Python | Introduction

Basics. Python programming language, interpreter and overview of popular IDEs, basic data types (int, float, string, bool, list), operators (arithmetic, relational, logical)

Today’s Goals!

□ understand what programming is & why python
□ install python & set up working environment
□ first program in python!
□ operators in python
□ identify basic data types

Programming

Programming is telling machines to do things.

Machines can only understand binary.

We don’t.

The solution?

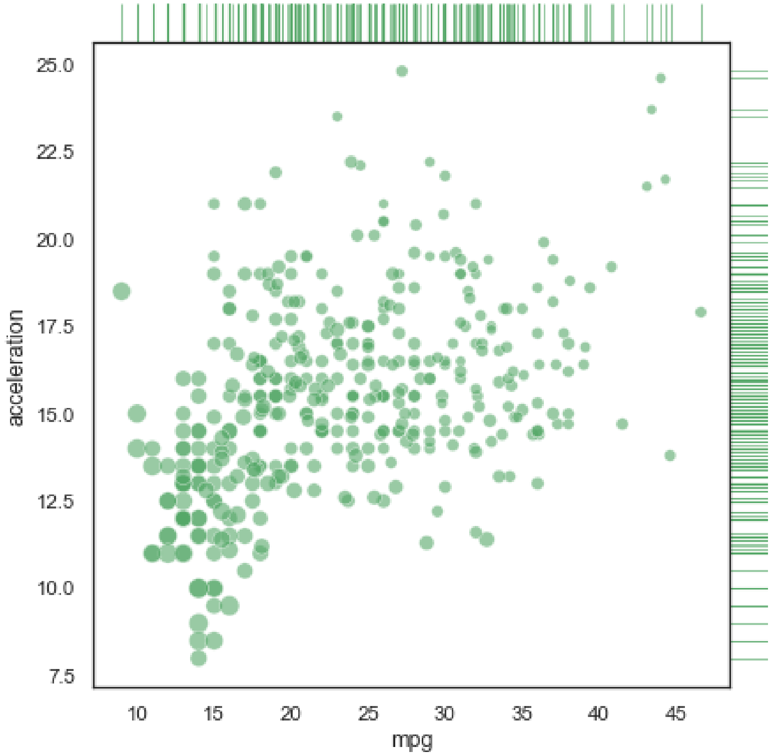
Interpreters!

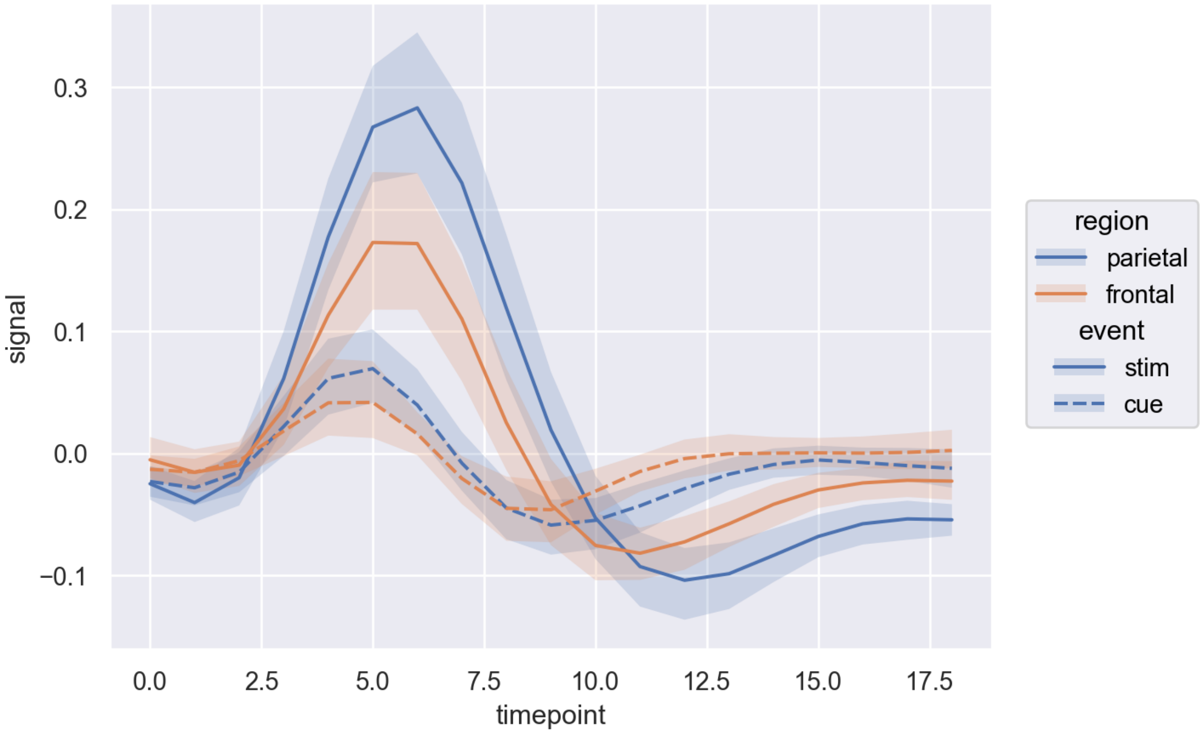
We write in a language we understand, and interpreters will compile it into machine code. A compiler for Python is included in the standard installations.

