

**SUP MTI**  
École Supérieure de Management,  
de Télécommunication et d'Informatique

**RECONNUE PAR L'ÉTAT**

# **PYTHON**

## **SÉANCE NUM 1**



Enseignant : GHANEM

# Plan de la Séance 1 : Introduction et Premiers Pas en Python

## Sommaire :

- Qu'est-ce que la programmation ?
- Langage interprété vs langage compilé
- Découverte de Google Colab
- Installation de Python
- Python Console
- Python IDLE
- Premiers programmes avec print, variables et input
- Les commentaires en Python

# Qu'est-ce que la programmation ?

- La programmation consiste à donner des instructions à un ordinateur
- Un programme est une suite d'instructions logiques
- Le code source est écrit dans un langage de programmation
- L'ordinateur exécute ces instructions séquentiellement
- Analogie : une recette de cuisine pour un ordinateur

## Qu'est-ce qu'un langage interprété ?

- Un langage interprété est exécuté ligne par ligne par un interpréteur
- Le code source est traduit en instructions machine à la volée
- Aucune étape de compilation séparée n'est nécessaire
- L'interpréteur lit, analyse et exécute directement le code
- Exemples : Python, JavaScript, Ruby, PHP

## Avantages :

- Portabilité (même code sur différentes plateformes)
- Développement plus rapide (test immédiat)
- Debuggage facilité

## Inconvénients :

- Performance généralement inférieure aux langages compilés
- Nécessite l'installation de l'interpréteur

# Langages interprétés vs compilés

## Qu'est-ce qu'un langage compilé ?

- Un langage compilé nécessite une transformation du code source en code machine
- La compilation crée un fichier exécutable binaire autonome
- Cette étape est effectuée avant l'exécution du programme
- Le compilateur analyse tout le code en une seule fois
- Exemples : C, C++, Rust, Go, Swift

## Avantages :

- Performance optimisée (exécution directe par le processeur)
- Pas besoin d'environnement d'exécution supplémentaire
- Détection des erreurs avant l'exécution

## Inconvénients :

- Compilation nécessaire après chaque modification
- Code spécifique à chaque plateforme (OS, architecture)
- Débogage plus complexe

# Langages interprétés vs compilés

Tableau comparatif :

Langages Compilés	Langages Interprétés	Mixte (Compilé + Interprété)
C, C++	Python, JavaScript	Java, C#
Rust, Go	Ruby, PHP	Scala, Kotlin

# Qu'est-ce qu'un bytecode ?

- Code intermédiaire entre le code source et le code machine
- Représentation binaire compacte du programme
- Non exécutable directement par le processeur
- Nécessite une machine virtuelle pour être interprété
- Plus abstrait que le code machine, plus concret que le code source

## Caractéristiques :

- Portable entre différentes plateformes
- Plus rapide à interpréter que le code source
- Généré lors de la première exécution du programme

## Exemples :

- Python → .pyc files (Python bytecode)
- Java → .class files (Java bytecode)
- .NET → CIL (Common Intermediate Language)

# Python - Langage interprété

- Python est principalement interprété
- Compilation en bytecode transparente
- Avantage : portabilité et simplicité
- Inconvénient : performance moindre que les langages compilés
- Machine Virtuelle Python (PVM) exécute le bytecode



# Google Colab – Introduction

- Environnement de développement en ligne
- Nécessite seulement un navigateur web
- Idéal pour débuter sans installation
- Basé sur Jupyter Notebooks

(cherchez sur google le mot : Colab)



colab



Mode IA

Tous

Images

Vidéos

Actualités

Vidéos courtes

Maps

Plus ▾

Outils ▾



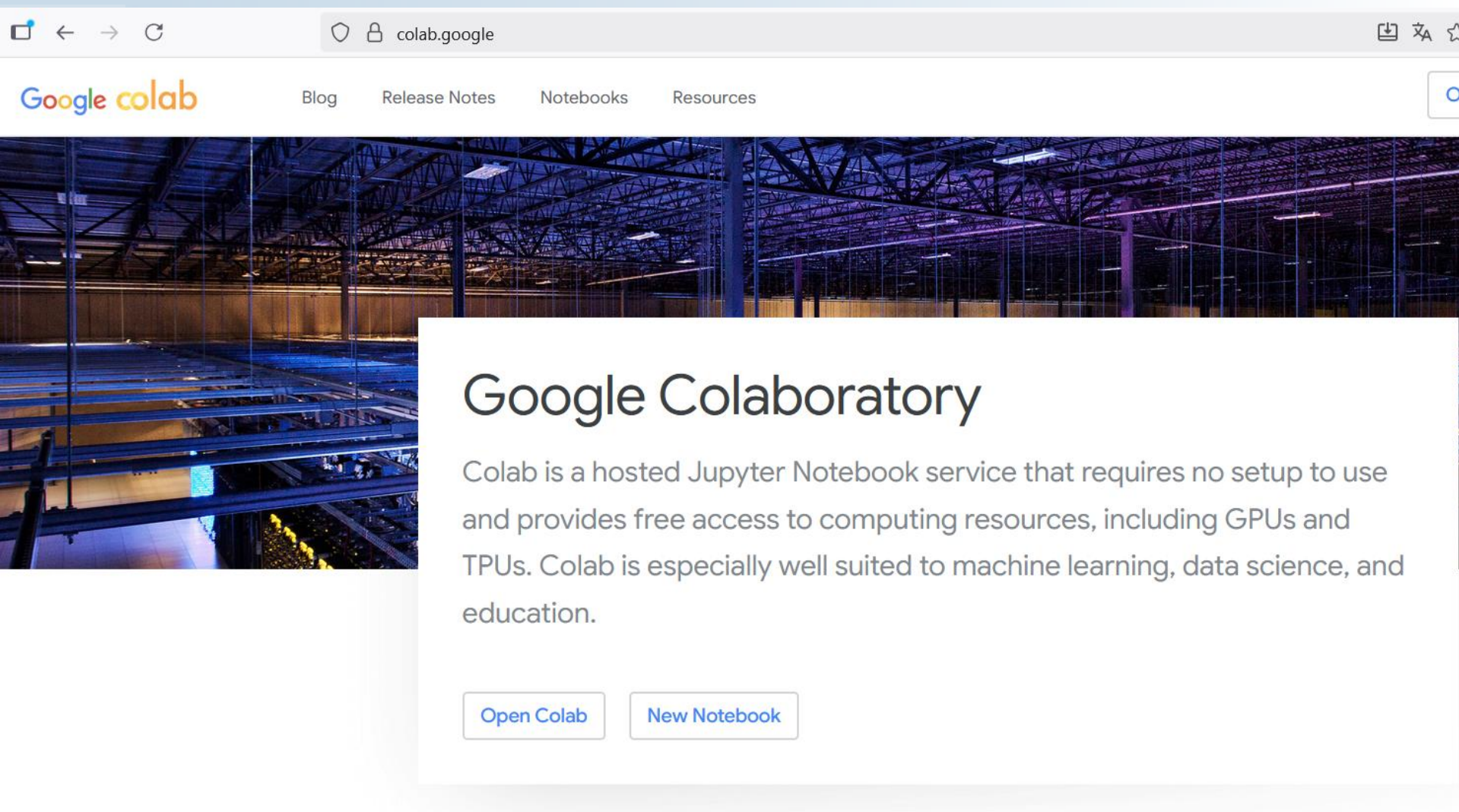
colab.google

<https://colab.google> · [Traduire cette page](#) ⋮

## Google Colab

Colab is a hosted Jupyter Notebook service that requires no setup to use and provides free access to computing resources, including GPUs and TPUs.

# Ouvrez le premier site :



The screenshot shows the Google Colaboratory website in a web browser. The browser's address bar displays 'colab.google'. The website's header includes the 'Google colab' logo and navigation links for 'Blog', 'Release Notes', 'Notebooks', and 'Resources'. A large background image of a server room is visible. On the right side, a white box contains the title 'Google Colaboratory' and a descriptive paragraph. At the bottom of this box are two buttons: 'Open Colab' and 'New Notebook'.

Google colab

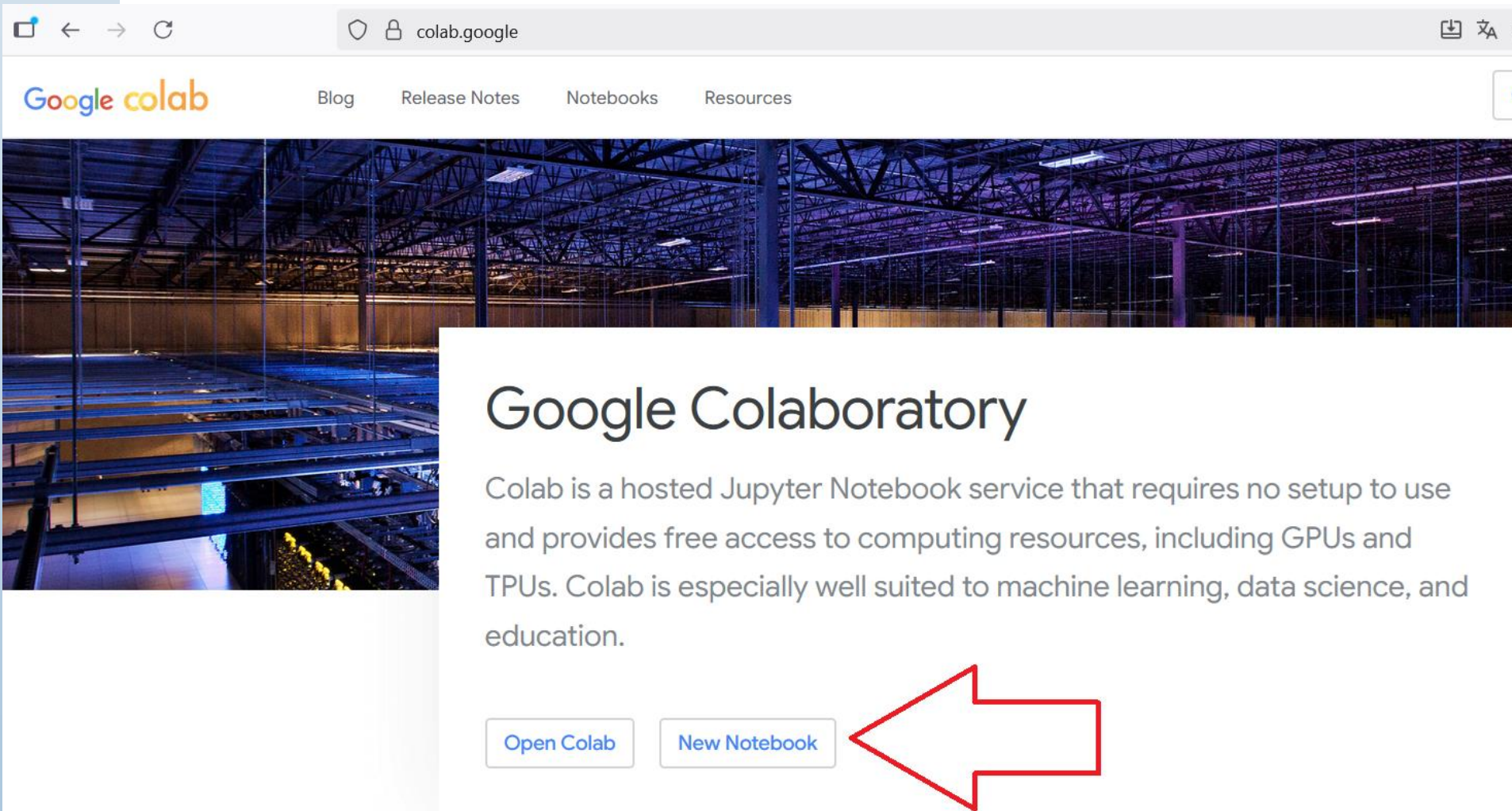
Blog Release Notes Notebooks Resources

## Google Colaboratory

Colab is a hosted Jupyter Notebook service that requires no setup to use and provides free access to computing resources, including GPUs and TPUs. Colab is especially well suited to machine learning, data science, and education.

[Open Colab](#) [New Notebook](#)

# Cliquez sur New Notebook



The screenshot shows the Google Colaboratory website interface. At the top, there is a navigation bar with the Google Colab logo and links for Blog, Release Notes, Notebooks, and Resources. Below the navigation bar is a large banner image of a server room. On the right side of the banner, the text 'Google Colaboratory' is displayed, followed by a description: 'Colab is a hosted Jupyter Notebook service that requires no setup to use and provides free access to computing resources, including GPUs and TPUs. Colab is especially well suited to machine learning, data science, and education.' At the bottom of the banner, there are two buttons: 'Open Colab' and 'New Notebook'. A large red arrow points from the right towards the 'New Notebook' button.

Google colab

Blog Release Notes Notebooks Resources

## Google Colaboratory

Colab is a hosted Jupyter Notebook service that requires no setup to use and provides free access to computing resources, including GPUs and TPUs. Colab is especially well suited to machine learning, data science, and education.

