2. PyLint

Fig. 2.1 Photo by Timo Strüker on Unsplash

Note

Outline

2.1. Overview

1. PyLint is a popular Python static code analysis tool that helps you identify and fix issues in your Python code.
2. PyDev is a Python IDE for Eclipse that provides integration with various Python tools, including PyLint.
3. Here's a step-by-step tutorial on how to set up and use PyLint in the Eclipse IDE with the PyDev plugin.
4. Prerequisites

    a. Install Python: Make sure you have Python installed on your system.

    b. Install Eclipse: Download and install the Eclipse IDE for Python development.

    c. Install PyDev: Install the PyDev plugin for Eclipse using the Eclipse Marketplace or update sites.

2.1.1. Procedure

1. Step 1: Install PyLint

    a. You need to install PyLint globally or within your Python virtual environment.

    b. You can install it using pip:

```
pip install pylint
```

2. Step 2: Create a Python Project in Eclipse

    a. Open Eclipse.

    b. Go to File > New > PyDev Project.

    c. Enter a project name and choose your Python interpreter.

    d. Click "Finish."

3. Step 3: Configure PyLint in Eclipse

    a. Go to menu and choose [Window] >> [Preferences].

    b. Expand the [PyDev] section and select [Editor].

    c. Click on [Code Analysis] >> [PyLint].

    d. Check the [Use PyLint] option.

e. Check the [Redirect PyLint output to Console?] option. << Optional

f. Suggest change some minor options as shown as below.

i. set [FATAL Severity] option to [Error] level.

ii. set [ERRORS Severity] option to [Error] level.

iii. set [WARNING Severity] option to [Warning] level.

iv. set [CONVENTIONS Severity] option to [Warning] level.

v. set [REFACTOR Severity] option to [Warning] level.

vi. set [INFORMATIONAL Severity] option to [Error] level.

