What's Next: Stepping into Python 3 Programming

Python for Absolute Beginners · Lesson 17

What you'll learn

- Everything you can now do, collected in one place
- How this course connects to Python 3 Programming
- Which of its lessons will feel familiar, and which will be new
- How to keep learning after this course ends

What you can do now

- Run Python and use IDLE (Lessons 2–3): install, open the shell, save and run program files
- Store values (Lesson 4): variables, naming rules, updating a value from its old self
- Do arithmetic (Lesson 5): integers, floats, division, remainders, powers, order of operations
- Handle text (Lesson 6): strings, quotes, joining, repeating, length, and positions
- Decide and repeat (Lessons 8–10): booleans, comparisons, if chains, for loops, and while loops

What you can do now, continued

- Draw with code (Lesson 11): Turtle shapes, colors, and loops you can see
- Organize data (Lessons 12–13): lists by position, dictionaries by label
- Reuse code (Lesson 15): functions with parameters and return values
- Build a complete program (Lesson 16): a guessing game with random numbers, player input, hints, and a score count

The recap program

```
name = "Coder"
score = 0
for round in range(3):
    score = score + 10
    print(name + " has " + str(score) + " points.")
if score >= 30:
    print("Finished!")

Coder has 10 points.
Coder has 20 points.
Coder has 30 points.
Finished!
```

Every line uses something from this course, and nothing else. If you can read it line by line, you have genuinely learned the foundations.

Stepping into Python 3 Programming

- This course was designed as the on-ramp to Python 3 Programming, and the step is deliberately smooth
- You already know its opening ground: setup, IDLE, printing, variables, numbers, strings, comparisons, decisions, and loops
- Its opening lessons will therefore move faster for you than for someone starting cold
- You will recognize the ideas and spend your energy on the new material instead of the setup

New ground ahead

- Files — reading from and writing to files, so programs can remember things between runs
- Object-oriented programming — bundling data and the code that works on it into objects of your own design
- Networking — programs that talk to other computers: fetching web pages, sending requests, running simple servers
- None of that requires anything you do not already have — the same thinking, applied to larger problems

How to keep learning

- Change working programs: take the guessing game and alter one thing at a time
- Read error messages: they point at the exact problem — the computer helping you, not punishing you
- Build small things often: a calculator, a quiz game, a to-do list, a Turtle drawing
- Small finished programs beat big unfinished ones, every time
- If you feel stuck, go back one lesson, not back to the start

Hands-on exercise

- Run the recap program above and explain each line out loud, in your own words
- Extend it: add a dictionary holding your name and score, and print a final message built from it
- Write down the first thing you want to build next — a game, a tool, or a drawing — and which lessons' ideas it will need
- Open the Python 3 Programming course and read its first lesson; mark every idea you already recognize

Key takeaways

- You now command the foundations: setup, variables, numbers, strings, decisions, loops, Turtle, lists, dictionaries, functions, and a complete game.
- Python 3 Programming builds directly on this course, with the same IDLE toolchain — its opening lessons will feel familiar.
- New ground there: files, object-oriented programming, and networking.
- Keep learning by changing working programs, reading error messages, and building small things often.
- You started with zero experience and finished with a real program. That is what learning to code looks like.

Take the quiz

Take the Lesson 17 quiz — 5 quick questions to check what you learned.

Course complete

Course complete — continue into Python 3 Programming, where files, objects, and networking build on everything you learned here. See you there!