

Find Your Feature Fit

How to pick a text editor for Python programming

Gregory M. Kapfhammer
Madelyn M. Kapfhammer

PyOhio 2019

Hi! Glad to be Back at PyOhio!

GREGORY M. KAPFHAMMER



@GregKapfhammer



www.gregorykapfhammer.com

Hello! It's Great to Meet You!

MADELYN M. KAPFHAMMER



@MaddyKapfhammer



www.madelynkapfhammer.com

Why Care About a Text Editor?

Developer Productivity

Programming Fun

Effective Code

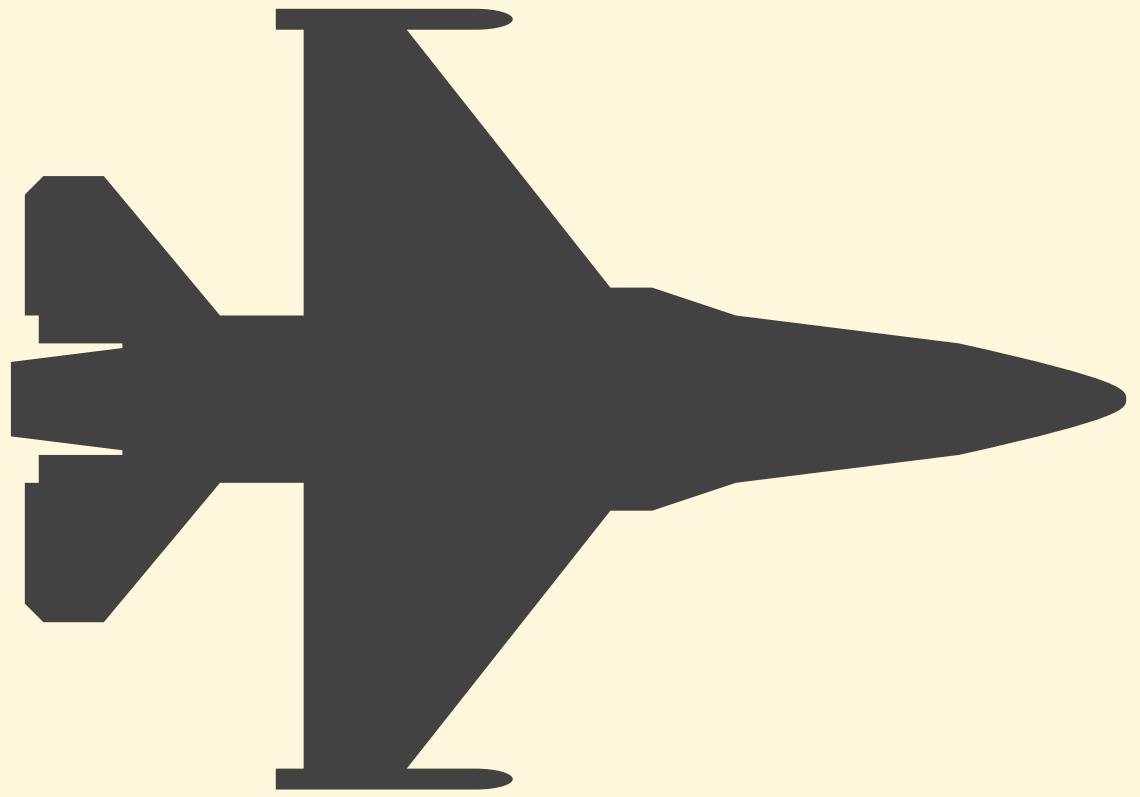
VS CODE



VIM



A
FAST
PACE
FIELD



Key Goals

- Showcase noteworthy features
- Compare and contrast text editors
- Highlight less available information
- Share the thrill of customization

Evaluation Metrics

Ease of ...

