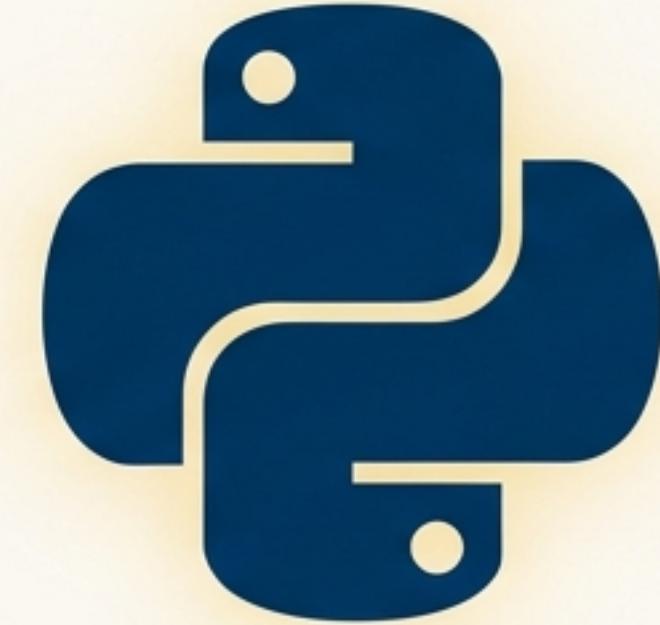
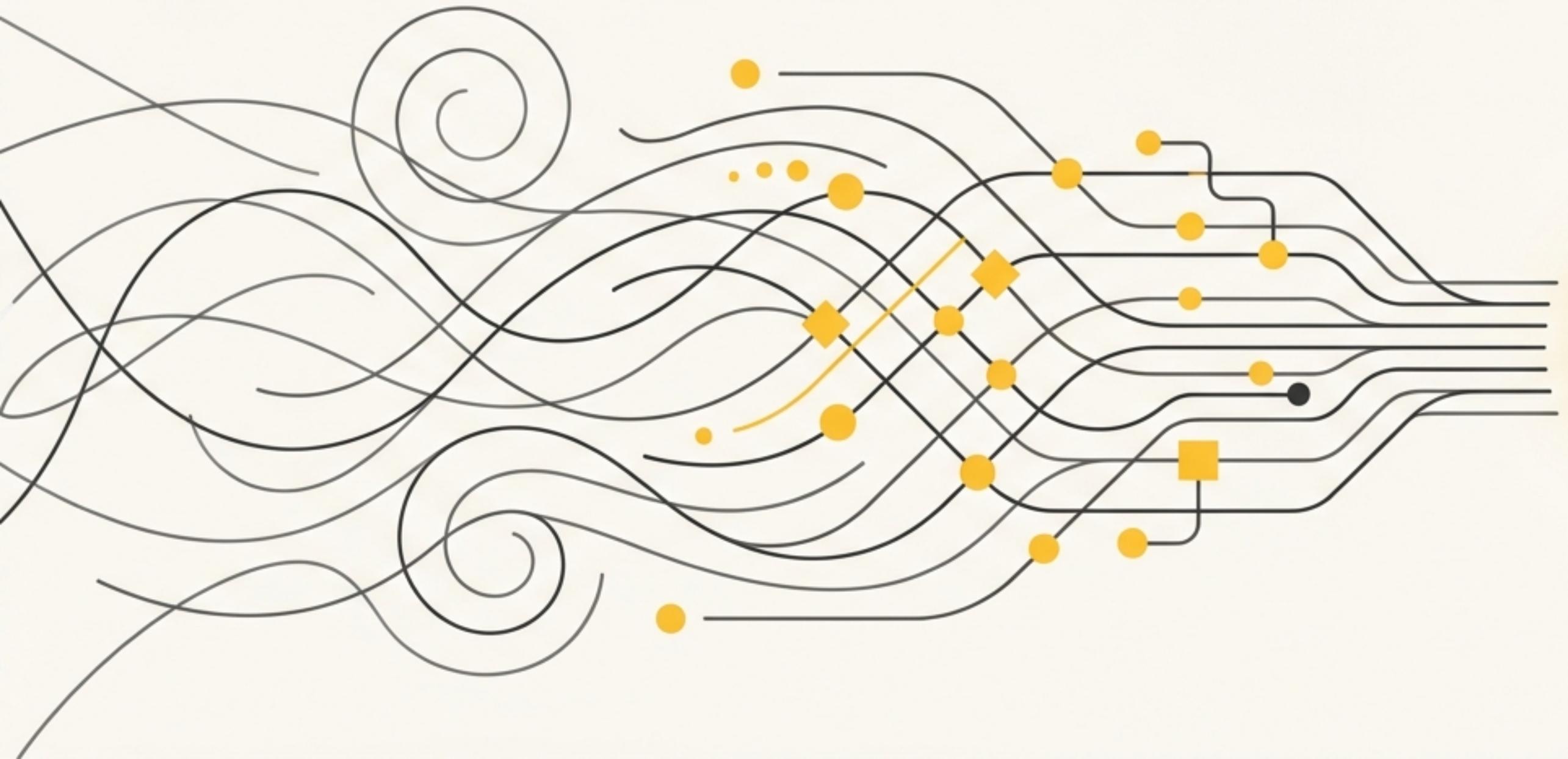