Open Colab New Notebook

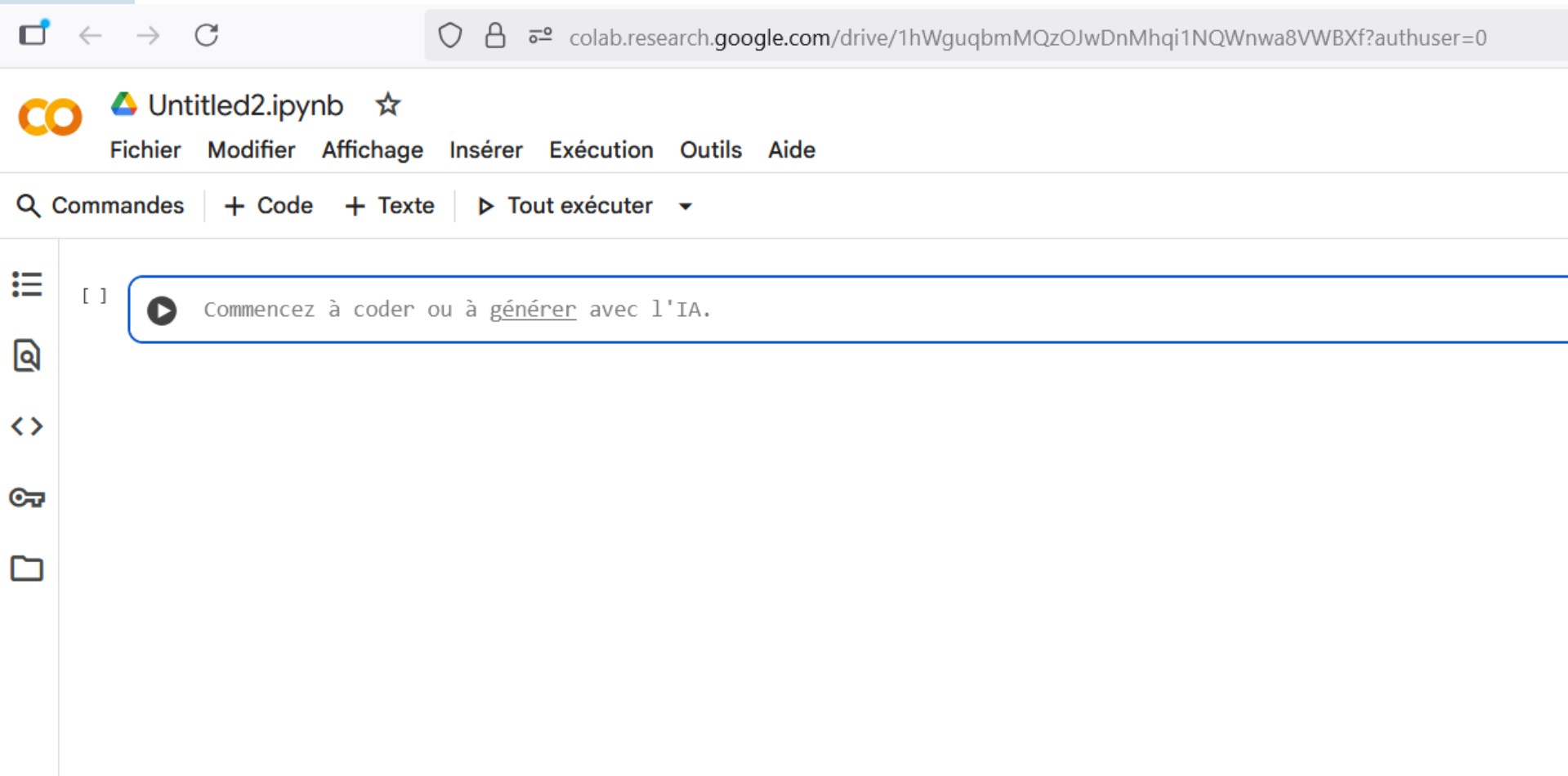
# Premier programme sur Colab

## Connexion à Google obligatoire

Vous devez être connecté à un compte Google pour continuer.

Connexion

# Premier programme sur Colab



The screenshot displays the Google Colab web interface. At the top, the browser address bar shows the URL: `colab.research.google.com/drive/1hWguqbmMQzOJwDnMhqi1NQWnwa8VWBXf?authuser=0`. Below the browser, the Colab logo is followed by the notebook title 'Untitled2.ipynb' and a star icon. A menu bar contains the following items: 'Fichier', 'Modifier', 'Affichage', 'Insérer', 'Exécution', 'Outils', and 'Aide'. Below the menu bar, there is a search bar and a row of buttons: 'Commandes', '+ Code', '+ Texte', and 'Tout exécuter' with a dropdown arrow. On the left side, a vertical sidebar contains icons for a menu, a document, a code editor, a key, and a folder. The main workspace area shows a single code cell with a play button icon and the text: 'Commencez à coder ou à générer avec l'IA.'

# Premier programme sur Colab



Untitled2.ipynb ☆ ☁

Fichier Modifier Affichage Insérer Exécution Outils Aide

🔍 Commandes | + Code | + Texte | ▶ Tout exécuter

2



[ ]



```
print("bonjour l'école")
```

1



# Premier programme sur Colab

The screenshot displays the Google Colab web interface. At the top, the browser address bar shows the URL `colab.research.google.com/drive/1hWguqbmMQzOJwDnMh`. Below the browser, the Colab header includes the logo, the file name `Untitled2.ipynb`, a star icon, and a link icon with the text `Enregistrement...`. A menu bar contains the options: `Fichier`, `Modifier`, `Affichage`, `Insérer`, `Exécution`, `Outils`, and `Aide`. Below the menu, a toolbar features a search icon, the text `Commandes`, and buttons for `+ Code`, `+ Texte`, and `Tout exécuter` (with a dropdown arrow). A dark button labeled `Exécuter toutes les cellules du notebook` is positioned above the code cell. The code cell itself is labeled `[1]` and shows a green checkmark and `0 s` indicating successful execution. The code `print("bonjour l'école")` is displayed in a monospace font. Below the code, the output `bonjour l'école` is shown. On the left side of the interface, a vertical sidebar contains icons for a menu, search, navigation, and file management.



# Premier programme sur Colab

- Vous pouvez exécuter avec Ctrl+Enter

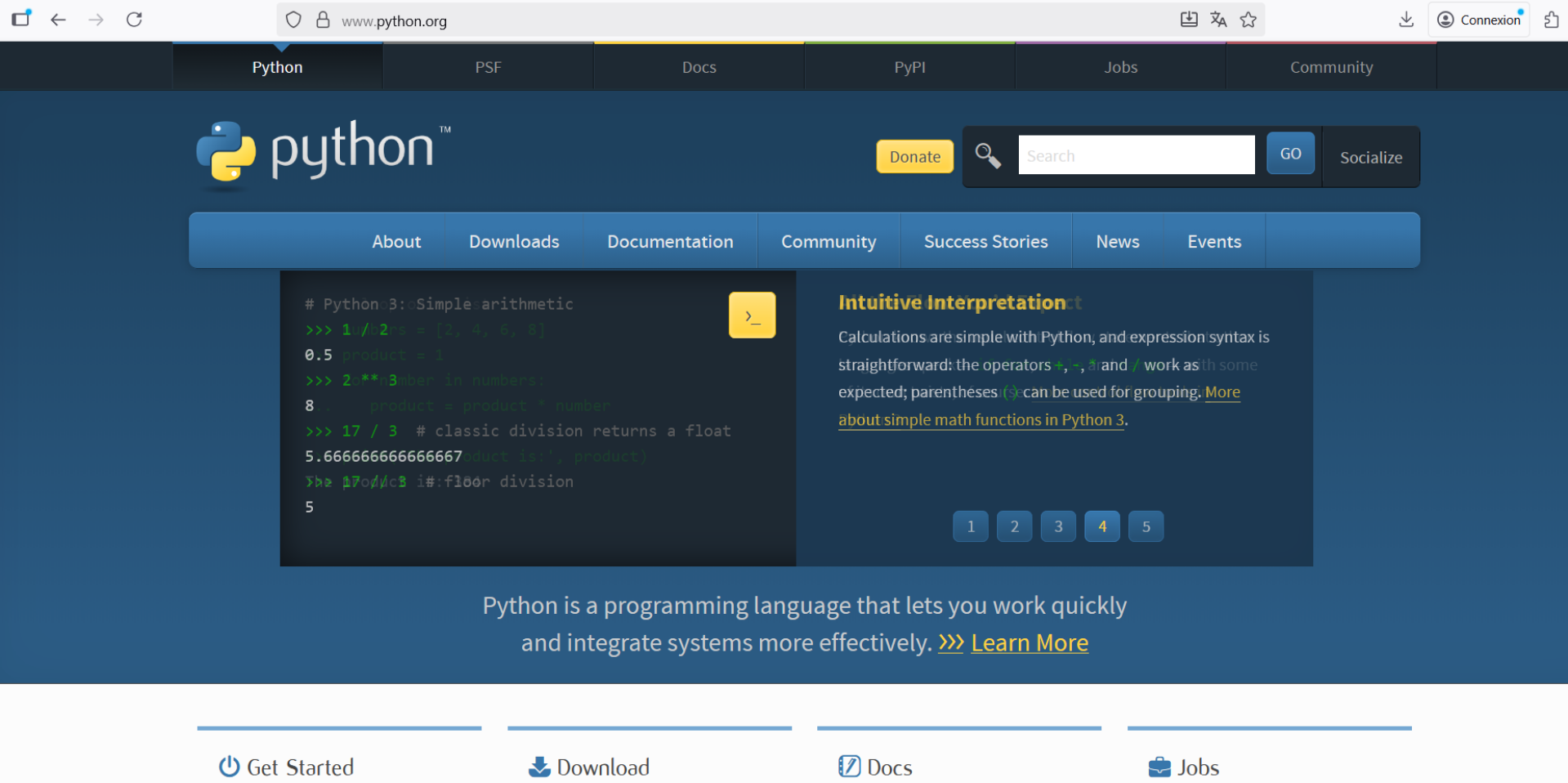
# Python – Présentation

- Créé par Guido van Rossum en 1991
- Géré par la Python Software Foundation
- Organisme à but non lucratif
- Financement : dons, sponsors, conférences
- Philosophie : simplicité et lisibilité

- L'organisme qui maintient le langage de programmation Python est la Python Software Foundation (PSF). Il s'agit d'une organisation à but non lucratif qui a pour mission de faire progresser le langage de programmation Python.
- Fondation : La Python Software Foundation (PSF) a été créée en 2001.
- Rôle : Elle est responsable du développement et de la maintenance du langage, de ses bibliothèques standard et de l'écosystème qui l'entoure.
- Financement : Le financement de la PSF provient de dons, de subventions et de partenariats avec des entreprises du monde entier.

- Site officiel : [python.org](https://python.org)
- Documentation complète
- Téléchargement de l'interpréteur
- Communauté et ressources

# (Capture d'écran : page d'accueil [python.org](https://python.org))



The screenshot shows the Python.org homepage. At the top, there's a navigation bar with links: Python, PSF, Docs, PyPI, Jobs, and Community. Below this is a dark blue header with the Python logo, a 'Donate' button, a search bar with a 'GO' button, and a 'Socialize' button. A secondary navigation bar contains links: About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area features a code snippet on the left demonstrating simple arithmetic in Python, and a text block on the right titled 'Intuitive Interpretation' explaining that calculations are straightforward. Below the text is a pagination bar with numbers 1 through 5, where '4' is highlighted. At the bottom, a large blue banner states: 'Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)'. The footer contains four icons with labels: 'Get Started', 'Download', 'Docs', and 'Jobs'.

Python

PSF

Docs

PyPI

Jobs

Community

python™

Donate

Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

```
# Python 3: Simple arithmetic
>>> 1 + 2 * 3 = (2, 4, 6, 8)
0.5 product = 1
>>> 2 ** 3
8
product = product * number
>>> 17 / 3 # classic division returns a float
5.666666666666667
>>> 17 // 3 # floor division
5
```

**Intuitive Interpretation**

Calculations are simple with Python, and expression syntax is straightforward: the operators `+`, `*`, `^` and `/` work as th some expected; parentheses `()` can be used for grouping. [More about simple math functions in Python 3.](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

Get Started

Download

Docs

Jobs

Python

PSF

Docs

PyPI

Jobs



Donate



Search

About

Downloads

Documentation

Community

Success Stories

News

Events

```
# Simple output
>>> print("Hello, I'm Python")
Hello, I'm Python

# Input, assign a variable
>>> name = input("What is your name? ")
What is your name? Python

>>> print(f'Hi, {name}!')
Hi, Python.
```

All releases

Source code

Windows

macOS

Android

Other Platforms

License

Alternative Implementations

## Download for Windows

Python 3.14.0

**Note that Python 3.9+ *cannot* be used on Windows 7 or earlier.**

Not the OS you are looking for? Python can be used on many operating systems and environments.

[View the full list of downloads.](#)

and integrate systems more effectively. >>> [Learn More](#)

www.python.org

Python PSF Docs

python™

About Downloads Documentation Community Success Stories News Events

```
# Simple output (with Unicode)
>>> print("Hello, I'm Python!")
Hello, I'm Python!
# Input, assignment
>>> name = input('What is your name?\n')
What is your name?
Python
>>> print(f'Hi, {name}.')
Hi, Python.
```

**Quick & Easy to Learn**

Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. [Whet your appetite](#) with our Python 3 overview.

1 2 3 4 5

python-3.14.0-amd64(1).exe  
Temps restant : 2 min 48 s — 3,6 sur 28,5 Mo (115 Ko/s)  
Afficher tous les téléchargements

PSF

Docs

F



python-3.14.0-amd64(1).exe

Ouvrir le fichier



Afficher tous les téléchargements

Donate

Search

GO

Socialize

out

Downloads

Documentation

Community

Success Stories

News

Events

List comprehensions

```
fruits = ['Banana', 'Apple', 'Lime']  
fruits = [fruit.upper() for fruit in
```

```
fruits]
```

```
print(fruits)
```

the enumerate function

```
for index, fruit in enumerate(fruits):  
    print(index, fruit)
```

>\_

## Compound Data Types

Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions. [More about lists in Python 3](#)

