

TD 1c – Configurer l'éditeur de code, et lancer un programme

Guide détaillé, étape par étape

Le TD prépare l'éditeur de code du module, Visual Studio Code (VS Code), puis s'en sert pour exécuter un petit programme Python de trois façons. VS Code sait éditer du texte dans n'importe quel langage, mais il ne sait pas d'avance quel interpréteur Python employer, ni comment lancer un programme : il faut lui ajouter l'extension Python et lui désigner l'interpréteur installé par Anaconda. Le TD dure environ vingt-cinq minutes.

La configuration de VS Code se fait une fois par poste et par compte, et reste valable toute l'année. Le programme exécuté, `altitudes.py`, calcule la moyenne de trois altitudes : il est lancé en entier, puis son calcul est refait ligne à ligne dans une session interactive, puis son exécution est suivie pas à pas avec le débogueur. Le TD nécessite Anaconda Navigator, VS Code, et une connexion réseau pour installer l'extension.

Étape	Objectif	Ce qu'on fait
1	démarrer l'IDE avec le Python d'Anaconda	lancer VS Code depuis Anaconda Navigator
2	donner à l'IDE le langage du projet	ouvrir le dossier du TD, et installer l'extension Python
3	choisir le Python qui exécute le code	choisir l'interpréteur, et régler le terminal
4	exécuter un programme en entier, puis ligne à ligne	lancer le programme en entier, puis ligne à ligne
5	suivre l'exécution et la valeur des variables	exécuter le programme pas à pas

Chaque étape commence par un encadré qui la résume. Ce que chaque étape fait constater est expliqué à la fin du guide, dans « Ce que le TD fait constater » : faire l'étape d'abord, et noter ce qu'on observe, avant de lire l'explication.

VS Code s'affiche en anglais, et le module ne demande pas d'en changer : les noms de menus et de commandes de ce guide sont ceux de l'interface anglaise.

1 · Lancer VS Code depuis Anaconda Navigator

À faire : ouvrir Anaconda Navigator ; lire l'environnement affiché en haut ; lancer VS Code depuis sa fiche ; ouvrir un terminal et y taper `python --version`.

À obtenir : VS Code ouvert, et un terminal en bas de sa fenêtre qui répond à `python --version`.

Le dossier du TD

Le dossier du TD est dans l'archive de la séance, extraite dans `C:\Users\eleve\Desktop\info01\` (la page « Récupérer les fichiers d'une séance » du site du cours détaille cette opération). Il contient :

```
lc_vscode_python\  
├─ altitudes.py          la moyenne de trois altitudes  
├─ bonjour.py            un programme qui affiche une phrase  
├─ README.md             la présentation du TD  
├─ td_lc_vscode_python.pdf la feuille du TD  
└─ guide_lc_vscode_python.pdf ce guide
```

Son chemin complet est `C:\Users\eleve\Desktop\info01\cours1\lc_vscode_python\`. Le TD ne modifie aucun de ces fichiers, et n'en crée pas : il n'a ni `depart\` ni `travail\`.

Anaconda Navigator

1. Menu Démarrer, taper `anaconda navigator`, puis Entrée. Une fenêtre « Loading applications... » s'affiche, puis la page d'accueil. Le premier lancement de l'année peut prendre jusqu'à deux minutes : cliquer une seule fois, puis attendre.
2. Si Navigator propose une mise à jour, répondre No.
3. En haut de la page d'accueil, lire la liste déroulante des environnements, sans la changer.
4. En dessous, une fiche par application, chacune avec son bouton Launch. Trouver la fiche VS Code, et cliquer sur Launch.

Les fiches ne sont pas les mêmes d'un poste à l'autre. Une application qui n'est pas installée porte un bouton Install à la place de Launch. Seule la fiche VS Code sert dans ce TD.

À noter : ce que contient la liste déroulante du haut, et ce qui y est sélectionné.

