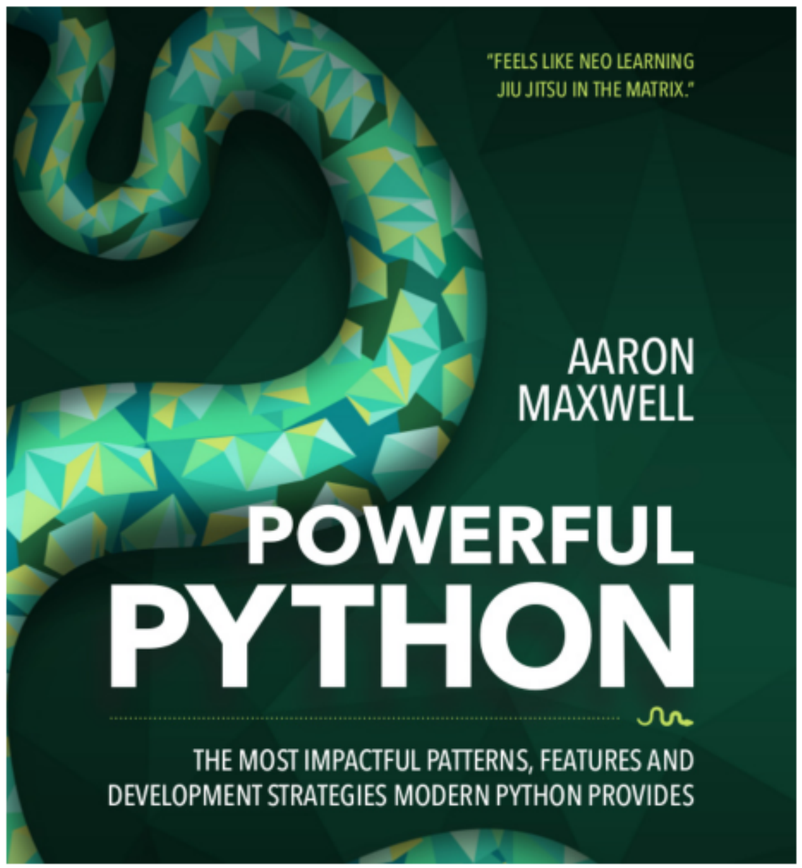
Three Keys To Advanced Python

Welcome

I'm your host, Aaron Maxwell.

- Author of **Powerful Python**
- @powerfulpython on twitter



Our focus in this course: The three most important topics to focus on, as you go "beyond the basics" of Python.

Key #1: Object-Oriented Programming

Object-oriented programming is the foundation.

EVERYTHING for intermediate and advanced Python builds on OOP.

Even if you have written classes in Python before, there are deeper, more powerful principles for you to discover.

Key #2: Writing Unit Tests

Writing automated tests is a **superpower**.

It is a key skill that separates average developers from the *very best* on the face of the earth.

Gaining this skill is **transformative**.

Do not underestimate the impact this will have on your career.

Key #3: Scalable Generators

A rich and important feature of modern Python: the **generator**.

Among other benefits, this is a valuable tool for making your Python code **scalable**...

So it can *gracefully* handle increasing amounts of data, that would grind other programs to a halt.

This will only become MORE important over time.

How we will proceed

Download courseware ZIP:
Courseware-3KEYS.zip

What's included:

- Slides
- Text files
- Labs (i.e., programming exercises - more on that later)

What makes perfect?

Practice, practice, practice.

To give you the ABILITY to do useful, valuable things you could not do before.

- Practice syntax (typing things in)
- Practice programming (higher-level labs)

I expect you to do your part!

You **exponentially** get out of this what you put into it.

Running the labs

Labs are the main programming exercises. You are given a failing automated test; your job is to write Python code to make it pass.

Simply run it as a Python program, any way you like. (For example, "python3 helloworld.py")

Run unmodified first, so you can see the failure report.

When done, congratulate yourself! (Ideally, find someone to high-five.)

Lab Demo

Here’s how it works.

Solutions

You have solutions! Use them wisely, not foolishly:

- After you get the lab passing, compare it to the official solution. Is it different?
- Other than that, don't look at the solution if you can avoid it.
- If you need help on a lab, **peek** at the solution - just enough to make your light bulb go off!
- The more you do on your own, the more you will learn. Peek at the solution to get a hint when you really need it.

Lab: helloworld.py

Now it's your turn! Do your first lab now: `helloworld.py`

This is in the `labs` folder in your courseware.

Instructions are in `LABS.txt`. You'll know the tests pass when you see:

```
*** ALL TESTS PASS ***
Give someone a HIGH FIVE!
```