Langage Python

ISUP, Sorbonne Université

Etienne Guével
Ingénieur de Recherche - SCAI
etienne.guevel@sorbonne-universite.fr

Septembre-Novembre 2025

Objectifs du cours

Thématiques abordées

- Fondamentaux du langage et outils associés
- Programmation orientée objet
- Pratique des librairies scientifiques

Chaque semaine se déroulera en 2 parties :

- CM
- TP sur machine

Evaluation finale sur projet

Contenu du cours

Partie 1 – Fondamentaux

- Revue des outils, installation et prise en main
- Syntaxe, structures de contrôle
- Types, structures de données

Partie 2 – Programmation orientée objet

- Classe, objets
- Attributs, méthodes
- Héritage

Partie 3 – Librairies scientifiques : NumPy, Pandas, Matplotlib

Partie 4 – Packaging, tests unitaires et bonnes pratiques

Quelques références

- The Python Language Reference
  docs.python.org/3/reference/index.html
- Luciano Ramalho (2019) Fluent Python: Clear, concise and effective programming, O'Reilly.
- www.w3schools.com/python
- realpython.com

Introduction

Introduction

Historique

Le langage Python a été créé par Guido van Rossum en 1989 et rendu public en 1991. Le nom fait référence aux *Monty Python*.

G. van Rossum a été jusqu'à 2018 "Benevolent Dictator for Life".

`www.python.org`

Un langage facile à interpréter

- Simplificité d'écriture et flexibilité

duck.py

part_1 > example_code > duck.py > ...

```python
class Duck:
    def __init__(self):
        self.name = 'duck'

    def quack(self):
        print('Duck says quack!')

    def fly(self):
        print('Duck flies away!')

duck = Duck()
duck.quack()
duck.fly()

```

Un langage facile à interpréter

- Simplificité d'écriture et flexibilité
- Exécution par un interpréteur ligne par ligne

duck.py

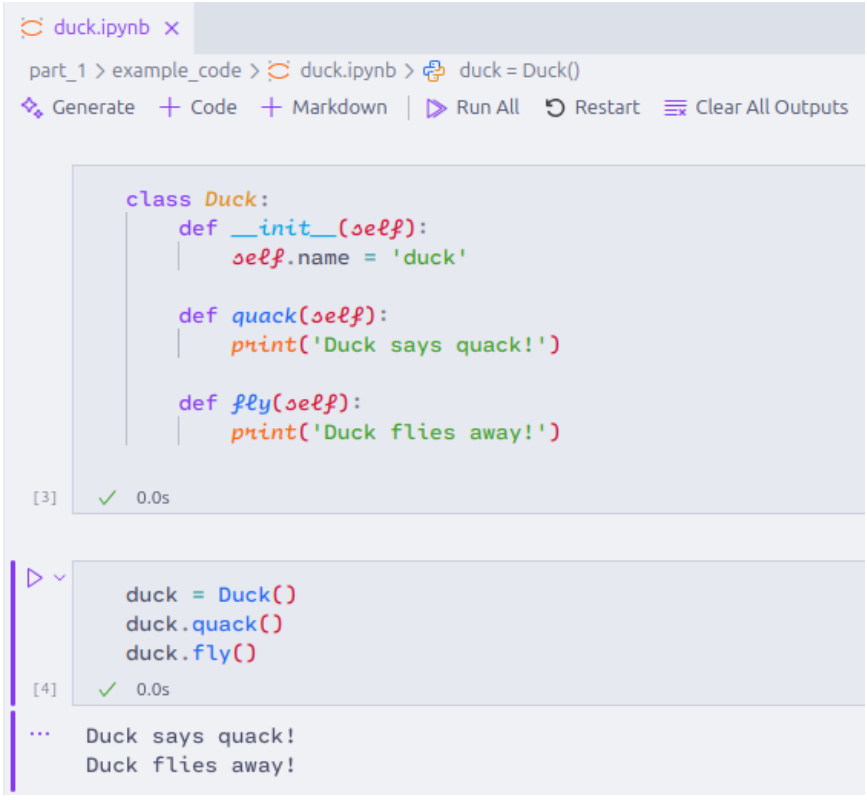
part_1 > example_code > duck.py > ...

```
1   class Duck:
2       def __init__(self):
3           self.name = 'duck'
4
5       def quack(self):
6           print('Duck says quack!')
7
8       def fly(self):
9           print('Duck flies away!')
10
11  duck = Duck()
12  duck.quack()
13  duck.fly()
14
```

Très adaptable

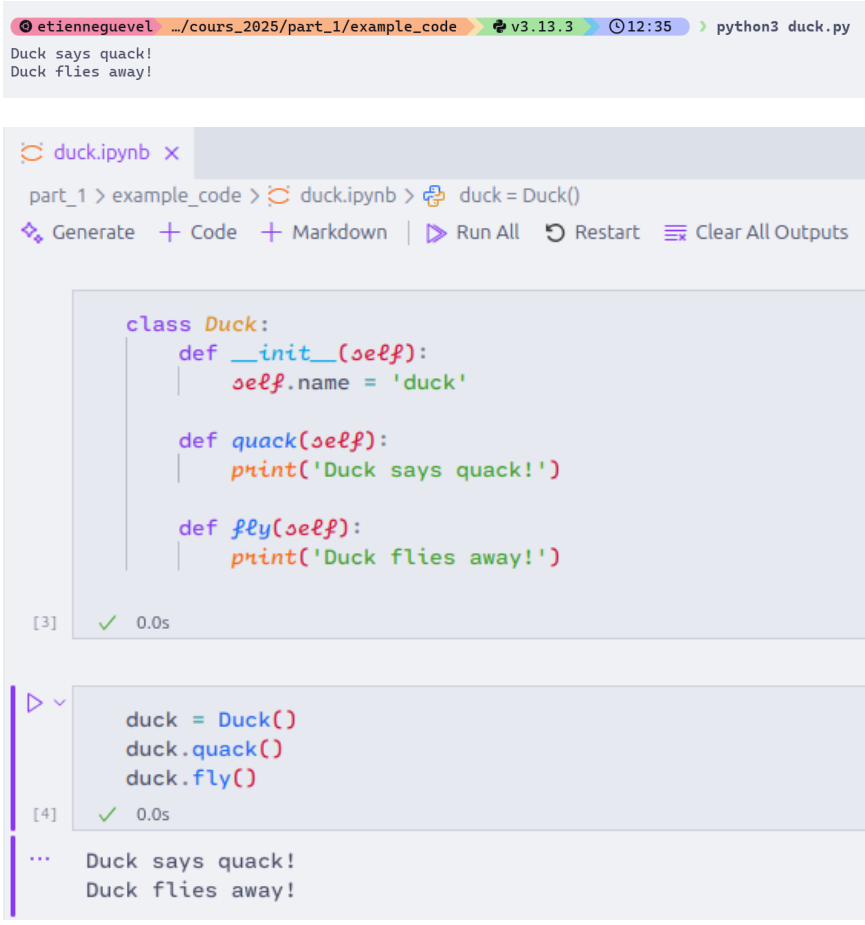
- Exécution intéractive ou par script

```
etienneguevel …/cours_2025/part_1/example_code v3.13.3 12:35 python3 duck.py
Duck says quack!
Duck flies away!
```



