Your First Program: Print, and Running Code

Python for Absolute Beginners · Lesson 3

What you'll learn

- What the IDLE shell is for, and what a program file is for
- How to print messages with print
- How to write a comment with #
- How to save a program file and run it in IDLE
- The two most common first-day mistakes

Two ways to run Python

- The shell is for trying one line at a time: type a line, press Enter, see the result at once — perfect for experiments
- A program file is for code you want to keep: write several lines, save them in a file ending in .py, and run the whole file
- Start in the shell

In the shell: your first line

```
print("Hello, world!")
```

```
Output:
Hello, world!
```

print means “show this on the screen.” The words inside the quotes are called a string — text, as opposed to a number.

The first common mistake

```
                        print(Hello)

Error:
NameError: name 'Hello' is not defined
```

Without quotes, Python thinks Hello is a name, not text. The error means: you used the name Hello, but I have no value stored under that name. Text needs quotes; names do not.

Comments: notes that do not run

- A comment is a note to yourself, or to anyone reading your code, that Python ignores
- It starts with a # sign
- Use comments to explain why you wrote something, not what it does — the code itself already says what it does

A comment in code

```
    # This line is a note. Python skips it completely.
print("Comments help future you remember why you wrote
this.")

Output:
Comments help future you remember why you wrote this.
```

Only the print line runs.

Your first program file

- Shell experiments vanish when you close IDLE — real programs live in files
- In IDLE, choose File → New File; a blank editor window opens
- Type your program, then choose File → Save, and save it as hello.py — the .py ending tells the computer this is a Python program
- Choose Run → Run Module, or press F5; IDLE runs the whole file, and the output appears in the shell
- If IDLE asks you to save before running, that is normal: it always runs the saved version of your file

hello.py

```
                        # My first program
print("Hello, world!")
print("My name is Sam.")
print(2 + 3)

Output:
Hello, world!
My name is Sam.
5
```

Notice the third line: 2 + 3 has no quotes, because it is arithmetic, not text. Python works it out and prints 5.



Text in quotes, numbers bare

- Quoted text is printed as-is; unquoted arithmetic is calculated first
- That distinction — text in quotes, numbers bare — is one you will use in every program you ever write

The two most common first-day mistakes

- Forgetting the quotes: print(Hello) gives a NameError, because Python reads Hello as a name, not text
- Running before saving: if you change a file and run it without saving, IDLE runs the old saved version, and your change seems to have no effect
- Save, then run, every time

Hands-on exercise

- In the shell, print your own name and the result of 10 * 5
- Create a file called first.py that prints three lines: a greeting, your name, and one arithmetic result
- Run it with Run → Run Module (F5) and confirm all three lines appear
- Add a comment at the top saying what the program does, save, and run it again to confirm the output is unchanged

Key takeaways

- The shell is for experiments; files are for programs you keep
- print shows output; text goes in quotes, numbers do not
- # starts a comment, which Python skips
- Save as .py, then run with F5 — save first, every time
- A NameError usually means you forgot quotes around text

Take the quiz

Take the Lesson 3 quiz — 5 quick questions to check what you learned.

Next lesson

Variables: Storing Things