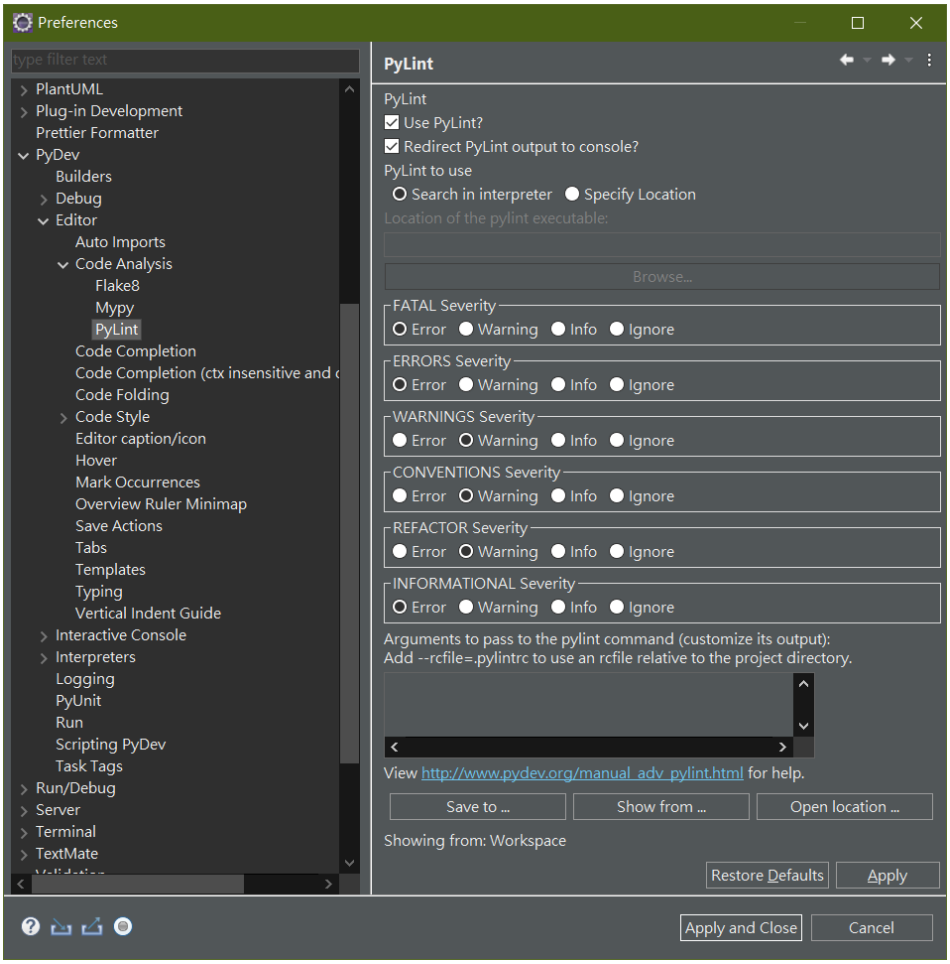


Fig. 2.1.1.1 PyLint: Eclipse: Preferences: PyLint Setup

4. Step 4: Configure Your Python Project

a. Right-click on your Python project in the “Project Explorer.”

b. Select [Properties].

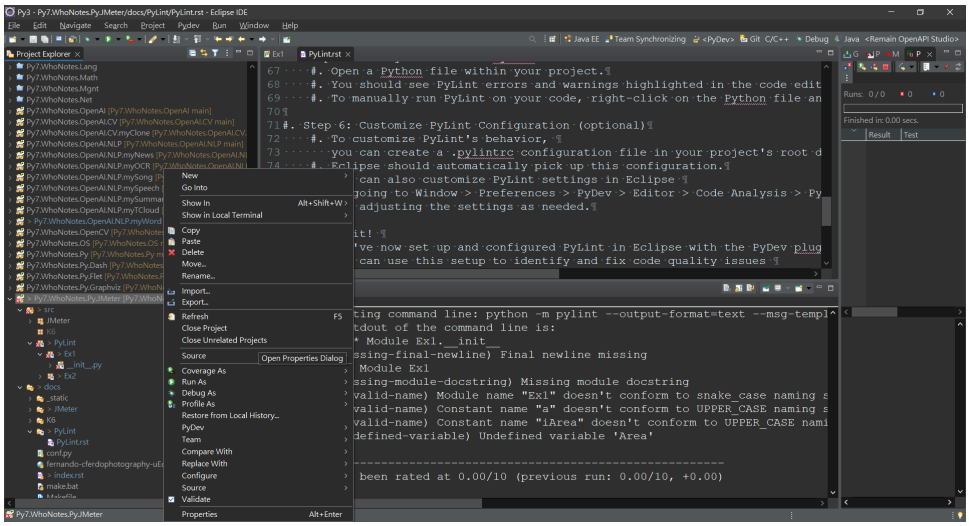


Fig. 2.1.1.2 PyLint: Eclipse: Project Properties

c. Under the [PyDev - PYTHONPATH] section, make sure your project's [Source Folders] is listed as a source folder.