Python: From Idea to Impact

Your Journey into Programming Begins Here



Python is the language behind the tools you use every day.



Web Applications & APIs



Data Science & Machine Learning



Automation & Scripting



Cybersecurity Tools



Desktop & Mobile Applications



IoT & Game Development

A single language unlocks countless possibilities in technology.

Python was designed for humans to read and write.

Python is a high-level, general-purpose programming language.

High-Level

You focus on your ideas, not on complex computer memory management.

Interpreted

Code runs line-by-line, which makes finding and fixing errors much faster.

Dynamically Typed

Offers more flexibility, as you don't have to declare a variable's type before you use it.



Simple, Powerful, and Versatile by Design



Readable Syntax

Its structure is clean and reads almost like plain English.



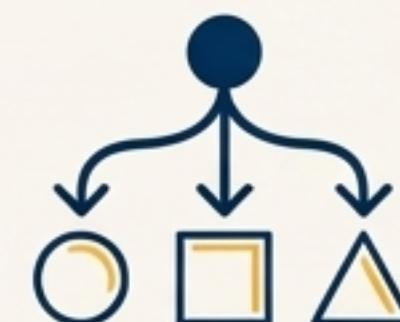
Large Standard Library

Comes with a huge collection of pre-built tools and modules for nearly any task.