Language	Meaning	Difficulty	Example
High level	Abstractions to hide implementation details	Easy	JavaScript, python
Low level	Need to interact with machines	Harder	Rust, C, assembly

Why Python?

- Very easy to learn
- Widely used for data analysis, visualizations, building apps and servers, and building automated systems
- Huge community of developers, tons of libraries for data analysis, bioinformatics, and machine learning
- Integrates well with existing bioinformatics tools and pipelines


acceleration
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
10
15
20
25
30
35
40
45
mpg


signal
0.3
0.2
0.1
0.0
−0.1
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
timepoint
region
parietal
frontal
event
stim
cue

Python

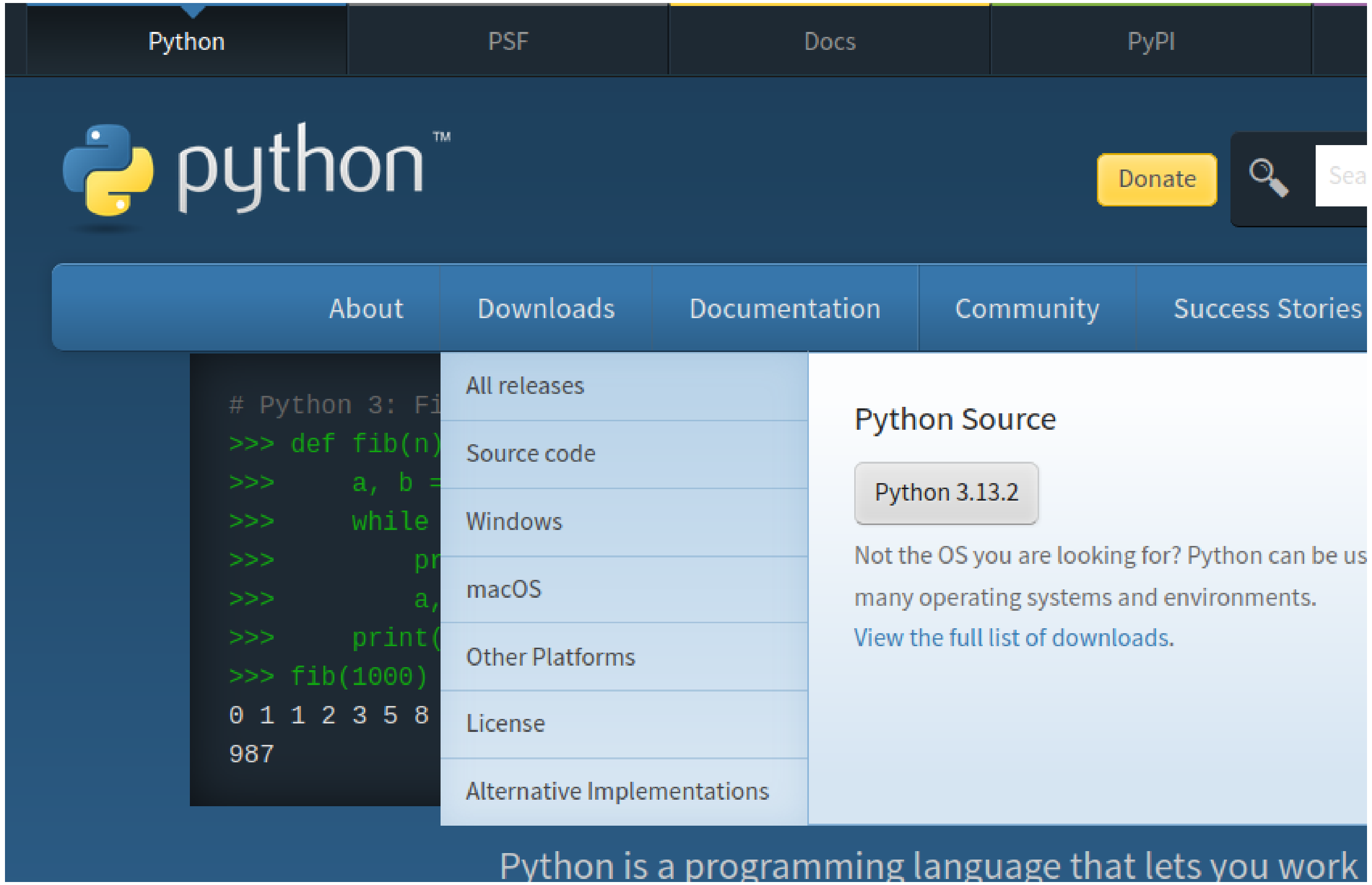
- Python is a high-level programming language for general purpose.
- It is open-source, and provides a built-in standard *library.*
- Python is very powerful, as you can not only write your own programs in an intuitive manner, but also leverage programs written by millions of programmers.
- It’s commonly used for automation, data analytics, databases, documentation, graphical user interfaces, machine learning, mobile apps, multimedia, networking, scientific computing, system administration, text processing, web frameworks, web scraping, and more.