1

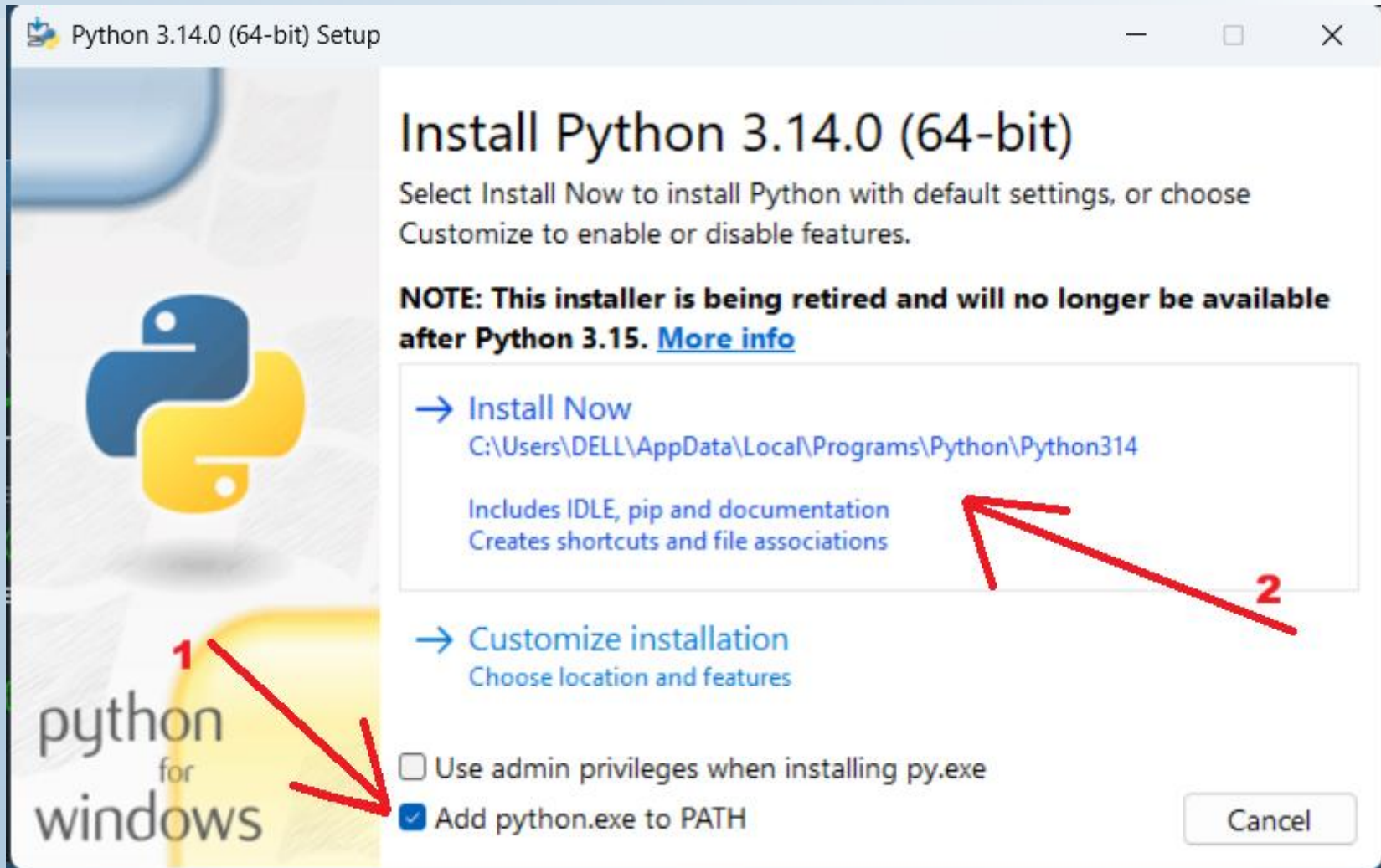
2

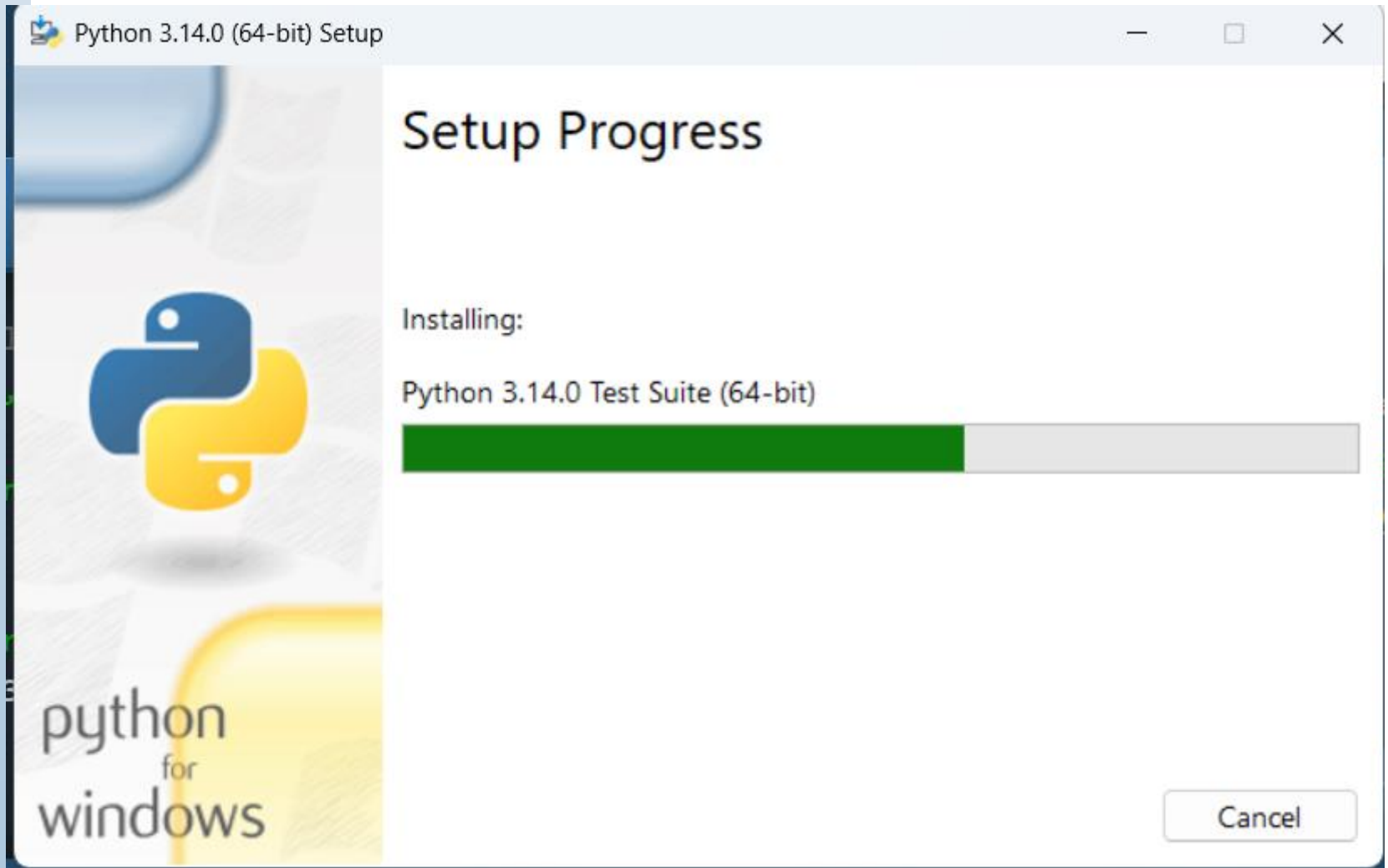
3

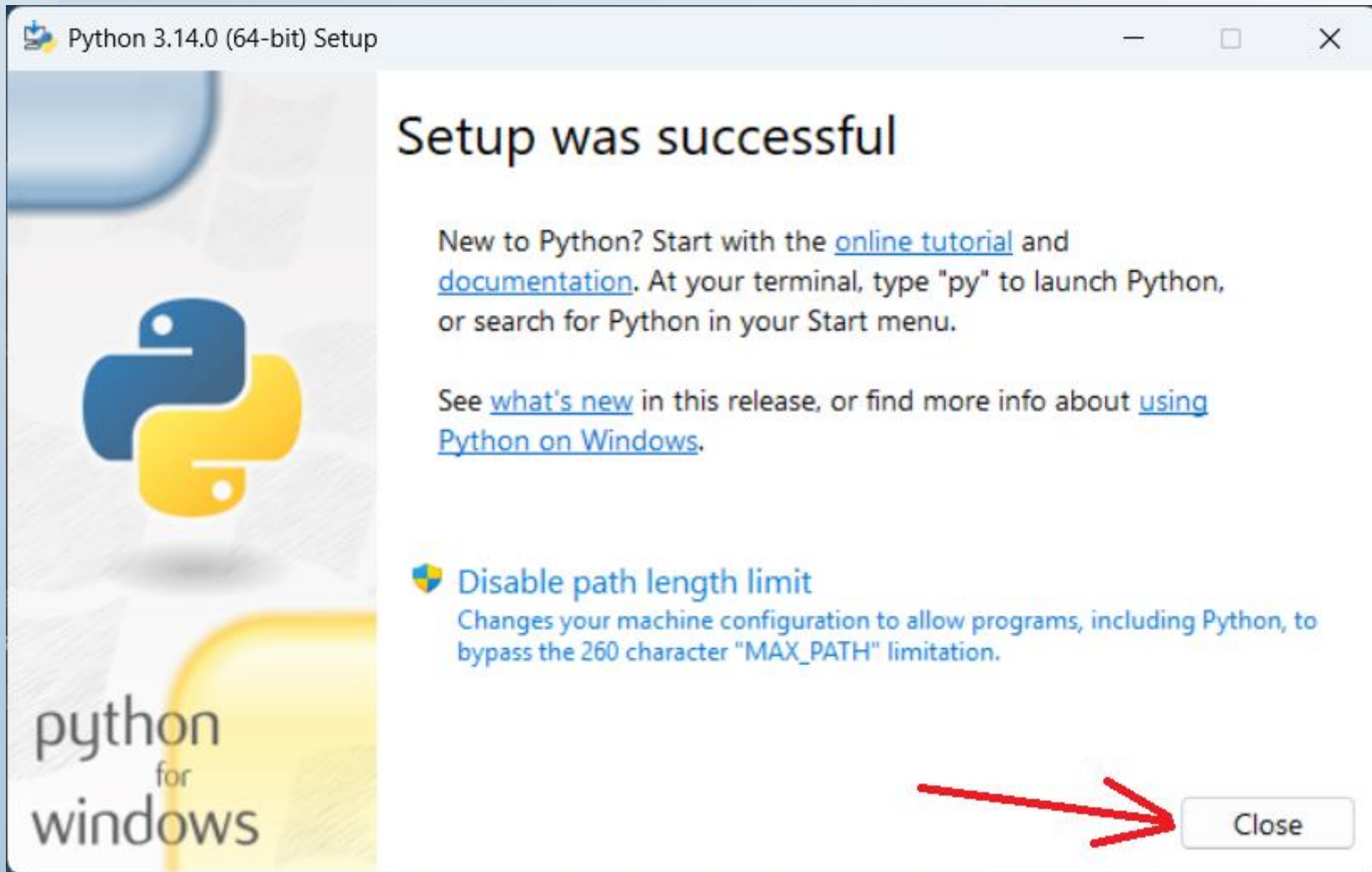
4

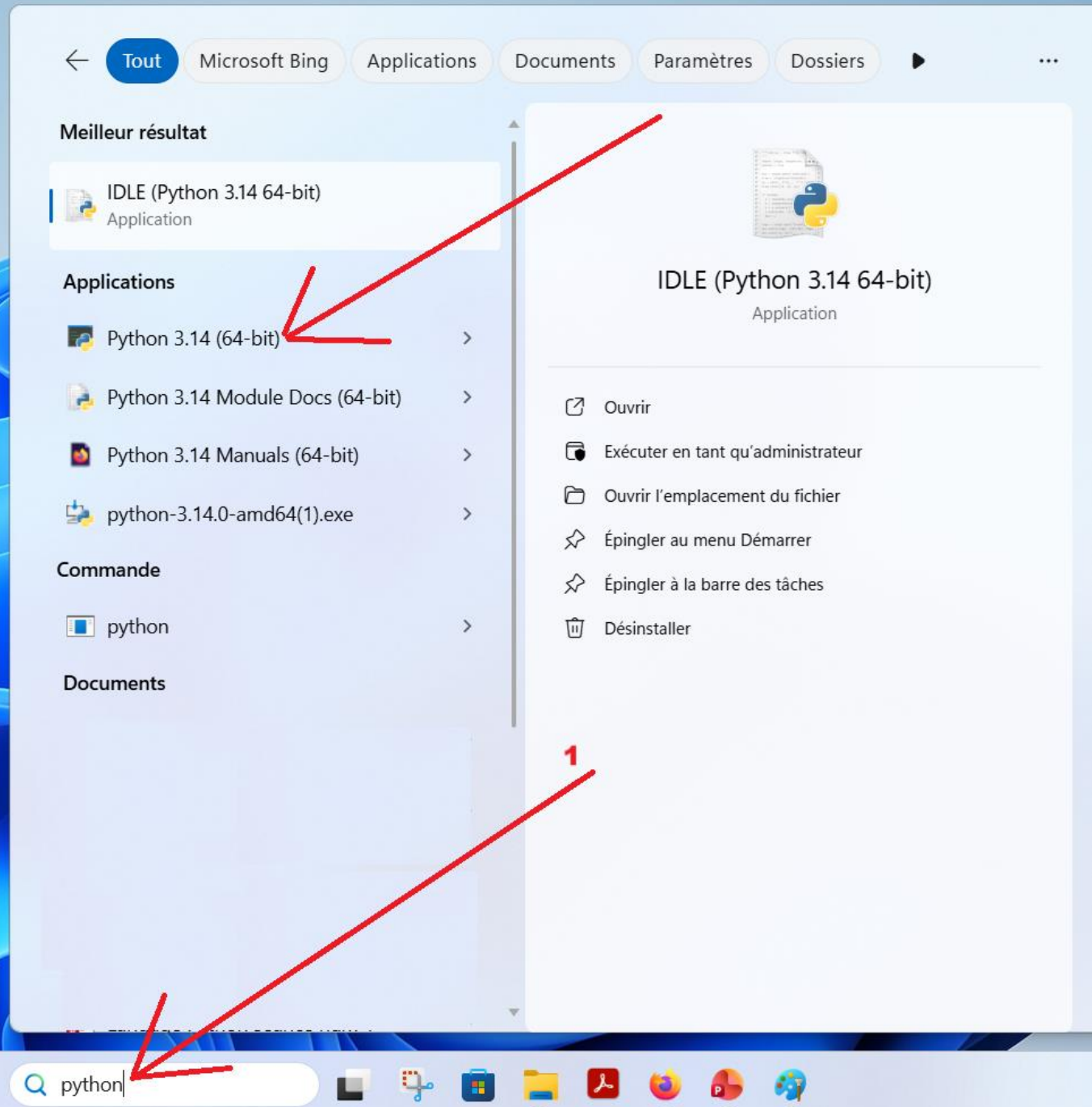
5



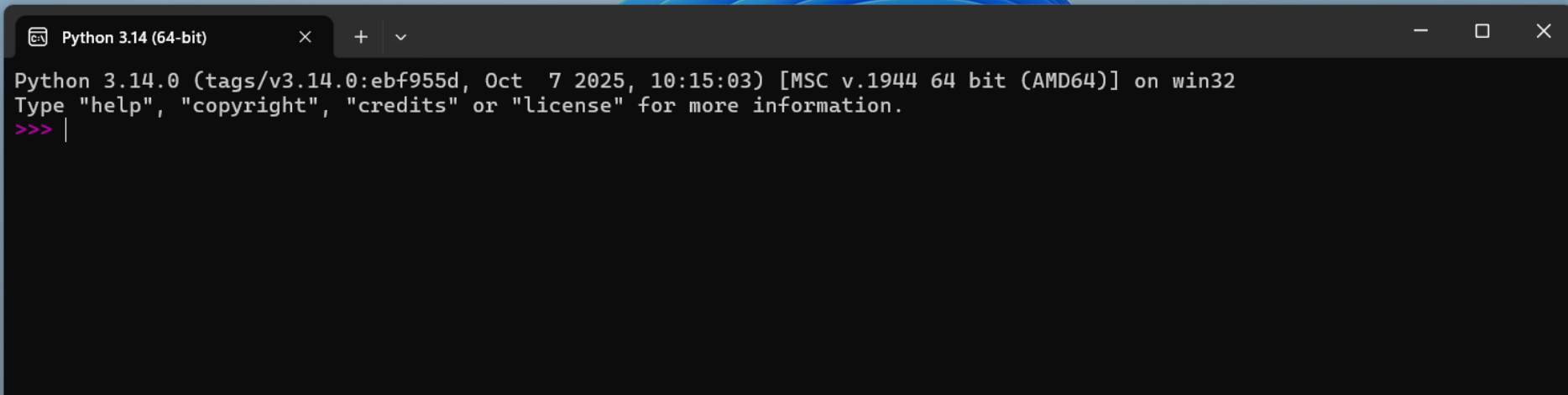






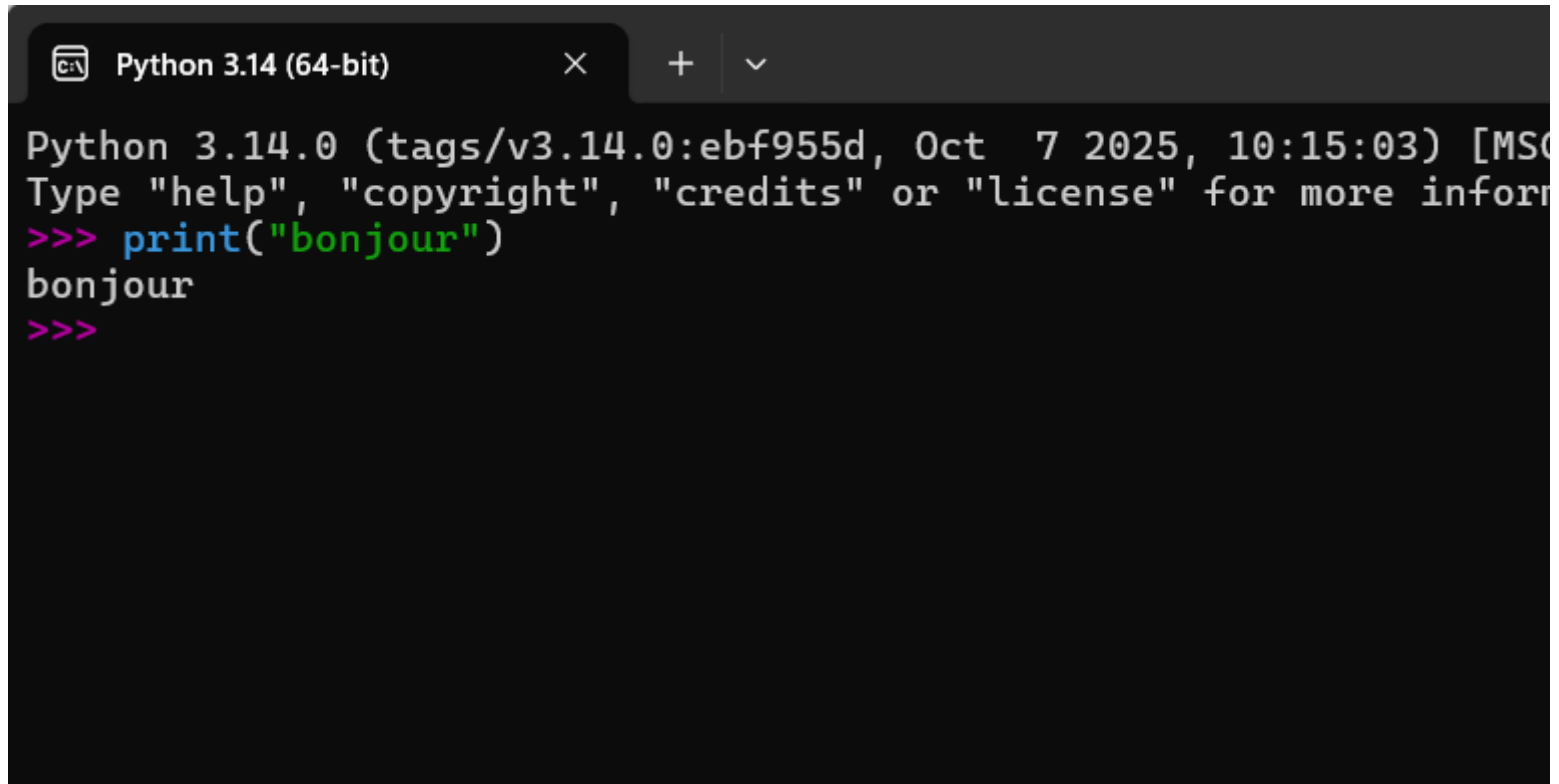


# Python Console



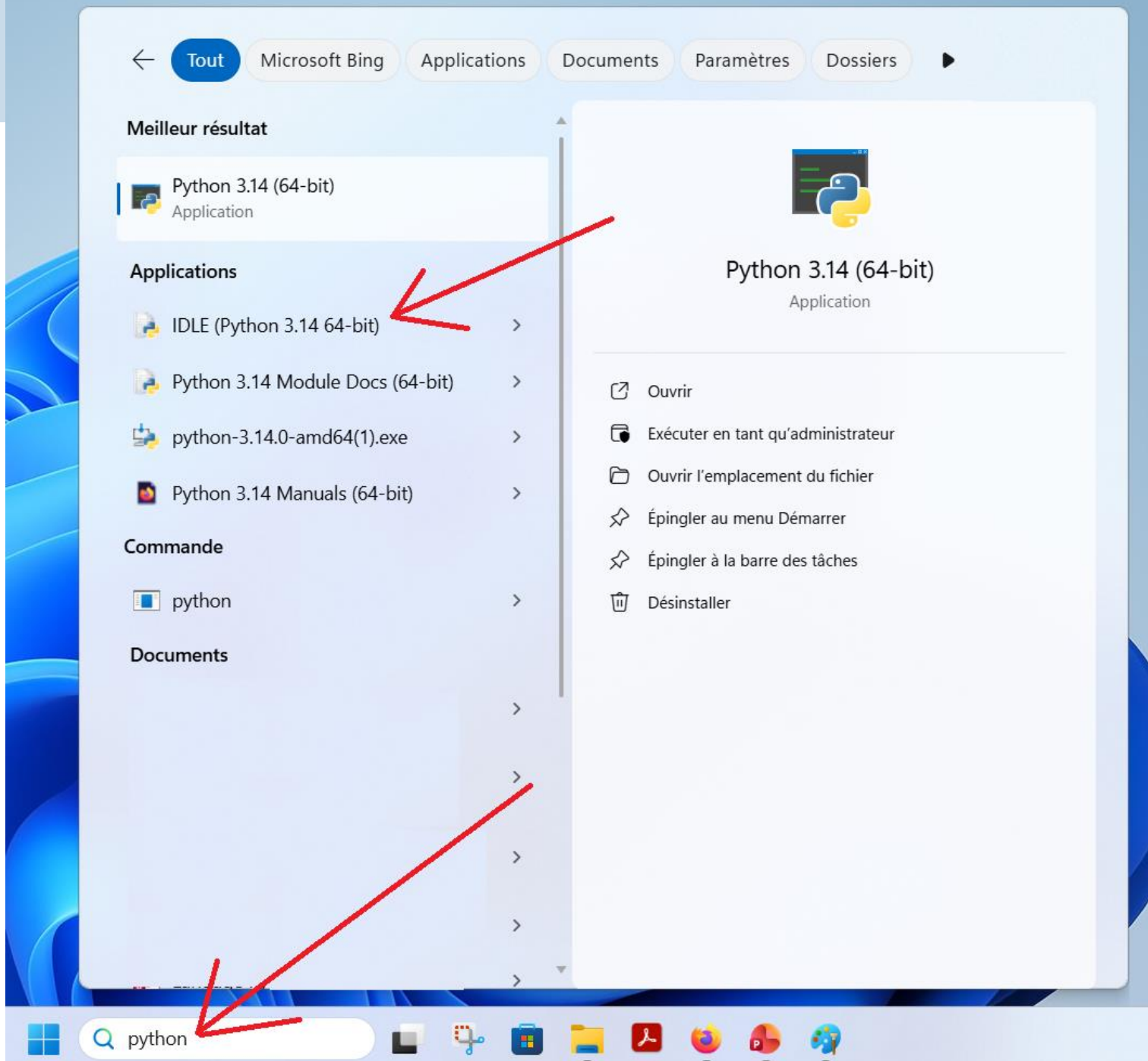
```
Python 3.14 (64-bit) × + ∨  
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
>>> |
```

# Python Console : écrivez et cliquez sur la touche Entrée

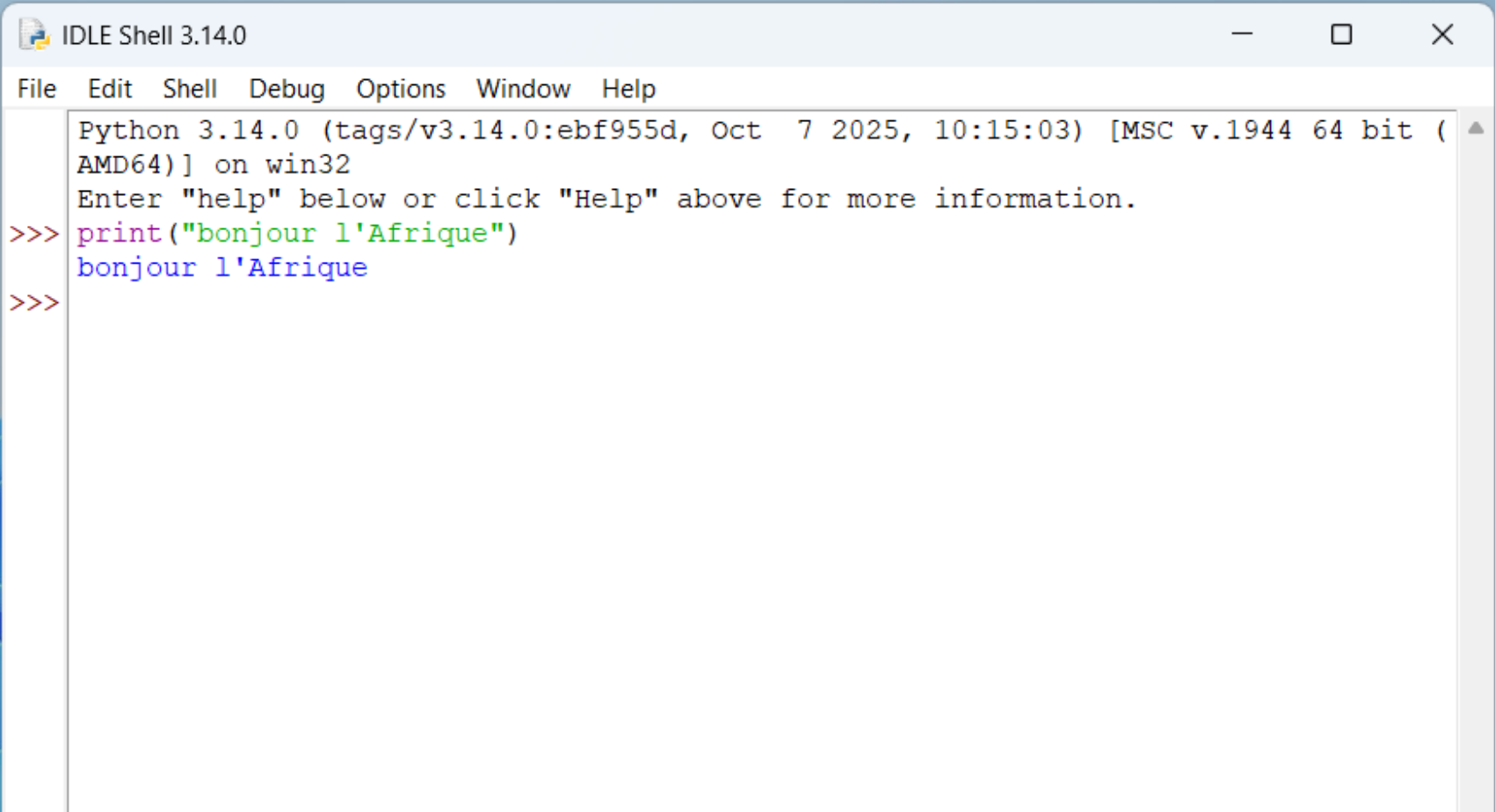


The screenshot shows a terminal window titled "Python 3.14 (64-bit)". The window contains the following text:

```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 7 2025, 10:15:03) [MSO
Type "help", "copyright", "credits" or "license" for more inform
>>> print("bonjour")
bonjour
>>>
```







```
IDLE Shell 3.14.0
File Edit Shell Debug Options Window Help
