Global Knowledge®

Idiomatic Python

Estimated time for completion: 45 minutes

Overview:

In this lab, you will take some poorly written code and correct it using the PEP8 guidelines and Pylint tooling. You will start with code from the object-oriented tic-tac-toe game which has been somewhat mangled to highlight additional errors and warnings. You will change this code to remove the Pylint warnings.

These instructions assume you have **Python 3** installed. You can use any OS you choose (OS X, Windows, or Linux).

Goals:

- Install Pylint
- Change code to conform to community standards
- Find and remove code errors and bugs

Part 1 – Install Pylint

Start by installing Pylint from PyPI with PIP. You can either use PyCharm's GUI tools or the command-line:

Command-line install:

```
$prompt> pip install Pylint
```

Note: You must be careful that you are running pip from Python3 not pip from Python2. You can check by typing **`pip3 --version`**

If you have the wrong version or pip isn't found, you might need to directly to the folder containing the correct version and run it there.

On **OS X** that is something like:

```
$prompt> cd /Library/Frameworks/Python.framework/Versions/3.4/bin/

$prompt> ./pip3 install Pylint
```

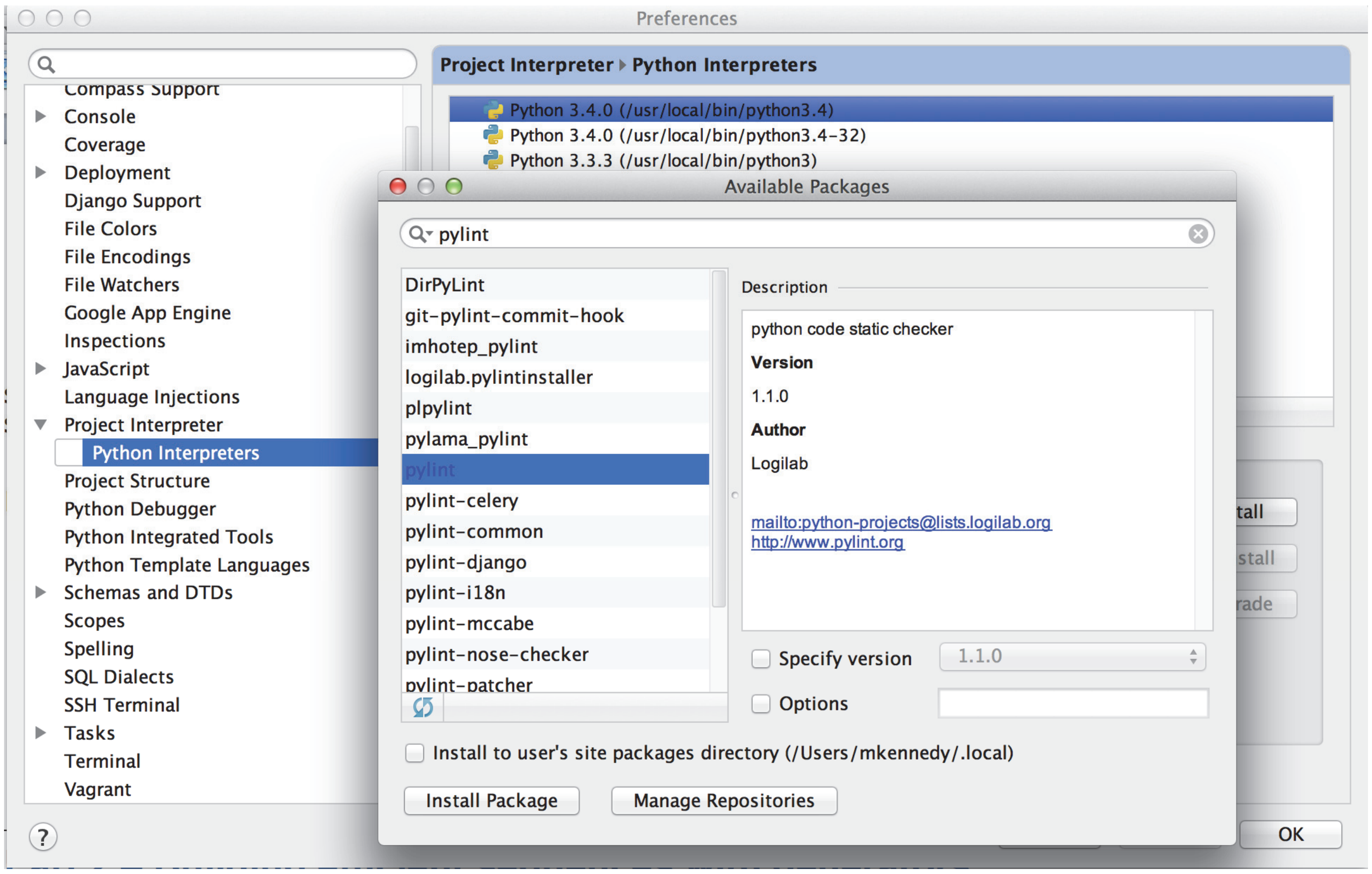
On **Windows**, this is something like:

```
$prompt> cd C:\python\Python34\Scripts\

$prompt> pip3 install Pylint
```