History

- Guido van Rossum was Python’s benevolent dictator for life (BDFL) until 2018.
- He started working on Python in late 1980 as a successor to ABC programming language and released Python 0.9.0 in 1991.
- Subsequent versions – Python 2.0 (2000) and Python 3.0 (2008) are widely in use today.
- The design philosophy (simplicity and usability) of Python has inspired languages like Go, Groovy, Julia, Mojo, Ruby, Kotlin (Android), and Swift (Apple/ iOS).

Installing Python

Download latest version from the official site:
https://www.python.org/downloads/



```
# Python 3: Fi
>>> def fib(n)
>>>     a, b =
>>>     while
>>>         pr
>>>         a,
>>>     print(
>>> fib(1000)
0 1 1 2 3 5 8
987
```

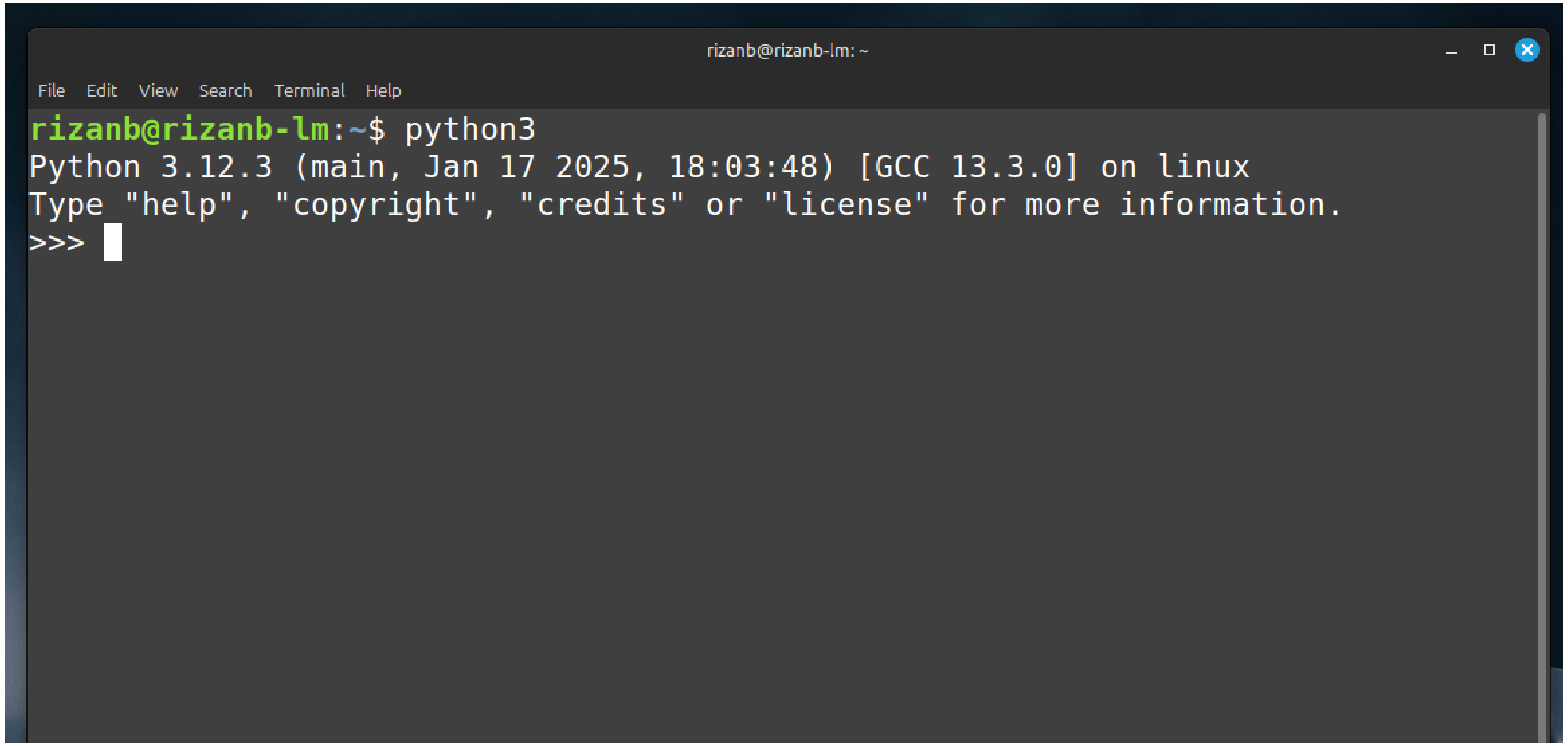
Python Source

Not the OS you are looking for? Python can be us
many operating systems and environments.