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>> print("bonjour l'Afrique")
      bonjour l'Afrique
>>>
```



# Différence entre Python Console et IDLE

## Python Console

- Interface en ligne de commande interactive
- Exécution immédiate instruction par instruction
- Environnement minimaliste
- Idéal pour tests rapides et exploration

## IDLE

- Éditeur de code avec coloration syntaxique
- Possibilité de créer et sauvegarder des fichiers .py
- Interface graphique complète

## Quand utiliser lequel :

- **Console** : tests rapides, exploration interactive
- **IDLE** : développement de scripts complets, apprentissage structuré

# La fonction print()

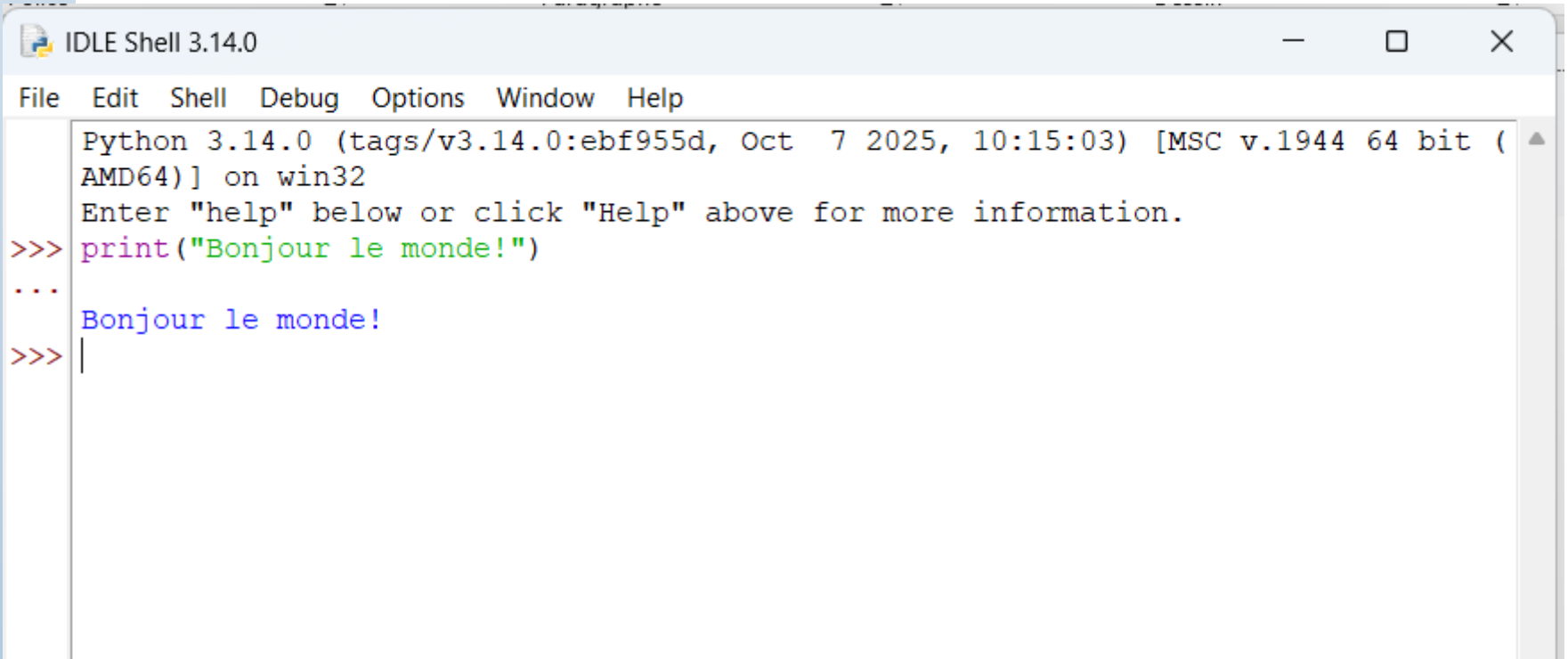
- Fonction pour afficher du texte
- Syntaxe : `print(valeur)`
- Peut afficher texte, nombres, variables
- Exemples :

```
print("Bonjour le monde!")
```

```
print(42)
```