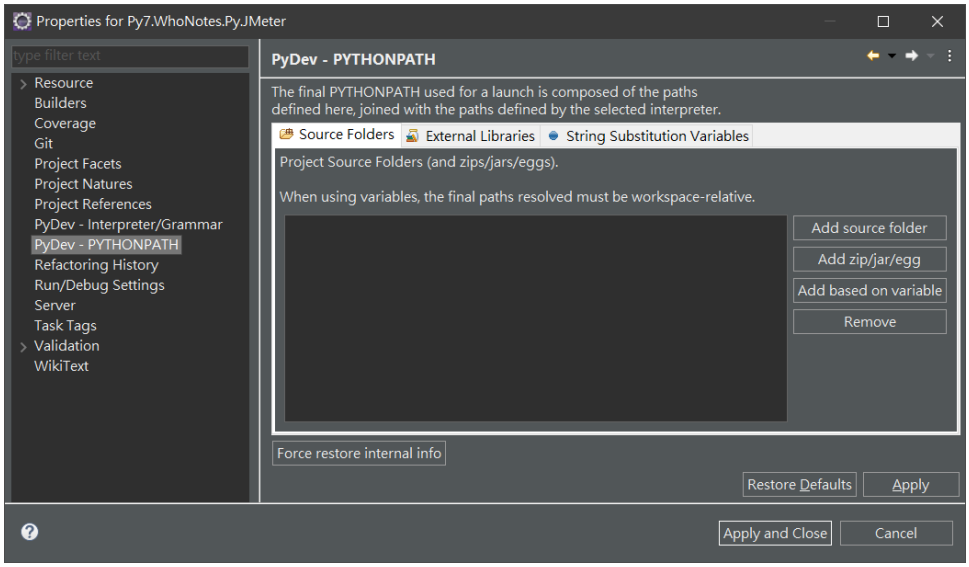


Fig. 2.1.1.3 PyLint: Eclipse: PyDev - PYTHONPATH

d. If NOT existed, please click [Add Source Folder] button.

e. Then, a [Selection Dialog] is shown and click [src] folder in the folder tree and [OK] button to return.

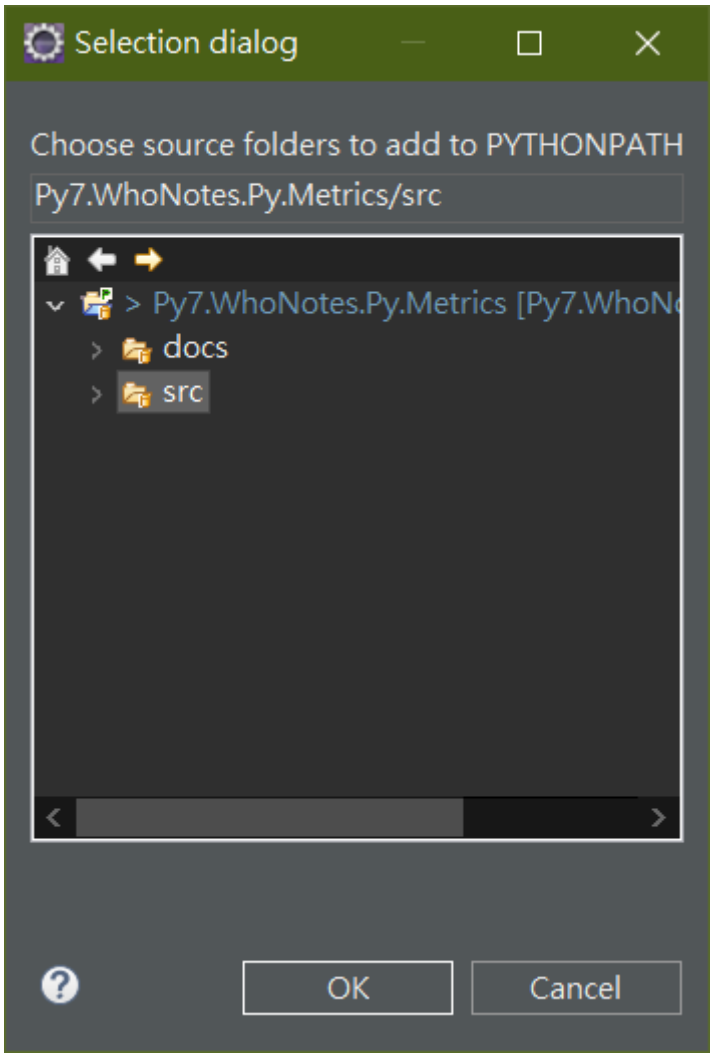
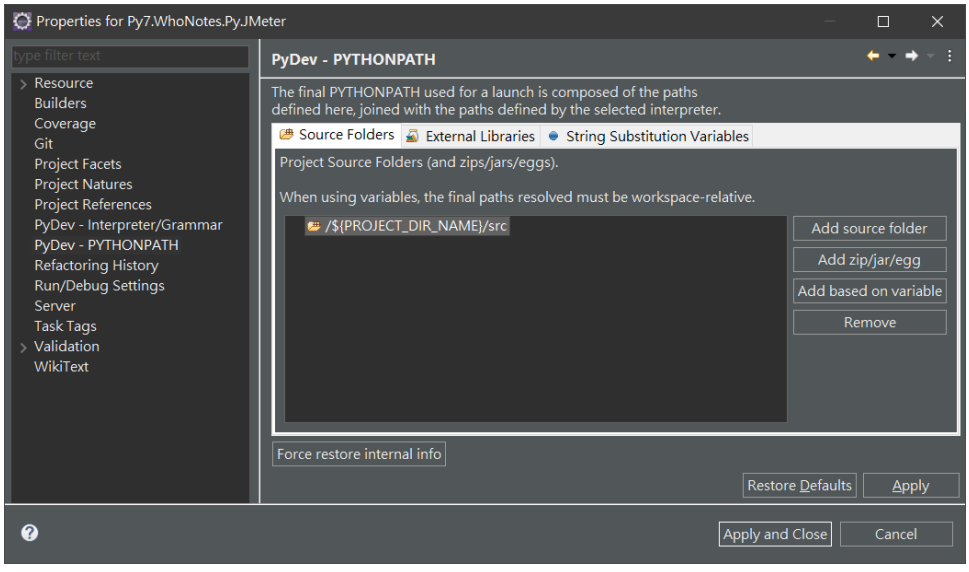


