitez revenir au point de contrôle ultérieurement, vous pouvez le faire en utilisant l'option Fichier → Revenir au point de contrôle.



## Étape 8 : Exporter le notebooks

Vous pouvez exporter le bloc-notes en cliquant sur Fichier → Télécharger en tant qu'options.

Choisissez l'un des formats proposés dans la liste, allant de .ipynb à .pdf.



## Résumé des raccourcis de commande du bloc-notes Jupyter

Les notebooks Jupyter disposent de quelques astuces clavier rapides qui vous permettent de travailler plus facilement. Voici quelques-unes des touches clés que vous devez connaître :

### Naviguer dans le bloc-notes

- **Esc** : Basculer entre le mode édition et le mode commande dans une cellule
- **Enter** : Basculer vers le mode édition dans une cellule
- **Flèche haut/bas** : Déplacer vers la cellule au-dessus ou en dessous
- **A ou B** : Ajouter une nouvelle cellule au-dessus ou en dessous de la cellule actuelle
- **X** : Couper la cellule sélectionnée
- **C** : Copier la cellule sélectionnée
- **V** : Coller la cellule copiée ou coupée
- **D, D** (appuyer deux fois sur la touche D) : Supprimer la cellule sélectionnée
- **Z** : Annuler la dernière suppression de cellule
- **Shift** : Sélectionner plusieurs cellules à la fois

## Modification des cellules

- **Tab** : Aide au code ou ajouter un retrait
- **shift + Tab** : Supprimer un retrait
- **Ctrl + /** : Créer ou supprimer un commentaire sur une ligne

## Exécution de la cellule

- **Ctrl + Enter** : Exécuter la cellule actuelle
- **Alt + Enter** : Exécuter la cellule actuelle et en ajouter une nouvelle en dessous
- **00 (appuyer deux fois sur la touche 0)** : Redémarrer le moteur du bloc-notes
- **I, I (appuyer deux fois sur la touche I)** : Arrêter le moteur du notebook s'il est bloqué

## Informations sur le noyau

- **I** - Arrêter le moteur s'il tourne trop longtemps
- **0, 0** (appuyer deux fois sur la touche 0) - Redémarrer le moteur
- **shift + M** : Combiner les cellules sélectionnées ensemble

# Manipulation de données '**Pandas**';

Installation '**pip install pandas**'

{

— **Importation ()**

```
import pandas as pd.
```

— **Creation de dataframe1()**

```
Variable_nom = pd.DataFrame ()
```

— **Attribution de valeur()**

```
Variable_nom ['Nom'] = ['Onesime', 'cyrille', 'Michel', 'Aimee']
```

```
Variable_nom ['Nom'] = [23, 12, 34, 90]
```

```
Variable_nom ['Nom'] = ['Vrai', 'Faux', 'Faux', 'Vrai']
```

```
Print(Variable_nom)
```

}

Pratique : **‘Voir Jupyter Notebook’** ;

Link : ;

THANK YOU!