Cross-Platform

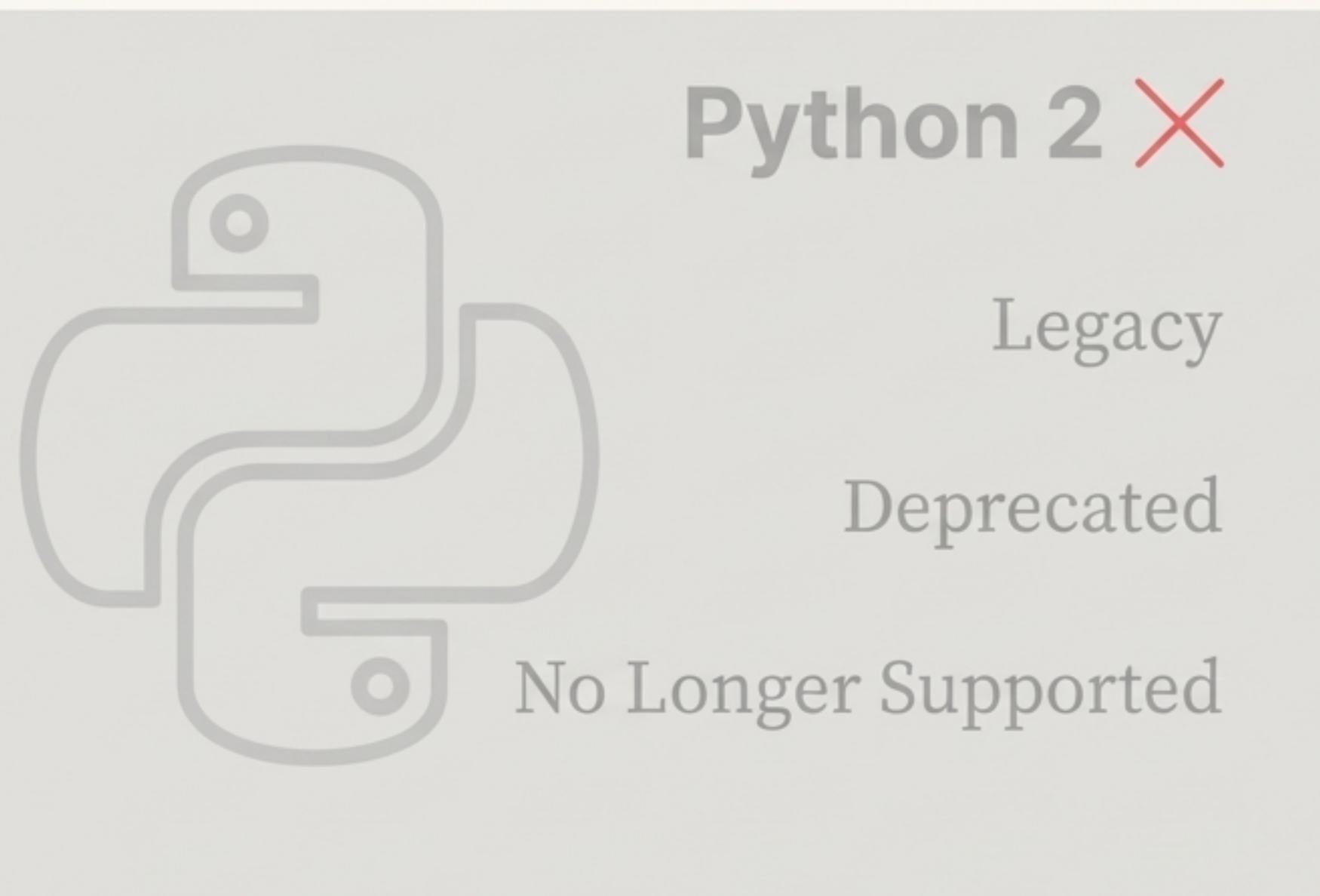
Write your code once and run it anywhere—Windows, macOS, or Linux.



Multi-Paradigm Support

It flexibly supports different coding styles, including object-oriented, functional, and procedural programming.

We will use the modern standard: Python 3.

