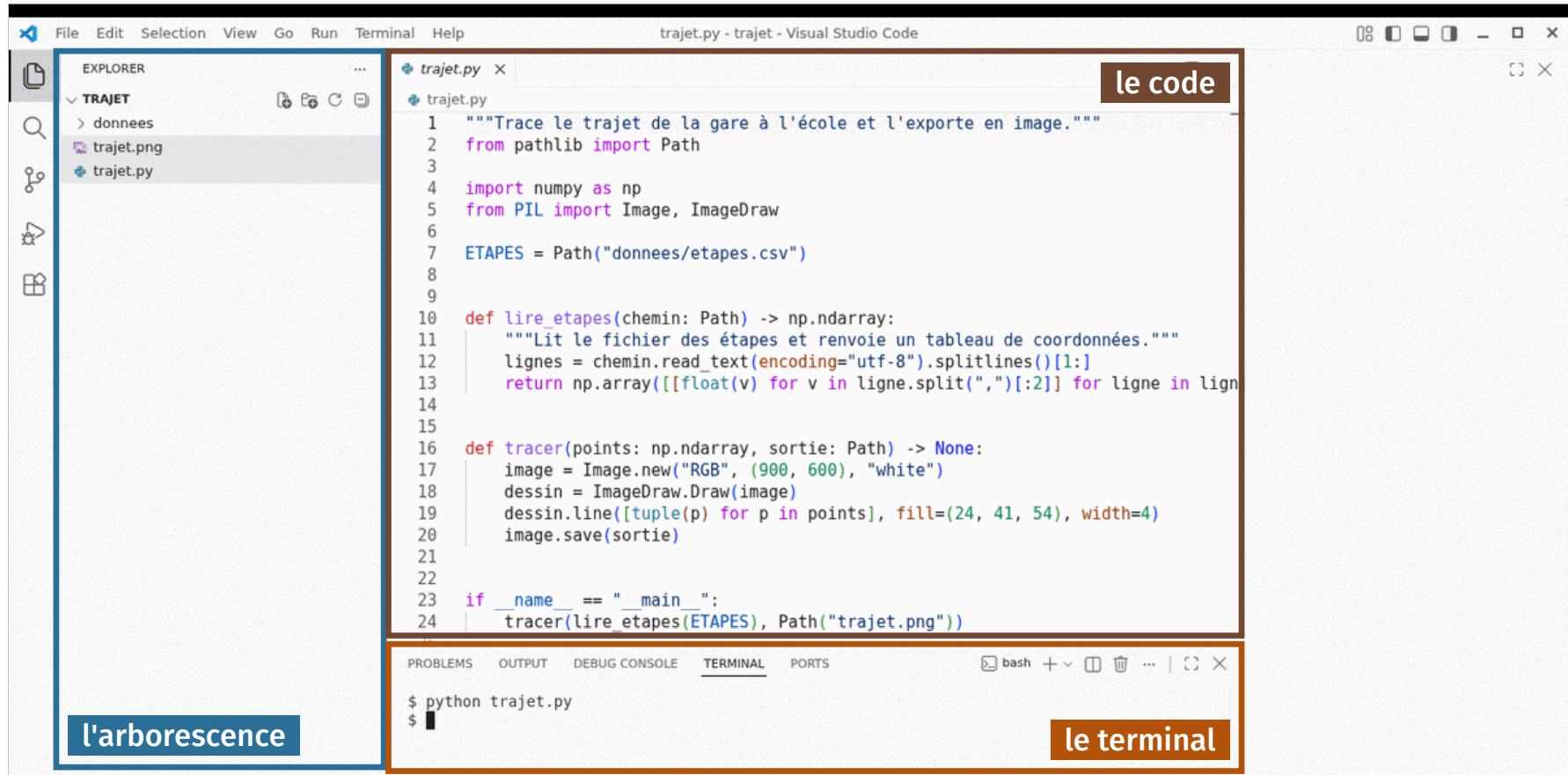


Configurer l'éditeur de code, et lancer un programme

Objectif : configurer un IDE pour Python, puis exécuter un programme de trois façons : en entier, ligne à ligne, pas à pas

`cours1/1c_vscode_python/` $\approx 25'$

Visual Studio Code



Les deux temps du TD

VS Code ne connaît d'avance ni le langage du fichier ouvert, ni l'interpréteur Python à utiliser. Il faut donc configurer l'IDE avant de pouvoir l'utiliser pleinement.