La fenêtre de VS Code

VS Code s'ouvre, parfois sur une page de bienvenue qu'on peut fermer. Sa fenêtre a trois zones, qui servent toutes dans ce TD :

Zone	Où	Ce qu'elle montre
l'arborescence	à gauche	les dossiers et les fichiers du dossier ouvert
le code	au centre	le fichier ouvert, avec ses numéros de ligne
le terminal	en bas	une fenêtre de commandes, ouverte à la demande

Ouvrir le terminal : menu Terminal, New Terminal. Le raccourci `Ctrl + `` fait la même chose sur un clavier français. Dans le terminal, taper la commande suivante, puis Entrée :

```
python --version
```

Si le terminal affiche, en rouge, un message qui contient `activate.ps1` et `running scripts is disabled on this system`, poursuivre : l'étape 3 remplace ce terminal par un terminal qui n'affiche pas ce message.

À noter : la réponse de `python --version`.

2 · Ouvrir le dossier du TD, et installer l'extension Python

À faire : ouvrir le dossier `cours1\lc_vscode_python\` dans VS Code ; ouvrir `bonjour.py` ; installer l'extension `ms-python.python`.

À obtenir : l'extension Python installée ; en bas à droite de la fenêtre, le nom d'un interpréteur Python, ou une invitation à en choisir un.

Ouvrir le dossier

1. Menu File, Open Folder.
2. Choisir `C:\Users\eleve\Desktop\info01\cours1\lc_vscode_python`, puis « Sélectionner un dossier ».
3. À la première ouverture d'un dossier, VS Code demande si on fait confiance à ses auteurs. Répondre « Yes, I trust the authors ». Sans cette réponse, VS Code reste en « Restricted Mode », et l'extension Python ne se charge pas.

VS Code travaille sur un dossier : le dossier ouvert devient le projet, et l'arborescence de gauche montre son contenu. Un fichier ouvert seul, par File, Open File, ne donne à l'éditeur aucun dossier de projet ; le terminal, en particulier, ne s'ouvrirait pas dans le dossier du fichier.

Vérification : l'arborescence montre `altitudes.py`, `bonjour.py`, `README.md` et la feuille du TD, sous le titre `2A_VSCODE_PYTHON`.

Ouvrir un programme avant l'installation

Cliquer sur `bonjour.py` dans l'arborescence. Le fichier s'ouvre dans la zone du code :

```
"""Le plus petit programme Python qui fasse quelque chose de visible."""

nom = "géomatique"
print(f"Bonjour, {nom} !")
```

À noter : si le texte est coloré, alors qu'aucune extension Python n'est encore installée.

Installer l'extension

1. Ouvrir le panneau Extensions : `Ctrl + Maj + X`, ou l'icône des quatre carrés dans la barre de gauche.
2. Dans la zone de recherche en haut du panneau, taper l'identifiant de l'extension, `ms-python.python`.
3. L'extension « Python », publiée par Microsoft (le nom de l'éditeur est écrit sous celui de l'extension, avec une coche bleue), vient en tête de la liste. Cliquer dessus : sa page s'ouvre à droite.
4. Vérifier son identifiant : à droite de la page, en bas de la colonne « Marketplace Info », la ligne « Identifier » doit indiquer `ms-python.python`.
5. Cliquer sur Install, et attendre la fin de l'installation.

Plusieurs extensions s'appellent « Python », et certaines ne viennent pas de Microsoft. Seul l'identifiant désigne sans ambiguïté la bonne, d'où la recherche par identifiant plutôt que par nom.

L'installation télécharge l'extension depuis un serveur. Sans réseau, le panneau reste vide ou l'installation ne finit pas : vérifier que la session réseau est ouverte, puis recommencer. Une extension installée le reste pour les séances suivantes.

À noter : les extensions que la liste des extensions installées montre après l'installation, et ce qui apparaît en bas à droite de la fenêtre quand `bonjour.py` est affiché.

