

Getting started

Python Reference

Your first program

Python runs your code one line at a time. Each line is a statement . A program is just a list of statements that run from top to bottom.

The `print()` function shows text on the screen. This is called output . Text inside quotes is a string .

```
print("Hello, world!")
print("I am learning Python")
```

- Each `print()` starts a new line.
- Quotes can be "double" or 'single' — both make a string.
- A program does nothing until you run it.

Your first program: write → run → see



Exam tip: `print` shows a value; quotes mark a string; indentation is part of the syntax

Source code runs through the interpreter to produce output

Comments & code style

A comment starts with `#`. Python ignores everything after the `#` on that line. Comments explain your code to people; they do not change what the code does.

```
# This line is a note for humans
print("Hi")           # you can also comment at the end of a line
```

Good style makes code easy to read:

- Use clear names that say what a value means.
- Put one statement on each line.
- Do not add spaces at the start of a normal line. In Python, spacing at the start (indentation) has a special meaning, so a wrong space gives an error .

Input, process, output

Many programs follow a simple plan: **input** → **process** → **output**. You get some data, do something with it, then show a result.

The `input()` function reads text that the user types. It always gives back a string.

```
name = input("What is your name? ")
print("Hello, " + name)
```

- `input()` waits for the user to type and press Enter.
- Store the typed text in a variable so you can use it later.
- Because `input()` returns a string, change it with `int(...)` first if you need a number.

Common mistakes

- Forgetting the quotes: `print(Hello)` looks for a variable named `Hello` and raises a `NameError`. Text needs quotes: `print("Hello")`.
- A stray space at the **start** of a line: Python reads indentation as structure, so it raises an `IndentationError`.
- Assuming `input()` returns a number. It always returns a string, so wrap it in `int(...)` before doing any maths.