# When Things Go Wrong: Debugging

Jamie Saxon

Introduction to Programming for Public Policy

October 12, 2016

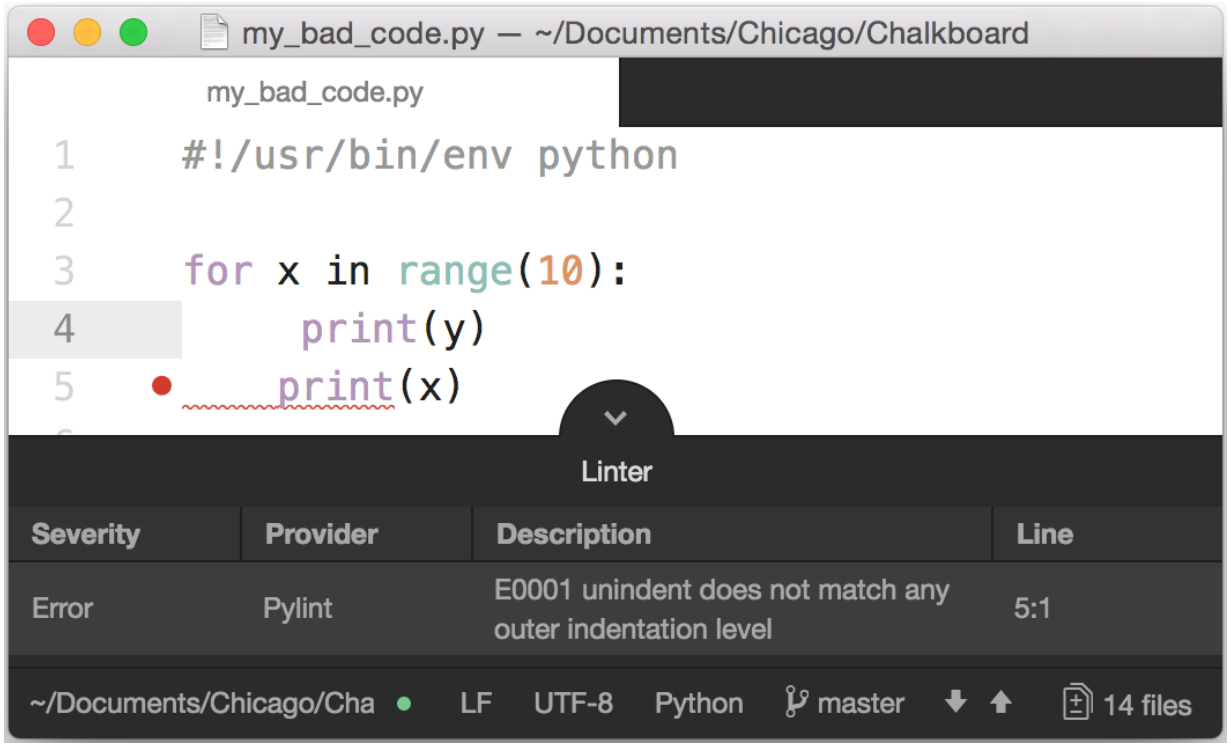
# Debugging

> Everyone knows that debugging is twice as hard as writing a program in the first place. So if you're as clever as you can be when you write it, how will you ever debug it?
>
> – Brian Kernighan

- Bugs are inevitable; debugging is hard.
- Let's talk briefly about how to deal with bugs.

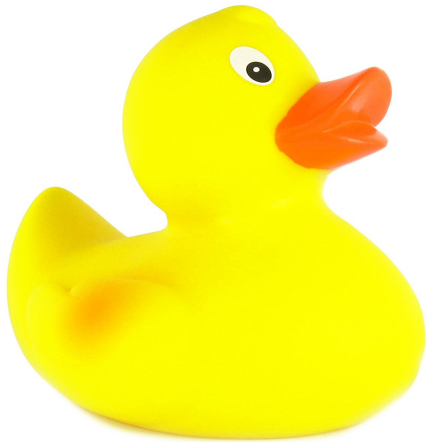
# Avoid Errors in the First Place

- You should have installed a "linter" in Atom. Pay attention to it!
- If you didn't, then revisit the instructions from HW1. [**PC**, **Mac**]



# Problems Running Code? Steps

1. **Read the error!** Python tells you the location of and code for syntactical errors. That may be enough.
2. If you don't understand the error, google it.
3. For semantic errors, add lots of `print` statements, near where you believe the code is failing, to understand the state of the program.
   - Or, possibly, use `pdb`: covered next.
4. Build the minimal piece of code that reproduces the bug.
   - If your 'minimal example' works, then build up from there.
5. Explain your code to a friend (or an inanimate object = duck).



# The Python Debugger

A fast way to see what the computer's actual state is to use pdb

```
python -m pdb jamie_spirograph.py
```

**n:** to go to next line of code

**l:** list source code for the current file (or `ll`).

**b:** set a breakpoint

**c:** continue debugging until you hit a breakpoint

**s:** step into a function

**p:** to print the value of an expression in the current context

# Ask for help – but ask well!

Many forums where you can get help, but they can be snarky.

- In this class, we have Canvas; in the real world, stack**overflow**.
- Of course – start by searching the forum (indexed on google).

Preparing a good question will often lead you to the answer.
Here are some common tips [1, 2, 3, 4]:

1. Be specific about what you wanted (what are you trying to do), expected, and got.
2. Provide your "minimal example," so others can reproduce your error.
3. List what you've already tried (demonstrate commitment).
4. Ask in public, in the right place. Tag it so people can find it.
5. Meaningful subject line – not 'python problem' or 'help!'

Good questions get answered; bad ones get **LMGTFY** (or RTFM).