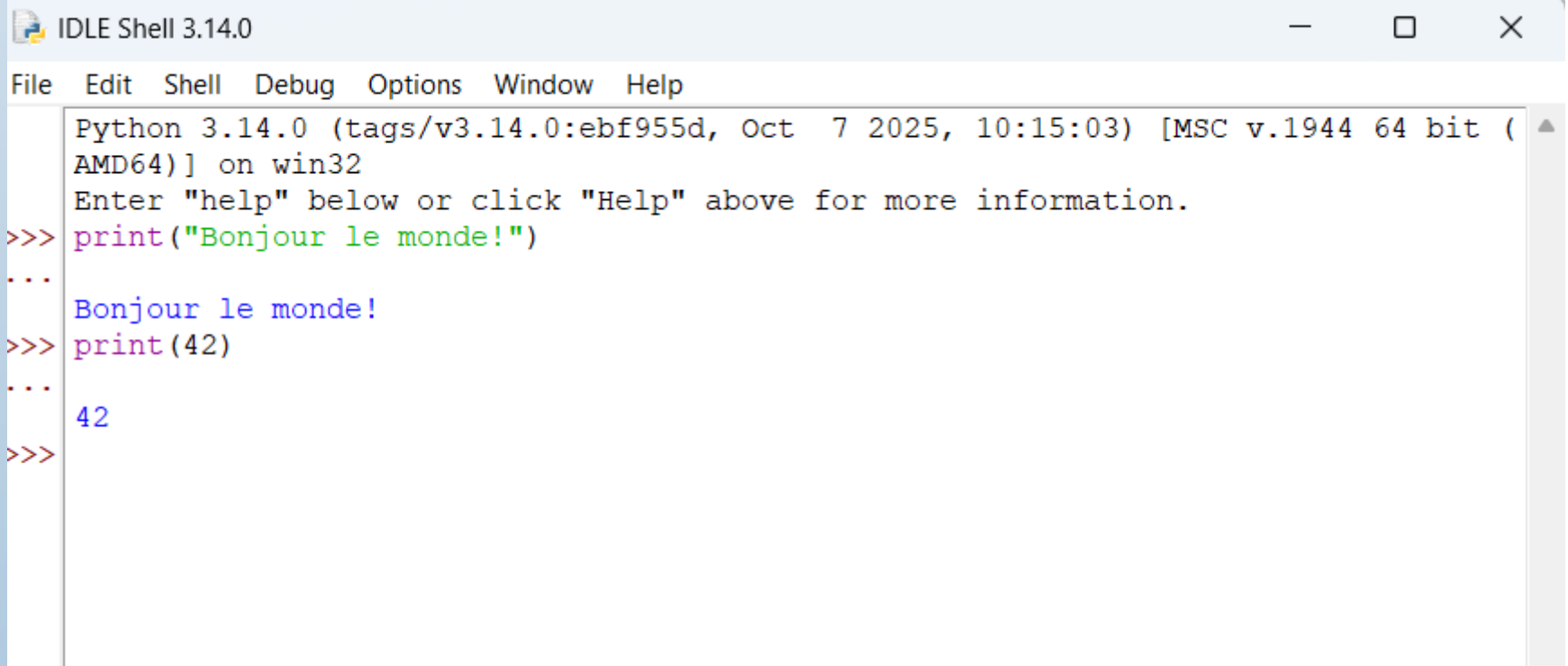
```
print(3.14)
```

# 1ère instruction



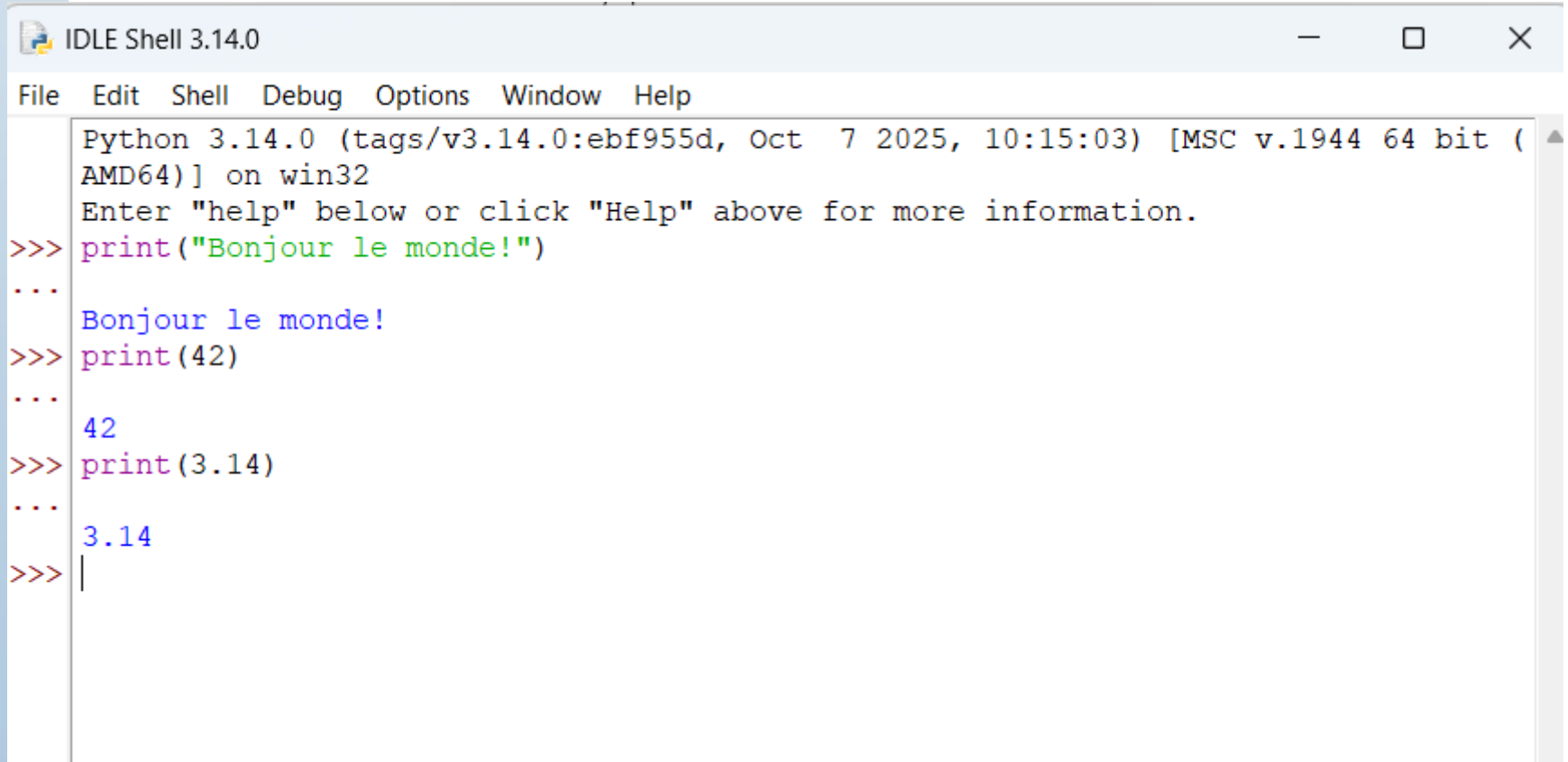
```
IDLE Shell 3.14.0
File Edit Shell Debug Options Window Help
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>> print("Bonjour le monde!")
...
Bonjour le monde!
>>> |
```

# 2ème instruction



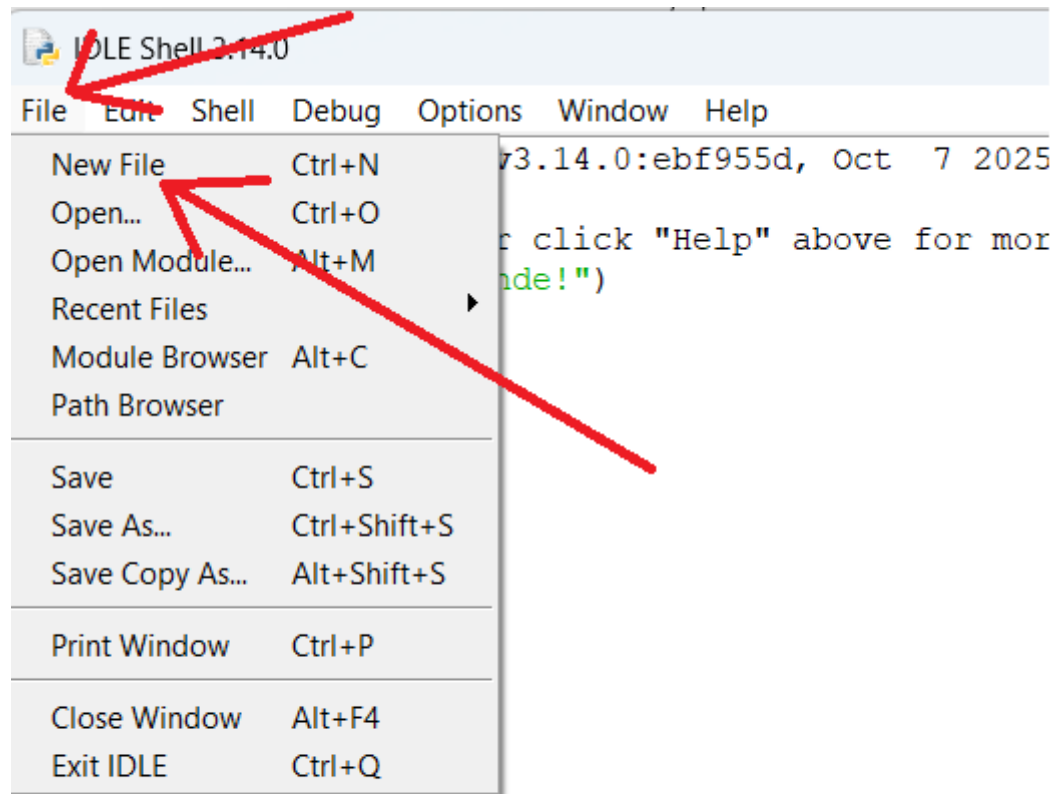
```
IDLE Shell 3.14.0
File Edit Shell Debug Options Window Help
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>> print("Bonjour le monde!")
...
Bonjour le monde!
>>> print(42)
...
42
>>>
```