Fig. 2.1.1.4 PyLint: Eclipse: Selection Dialog

f. After return to the [PyDev - PYTHONPATH] section, we find a [/${PROJECT_DIR_NAME}/src] item is inserted in the [Source Folders] list as shown as below.



> *Fig. 2.1.1.5 PyLint: Eclipse: PyDev - PYTHONPATH: After*

g. Click [Apply and Close] to save your project settings.

h. Restart your Eclipse IDE now to enable the PyLint features.

5. Step 5: Run PyLint on Your Python Code

   a. Open a Python file within your project.

   b. You should see PyLint errors and warnings highlighted in the code editor.

   c. To manually run PyLint on your code, right-click on the Python file and select PyLint > Run PyLint.

6. Step 6: Customize PyLint Configuration (optional)

   a. To customize PyLint's behavior, you can create a .pylintrc configuration file in your project's root directory.

   b. Eclipse should automatically pick up this configuration.

   c. You can also customize PyLint settings in Eclipse by going to Window > Preferences > PyDev > Editor > Code Analysis > PyLint and adjusting the settings as needed.

7. That's it!

   a. You've now set up and configured PyLint in Eclipse with the PyDev plugin.

   #. You can use this setup to identify and fix code quality issues in your Python projects within the Eclipse IDE.

2.2. Ex1: Area

1. We demo a very easy example for calculating area.
2. Code+Output 1: Incorrect Codes

> **Code: Incorrect** | **Output: Console**
>
> a. Note that line 1-4 have at least 3 warnings.

Listing 2.2.1 /src/PyLint/ex1/__init__.py

```
1  a = 4
2  B = 5
```

```
3  iArea = a * a
4  print(Area)
```

3. Code+Output 2: Rector Codes

Code: Correct **Output: Console**

a. Moudle name use [snake_name_case.py] to use and follow snake_case.

b. Line 1-4: Add comment to resolve [missing-module-docstring] warning.

c. Line 5: Named constant [A] to use upper case naming.

d. Line 7: Named constant [AREA] to use upper case naming.

e. Line 8: Rename variable [AREA].

f. Line 9: Add a new line as the last line.

Listing 2.2.2 /src/PyLint/ex1/snake_name_case.py

```
1  '''
2  author: CPH
3  since:  20230901
4  '''
5  A = 4
6  B = 5
7  AREA = A * B
8  print(AREA)
```

2.3. Ex2: File: Write

1. We demo another easy example for writing a file.
2. Code+Output 1: Incorrect Codes

Code: Incorrect **Output: a.txt** **Output: Console**

a. Note that line 5 have at least 2 warnings.