View the full list of downloads.

Python is a programming language that lets you work

Check Python installation

Open a terminal and type: python or python3



```
rizanb@rizanb-lm:~$ python3
Python 3.12.3 (main, Jan 17 2025, 18:03:48) [GCC 13.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

IDE

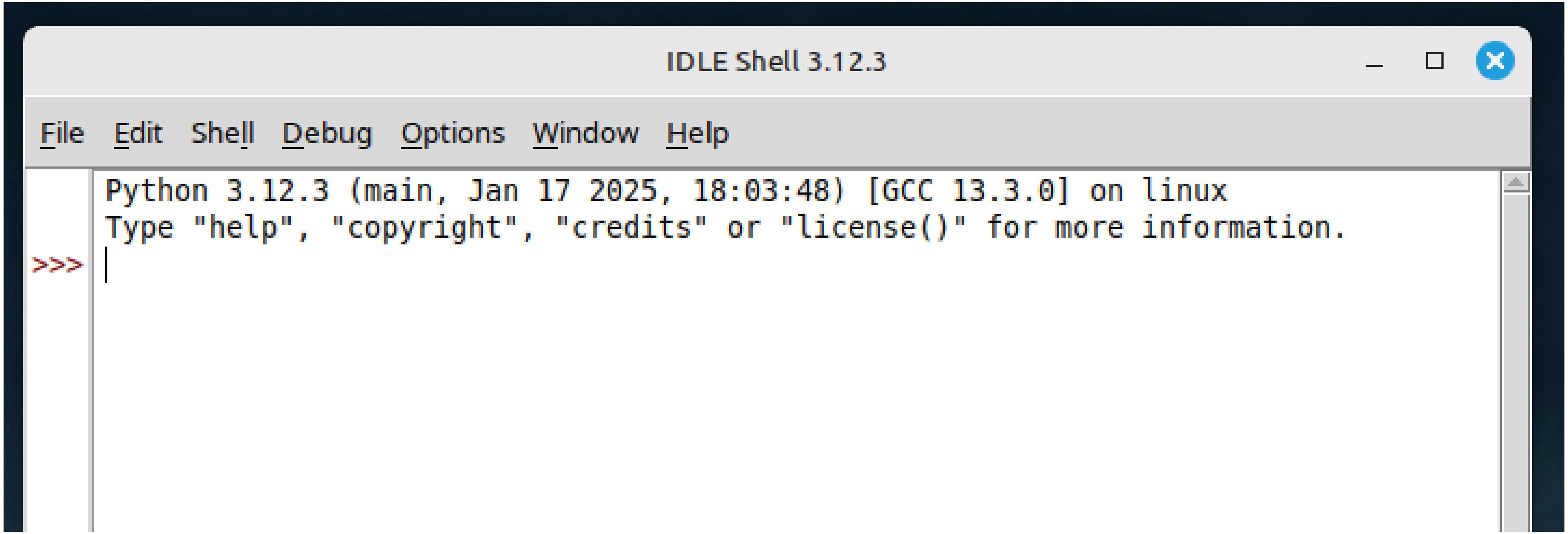
IDE (Integrated Development Environment) is where we write code. IDEs make our life easier by offering syntax highlighting, auto-completion, debugging tools, and lots of features.

Common IDEs/ code editors are IDLE, Visual Studio Code, PyCharm, Spyder, Jupyter Notebook, and Google Colab.

Anaconda distribution also offers many of these tools.

IDLE

IDLE comes bundled with Python installations (on Windows). It's a simple IDE with basic features- great if you're working on small projects.

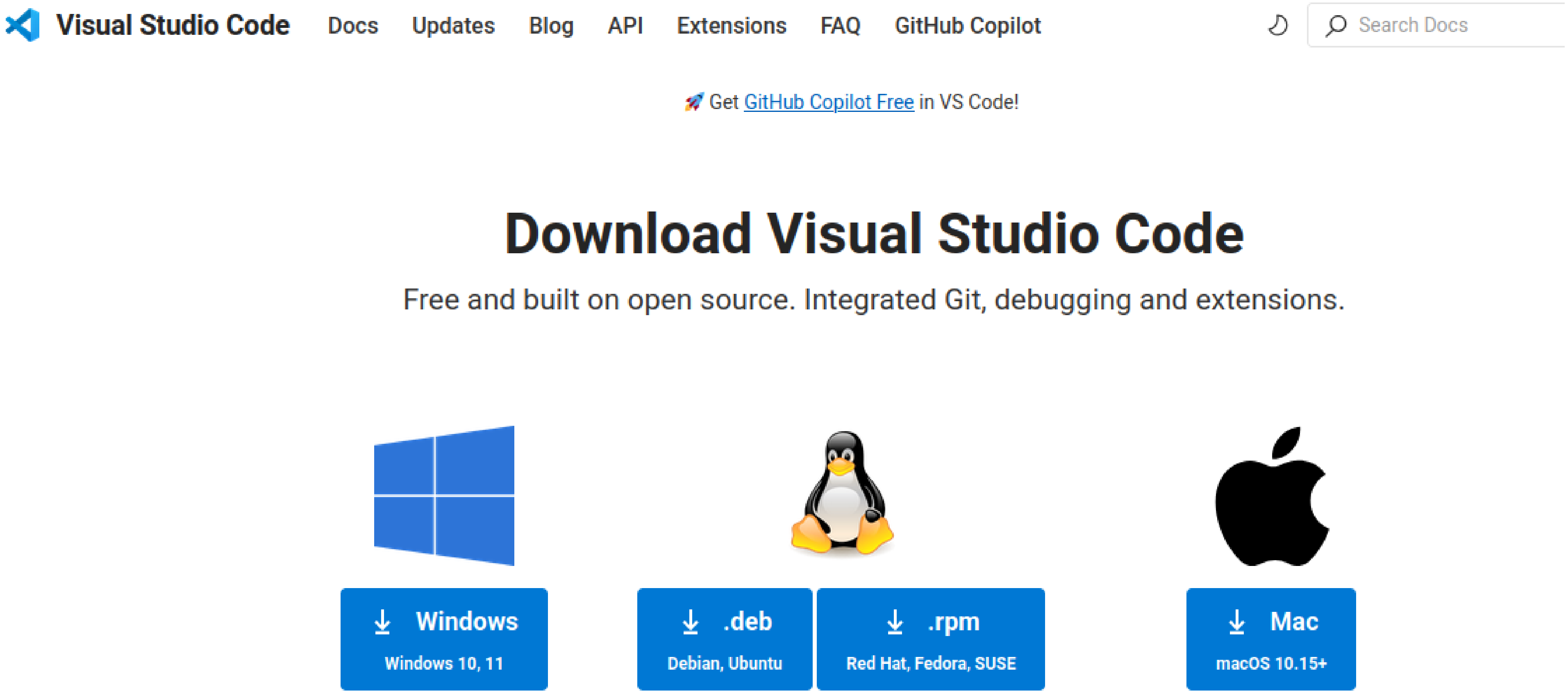