3 · Choisir l'interpréteur, et régler le terminal

À faire : repérer la palette de commandes et les réglages ; choisir l'interpréteur d'Anaconda par « Python: Select Interpreter » ; ouvrir un nouveau terminal, et le régler s'il n'est pas activé.

À obtenir : un terminal dont l'invite commence par (base).

La palette de commandes et les réglages

Les commandes et les réglages de VS Code se cherchent par leur nom, sans parcourir les menus. Quatre raccourcis ouvrent les panneaux qui servent dans le module :

Panneau	Raccourci	Ce qu'on y fait
la palette de commandes	Ctrl + Maj + P	taper le début du nom d'une commande, par exemple « Python: Select Interpreter »
les réglages	Ctrl + ,	chercher un réglage par un mot, par exemple « render whitespace »
les extensions	Ctrl + Maj + X	chercher une extension par son identifiant, ms-python.python
le terminal	Ctrl + `	ouvrir, masquer, rouvrir le terminal

Ouvrir les réglages (Ctrl + ,), et taper `render whitespace` dans la zone de recherche en haut. Le réglage trouvé s'affiche sous le titre « Editor: Render Whitespace » ; son nom complet, `editor.renderWhitespace`, est fait de plusieurs parties séparées par des points. En haut de la page, deux onglets, User et Workspace, correspondent aux deux niveaux de réglages : User pour le compte sur ce poste, Workspace pour le dossier ouvert. Ne rien changer, et fermer l'onglet des réglages.

Les mêmes réglages sont écrits dans un fichier texte, `settings.json`. Palette (Ctrl + Maj + P), taper « Preferences: Open User Settings (JSON) », puis Entrée : le fichier du compte s'ouvre dans la zone du code. Le regarder, puis le fermer, sans rien y écrire pour l'instant.

Choisir l'interpréteur

1. Garder `bonjour.py` ouvert dans la zone du code.
2. Palette (Ctrl + Maj + P), taper « Python: Select Interpreter », puis Entrée.
3. Une liste apparaît, avec pour chaque Python trouvé sur le poste son nom, son chemin et son type. Choisir la ligne base, de type Conda, dont le chemin contient `anaconda3` : sur les postes de la salle, `C:\ProgramData\anaconda3\python.exe`. Lire le chemin, et pas seulement le nom.

Le nom de l'interpréteur choisi s'affiche en bas à droite de la fenêtre, et VS Code le retient pour ce dossier.

Si la liste ne contient ni base ni aucun chemin avec `anaconda3`, choisir « Enter interpreter path... », puis « Find... », et désigner le fichier `C:\ProgramData\anaconda3\python.exe`.

Ouvrir un terminal activé

Menu Terminal, New Terminal. Un nouveau terminal s'ouvre en bas, dans le dossier du TD. L'extension Python y écrit et y exécute elle-même la commande qui active l'interpréteur choisi. Quand l'activation réussit, l'invite, la ligne qui attend une commande, commence par (base) :

```
(base) C:\Users\eleve\Desktop\info01\cours1\1c_vscode_python>
```

Si l'invite commence par (base), l'étape est finie. Si elle commence par PS, sans (base), ou si le message ... activate.ps1 cannot be loaded because running scripts is disabled on this system s'affiche, le terminal est un PowerShell, et le poste lui interdit d'exécuter le script d'activation. Ce refus se produit toujours quand VS Code a été lancé depuis le menu Démarrer ou depuis le bureau, et il peut se produire aussi quand VS Code a été

lancé depuis Navigator. La section suivante remplace PowerShell par l'invite de commandes cmd, sans demander de droits d'administrateur.