Très adaptable

- Exécution intéractive ou par script
- Multiplateforme et open source

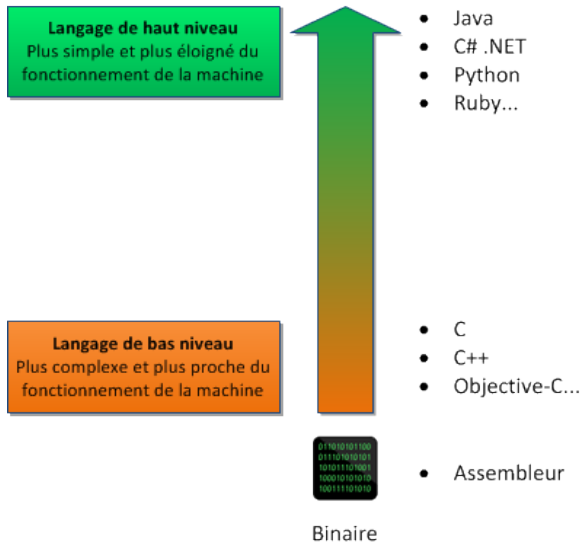


Très adaptable

- Exécution intéractive ou par script
- Multiplateforme et open source
- Correction de bugs **relativement simple**

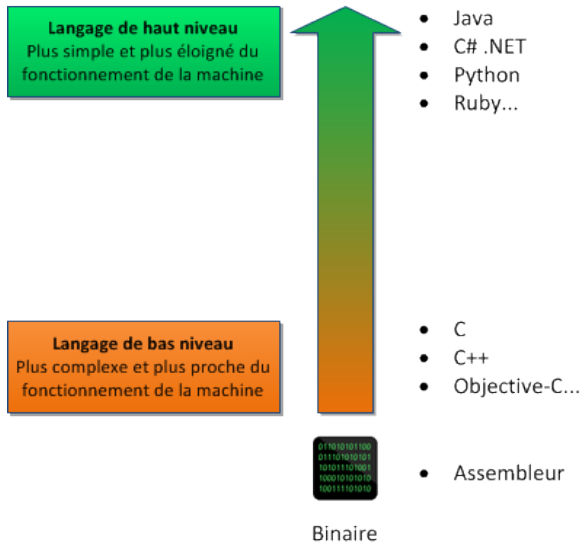
Mais relativement peu performant

- Interprétation vs compilation



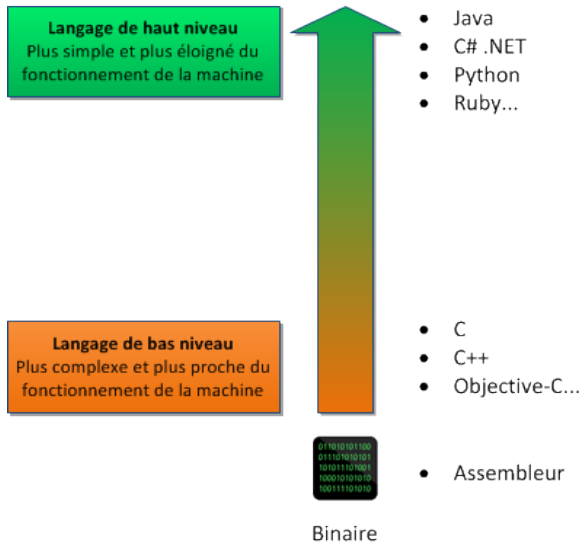
Mais relativement peu performant

- Interprétation vs compilation
- Gestion de données et bibliothèques de haut niveau



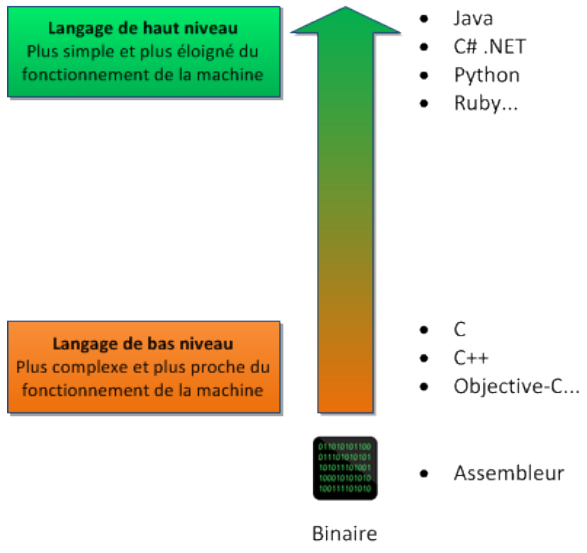
Mais relativement peu performant

- Interprétation vs compilation
- Gestion de données et bibliothèques de haut niveau
- Typage dynamique