Listing 2.3.1 /src/PyLint/ex2/__init__.py

```
'''
author: CPH
since:  20230901
'''
file = open('a.txt', 'w')
file.write('Hello, Python...')
```

3. Code+Output 2: Rector Codes

Code: Correct | **Output: a.txt** | **Output: Console**

a. Moudle name use [write_file.py] to use and follow snake_case.

b. Line 5: Use [with] statement.

c. Line 5: add [encoding='utf-8'].

d. Line 6: refactor it.

Listing 2.3.3 /src/PyLint/ex2/write_file.py

```
'''
author: CPH
since:  20230901
'''
with open('a.txt', 'w', encoding='utf-8') as f:
    f.write('Hello, Python...')
```

Note

1. Start: 20120311
2. System Environment:

Listing 2.3.5 requirements.txt

```
sphinx==7.2.5                           # Sphinx
graphviz>=0.20.1                        # Graphviz
sphinxcontrib-plantuml>=0.26            # PlantUML
sphinxcontrib.bibtex>=2.5.0             # Bibliography
sphinx-autorun>=1.1.1                   # ExecCode: pycon
sphinx-execute-code-python3>=0.3        # ExecCode
btd.sphinx.inheritance-diagram>=2.3.1   # Diagram
sphinx-copybutton>=0.5.2                # Copy button

#-- Theme #1 ------------------------------------------------
sphinxbootstrap4theme>=0.6.0            # Theme: Bootstrap4

#-- Theme #2 ------------------------------------------------
sphinx-material>=0.0.36                 # Theme: Material

#-- Theme #3 ------------------------------------------------
sphinx-immaterial>=0.11.6               # Theme: Immaterial:
Tabs
sphinx_code_tabs>=0.5.3                 # Tabs

#-- Theme #4 ------------------------------------------------
sphinx-revealjs==2.6.0                  # Theme: Reveal.js:
Slide Maker
sphinxcontrib-mermaid==0.9.2            # Slide Maker:
required

#-- Theme #5 ------------------------------------------------
sphinx_rtd_theme==1.3.0                 # bind with
sphinx_revealjs theme

#-------------------------------------------------------
#-- Minor Extension
#-------------------------------------------------------
sphinxcontrib.httpdomain>=1.8.1         # HTTP API
pydantic==2.3.0                         # Data Validation
sphinx-mdinclude>=0.5.3                 # Markdown Format
pandoc>=2.3                             # nbsphinx
IPython>=8.14.0                         # nbsphinx
ipykernel>=6.24.0                       # nbsphinx
nbsphinx>=0.9.2                         # nbsphinx
#sphinxcontrib-blockdiag>=3.0.0         # Diagram: block
#sphinxcontrib-actdiag>=3.0.0           # Diagram: activity
#sphinxcontrib-nwdiag>=2.0.0            # Diagram: network
#sphinxcontrib-seqdiag>=3.0.0           # Diagram: sequence

#-------------------------------------------------------
```

```
#-- Library Upgrade Error by Library Itself
#   >> It needs to fix by library owner
#   >> After fixed, we need to try it later
#------------------------------------------------------

#------------------------------------------------------
#-- Still Wait For Upgrading Version
#------------------------------------------------------

#------------------------------------------------------
#-- Still Under Testing
#------------------------------------------------------
#numpy>=1.24.2                          # Figure: numpy

#------------------------------------------------------
#-- NOT Workable
#------------------------------------------------------
#sphinxcontrib.jsdemo==0.1.4    # ExecCode: Need replace
add_js_file()
#jupyter-sphinx==0.4.0          # ExecCode: Need gcc compiler
#sphinxcontrib.slide==1.0.0     # Slide: Slideshare
#hieroglyph==2.1.0              # Slide: make slides
#matplotlib>=3.7.1              # Plot: Need Python >= v3.8
#manim==0.17.2                  # Diagram: scipy, numpy need
gcc
#sphinx_diagrams==0.4.0         # Diagram: Need GKE access
#sphinx-tabs>=3.4.1             # Tabs: Conflict w/ sphinx-
material
```

Was this page helpful?

☺ ☹