Régler le terminal

1. Palette, « Preferences: Open User Settings (JSON) ».
2. Si le fichier est vide, y recopier le bloc suivant en entier. S'il contient déjà des réglages, ajouter les lignes du milieu avant l'accolade fermante, en mettant une virgule à la fin de la ligne qui les précède.

```
{
  "python.condaPath": "C:\\ProgramData\\anaconda3\\Scripts\\conda.exe",
  "python.defaultInterpreterPath": "C:\\ProgramData\\anaconda3\\python.exe",
  "python-envs.defaultEnvManager": "ms-python.python:conda",
  "python-envs.defaultPackageManager": "ms-python.python:conda",
  "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"
}
```

3. Enregistrer (Ctrl + S).
4. Palette, « Developer: Reload Window ». La fenêtre se recharge.
5. Fermer l'ancien terminal (l'icône de corbeille en haut à droite du panneau du terminal), puis menu Terminal, New Terminal.

Les guillemets, les virgules et les doubles barres obliques inverses sont ceux du format JSON, et se recopient tels quels. VS Code souligne en rouge la ligne qui contient une erreur de copie. Les dernières lignes décrivent à VS Code le terminal de l'invite de commandes d'Anaconda, un cmd déjà activé à l'ouverture, et en font le terminal par défaut ; les quatre premières indiquent à l'extension Python où est conda, et quel interpréteur employer tant qu'on n'en a pas choisi.

Vérification : l'onglet du terminal s'appelle « Anaconda Prompt », et l'invite commence par (base).

Les chemins C:\\ProgramData\\anaconda3 sont ceux de l'installation de la salle. Sur un autre ordinateur, ils se lisent dans le raccourci de l'invite de commandes d'Anaconda : menu Démarrer, clic droit sur « Anaconda Prompt », Ouvrir l'emplacement du fichier, puis clic droit sur le raccourci, Propriétés, champ Cible. Sous macOS et Linux, aucun de ces réglages n'est nécessaire : le terminal de VS Code y est un terminal ordinaire, que l'extension active sans difficulté.

À noter : le nom écrit entre parenthèses au début de l'invite.

4 · Lancer le programme en entier, puis ligne à ligne

À faire : lancer `altitudes.py` dans le terminal ; refaire son calcul dans une session interactive de Python.

À obtenir : la moyenne affichée par le programme, puis la même valeur obtenue dans la session interactive.

Le programme

Ouvrir `altitudes.py` dans la zone du code. Le fichier compte huit lignes : une chaîne de documentation, une ligne vide, puis six lignes de code.

```
"""Moyenne d'une série d'altitudes, à la main plutôt qu'avec sum()."""

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")
```

Le programme range trois altitudes dans une liste, les additionne une à une dans `total`, divise la somme par le nombre d'altitudes, puis affiche le résultat arrondi à une décimale.

Lancer le programme en entier

1. Vérifier que l'invite du terminal commence par `(base)` et se termine par `1c_vscode_python>`. Sinon, reprendre l'étape 3.
2. Taper la commande suivante, puis Entrée :

```
python altitudes.py
```

La sortie est :

```
moyenne : 129.0 m
```

Deux autres commandes de VS Code lancent le même programme : le bouton d'exécution, un triangle en haut à droite de la zone du code, et le menu Run, Run Without Debugging (`Ctrl + F5`). Les essayer après la commande tapée à la main, et regarder ce qu'ils écrivent dans le terminal avant d'exécuter le programme.

À noter : le nombre de lignes affichées par le programme ; ce qui change dans l'arborescence après son exécution.

Refaire le calcul ligne à ligne

Dans le même terminal, taper `python`, sans nom de fichier, puis Entrée. Python affiche sa version sur quelques lignes, puis son invite, trois chevrons `>>>`. Taper ensuite les lignes suivantes, une par une, chacune suivie d'Entrée :

```
>>> 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()
```

Les chevrons `>>>` et les points `...` sont écrits par Python : ne pas les taper. Après la ligne `for`, Python affiche `...`, pour signaler que le bloc n'est pas fini : taper quatre espaces, puis `total = total + altitude`, Entrée, puis Entrée :

sur la ligne vide suivante pour terminer le bloc. Si Python a déjà placé le curseur en retrait après les `...`, ne pas ajouter d'espaces. `exit()` ferme la session, et l'invite du terminal revient.