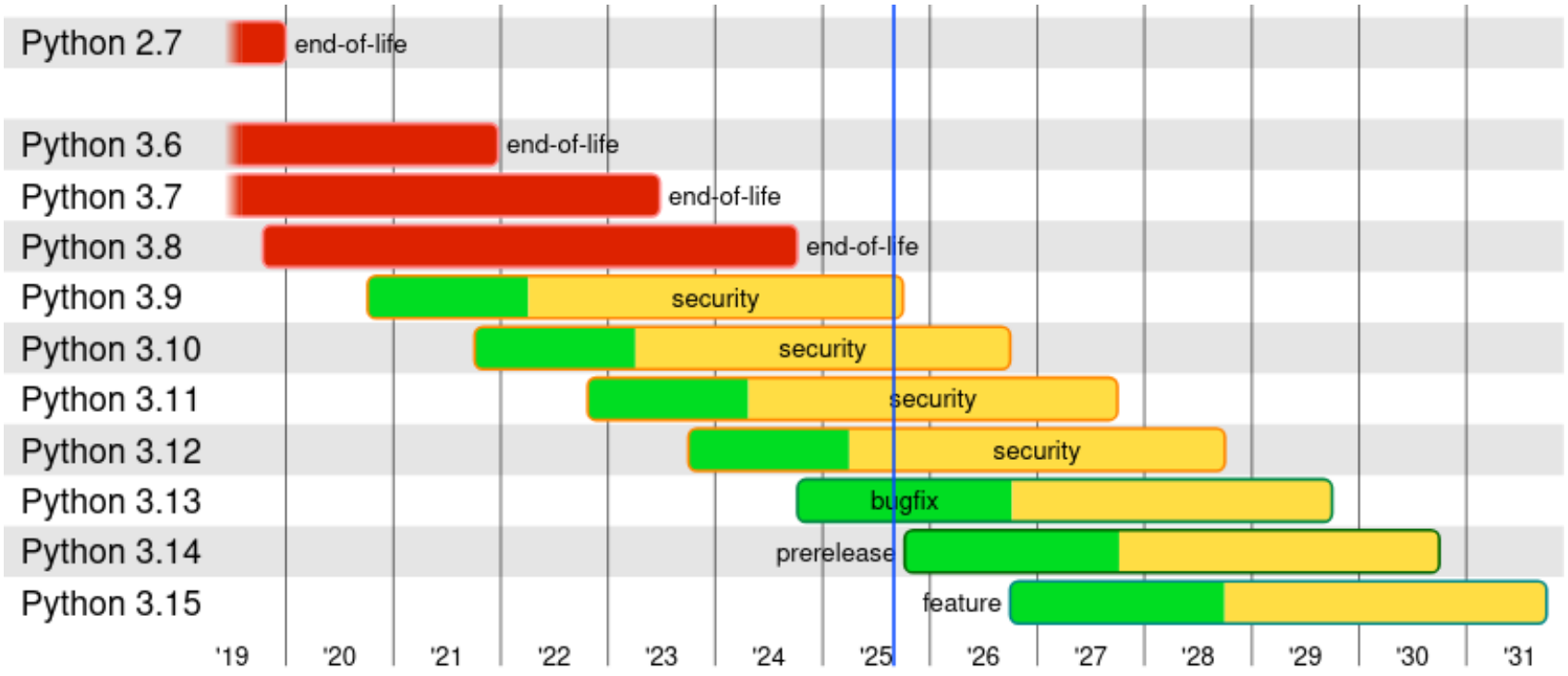


Mais relativement peu performant

- Interprétation vs compilation
- Gestion de données et bibliothèques de haut niveau
- Typage dynamique
- Gestion automatique de mémoire



Une histoire de versions



Installation et revue des outils

Pré-requis pour le cours

- Un terminal

Pré-requis pour le cours

- Un terminal
  - Linux / MacOS : le terminal par défaut

Pré-requis pour le cours

- Un terminal
  - Linux / MacOS : le terminal par défaut
  - Windows : Powershell, intégré à Anaconda

Pré-requis pour le cours

- Un terminal
  - Linux / MacOS : le terminal par défaut
  - Windows : Powershell, intégré à Anaconda
  - Windows 10 ou 11 (conseil) : Windows Subsystem for Linux (wsl)

Pré-requis pour le cours

- Un terminal
  - Linux / MacOS : le terminal par défaut
  - Windows : Powershell, intégré à Anaconda
  - Windows 10 ou 11 (conseil) : Windows Subsystem for Linux (wsl)
- Python, Anaconda, Miniconda version $\geq$ 3.10

Pré-requis pour le cours

- Un terminal
  - Linux / MacOS : le terminal par défaut
  - Windows : Powershell, intégré à Anaconda
  - Windows 10 ou 11 (conseil) : Windows Subsystem for Linux (wsl)
- Python, Anaconda, Miniconda version ≥ 3.10
- jupyter-lab ou jupyter-notebook

Pré-requis pour le cours

- Un terminal
  - Linux / MacOS : le terminal par défaut
  - Windows : Powershell, intégré à Anaconda
  - Windows 10 ou 11 (conseil) : Windows Subsystem for Linux (wsl)
- Python, Anaconda, Miniconda version ≥ 3.10
- jupyter-lab ou jupyter-notebook
- Un IDE (ex. VSCode, PyCharm)

Commandes Linux à connaître

Linux command	Description	Linux command example
cd	Change directory with a specified path	cd */path/directory1*
clear	Clear the screen	clear
cp	Copy file(s)	cp */path1/file1 /path2/file1*
diff	Compare the contents of files	diff *file1 file2*
exit	Log out of Linux	exit
grep	Find a string of text in a file	grep "word or phrase" *file1*
head	Display beginning of a file	head *file1*
less	View a file	less *file1*
ls	List contents of a directory	ls */path/directory1*
mv	Move file(s) or rename file(s)	mv */path1/file1 /path2/file2*
mkdir	Create a directory	mkdir *directory*
rm	Delete file(s)	rm *file1*
rmdir	Remove a directory	rmdir *directory*
tail	Display end of a file	tail *file1*
tar	Store, list or extract files in an archive	tar *file1*
vi	Edit file(s) with simple text editor	vi *file1*

Installation de Python

Python3 est en général déjà installé, sous Linux **Python2 est systématiquement installé**.

Il existe plusieurs façons d'installer python (et ses outils) :

1/ Paquets Python

Pour Linux (ou wsl) :

```
$ sudo apt-get install python3 python3-pip
```

Pour MacOS :

```
$ brew install python
```

Installation de Python

Python3 est en général déjà installé, sous Linux **Python2 est systématiquement installé**.

Il existe plusieurs façons d'installer python (et ses outils) :

2/ Anaconda

- Installation complète de l'environnement
- Un très grand nombre de librairies pré-installées
- anaconda navigator installé
- Multiplateforme : Windows, Linux, MacOS

Voir : `https://docs.anaconda.com/anaconda/install`

Installation de Python

Python3 est en général déjà installé, sous Linux **Python2 est systématiquement installé**.

Il existe plusieurs façons d'installer python (et ses outils) :

3/ Miniconda

- Installation minimale de l'environnement (480 MB contre 4.4GB)
- Aucune librairie installée
- anaconda navigator non installé
- Multiplateforme : Windows, Linux, MacOS

Voir : `https://docs.anaconda.com/miniconda/miniconda-install/`

Installation de Python

Que Choisir ?