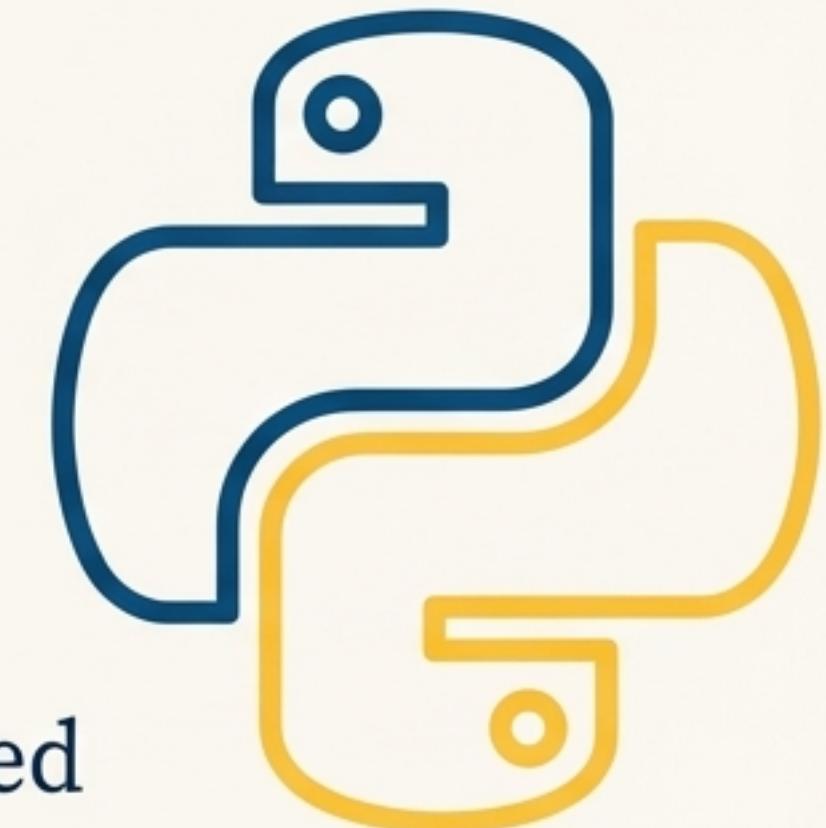


Python 3 ✓

Current

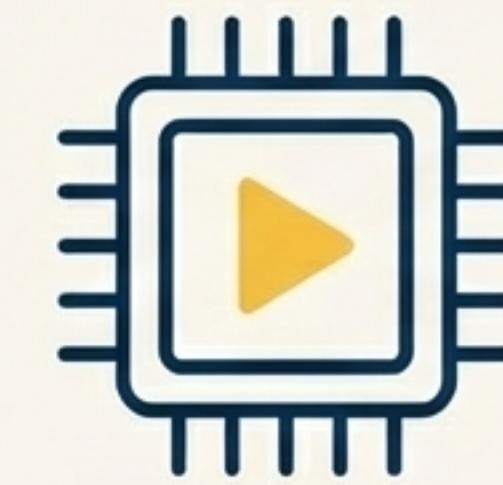
Recommended

Actively Developed



All examples and concepts in this guide use Python 3.

How your script becomes a running program.



your_code.py

Source Code

Python Interpreter

Compiles to Bytecode

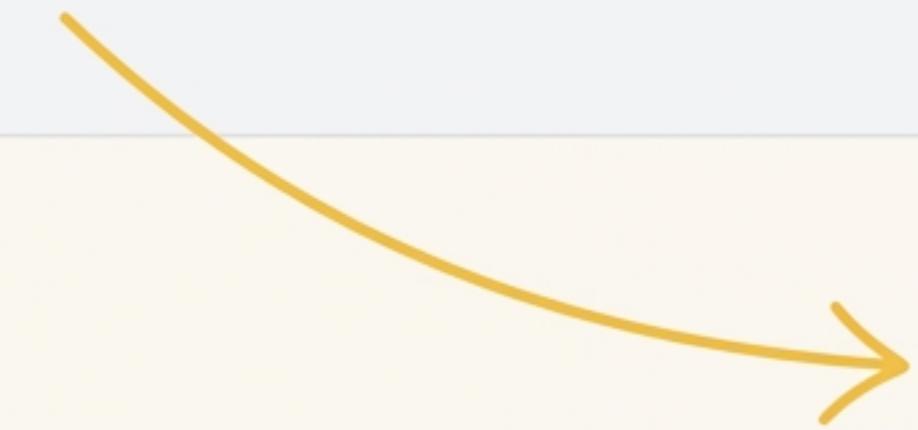
**Python Virtual
Machine (PVM)**

Executes Bytecode

The interpreter acts as a translator, converting your human-readable code into instructions the machine's PVM can understand and execute.

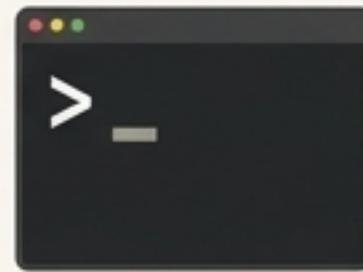
The journey begins with a single line of code.

```
# The '#' symbol starts a comment, which is ignored by Python.  
# This line of code prints a message to the screen.  
print("Hello, World")
```



This is a built-in Python function
that displays output.