```
Python 3.12.3 (main, Jan 17 2025, 18:03:48) [GCC 13.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

Visual Studio Code

Popular, by Microsoft, download from: https://code.visualstudio.com/download

Or check the open-source alternative: https://github.com/VSCodium/vscodium



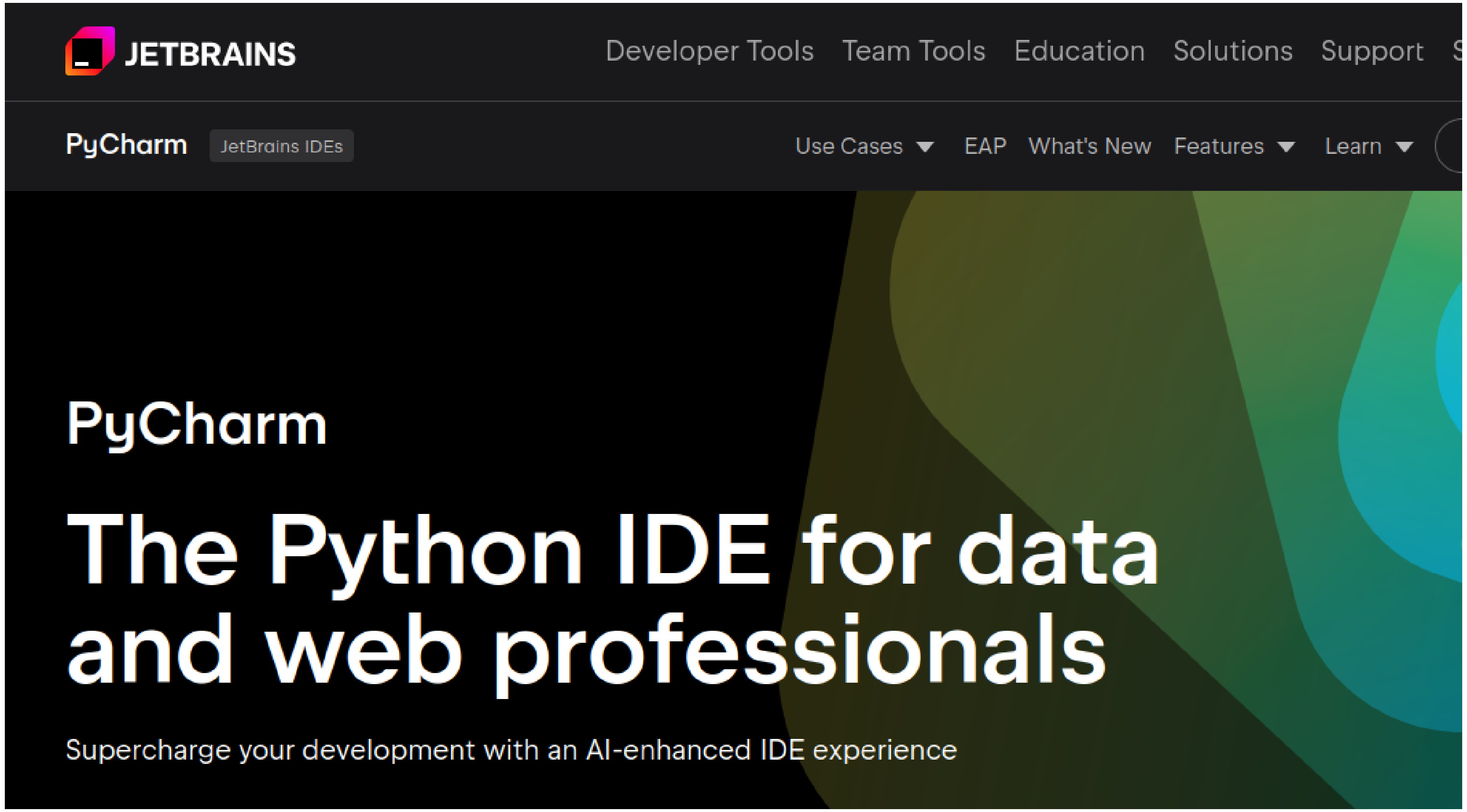
🚀 Get GitHub Copilot Free in VS Code!