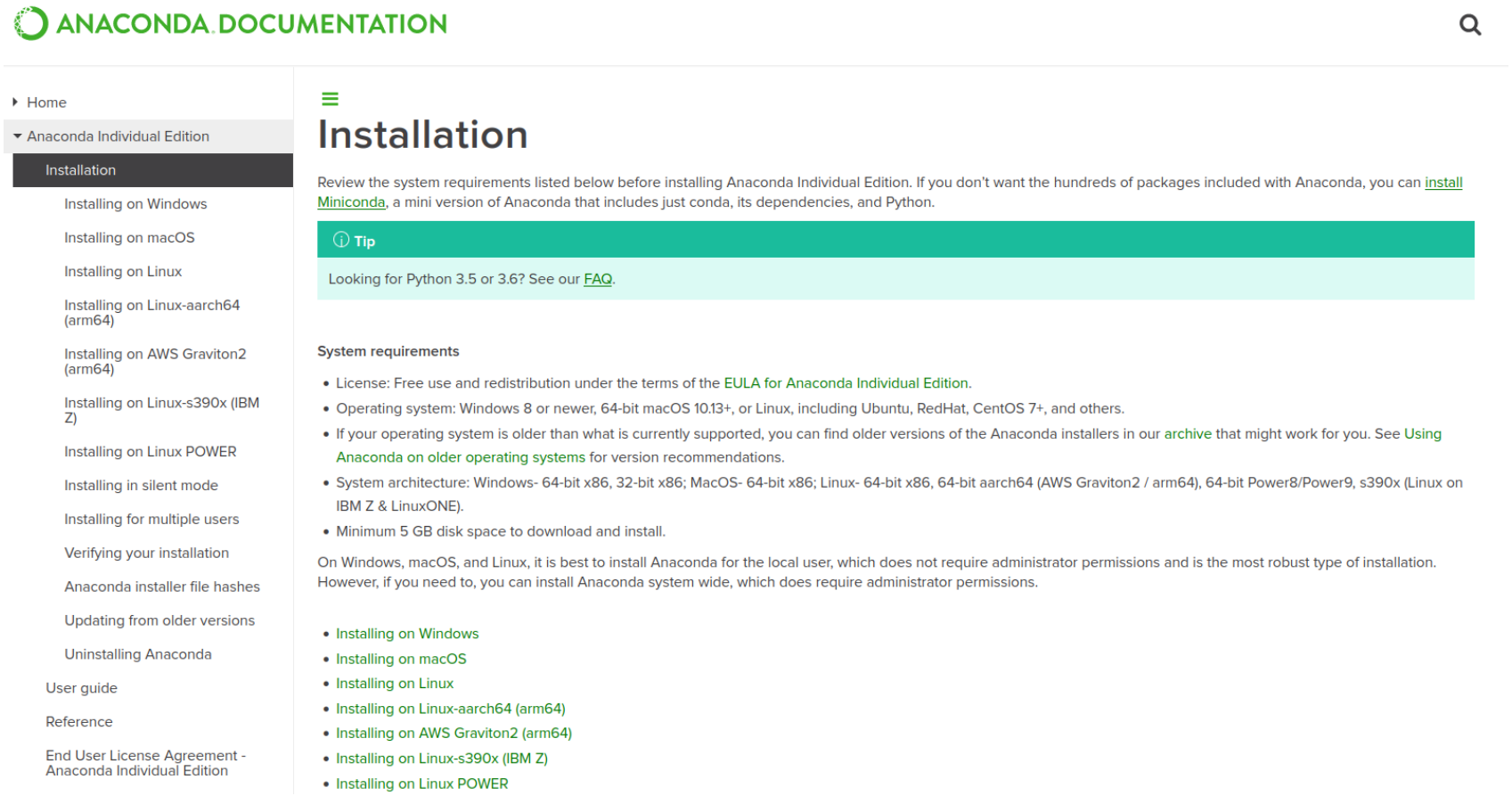
- Python officiel : pour avoir un environnement avec plus de contrôle sur les outils installés
- Anaconda : pour avoir un environnement clé en main
- Miniconda : pour avoir un environnement minimal avec la possibilité de créer des environnements virtuels avec conda

Installation de Python

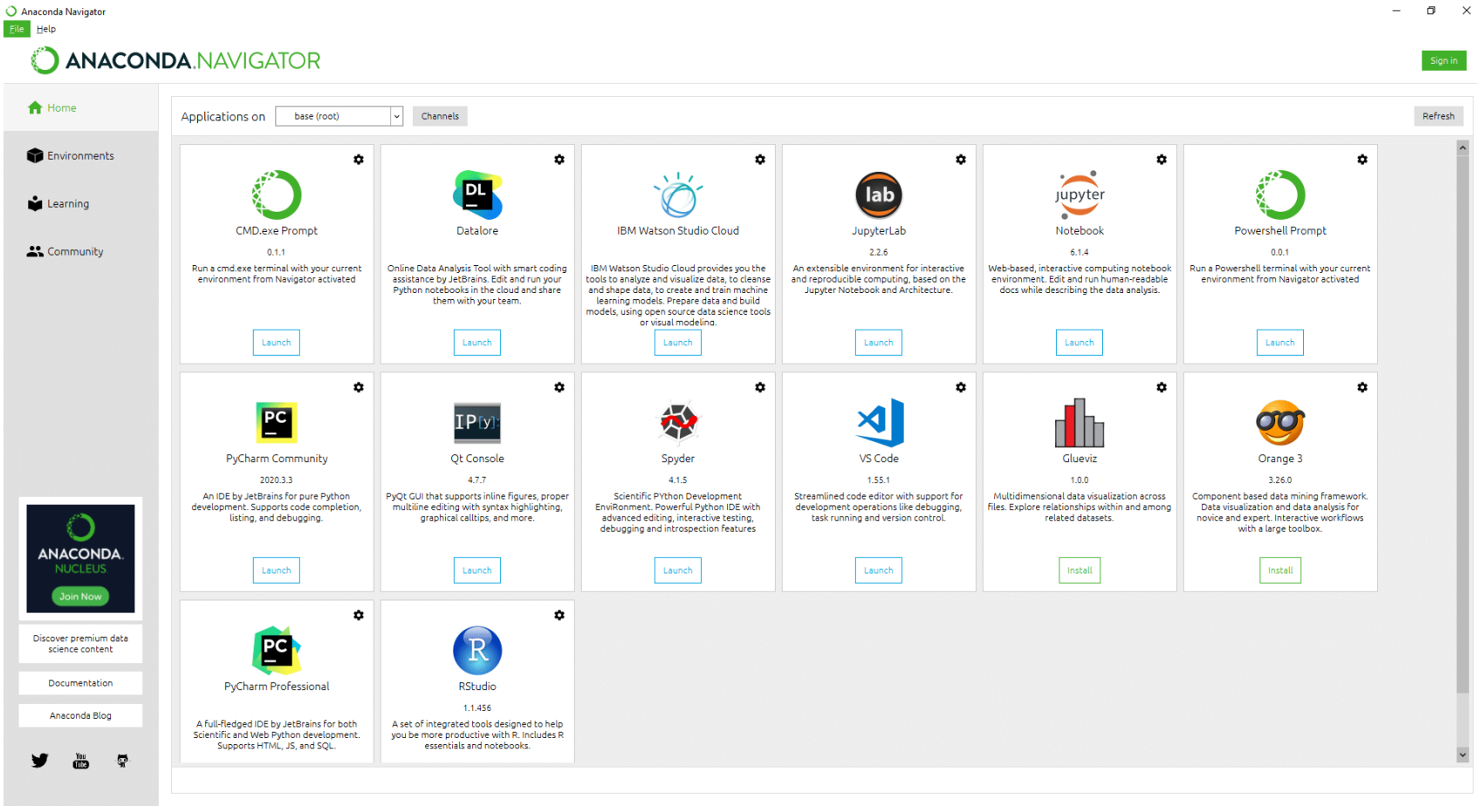
Une fois Python installé, ouvrir un terminal et taper

```
$ python --version
$ python3 --version
```

Installation de Python via Anaconda



Installation de Python via Anaconda



Environnements virtuels

Pour gérer des environnements virtuels, avec une installation spécifique

```
$ python -m venv myenv
$ source myenv/bin/activate
$ pip install numpy
$ deactivate
```

Voir ce lien pour les venvs.