Where to write and run your Python code.



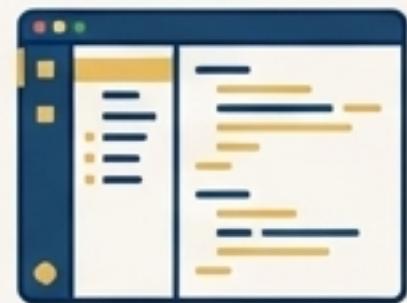
Terminal/Command Line

Use `python filename.py` to run script files you've saved.



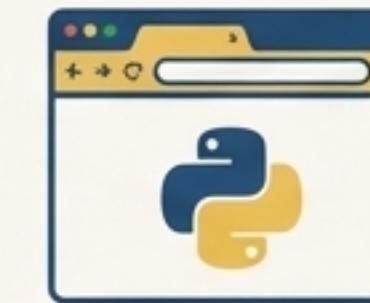
Interactive Shell (REPL)

Type `python` in your terminal to test code line-by-line.



Integrated Development Environments (IDEs)

Tools like PyCharm or VS Code provide a complete environment for larger projects.



Online Interpreters

Websites like [Replit](#) allow you to write and run code directly in your browser with no setup.

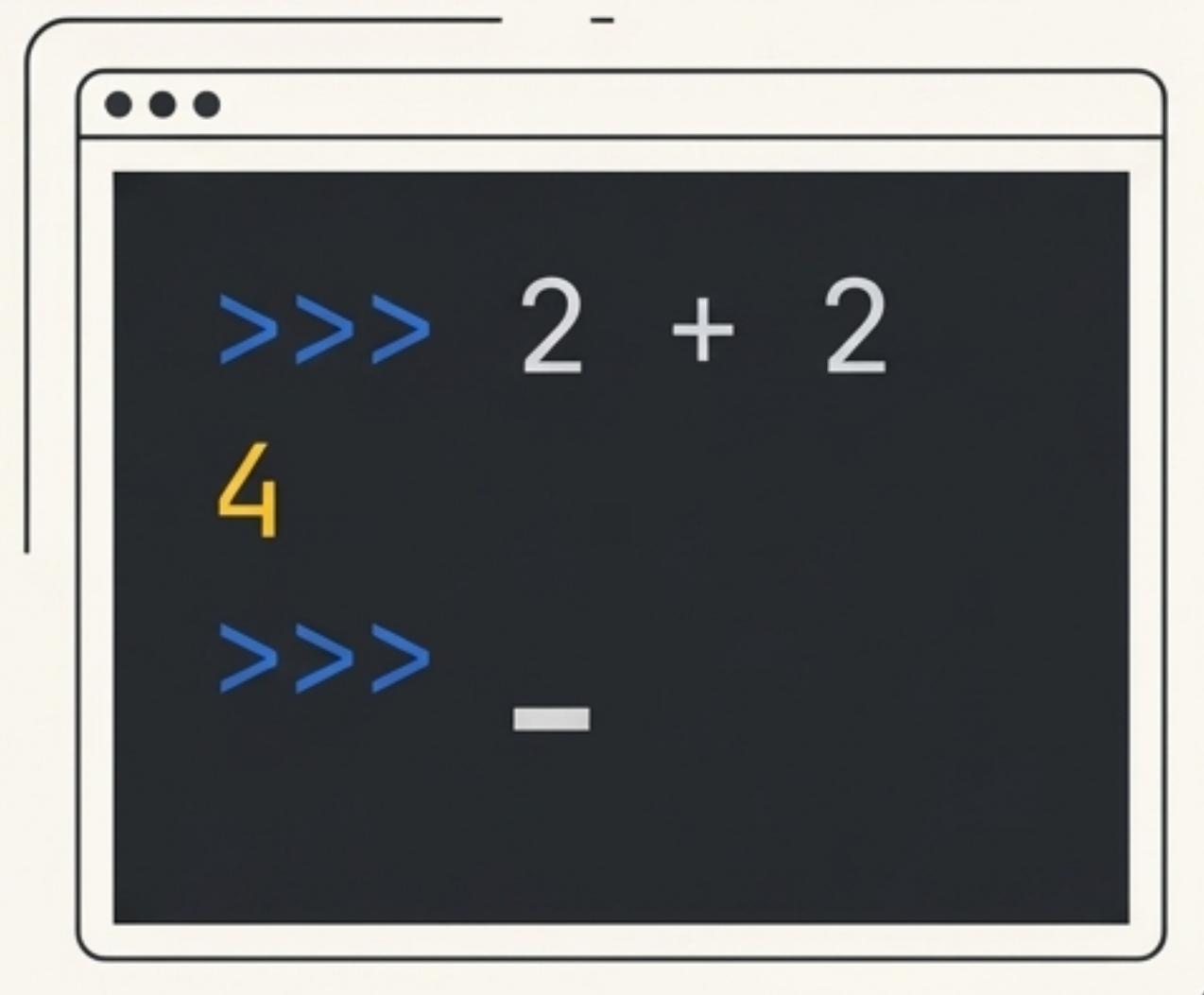
The REPL: An interactive conversation with Python.

