Errors and Exception Handling

Complete Python Bootcamp

- Errors are bound to happen in your code!
- Especially when someone else ends up using it in an unexpected way.
- We can use error handling to attempt to plan for possible errors.

Complete Python Bootcamp

- For example, a user may try to write to a file that was only opened in **mode='r'**
- Currently if there is any type of error in your code, the entire script will stop.
- We can use Error Handling to let the script continue with other code, even if there is an error.

Complete Python Bootcamp

- We use three keywords for this:
  - **try:** This is the block of code to be attempted (may lead to an error)
  - **except:** Block of code will execute in case there is an error in **try** block
  - **finally:** A final block of code to be executed, regardless of an error.

Unit Testing

Complete Python Bootcamp

- As you begin to expand to larger multi-file projects it becomes important to have tests in place.
- This way as you make changes or update your code, you can run your test files to make sure previous code still runs as expected.

Complete Python Bootcamp

- There are several testing tools, we will focus on two:
  - **pylint:** This is a library that looks at your code and reports back possible issues.
  - **unittest:** This built-in library will allow to test your own programs and check you are getting desired outputs.

Complete Python Bootcamp

- We'll begin by showing you how to use pylint to check your code for possible errors and styling.
- Python as a set of style convention rules known as "PEP 8".

Complete Python Bootcamp

- Afterwards we will explore how to test our code with the built-in **unittest** library.

Complete Python Bootcamp

- For this lecture we will be creating .py scripts in sublime.
- You can still use the associated notebook for code using the %%writefile magic jupyter command.
- Let’s get started!

Unit Testing - Part 2

PIERIAN DATA