Certaines installations de python viennent sans venv (plutôt pour linux), pour l'installer :

```
$ sudo apt install python3-venv # pour linux
```

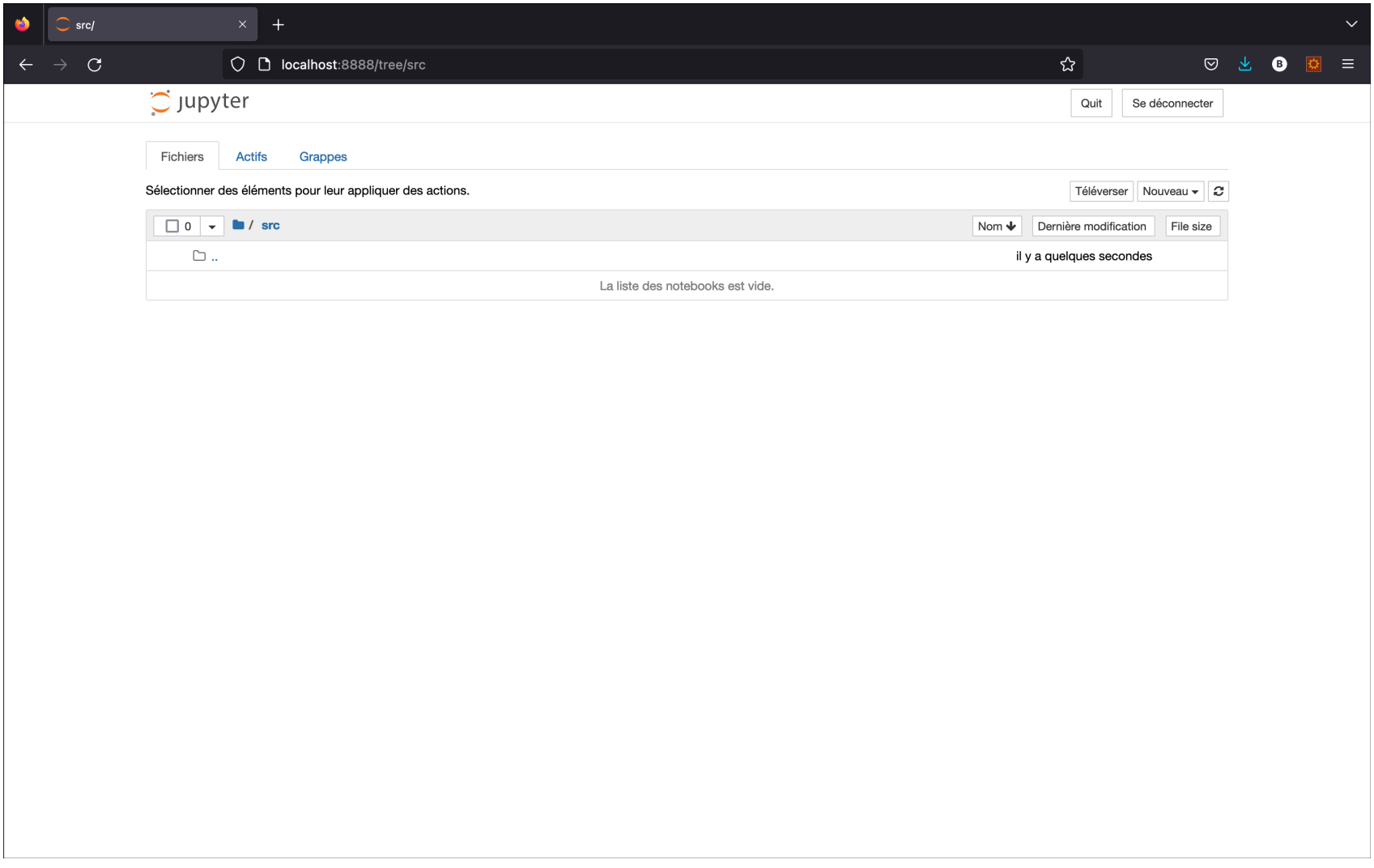
Environnements conda

Pour gérer des environnements virtuels, avec une installation spécifique

```
$ conda create --name myenv python=3.12
$ conda activate myenv
$ conda install pip
$ pip install numpy
$ conda deactivate
```

Voir ce lien pour les envs conda.

Jupyter


src/
localhost:8888/tree/src
jupyter
Quit
Se déconnecter
Fichiers
Actifs
Grappes
Sélectionner des éléments pour leur appliquer des actions.
Téléverser
Nouveau
0
/ src
Nom
Dernière modification
File size
..
il y a quelques secondes
La liste des notebooks est vide.

