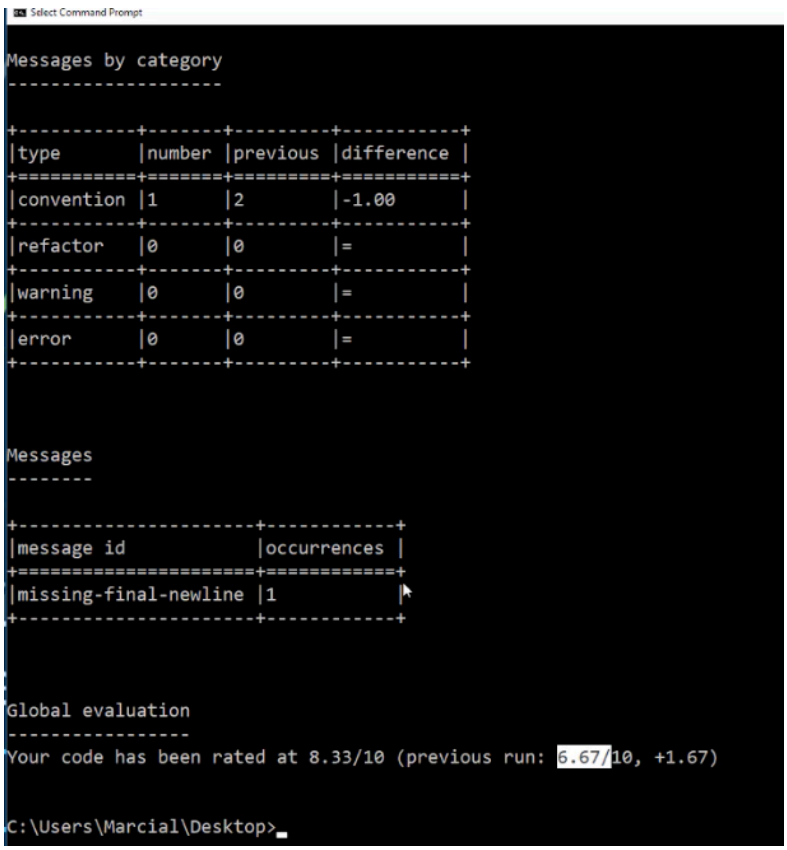
SECTION 10: ERRORS AND EXCEPTIONS HANDLING - 46 minutes, 6 parts

5/6 Pylint Overview

- **Unit testing**
  - -> working with multiple people
  - -> it is more important to have tests
  - -> testing tools
    - -> pylint <- this is a library which looks at the code and reports possible issues
    - -> unittest <- this is a library which lets us test our own programs and if we are getting the correct outputs or not
    - -> one which returns issues in the code, and then tests the programs which we write
  - -> PEP 8 <- Python convention rules
  - -> unittest <- how to test the code
  - -> we are creating .py scripts in sublime
  - -> we can use the associated notebook for code, with the %%writefile magic jupyter command

- **Tutorial**
  - -> in the terminal: pip install pylint
  - -> he's created a .py file
  - -> then deliberately written an error in the code <- print the value of a variable which doesn't exist
  - -> pylint file_name.py
  - -> running this in the terminal returns information about the code he's just written
  - -> report <- the number of classes, method, function
  - -> raw metrics <- how much code there is, docstring, comment, empty
  - -> duplication <- how many lines have been duplicated
  - -> messages <- what kind of messages rise up when we run the pylint report
  - -> in this case, one of the issues is the name of the variable which we have printed
  - -> the module is providing us metrics for the quality of the code



  - **-> he creates a new .py script**
    - -> he is using """ comments
    - -> then defining a function
    - -> in which there are more comments
    - **-> he is adding comments in to increase the pylint score for the .py file**
    - **-> this is a library which can be run in the terminal, to return information about the code**

  - **-> he makes changes to the code, and then runs .pylint in the terminal again**
    - -> we want to use spaces instead of tabs
    - -> it's like a spell checker in the terminal but for Python
    - -> it's returning errors in the code

- -> like the Python version of a CSS / HTML validator
- -> pylint will complain when we mix them together
- -> after he fixes the errors, he goes back and runs pylint again in the terminal
- -> in which case, the score of the code has increased
- -> this is for multiple users and larger pieces of code