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

PyCharm

Common IDE for data science, download from:
https://www.jetbrains.com/pycharm/



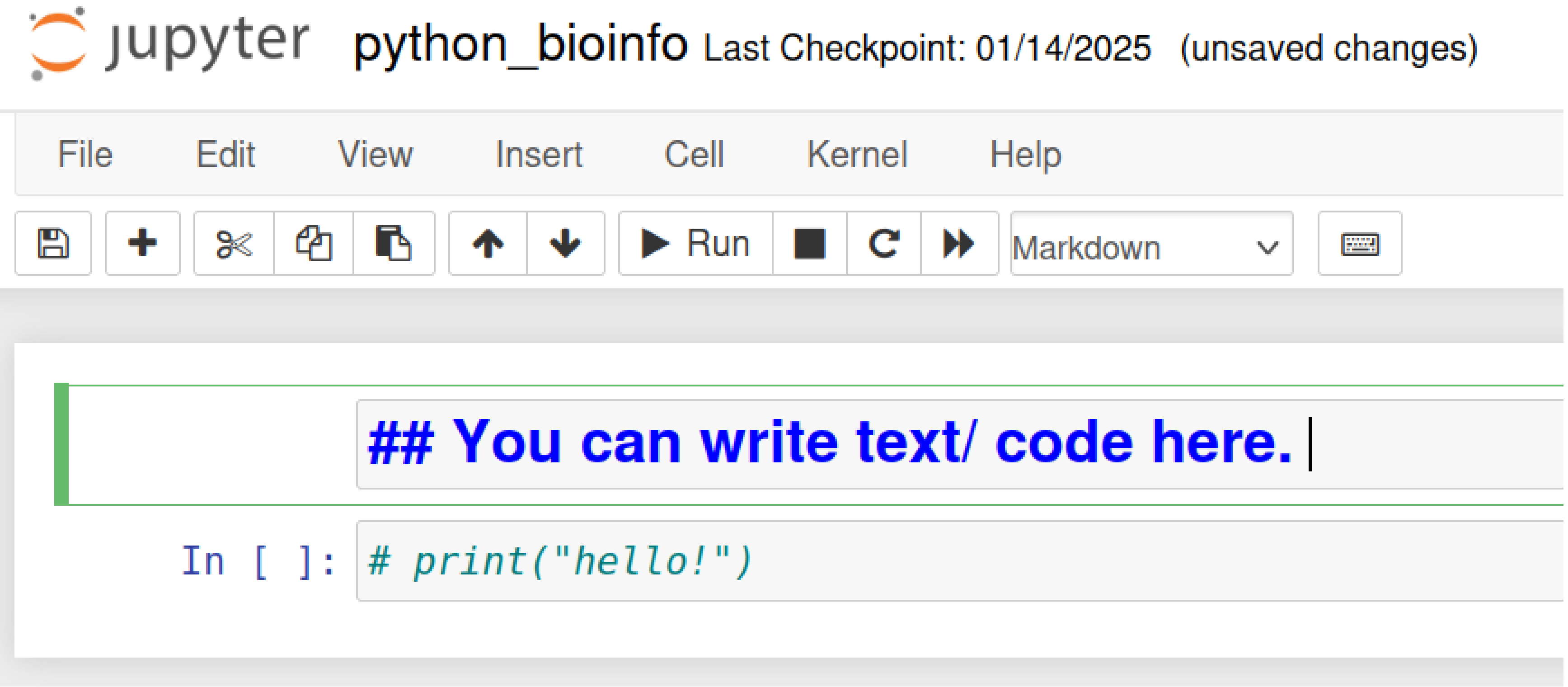
PyCharm

The Python IDE for data and web professionals

Supercharge your development with an AI-enhanced IDE experience

Jupyter Notebook