Jupyter

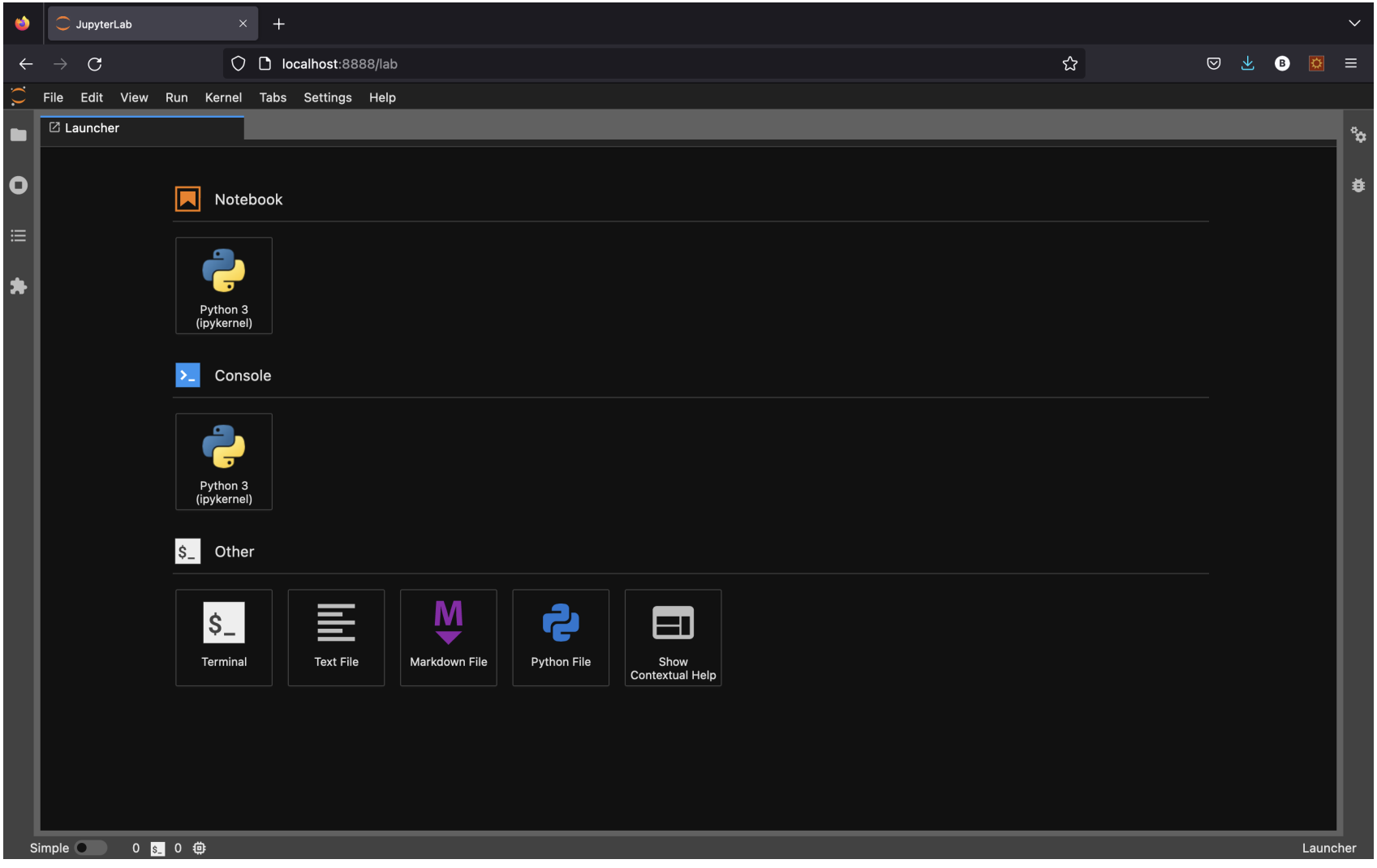
Pour installer jupyter dans un environnement python via pip :

```
$ source my_env/bin/activate
$ pip install jupyter # ou notebook ou jupyterlab
$ pip install ipython # package minimal pour vscode notebook
```

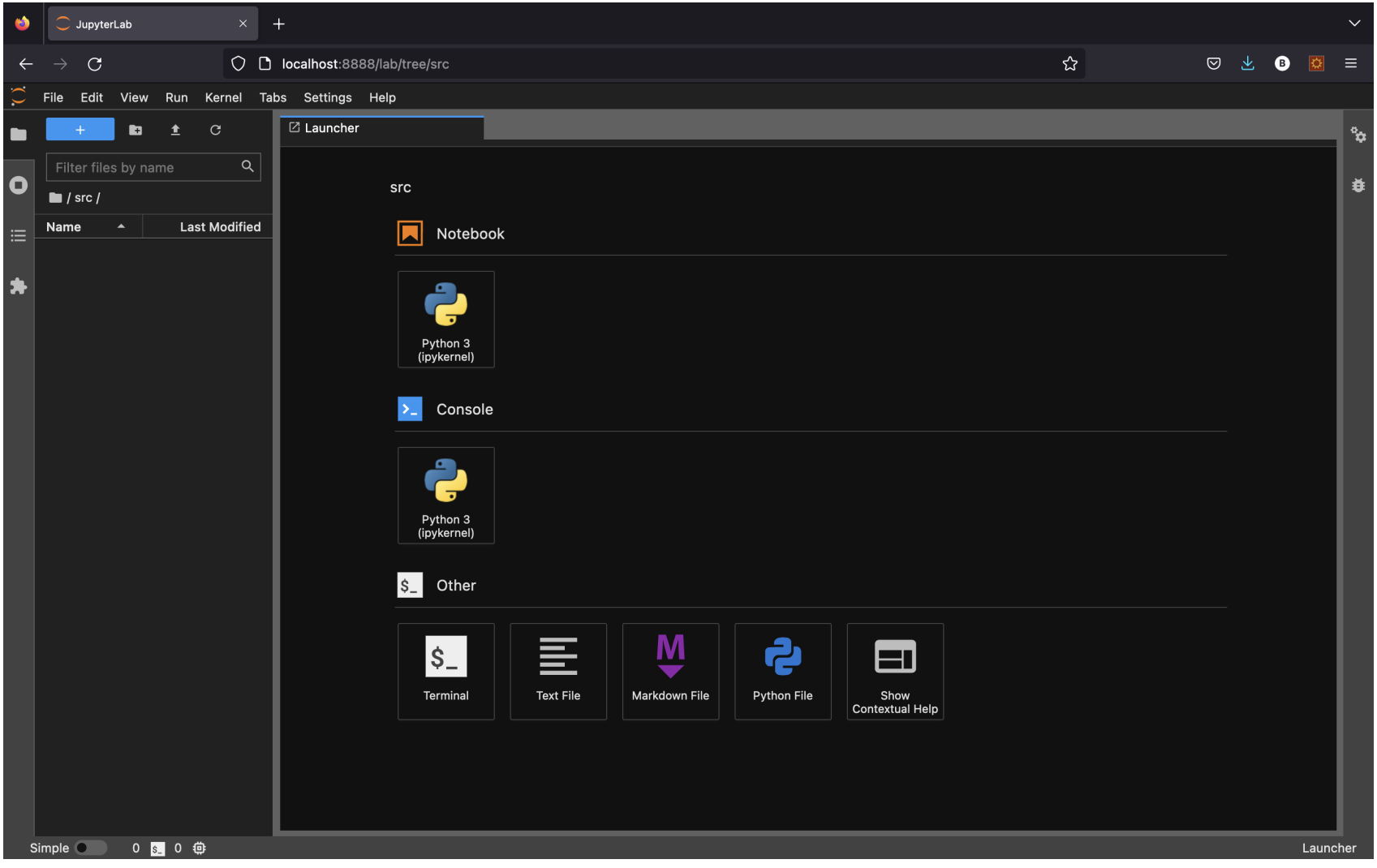
```
$ jupyter lab # ou jupyter notebook
```

Cela va ensuite ouvrir jupyter dans un navigateur. Si ce n'est pas le cas, un lien apparaitra dans le terminal : CTRL + click dessus.