Configurer l'IDE

Le langage

Installation de l'extension Python dans VS Code.

L'interpréteur

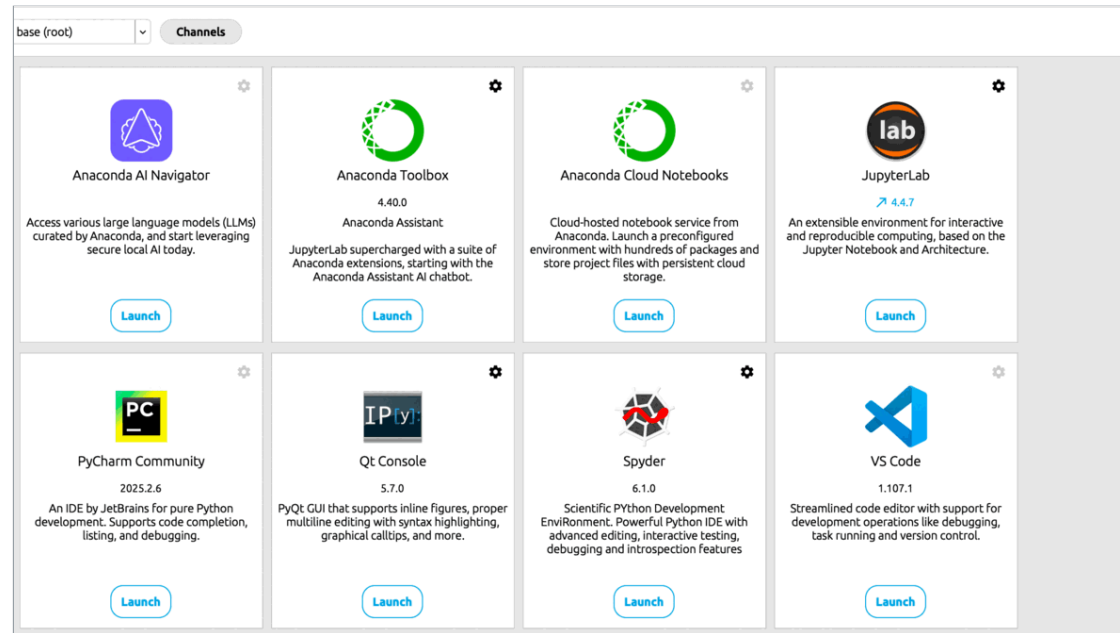
Configurer l'interpréteur Python pour utiliser celui fourni par Anaconda.

Lancer un programme

Ouvrir le dossier du TD. Trois façons : en entier, ligne à ligne, pas à pas.

La page d'accueil d'Anaconda Navigator

En haut, le Python dans lequel Navigator lancera ce qu'on ouvre ; en dessous, une fiche par application, chacune avec son bouton **Launch**.



Page d'accueil de Navigator, documentation Anaconda. Après l'installation, la liste du haut ne contient que `base (root)`.

Lancer VS Code depuis Anaconda

Sur les postes de la salle, VS Code se lance depuis Anaconda Navigator : il est alors configuré pour utiliser l'interpréteur Python affiché en haut de la page d'accueil, sans aucun réglage préalable.