Allows creating and sharing interactive notebooks. Very popular.



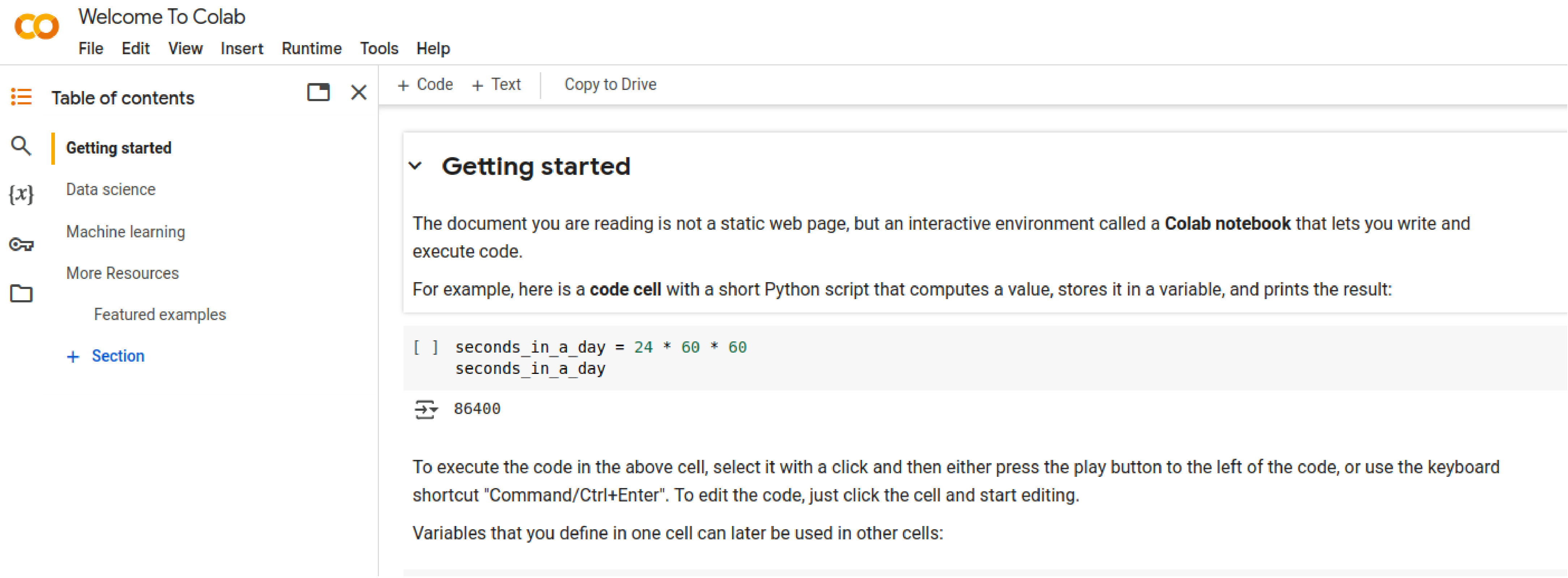
python_bioinfo Last Checkpoint: 01/14/2025 (unsaved changes)

```
## You can write text/ code here.
```

```
In [ ]: # print("hello!")
```

Google Colab

Hosted Jupyter notebook service that provides free access to GPUs and TPUs.



Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
    seconds_in_a_day
```

```
86400
```

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

Variables that you define in one cell can later be used in other cells:

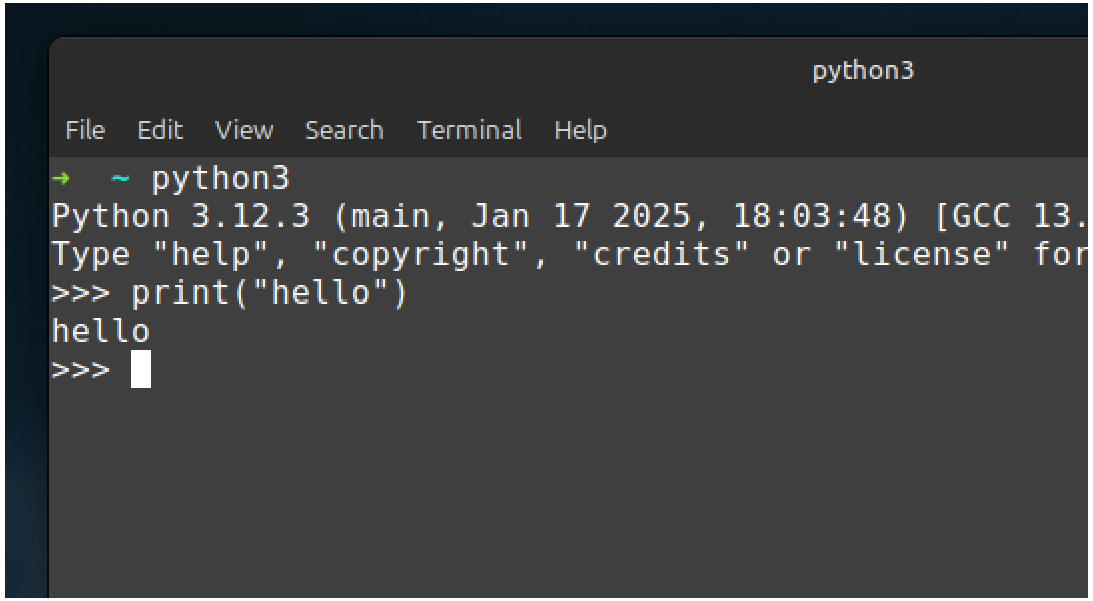
Choosing an IDE

Jupyter notebook is convenient for sharing and comes bundled with the Anaconda distribution, IDLE is simple and takes very little space.

First Python Program!

print(“hello”)

This prints hello as the output.



```
→  ~ python3
Python 3.12.3 (main, Jan 17 2025, 18:03:48) [GCC 13.
Type "help", "copyright", "credits" or "license" for
>>> print("hello")
hello
>>> 
```

Python as a calculator!

Python supports arithmetic operations like addition, subtraction, multiplication and division.

```
>> 2 + 3
```

Output: 5

Arithmetic operators: +, - , * , /

Logical Operators

Less than (<), greater than (>), less than or equal to (<=), greater than or equal to (>=)

```
>> 2 < 3
Output: True (2 is less than 5)
```

```
>> 5 >= 2
Output: True (5 is greater than or equal to 2)
```

Basic data types

int : integer values like 0, 1, 2, 3, … n

float: decimal values like 2.367, 3.1416, 9.1, etc.

string: text values like “apple”, “ACCGATTCCGA”, “protein A is …”

boolean: logical values like *True* or *False*

Notice how *strings* are enclosed in ‘single’ or “double” quotes but *integers*, *floats*, and *boolean* values aren’t.

Identify the data type:

a. 233

b. “1AKI: Lysozyme crystal structure”

c. 34.21

d. False

Assigning values to variables

We can store values inside *variables* by using the assignment operator (=), and print the stored values using *print*.

```
>> sequence_1 = “ATAAAGGCGAGGGAGGAGCCTTTTAA”
>> print(sequence_1)
Output: “ATAAAGGCGAGGGAGGAGCCTTTTAA”
```

```
>> acc_number = 12302
>> print(acc_number)
Output: 12302
```

List

List is a collection of elements (of any type). They are enclosed by big brackets [ ] .

```
>> my_bag = [1, “laptop”, “notebook”, 650, True]
>> print(my_bag)
Output: [1, “laptop”, “notebook”, 650, True]
```