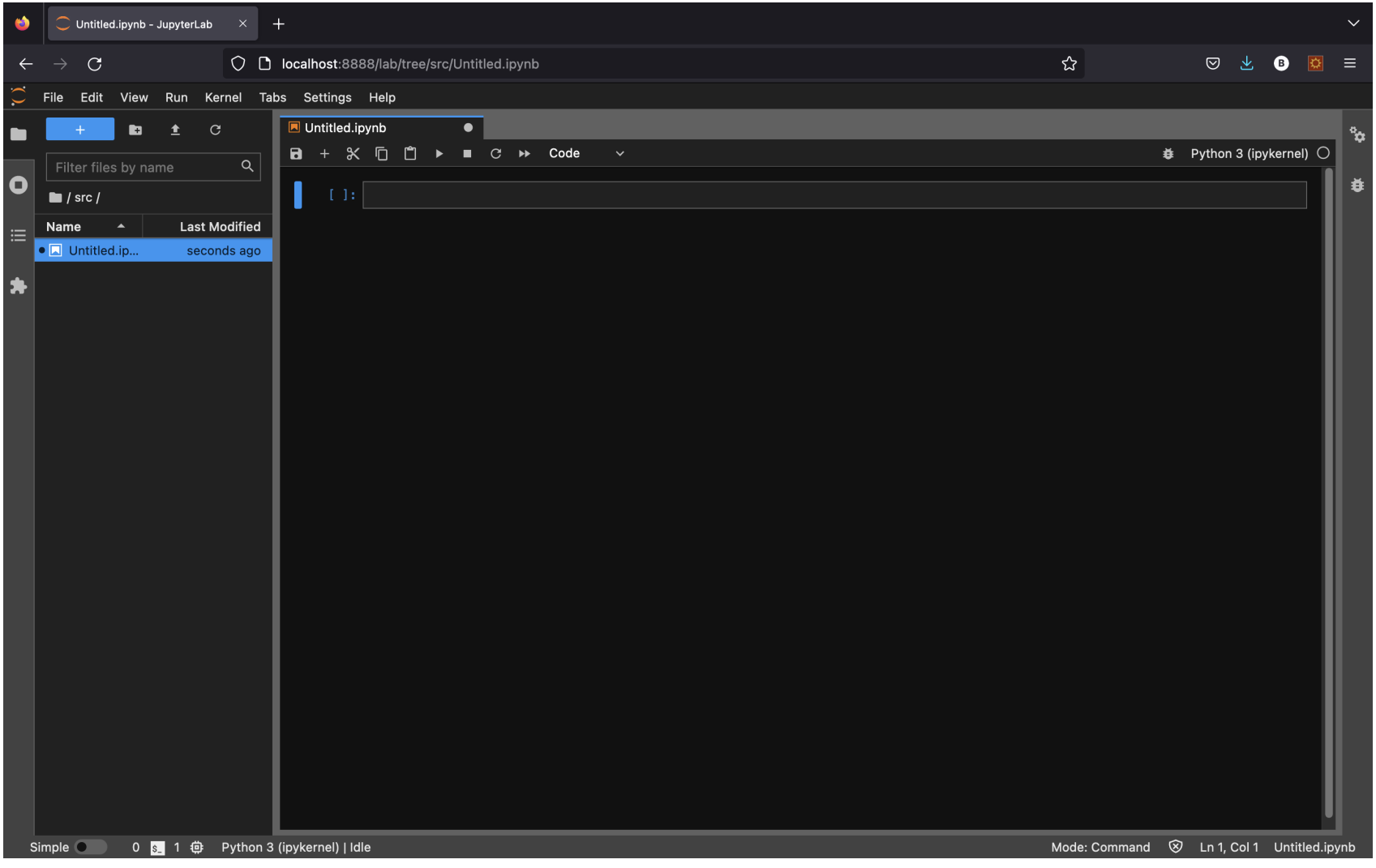
Jupyter


JupyterLab
localhost:8888/lab
File Edit View Run Kernel Tabs Settings Help
Launcher
Notebook
Python 3 (ipykernel)
Console
Python 3 (ipykernel)
Other
Terminal
Text File
Markdown File
Python File
Show Contextual Help
Simple
0 0
Launcher

Jupyter


JupyterLab
localhost:8888/lab/tree/src
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ src /
Name
Last Modified
Launcher
src
Notebook
Python 3 (ipykernel)
Console
Python 3 (ipykernel)
Other
Terminal
Text File
Markdown File
Python File
Show Contextual Help
Simple
0 0
Launcher

Jupyter


Untitled.ipynb - JupyterLab
localhost:8888/lab/tree/src/Untitled.ipynb
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ src /
Name
Last Modified
Untitled.ip...
seconds ago
Untitled.ipynb
Code
Python 3 (ipykernel)
[ ]:
Simple
0 1
Python 3 (ipykernel) | Idle
Mode: Command
Ln 1, Col 1
Untitled.ipynb

Exécution d'un programme python

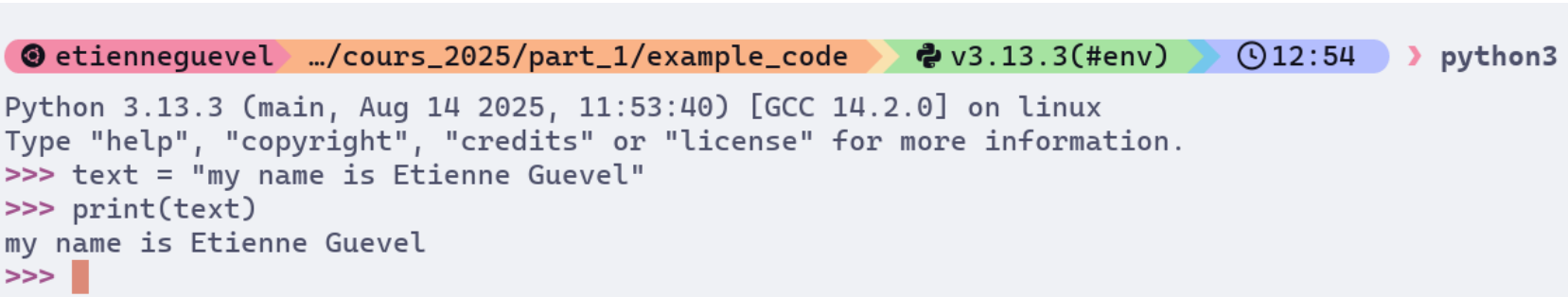
Cas 1 : via un shell python ou ipython

Exécution à la volée d'instructions python. Ouvrir un shell

```
$ python
```

Ecrire des instructions

```
>>> text = "my name is Etienne Guevel"
>>> print(text)
```



Exécution d'un programme python

Cas 2 : via un terminal

Dans un éditeur de texte, créer un fichier `program.py` et écrire

```
text = "my name is Etienne Guevel"
print(text)
```