Pour aller plus loin, refaire la boucle en affichant `total` à chaque tour :

```
>>> total = 0
>>> for altitude in altitudes:
...     total = total + altitude
...     print(total)
... 
```

À noter : comment `total` s'affiche, alors qu'aucun `print` ne le demande ; la différence entre l'invite `>>>` et celle du terminal.

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

Cette partie est facultative, pour qui a fini en avance. Elle lance le même programme depuis l'invite de commandes d'Anaconda, sans passer par l'éditeur.

1. Menu Démarrer, taper `anaconda` prompt, puis Entrée. Une fenêtre noire s'ouvre, et l'invite commence par `(base)`.
2. Taper `cd` suivi d'une espace, puis glisser le dossier `1c_vscode_python` depuis l'explorateur de fichiers dans la fenêtre : son chemin complet s'écrit à la suite. Appuyer sur Entrée.
3. Taper `python altitudes.py`, puis Entrée.
4. Taper `python`, puis les lignes de la session interactive, et `exit()`.

Si l'invite ne change pas à l'étape 2, le dossier est sur un autre disque que `C:` : refaire l'étape avec `cd /d` à la place de `cd`.

À noter : ce que l'invite affiche après l'étape 2 ; les sorties des étapes 3 et 4, comparées à celles obtenues dans VS Code.

5 · Exécuter le programme pas à pas

À faire : poser un point d'arrêt sur la ligne 6 d'`altitudes.py` ; lancer le débogueur ; avancer ligne par ligne en lisant la valeur de `total`.

À obtenir : le programme arrêté sur la ligne 6, puis mené jusqu'à la fin, avec la moyenne affichée dans le terminal.

Le débogueur exécute le programme et l'arrête sur une ligne choisie, pour qu'on puisse lire la valeur des variables sans modifier le code. Le repère posé sur la ligne où le programme doit s'arrêter est un **point d'arrêt**.

Poser un point d'arrêt, et lancer

1. Ouvrir `altitudes.py`. Passer la souris dans la marge, à gauche du numéro de la ligne 6, `total = total + altitude` : un point rouge pâle apparaît. Cliquer : le point devient rouge vif, et marque le point d'arrêt.
2. Appuyer sur F5.
3. Au premier lancement, VS Code demande en haut de la fenêtre quel débogueur employer : choisir « Python Debugger ». Il demande ensuite quelle configuration : choisir « Python File ». Si VS Code propose de créer un fichier `launch.json`, ne pas le créer.
4. Le programme démarre, puis s'arrête : une ligne de la zone du code est surlignée en jaune.
5. À gauche, le panneau Run and Debug s'ouvre de lui-même. S'il a été fermé, Ctrl + Maj + D le rouvre. Sa section Variables montre les variables du programme et leur valeur.

En haut de la fenêtre, une barre de boutons est apparue. Chaque bouton a son raccourci :

Bouton	Touche	Ce qu'il fait
Continue	F5	reprend l'exécution jusqu'au prochain point d'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 ; il ne sert pas dans ce TD
Restart	Ctrl + Maj + F5	recommence le programme depuis le début
Stop	Maj + F5	arrête le programme

À noter : la ligne surlignée, et les variables affichées avec leur valeur, dont `total`.

Avancer, et lire les variables

La ligne surlignée est la prochaine ligne à exécuter : elle n'a pas encore été exécutée.

1. Avant chaque appui sur F10, prédire la valeur que `total` aura ensuite ; puis appuyer sur F10, et lire `total` dans Variables.
2. Continuer avec F10 : la ligne surlignée passe de la ligne 6 à la ligne 5, puis revient à la ligne 6, tant que la boucle n'est pas finie. S'arrêter quand la ligne 7 est surlignée.
3. Appuyer sur F5 : le programme s'exécute jusqu'à la fin.
4. Cliquer sur le point rouge de la marge pour l'enlever.

À noter : les valeurs successives de `total` ; ce que le terminal affiche après F5 ; ce qui s'est passé dans le fichier `altitudes.py` depuis la pose du point d'arrêt.