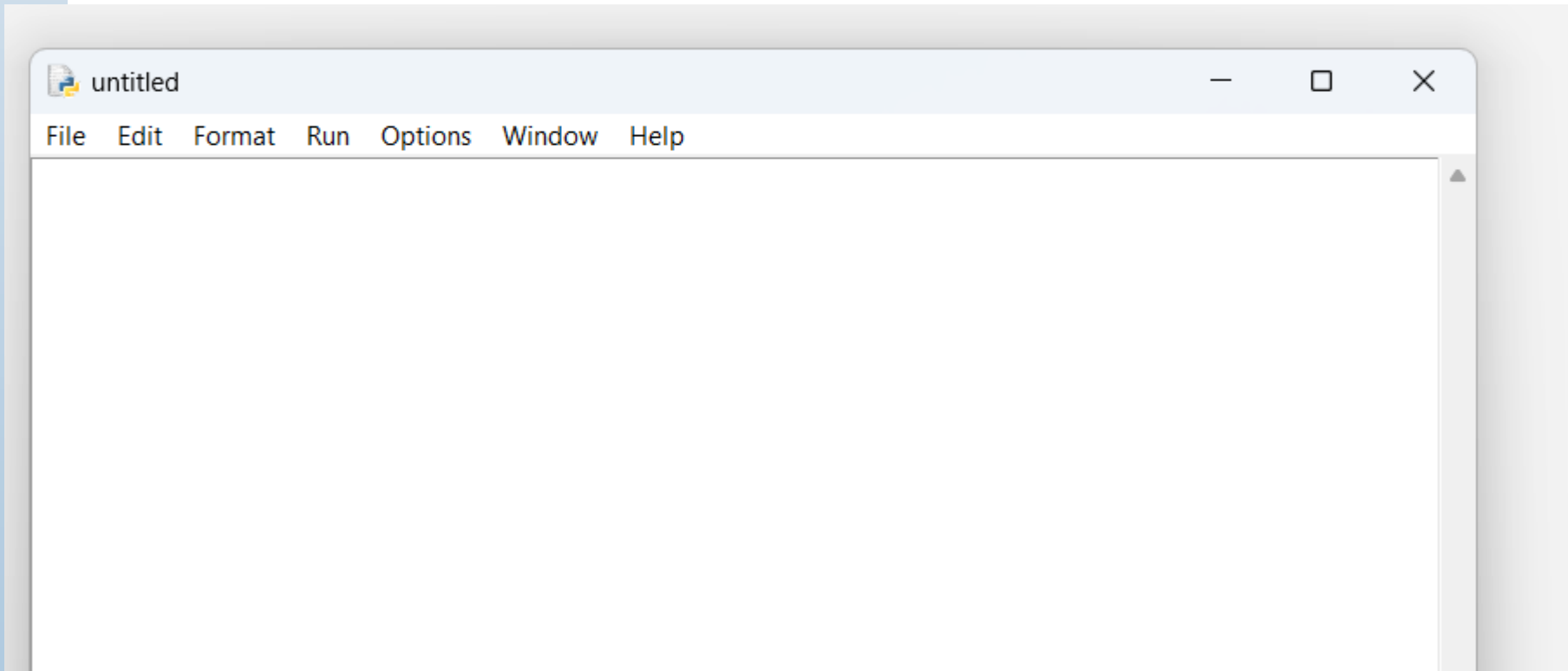
# 3ème instruction



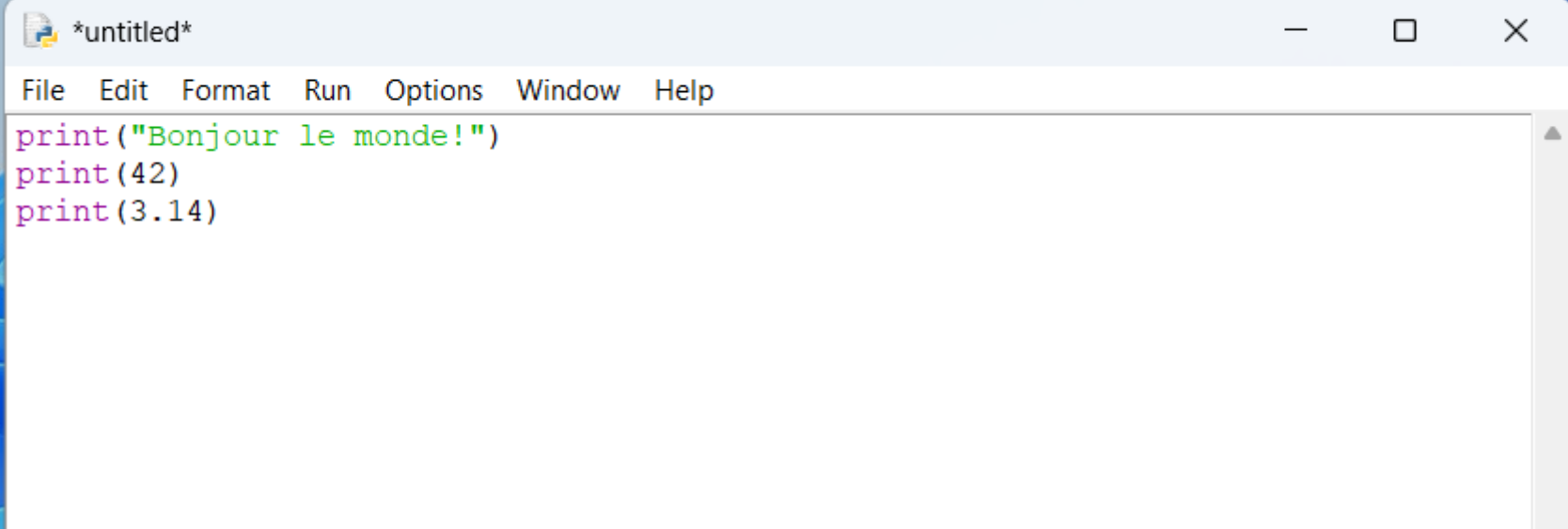
```
IDLE Shell 3.14.0
File Edit Shell Debug Options Window Help
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>> print("Bonjour le monde!")
...
Bonjour le monde!
>>> print(42)
...
42
>>> print(3.14)
...
3.14
>>> |
```

Et si on voulait exécuter les 3 instructions d'un seul coup !?

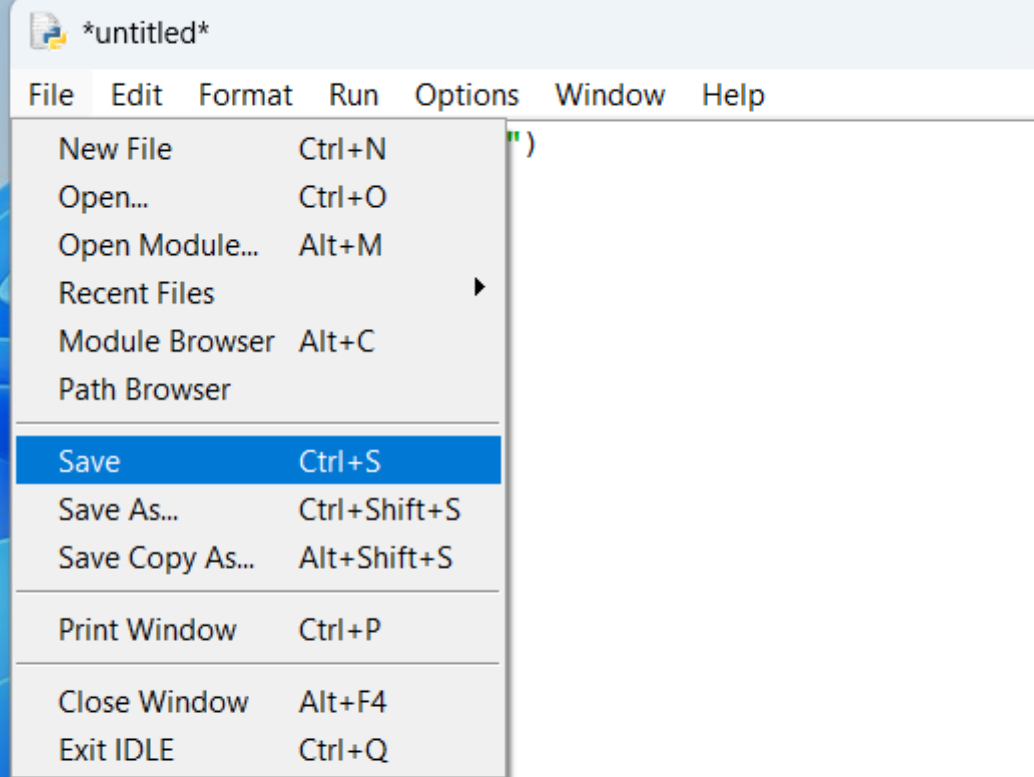


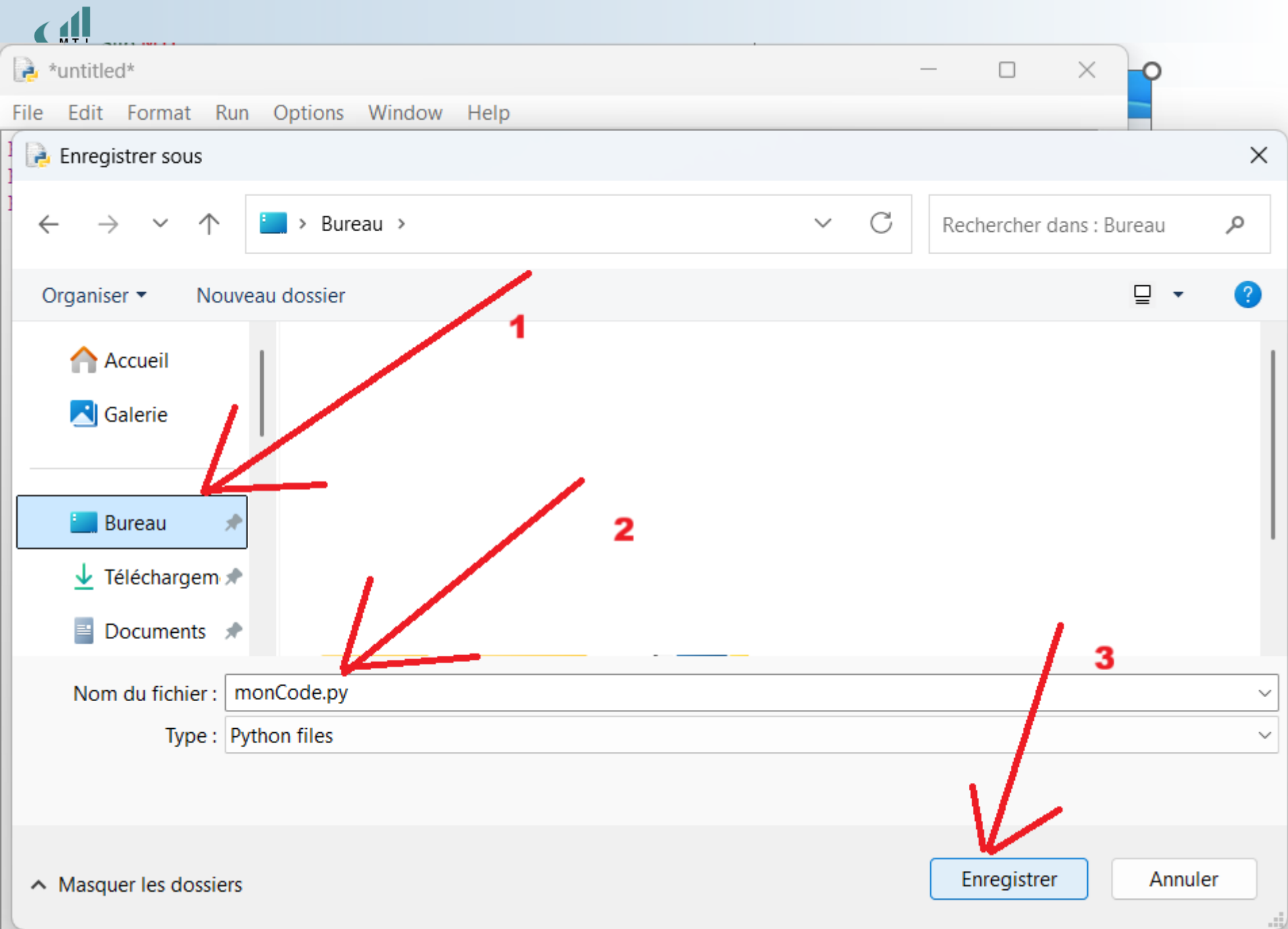


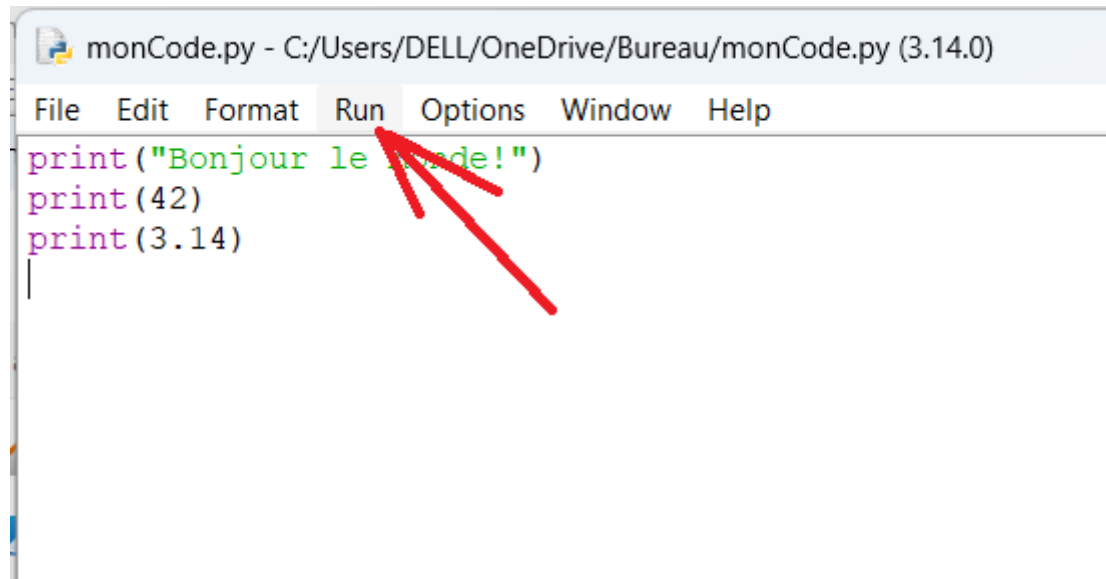




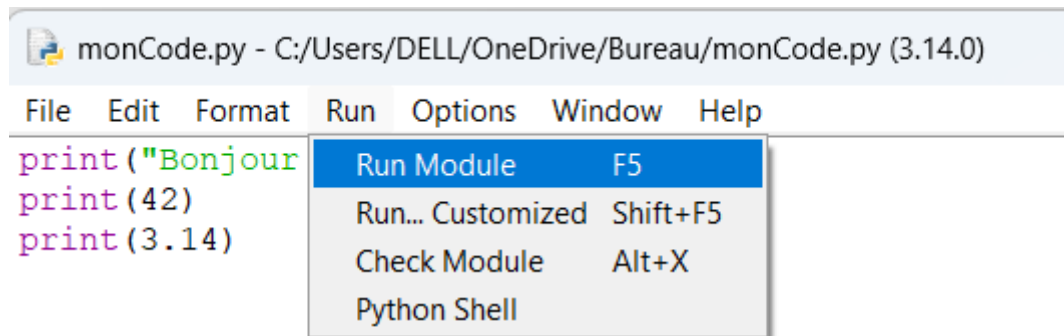
```
*untitled*  
File Edit Format Run Options Window Help  
print("Bonjour le monde!")  
print(42)  
print(3.14)
```







```
monCode.py - C:/Users/DELL/OneDrive/Bureau/monCode.py (3.14.0)
File Edit Format Run Options Window Help
print("Bonjour le monde!")
print(42)
print(3.14)
```



```
>>> ===== RESTART: C:/Users/DELL/OneDrive/Bureau/monCode.py ===  
Bonjour le monde!  
42  
3.14  
>>>
```

# Déclaration de variables

- Une variable est une boîte pour stocker une valeur
- Déclaration : `nom_variable = valeur`
- Typage dynamique
- Exemples :

`age = 25`

`nom = "Ali"`

`note = 19.99`

# Afficher des variables

- Combiner texte et variables

- Exemples :

```
age = 25
```

```
nom = "Ali"
```

```
print(nom)
```

```
print(age)
```

```
print(f"{nom} a {age} ans")
```

(attention il y a un : 'f' à ne pas oublier)



ok.py - C:/Users/DELL/AppData/Local/Programs/Python/Python314/ok.py (3.14.0)

File Edit Format Run Options Window Help

```
age = 25
nom = "Ali"
print(nom)
print(age)
print(f"{nom} a {age} ans")
```

IDLE Shell 3.14.0

File Edit Shell Debug Options Window Help

Python 3.14.0 (tags/v3.14.0:ebf955d, Oc  
AMD64) on win32  
Enter "help" below or click "Help" above