PyCharm install:

1. Open preferences
2. Go to Project interpreter > Python interpreters > select correct version
3. If Pylint is not installed, choose Install and search for Pylint and choose install.



python code static checker

Version

1.1.0

Author

Logilab

mailto:python-projects@lists.logilab.org
http://www.pylint.org

Specify version 1.1.0

Part 2 – Fixing coding issues and bugs

Open the files (or PyCharm project) in **Pythonic_Idiomatic_Python/Before/idiomatic_python_before**. You will go through the files, one by one, and analyze them with Pylint. Start with tic_tac_toe.py. Analyze it using the command:

```
pylint tic_tac_toe.py
```

You will see output somewhat like this:

```
No config file found, using default configuration

************* Module tic_tac_toe
```

```
C: 12, 0: Final newline missing (missing-final-newline)

C:  1, 0: Missing module docstring (missing-docstring)

C:  7, 0: Invalid function name "Main" (invalid-name)

C:  7, 0: Missing function docstring (missing-docstring)




Report

======

7 statements analysed.

...



Global evaluation

-----------------

Your code has been rated at 4.29/10
```

Look at the source file and fix these errors. After each change, rerun pylint to see the fix is accepted. You should be able to get a score of 10/10 on this file.

Next analyze **program.py**. This one most likely WILL CRASH. This is not your fault. Something is off with Pylint analyzing **game.py** which **program.py** depends upon. Open **program.py** and comment out the import game at the top.

Rerun Pylint and fix the errors (you cannot fix game undefined). You should be able to get a score of 8.84/10 on this file.