Exécuter la ligne de commande

```
$ python program.py
```

Exécution d'un programme python

Cas 3 : via Jupyter

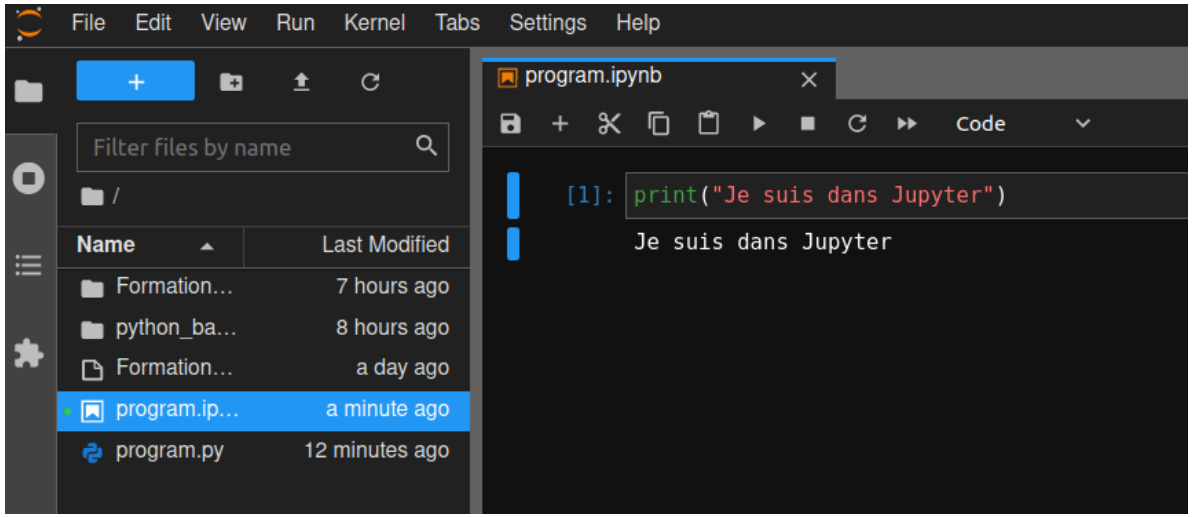
Lancer Jupyter depuis un terminal

```
$ jupyter notebook // ou jupyter lab
```

Dans un Jupyter notebook, écrire

```
print("Je suis dans Jupyter")
```

et exécuter la cellule correspondante



Revue des outils

Outils de base :

- python : exécuter du code à la volée
- pip : installer des packages
- pytest : tests unitaires
- pylint : vérification de la qualité du code source

Revue des outils

Outils de base :

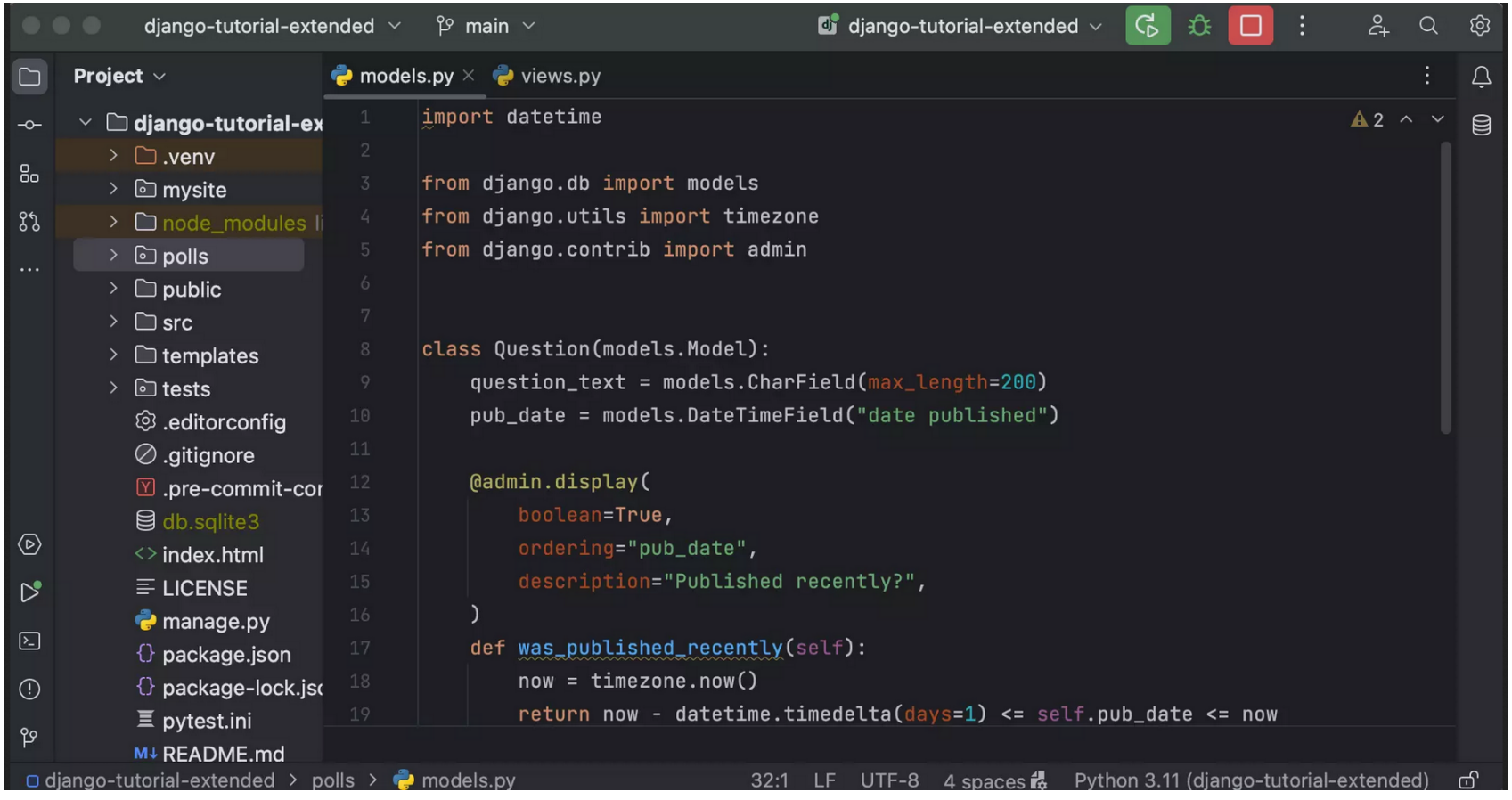
- python : exécuter du code à la volée
- pip : installer des packages
- pytest : tests unitaires
- pylint : vérification de la qualité du code source

Intégré au projet Anaconda :

- ipython : version interactive du shell python
- conda : installer des packages, gérer des environnements d'exécutions (voir ici), etc.
- Jupyter : environnement de développement intéractif et fléxible

Installer des IDEs

Les IDEs sont des environnements de développement intégrés (Integrated Development Environment) qui permettent de faciliter la création de programmes.



Installer des IDEs

Les IDEs sont des environnements de développement intégrés (Integrated Development Environment) qui permettent de faciliter la création de programmes.