Ce que le TD fait constater

Cette section se lit après avoir fait les étapes.

Étape 1 : VS Code lancé depuis Navigator

La liste déroulante du haut de Navigator ne contient qu'une entrée, `base (root)`, déjà sélectionnée. `base` est le nom du Python livré avec Anaconda, et le seul installé tant qu'on n'en a pas créé d'autre ; ce qu'est un environnement, et comment on en fabrique un second, est l'objet de la partie 4 du cours et du TD 1h. VS Code lancé depuis Navigator emploie l'interpréteur de l'environnement sélectionné en haut de la page.

`python --version` répond par un numéro de version, Python 3. suivi de deux nombres, sans message d'erreur. Le numéro exact dépend de la version d'Anaconda installée sur le poste.

VS Code enregistre du texte brut, comme le Bloc-notes et Notepad++ du TD 1a. Les couleurs et les numéros de ligne qu'il ajoute à l'écran ne sont pas enregistrés dans le fichier.

Étape 2 : ce qu'apporte l'extension

`bonjour.py` est coloré dès son ouverture, avant toute installation : la coloration syntaxique des langages courants est fournie par VS Code lui-même. L'extension Python ajoute les autres fonctions : le choix de l'interpréteur, l'exécution du programme, le débogueur, et la vérification des règles d'écriture du langage par Pylance. Cette vérification porte sur l'écriture du programme et ne dit rien de ce qu'il calcule : un programme peut la satisfaire entièrement et calculer autre chose que ce qu'on voulait.

D'autres extensions de Microsoft s'installent avec l'extension Python, dont Pylance et Python Debugger ; il n'y a qu'une extension à chercher. En bas à droite de la fenêtre, la barre d'état nomme un interpréteur Python, ou propose d'en choisir un.

La même démarche installe la prise en charge des autres langages : panneau Extensions, identifiant de l'extension, Install. Le TD 1e l'applique au C++.

Étape 3 : la palette, les réglages, l'interpréteur

La palette de commandes et la recherche dans les réglages permettent de trouver une commande ou un réglage par son nom, sans savoir dans quel menu il se trouve. Les consignes du module désignent les commandes et les réglages par ce nom.

Chaque réglage porte un nom en plusieurs parties séparées par des points, comme `editor.renderWhitespace` ou `terminal.integrated.defaultProfile.windows`, et sa valeur est rangée dans un fichier texte, `settings.json`. Ce fichier existe à deux niveaux :

Niveau	Portée	Fichier
User	le compte, pour tous les dossiers ouverts sur ce poste	<code>C:\Users\<nom>\AppData\Roaming\Code\User\</code>
Workspace	le dossier ouvert seulement	<code>.vscode\settings.json</code> , dans ce dossier

Quand un réglage est présent aux deux niveaux, celui du dossier l'emporte. Un réglage écrit dans un fichier texte peut être transmis par écrit, comme dans ce guide, et enregistré avec un projet, au niveau Workspace.

Plusieurs Python peuvent être installés sur une même machine, et « Python: Select Interpreter » désigne celui que VS Code emploie. Les terminaux ouverts ensuite dans VS Code sont activés pour ce même Python, et leur invite commence par `(base)`. Le nom écrit entre parenthèses au début de l'invite désigne le Python qui répondra à la commande `python` : c'est la seule marque visible de l'interpréteur actif, et il faut la relire avant d'installer un paquet. Un terminal qui emploie un autre Python que celui choisi dans l'éditeur produit, plus tard dans le module, des erreurs `ModuleNotFoundError` sur des paquets pourtant installés.

Le terminal PowerShell, celui que VS Code ouvre par défaut sous Windows, refuse d'exécuter le script d'activation d'Anaconda : sa stratégie d'exécution est `Restricted` par défaut, et les postes de la salle ne permettent pas de la

changer. Le cmd n'est pas soumis à cette stratégie ; c'est lui que lance le raccourci « Anaconda Prompt » du menu Démarrer, et c'est lui que le réglage du terminal donne à VS Code.