	Ce qu'il faut faire	Ce que vous observez
1	menu Démarrer → Anaconda Navigator	la page d'accueil, une fiche par application
2	en haut, lire la liste déroulante	
3	fiche VS Code → Launch	
4	Terminal → Nouveau terminal, taper <code>python --version</code>	

L'autre chemin, VS Code lancé depuis le bureau, marche aussi, mais son terminal demande un réglage, plus loin dans ce TD.

Installer l'extension d'un langage

Une extension s'installe depuis le panneau Extensions, en cherchant son identifiant. Il en faut une par langage.

Ce qu'il faut faire	Ce que vous observez
Fichier → Ouvrir le dossier, sur cours1/1c_vscode_python/	deux programmes, <code>bonjour.py</code> et <code>altitudes.py</code> , et un <code>README.md</code>
Ouvrir <code>bonjour.py</code> avant toute installation	
Ctrl + Maj + X, chercher <code>ms-python.python</code> , installer	

Chercher l'identifiant, `ms-python.python`, et jamais le nom affiché. Une extension installée le reste, pour les séances suivantes et les autres cours.

La palette de commandes et les réglages

Les commandes et les réglages se cherchent par leur nom, sans parcourir les menus. Quatre panneaux, et les raccourcis qui les ouvrent.

	Raccourci	Rôle
La palette de commandes	Ctrl + Maj + P	taper le début d'un nom : « Python: Select Interpreter »
Les réglages	Ctrl + ,	chercher un mot : « default profile », « render whitespace »
Les extensions	Ctrl + Maj + X	chercher un identifiant : <code>ms-python.python</code>
Le terminal	Ctrl + `	ouvrir, masquer, rouvrir
Le fichier <code>settings.json</code>		les mêmes réglages, écrits en texte

Intitulés de l'interface en anglais. Les réglages ont deux niveaux : **User**, pour vous sur ce poste, et **Workspace**, rangé avec le projet.

Choisir l'interpréteur Python

Plusieurs Python peuvent coexister sur une machine. Désigner celui de l'éditeur suffit : les terminaux qu'il ouvre ensuite emploient le même.

Où	Ce qu'il faut faire	Ce qui le prouve
Dans l'éditeur	Ctrl + Maj + P, « Python: Select Interpreter », choisir celui d'Anaconda	le terminal ouvert ensuite commence par (base)
Windows, hors éditeur	menu Démarrer, chercher « Anaconda Prompt »	l'invite commence par (base)
Linux, macOS	un terminal ordinaire suffit	l'invite commence par (base)

Le nom entre parenthèses, en tête d'invite, désigne le Python qui répondra. C'est la seule marque visible, et elle se relit avant chaque installation.

VS Code hors Anaconda : changer de terminal

Lancé depuis le bureau, VS Code ouvre un terminal PowerShell, qui refuse le script d'activation. Sans droits d'administrateur, on lui donne le terminal de l'invite de commandes d'Anaconda.

```
... \activate.ps1 cannot be loaded because running scripts is disabled on this system.
```

Palette, « Open User Settings (JSON) », puis ajouter :

```
"terminal.integrated.profiles.windows": {  
  "Anaconda Prompt": {  
    "path": "C:\\Windows\\System32\\cmd.exe",  
    "args": ["/K", "C:\\ProgramData\\anaconda3\\Scripts\\activate.bat", "C:\\ProgramData\\anaconda3"]  
  }  
},  
"terminal.integrated.defaultProfile.windows": "Anaconda Prompt"
```

Les chemins d'Anaconda sont ceux du raccourci « Anaconda Prompt » : clic droit → Propriétés → Cible. Puis ouvrir un nouveau terminal : l'invite commence par (base).

Le programme du TD

Six lignes qui calculent une moyenne d'altitudes.

altitudes.py

```
altitudes = [128.4, 131.0, 127.6]
total = 0
for altitude in altitudes:
    total = total + altitude
moyenne = total / len(altitudes)
print(f"moyenne : {moyenne:.1f} m")
```

Sortie réelle, Python 3.12. La moyenne de 128,4, 131,0 et 127,6 vaut bien 129,0.