>>>

==== RESTART: C:/Users/DELL/AppData/Lo

>>>

==== RESTART: C:/Users/DELL/AppData/Lo

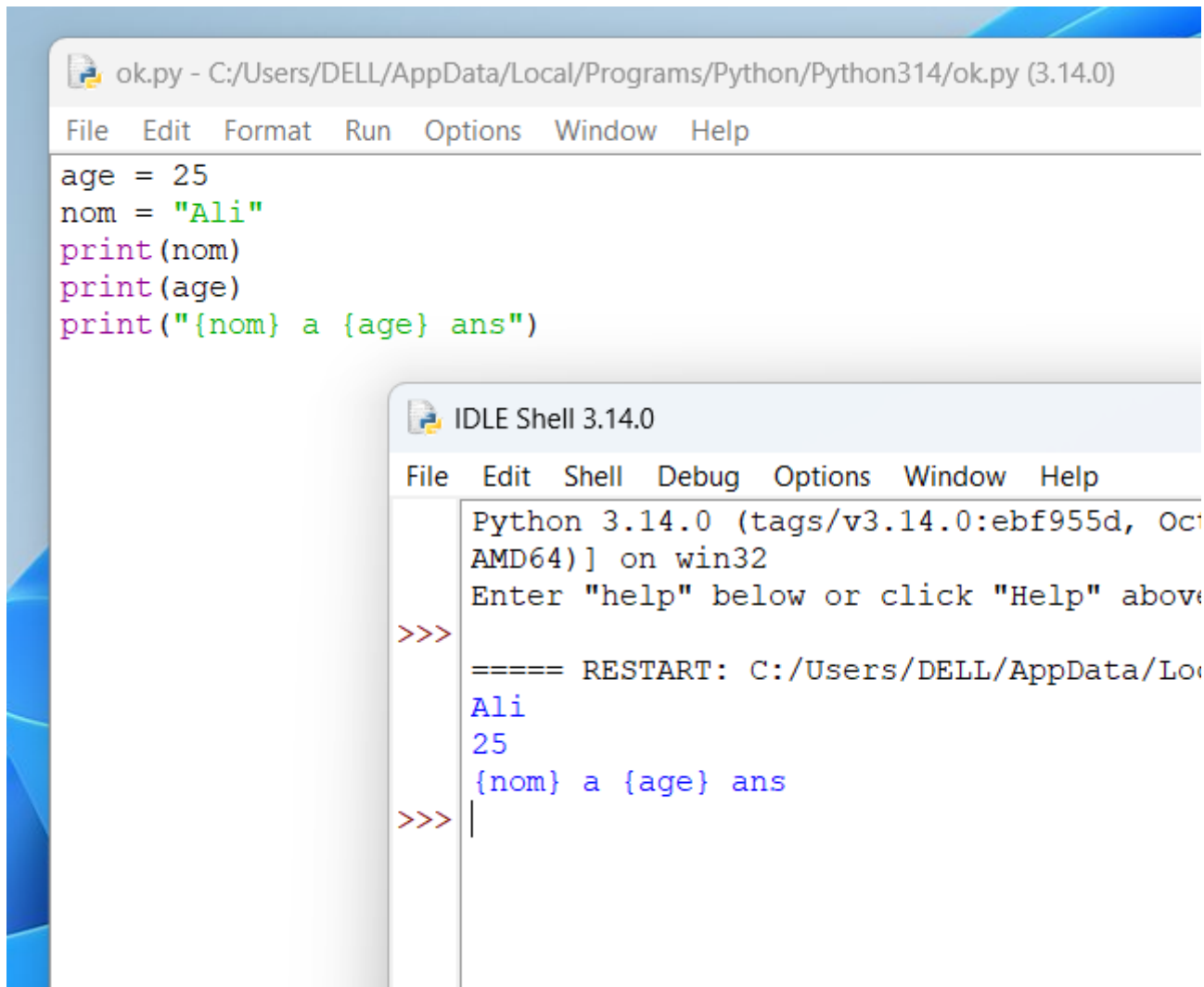
Ali

25

Ali a 25 ans

>>>

# Si vous oubliez le 'f' :



The screenshot shows a Python IDE window titled 'ok.py - C:/Users/DELL/AppData/Local/Programs/Python/Python314/ok.py (3.14.0)'. The code in the editor is:

```
age = 25
nom = "Ali"
print(nom)
print(age)
print("{nom} a {age} ans")
```

Below the editor is the 'IDLE Shell 3.14.0' window. It shows the output of the script after execution:

```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 14 2019, 15:50:00) on win32
Enter "help" below or click "Help" above
>>>
===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python314/IDLE Shell 3.14.0
Ali
25
{nom} a {age} ans
>>>
```

- Et si on ne veut pas utiliser 'f' ?!

Ecrivez donc ce code :

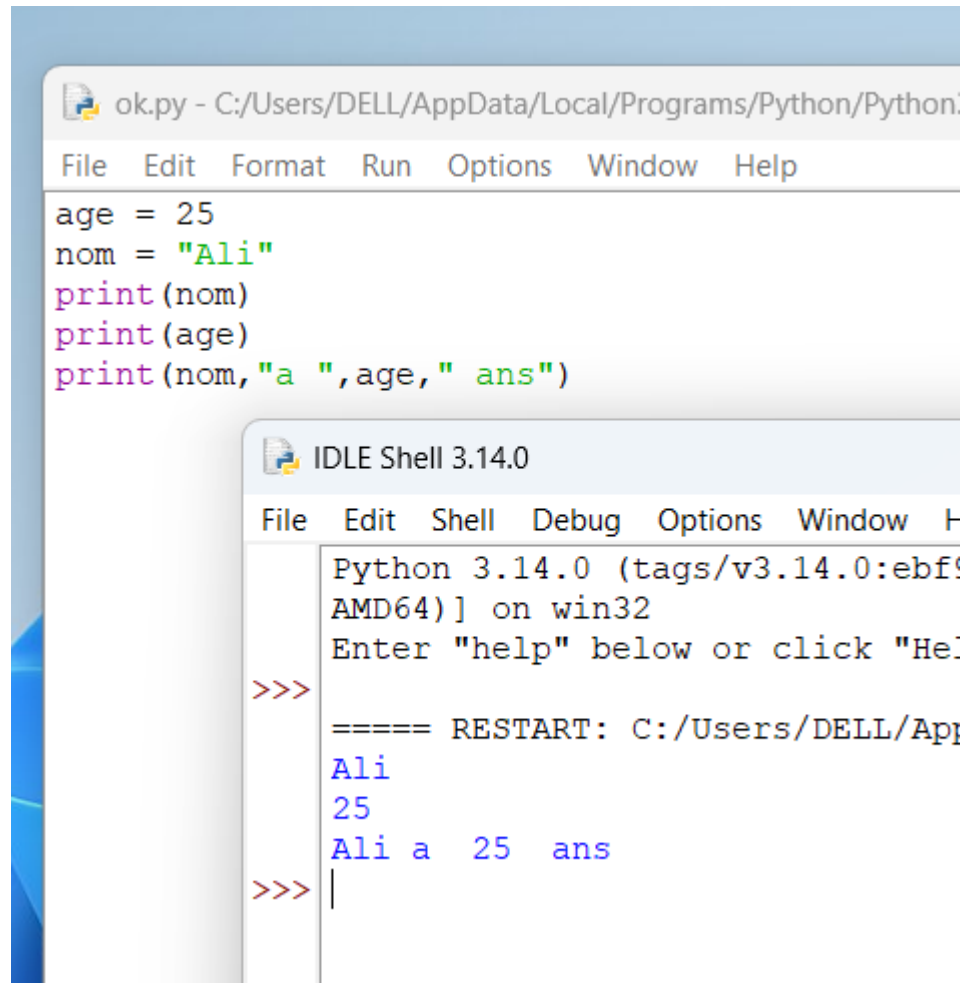
```
age = 25
```

```
nom = "Ali"
```

```
print(nom)
```

```
print(age)
```

```
print(nom,"a ",age," ans")
```



The screenshot displays the Python IDLE 3.14.0 interface. The main editor window, titled 'ok.py', contains the following Python code:

```
age = 25
nom = "Ali"
print(nom)
print(age)
print(nom, "a ", age, " ans")
```

Below the editor is the 'IDLE Shell 3.14.0' window. It shows the Python 3.14.0 startup message and the execution of the script. The output of the script is displayed as follows:

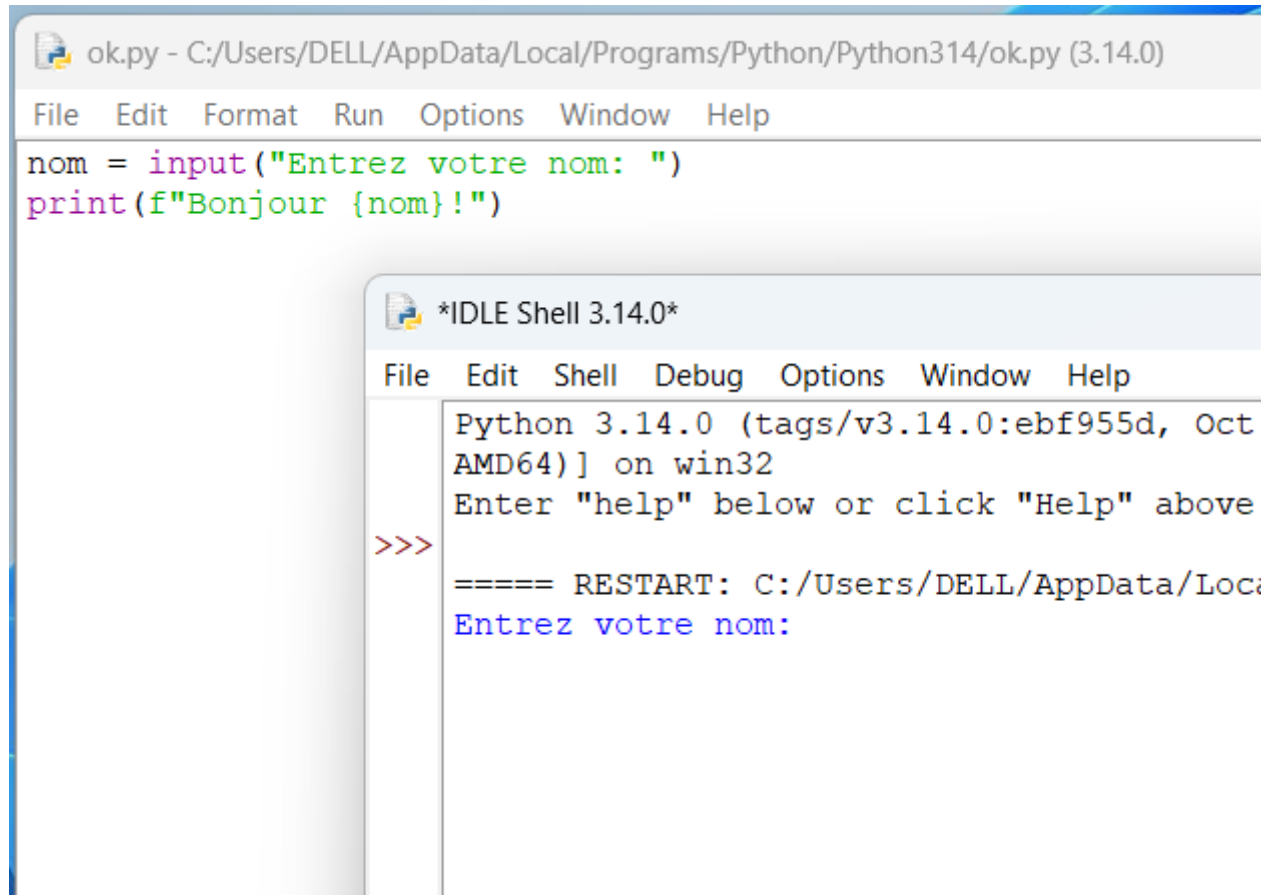
```
>>>
==== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python314/Python314.exe
Ali
25
Ali a  25  ans
>>> |
```

# La fonction input()

- Permet de récupérer une saisie
- Retourne toujours une chaîne de caractères
- Syntaxe : `variable = input("message")`
- Exemple :

```
nom = input("Entrez votre nom: ")
```

```
print(f"Bonjour {nom}!")
```

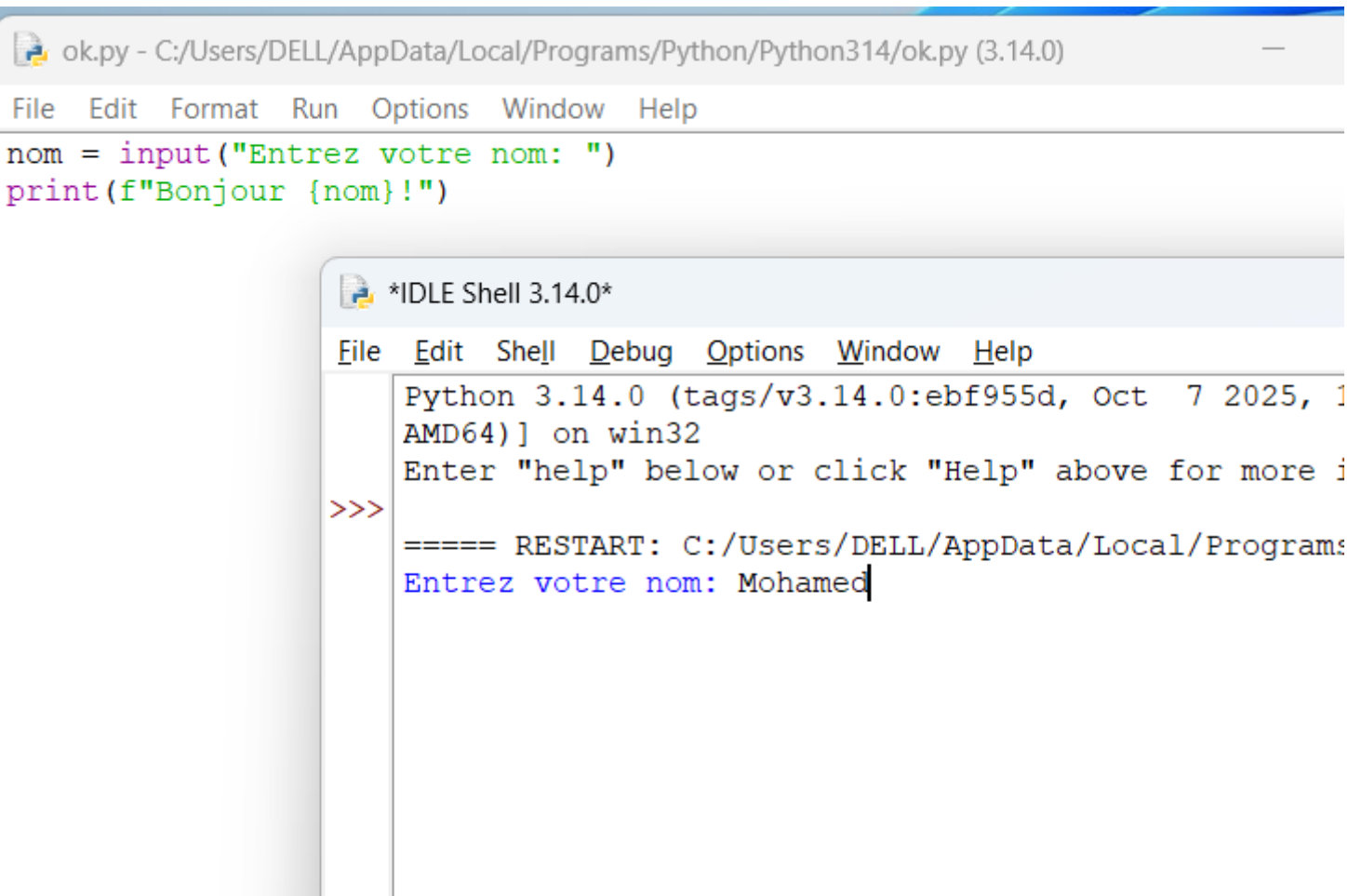


The screenshot displays the Python IDLE 3.14.0 interface. The main editor window, titled 'ok.py - C:/Users/DELL/AppData/Local/Programs/Python/Python314/ok.py (3.14.0)', contains the following Python code:

```
nom = input("Entrez votre nom: ")
print(f"Bonjour {nom}!")
```