Étape 4 : en entier, et ligne à ligne

Lancé en entier, le programme n'affiche qu'une ligne, moyenne : 129.0 m, celle du print final. Les valeurs intermédiaires de total ne sont pas visibles. La moyenne de 128,4, 131,0 et 127,6 vaut 129,0.

Rien n'apparaît dans l'arborescence : exécuter un programme Python ne crée aucun fichier sur le disque. Le TD 1e, facultatif, montre qu'un programme C++ en crée un, l'exécutable produit par le compilateur.

Le bouton d'exécution et le menu Run, Run Without Debugging écrivent leur commande dans le terminal avant de l'exécuter : c'est le chemin de l'interpréteur choisi, suivi du chemin du fichier. Ils emploient l'interpréteur sélectionné dans VS Code, et seule la lecture de cette commande indique lequel. La commande tapée à la main, `python altitudes.py`, est la même sous Windows, macOS et Linux, et elle est courte à relire.

Dans la session interactive, `total` et `total / len(altitudes)` s'affichent sans print : la session affiche la valeur de chaque expression tapée seule sur une ligne. Cet affichage permet de lire les valeurs intermédiaires du calcul. La boucle avec `print(total)` affiche 128.4, puis 259.4, puis 387.0.

Les trois chevrons `>>>` sont l'invite de Python, et non celle du terminal. Une commande du système, comme `cd` ou `python altitudes.py`, tapée derrière `>>>` produit une erreur `SyntaxError` ; il faut d'abord sortir de la session par `exit()`. Les trois points `...` signalent la suite d'un bloc commencé.

Un programme enregistré se relance à l'identique, et n'affiche que ce que ses print demandent ; une session interactive affiche chaque résultat, mais rien de ce qui y a été tapé n'est conservé sur le disque à sa fermeture. Le notebook, vu dans la dernière partie du cours, est une session de ce type, dont les cellules et le texte qui les entoure sont conservés dans un fichier.

Dans l'invite de commandes d'Anaconda, glisser le dossier dans la fenêtre colle son chemin complet, et `cd` fait de ce dossier le dossier courant : l'invite se termine ensuite par `1c_vscode_python>`. Les sorties sont les mêmes que dans VS Code, moyenne : 129.0 m et 129.0. Le terminal de VS Code et l'Anaconda Prompt lancent le même python ; VS Code ouvrait en plus le terminal directement dans le dossier du projet, ce que `cd` fait ici à la main.

Étape 5 : pas à pas

Au lancement, le programme s'arrête sur la ligne 6, surlignée en jaune, avant de l'avoir exécutée. Variables montre `altitudes`, la liste des trois valeurs, `total`, qui vaut 0, et `altitude`, qui vaut 128.4, la première valeur de la boucle.

Ligne surlignée après F10	altitude	total
5	128.4	128.4
6	131.0	128.4
5	131.0	259.4
6	127.6	259.4
5	127.6	387.0
7	127.6	387.0

`total` change chaque fois que la ligne 6 vient d'être exécutée : 128.4, puis 259.4, puis 387.0. La ligne 5, `for altitude in altitudes:`, est surlignée à chaque tour, parce que son exécution donne à `altitude` la valeur suivante de la liste, ou termine la boucle. Après F5, le programme se termine, et le terminal affiche moyenne : 129.0 m.

Le point d'arrêt est conservé par l'éditeur : le fichier `altitudes.py` n'a pas changé. Tant qu'il n'est pas enlevé, un nouveau F5 relance le programme et l'arrête au même endroit.

Les deux premières façons d'exécuter montrent le début et la fin du calcul ; le pas à pas en montre le milieu, une ligne après l'autre. Pour savoir ce que fait un programme, poser un point d'arrêt évite d'ajouter dans le code des print qu'il faudrait ensuite retirer. Le débogage est repris au cours 2.