Installation

Use

Customization

Learn Key Editor Features

Learn the basics before you copy
confusing configurations

Install Useful Text Editor Plugins

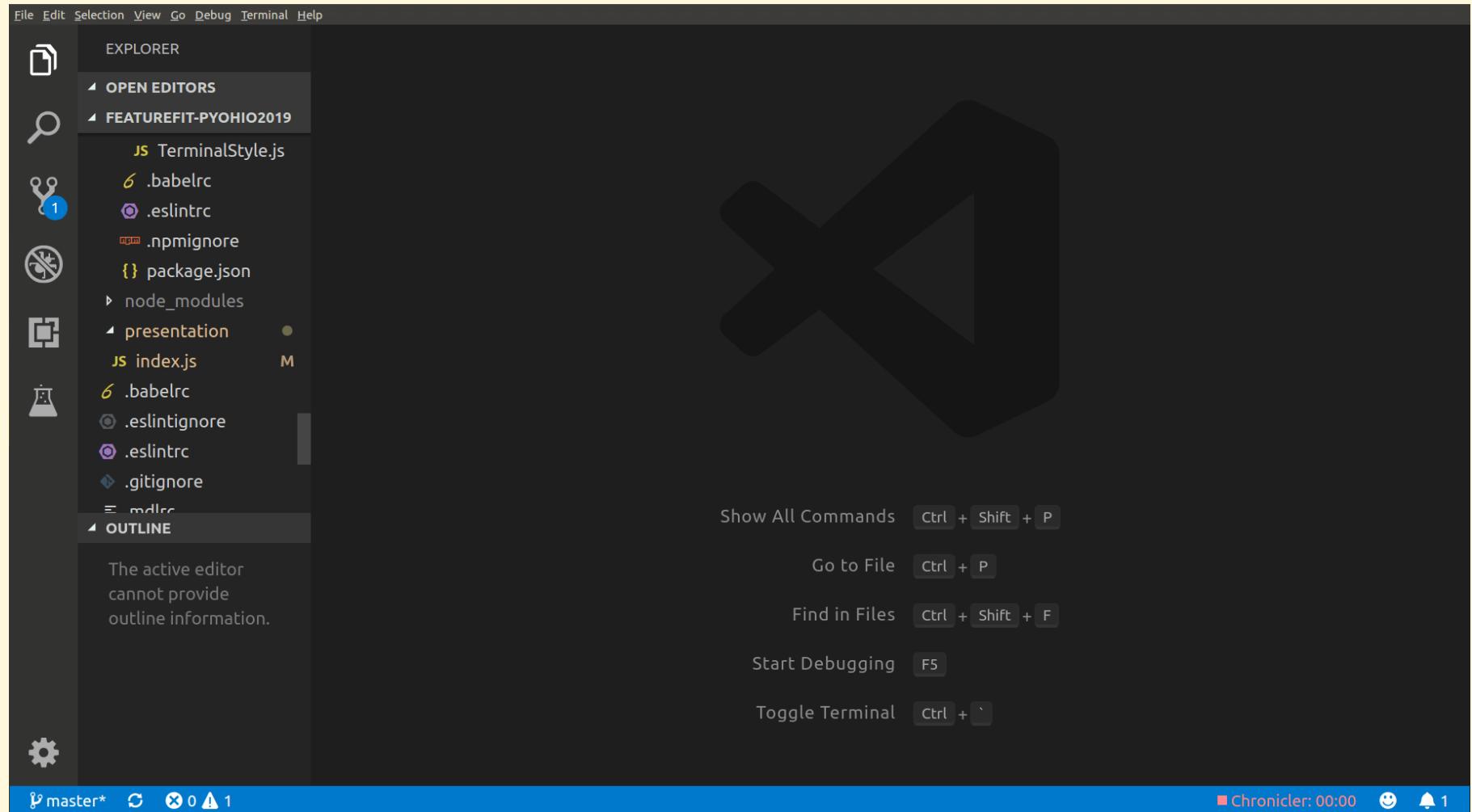
Discover new plugins and update them on a regular basis