R Read: It reads the line of code you type and press Enter.

E Eval: It evaluates (or executes) that line of code.

P Print: It prints the result of the evaluation back to you.

L Loop: It loops back and waits for your next command.



```
>>> 2 + 2
4
>>>
```

The rules for naming things: Identifiers.

Identifiers are the names you give to variables, functions, and classes.

DO (Valid Identifiers)

- Start with a letter (a-z, A-Z) or an underscore (_).
- Follow with letters, numbers (0-9), or underscores. (e.g., `user_name`, `score1`)
- Use descriptive names (`total_sales` is better than `ts`).
- Remember names are case-sensitive (`name` is different from `Name`).

DON'T (Invalid Identifiers)

- Start with a number (e.g., `1st_place`).
- Use special characters like !, @, #, \$, %.
- Use Python's reserved keywords.

The words Python reserves for itself: **Keywords**.

Keywords are the core vocabulary of the Python language. They have special meanings and cannot be used as identifiers for your variables or functions.

A large, stylized word cloud where the size of each word corresponds to its frequency or importance. The most prominent words are 'if', 'else', 'for', 'while', 'def', 'class', and 'return'. Other visible words include 'import', 'from', 'True', 'False', 'and', 'or', 'not', 'lambda', 'yield', 'pass', 'break', 'try', 'try', 'except', 'with', 'continue', and 'None'.

import from True
and if else False
is or not in try try
lambda for while except
yield for while except
pass def class with
break return continue
None

Leaving notes in your code with comments.

The Python interpreter ignores comments. They are for you and other programmers to understand what the code does.

Single-Line Comments

```
# This line calculates the user's  
age in days.  
  
age_in_days = 30 * 365
```

Multi-Line Comments (Docstrings)

```
"""  
This is a multi-line comment,  
often used at the beginning of a  
function or file to provide a  
detailed explanation of its  
purpose.  
"""
```

What we've accomplished in this chapter.

WHY PYTHON?

It's a versatile language used to build everything from websites and AI models to automation scripts.

WHAT IS PYTHON?

A simple, readable, and powerful interpreted language with a massive library of tools.

HOW IT WORKS?

You write code, the interpreter runs it, and you can start by following basic rules for identifiers, keywords, and comments.

Now, it's your turn to code.

Open a terminal, Python's interactive shell, or an online interpreter. Type this command, but with your own name, and press Enter:

```
print("Hello from [Your Name]!")
```

Congratulations. You are now a Python programmer.

The Journey Continues

Coming Up Next: Variables and Data Types

In the next chapter, we will learn how our programs can store, manage, and work with different kinds of information.