Lancé en entier

```
$ python altitudes.py
moyenne : 129.0 m
```

Une seule ligne de sortie, celle du `print` final. Ce qui s'est passé entre-temps n'est pas visible.

Lancer le programme

	Ce qu'il faut faire
1	Fichier → Ouvrir le dossier, puis choisir <code>cours1/1c_vscode_python/</code>
2	<code>Ctrl + Maj + P</code> , « Python: Select Interpreter », choisir celui d'Anaconda
3	Terminal → Nouveau terminal : il s'ouvre en bas, dans <code>1c_vscode_python/</code> , l'invite commence par <code>(base)</code>
4	taper <code>python altitudes.py</code> , puis Entrée

Deux autres chemins pour le même lancement : le bouton d'exécution en haut à droite, et le menu Run → Run Without Debugging (`Ctrl + F5`). L'un comme l'autre écrit sa commande dans le terminal avant de l'exécuter, et c'est cette commande qu'il faut savoir écrire : elle est identique sur les trois systèmes.

Python en interactif

Dans le même terminal, taper `python` sans nom de fichier ouvre une session interactive : chaque ligne est lue, exécutée, et son résultat affiché aussitôt.

```
$ python
Python 3.12.14 (main, Sep  2 2026, 23:27:36) [GCC 15.3.0] on linux
>>> altitudes = [128.4, 131.0, 127.6]
>>> total = 0
>>> for altitude in altitudes:
...     total = total + altitude
...
>>> total
387.0
>>> total / len(altitudes)
129.0
>>> exit()
```

Session réelle. `total` s'affiche sans `print` : c'est propre à la session interactive, et c'est ce qui permet de regarder à l'intérieur.

En option : sans VS Code, depuis l'invite de commandes d'Anaconda

L'éditeur n'est qu'une fenêtre autour du terminal. Tout ce qui précède se refait dans l'invite de commandes d'Anaconda seule, pour le voir.

	Ce qu'il faut taper	Ce que vous observez
1	menu Démarrer → Anaconda Prompt	l'invite commence par (base)
2	cd , puis glisser le dossier 1c_vscode_python dans la fenêtre, Entrée	
3	python altitudes.py	
4	python, les lignes de la session interactive, exit()	

Ce que VS Code faisait à votre place : se placer dans le dossier du projet, à l'étape 2.

Exécuter pas à pas : poser un arrêt, puis lancer

Le débogueur arrête le programme sur une ligne choisie et laisse regarder les variables, sans rien ajouter au code. D'abord l'arrêt, puis le lancement.

	Ce qu'il faut faire	Ce que vous observez
1	ouvrir <code>altitudes.py</code> , cliquer dans la marge à gauche du numéro de la ligne 4	un point rouge : le programme s'y arrêtera
2	F5	en haut, VS Code demande quoi déboguer
3	choisir « Python Debugger », puis « Python File »	
4	regarder à gauche, panneau Run and Debug (Ctrl + Maj + D)	

La question de l'étape 2 ne vient qu'au premier lancement : ne rien créer d'autre, ne pas enregistrer de configuration. Le point rouge est un réglage de l'éditeur, le fichier n'a pas changé.

Exécuter pas à pas : avancer et lire les variables

Une ligne à la fois, avec F10. À chaque arrêt, prédire la valeur de `total` avant de la lire.

Bouton	Touche	Ce qu'il fait
Continue	F5	reprend jusqu'au prochain arrêt, ou jusqu'à la fin
Step Over	F10	exécute la ligne surlignée et s'arrête à la suivante
Step Into	F11	entre dans la fonction appelée — pas aujourd'hui
Restart	Ctrl + Maj + F5	recommence du début
Stop	Maj + F5	arrête le programme

	Ce qu'il faut faire
5	F10, trois fois, en regardant Variables
6	F5