Below the editor is the 'IDLE Shell 3.14.0\*' window. It shows the Python interpreter's startup message and the execution of the script:

```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct
AMD64)] on win32
Enter "help" below or click "Help" above
>>>
===== RESTART: C:/Users/DELL/AppData/Local
Entrez votre nom:
```

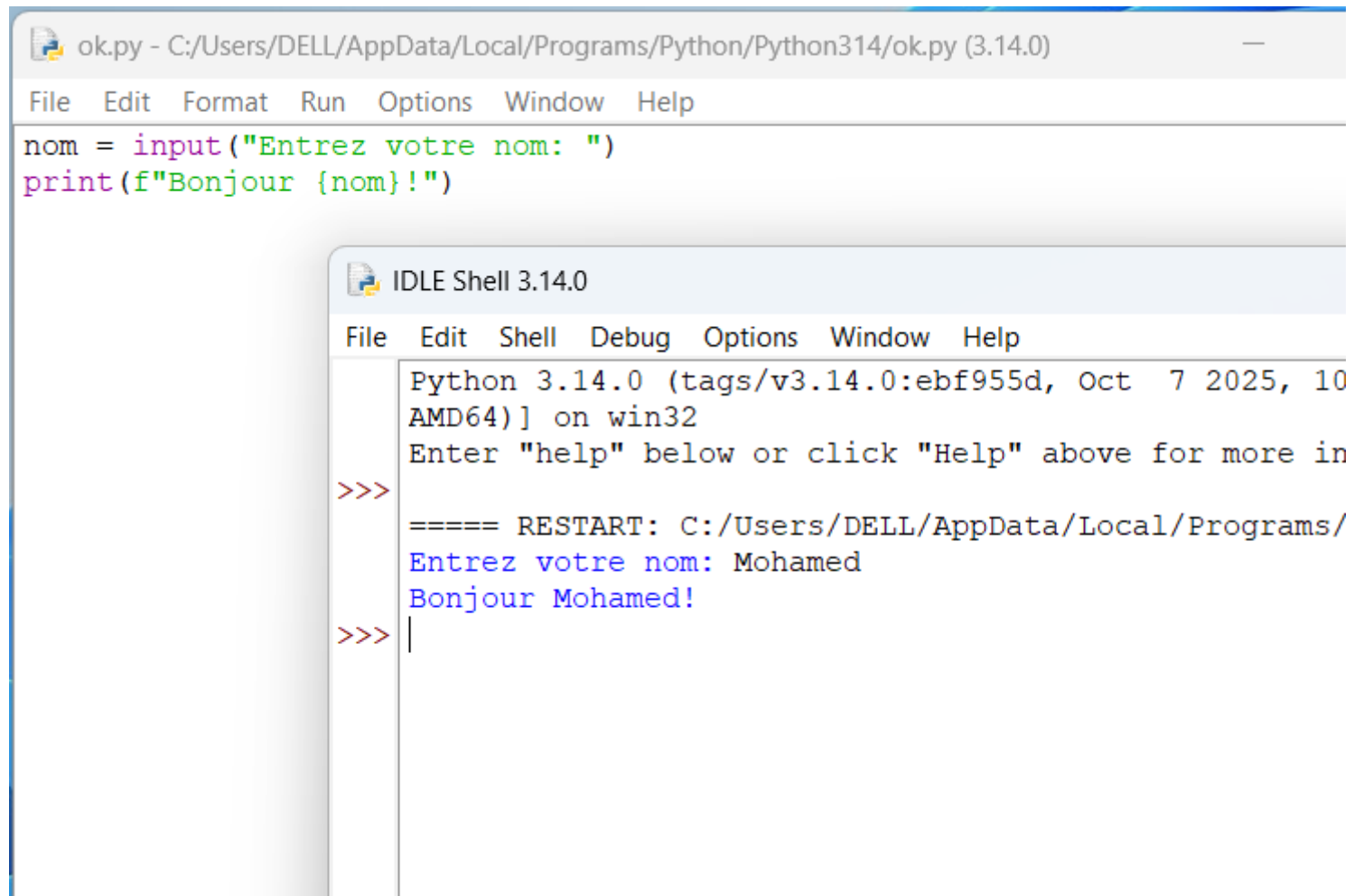


The image shows a screenshot of a Python script editor and its shell output. The editor window, titled 'ok.py - C:/Users/DELL/AppData/Local/Programs/Python/Python314/ok.py (3.14.0)', contains the following code:

```
nom = input("Entrez votre nom: ")
print(f"Bonjour {nom}!")
```

The shell window, titled '\*IDLE Shell 3.14.0\*', shows the execution of the script. It displays the Python version and architecture, followed by a prompt to enter help. The user has entered 'Mohamed' in response to the 'Entrez votre nom:' prompt.

```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 7 2025, 1
AMD64)] on win32
Enter "help" below or click "Help" above for more i
>>>
===== RESTART: C:/Users/DELL/AppData/Local/Programs
Entrez votre nom: Mohamed
```



The screenshot displays a Python IDE window titled 'ok.py - C:/Users/DELL/AppData/Local/Programs/Python/Python314/ok.py (3.14.0)'. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code in the editor is:

```
nom = input("Entrez votre nom: ")
print(f"Bonjour {nom}!")
```

Below the editor is an 'IDLE Shell 3.14.0' window with its own menu bar (File, Edit, Shell, Debug, Options, Window, Help). The shell output shows the program's execution:

```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 7 2025, 10
AMD64)] on win32
Enter "help" below or click "Help" above for more in
>>>
===== RESTART: C:/Users/DELL/AppData/Local/Programs/
Entrez votre nom: Mohamed
Bonjour Mohamed!
>>> |
```

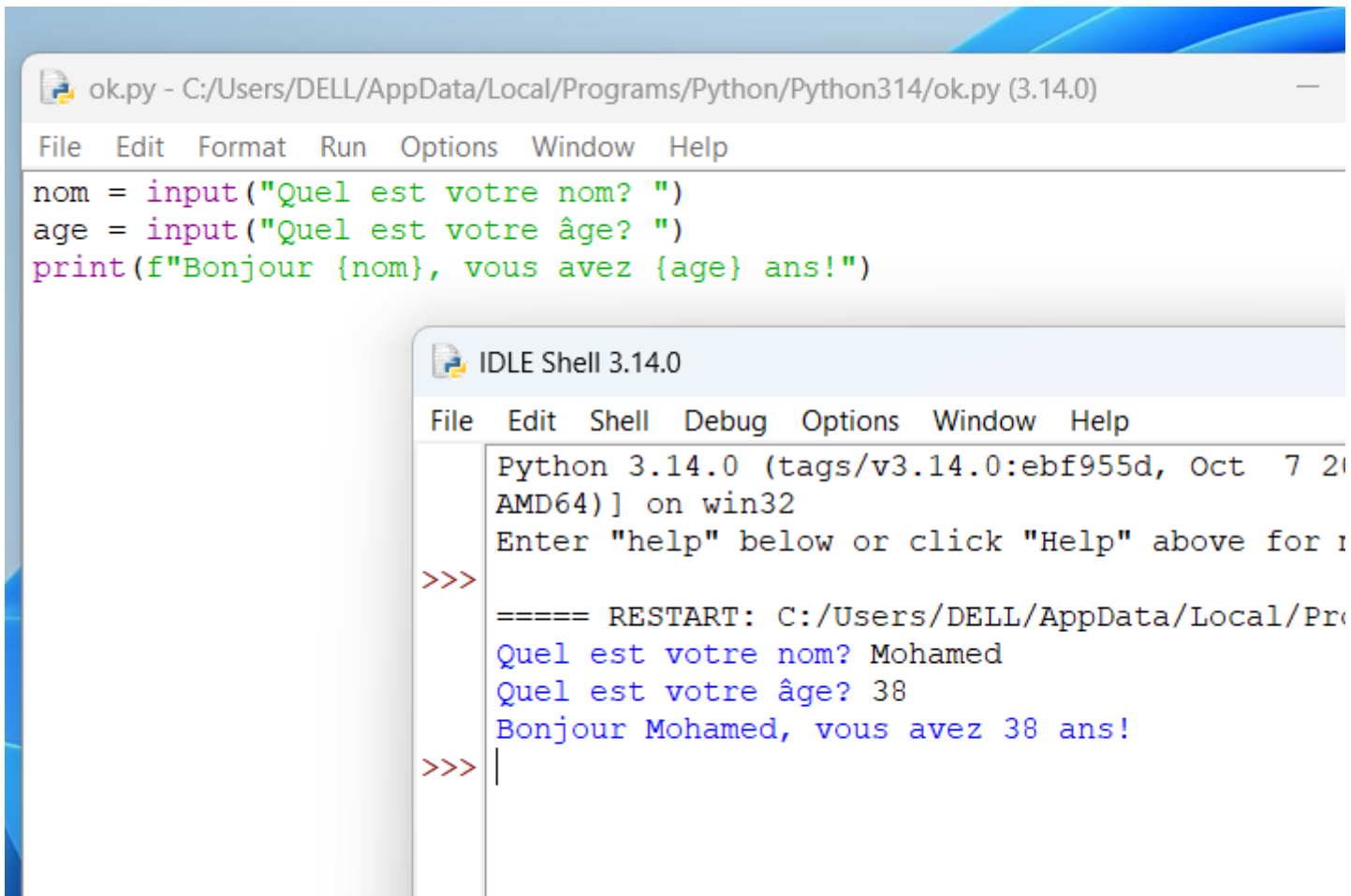


- Combinaison de tous les concepts

```
nom = input("Quel est votre nom? ")
```

```
age = input("Quel est votre âge? ")
```

```
print(f"Bonjour {nom}, vous avez {age} ans!")
```



The screenshot shows the Python IDLE 3.14.0 environment. The main window displays a Python script named 'ok.py' with the following code:

```
nom = input("Quel est votre nom? ")
age = input("Quel est votre âge? ")
print(f"Bonjour {nom}, vous avez {age} ans!")
```

The 'IDLE Shell 3.14.0' window is open, showing the execution of the script. It displays the Python version and architecture, followed by the execution of the script's output:

```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 7 2024, 12:00:00) on win32
Enter "help" below or click "Help" above for more
>>>
===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python314/IDLE Shell 3.14.0
Quel est votre nom? Mohamed
Quel est votre âge? 38
Bonjour Mohamed, vous avez 38 ans!
>>> |
```

# Les commentaires en Python

- Les commentaires sont ignorés par l'interpréteur Python
- Servent à expliquer le code pour les développeurs
- Deux types de commentaires :
  - Commentaires ligne unique avec #
  - Commentaires multi-lignes avec `"""` ou `'''`
- Bonnes pratiques :
  - Commenter le "pourquoi" plus que le "comment"
  - Garder les commentaires à jour
  - Éviter les commentaires évidents

## ■ **Commentaire simple :**

# Ceci est un commentaire sur une ligne

age = 25 # Commentaire en fin de ligne



The screenshot displays the Python IDLE 3.14.0 environment. The main editor window, titled 'ok.py - C:/Users/DELL/AppData/Local/Programs/Python/Python314/ok.py (3.14.0)', contains the following Python code:

```
nom = input("Quel est votre nom? ")

# Ceci est un commentaire sur une ligne
age = 25  # Commentaire en fin de ligne

print(f"Bonjour {nom}, vous avez {age} ans!")
```

The Shell window, titled 'IDLE Shell 3.14.0', shows the execution of the script. It displays the Python version and architecture, followed by the prompt 'Enter "help" below or click "Help" at'. The user has entered 'Rym' in response to the 'Quel est votre nom?' prompt, and the script has printed 'Bonjour Rym, vous avez 25 ans!'.

```
Python 3.14.0 (tags/v3.14.0:ebf955d, AMD64) on win32
Enter "help" below or click "Help" at
>>>
===== RESTART: C:/Users/DELL/AppData/
Quel est votre nom? Rym
Bonjour Rym, vous avez 25 ans!
>>> |
```

## ■ Commentaire multi-lignes :

""""

Ceci est un commentaire  
qui s'étend sur  
plusieurs lignes

""""

ok.py - C:/Users/DELL/AppData/Local/Programs/Python/Python314/ok.py (3.14.0)

File Edit Format Run Options Window Help

```
nom = input("Quel est votre nom? ")
age = input("Quel est votre âge? ")

"""
Ceci est un commentaire
qui s'étend sur
plusieurs lignes
"""
print(f"Bonjour {nom}, vous avez {age} ans!")
```

IDLE Shell 3.14.0

File Edit Shell Debug Options Window Help

Python 3.14.0 (tags/v3.14.0:ebf955  
AMD64) on win32

Enter "help" below or click "Help"

>>>

===== RESTART: C:/Users/DELL/AppDa

Quel est votre nom? Ali

Quel est votre âge? 20

Bonjour Ali, vous avez 20 ans!

>>>

## ■ Exemples de commentaires

### ■ Exemple pratique :

# Calcul de la surface d'un rectangle

longueur = 10

largeur = 5

surface = longueur \* largeur # Formule:  $L \times l$

print(f"Surface: {surface}") # Affichage du résultat



ok.py - C:/Users/DELL/AppData/Local/Programs/Python/Python314/ok.py (3.14.0)

File Edit Format Run Options Window Help

```
# Calcul de la surface d'un rectangle
longueur = 10
largeur = 5
surface = longueur * largeur # Formule: L × l
print(f"Surface: {surface}") # Affichage du résultat
```

IDLE Shell 3.14.0

File Edit Shell Debug Options Window Help

Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 7 2025,  
AMD64) on win32

Enter "help" below or click "Help" above for more

>>>

==== RESTART: C:/Users/DELL/AppData/Local/Progra  
Surface: 50

>>> |