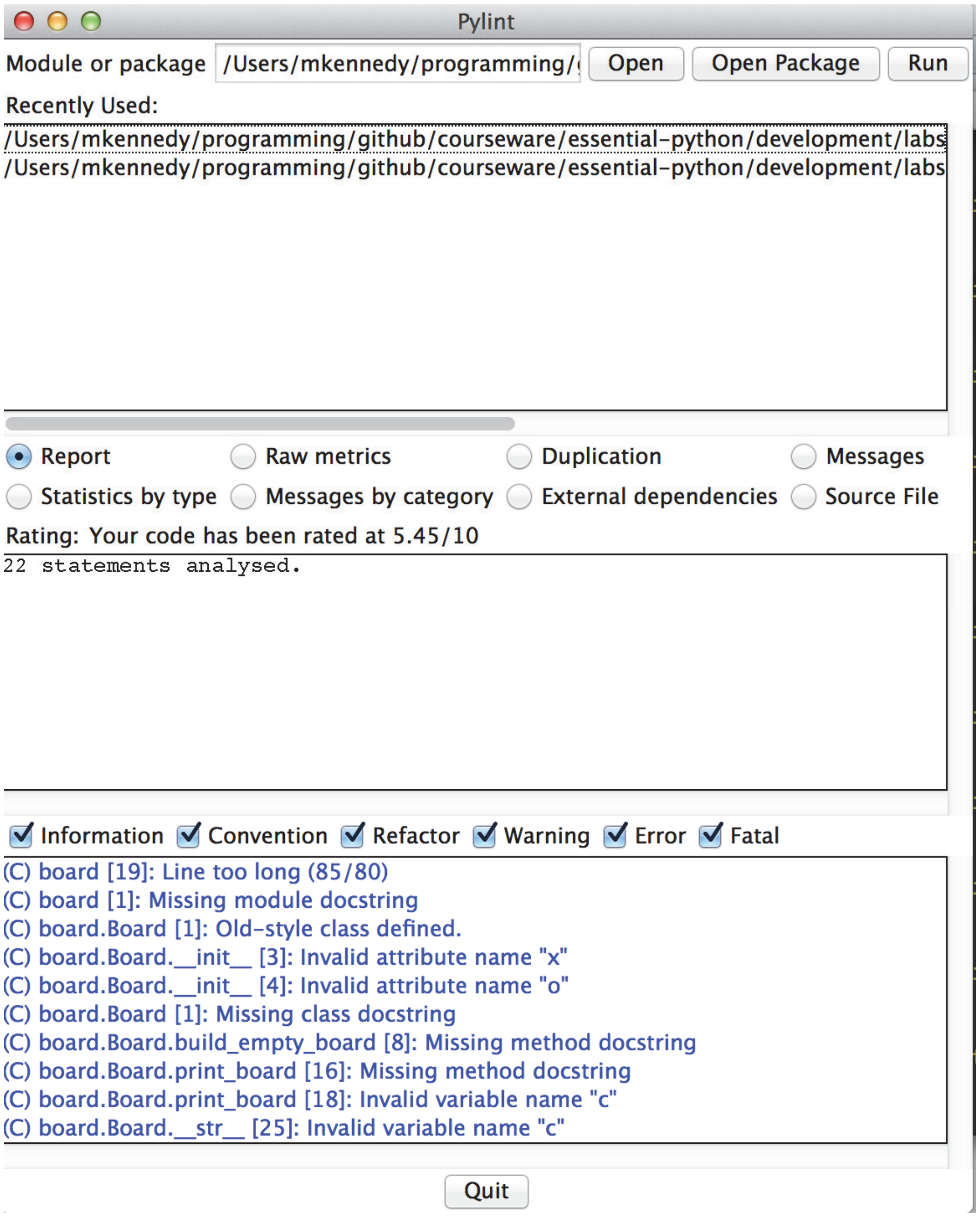
Now there is something wrong with Pylint and game.py (they don't get along) so skip that one. Analyze **board.py**. You should be able to get a score of 10.00/10 on this file.

The next part of the **generators.py** file works with an ordered tree class defined in **tree.py**. This is mostly done for you already but the iteration, which is supposed to return the items in order, is not done. We can navigate trees depth first with recursion, but sometimes a simple sequence is what you want (it is here).

Part 3 – Pylint GUI

Finally, you may want to see how the GUI for Pylint works. Just type pylint-gui on the command-line / terminal and you'll see a GUI appear. Don't get too excited, it is not a modern UI marvel, but it is a GUI. Try to run it on a few files and maybe reintroduce some errors to see the output.

Here is an example:



```
/Users/mkennedy/programming/github/courseware/essential-python/development/labs
/Users/mkennedy/programming/github/courseware/essential-python/development/labs
```

[● Report] Raw metrics Duplication Messages Statistics by type Messages by category External dependencies Source File

Rating: Your code has been rated at 5.45/10

```
22 statements analysed.
```

[✓ Information] [✓ Convention] [✓ Refactor] [✓ Warning] [✓ Error] [✓ Fatal]

```
(C) board [19]: Line too long (85/80)
(C) board [1]: Missing module docstring
(C) board.Board [1]: Old-style class defined.
(C) board.Board.__init__ [3]: Invalid attribute name "x"
(C) board.Board.__init__ [4]: Invalid attribute name "o"
(C) board.Board [1]: Missing class docstring
(C) board.Board.build_empty_board [8]: Missing method docstring
(C) board.Board.print_board [16]: Missing method docstring
(C) board.Board.print_board [18]: Invalid variable name "c"
(C) board.Board.__str__ [25]: Invalid variable name "c"
```