The Basics

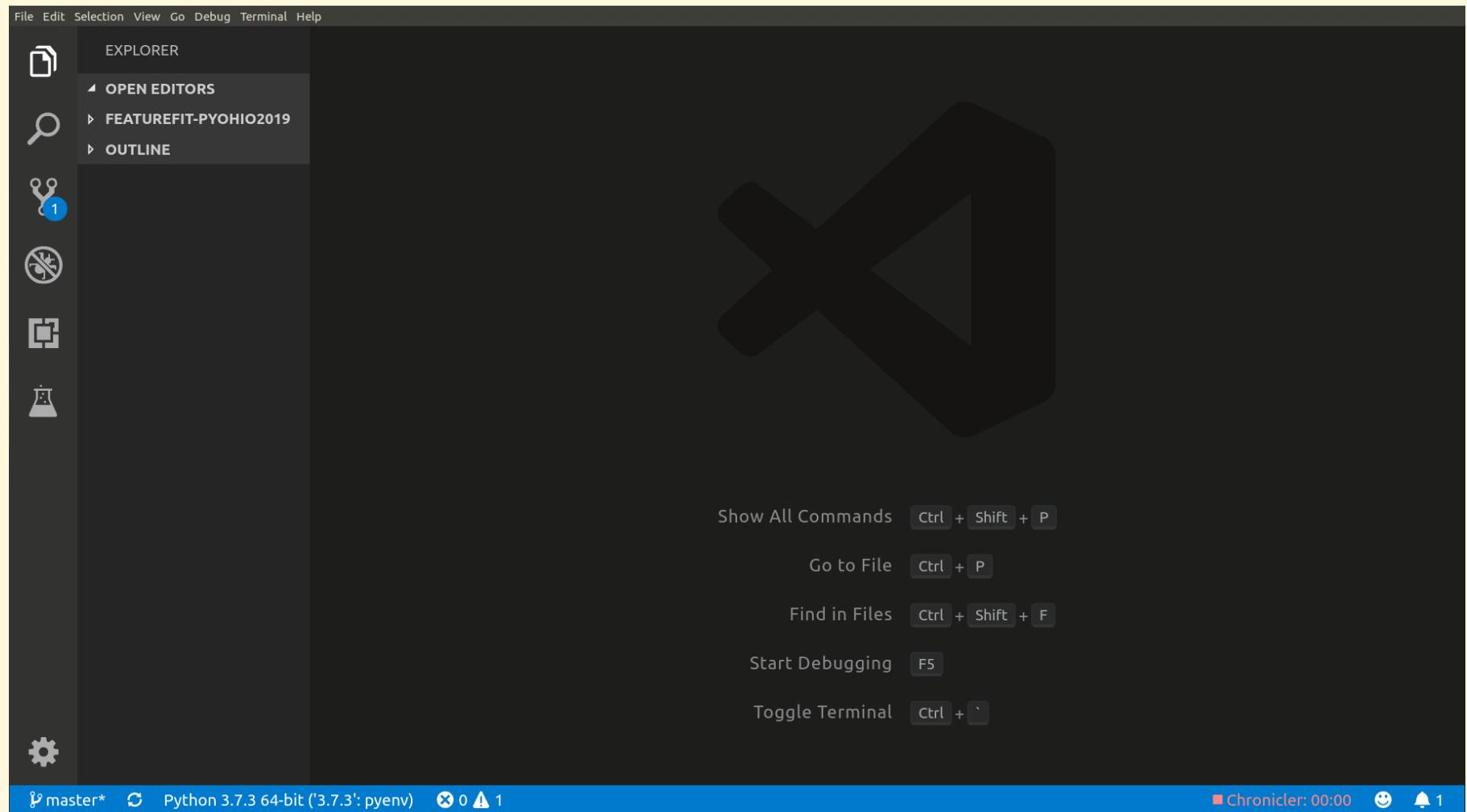
Visual Studio Code

- Open Command Palette: `Ctrl+Shift+P`
- IntelliSense: Uses Language Server Protocol
- Save configuration in the User Settings
- Remember to download the Python Extension

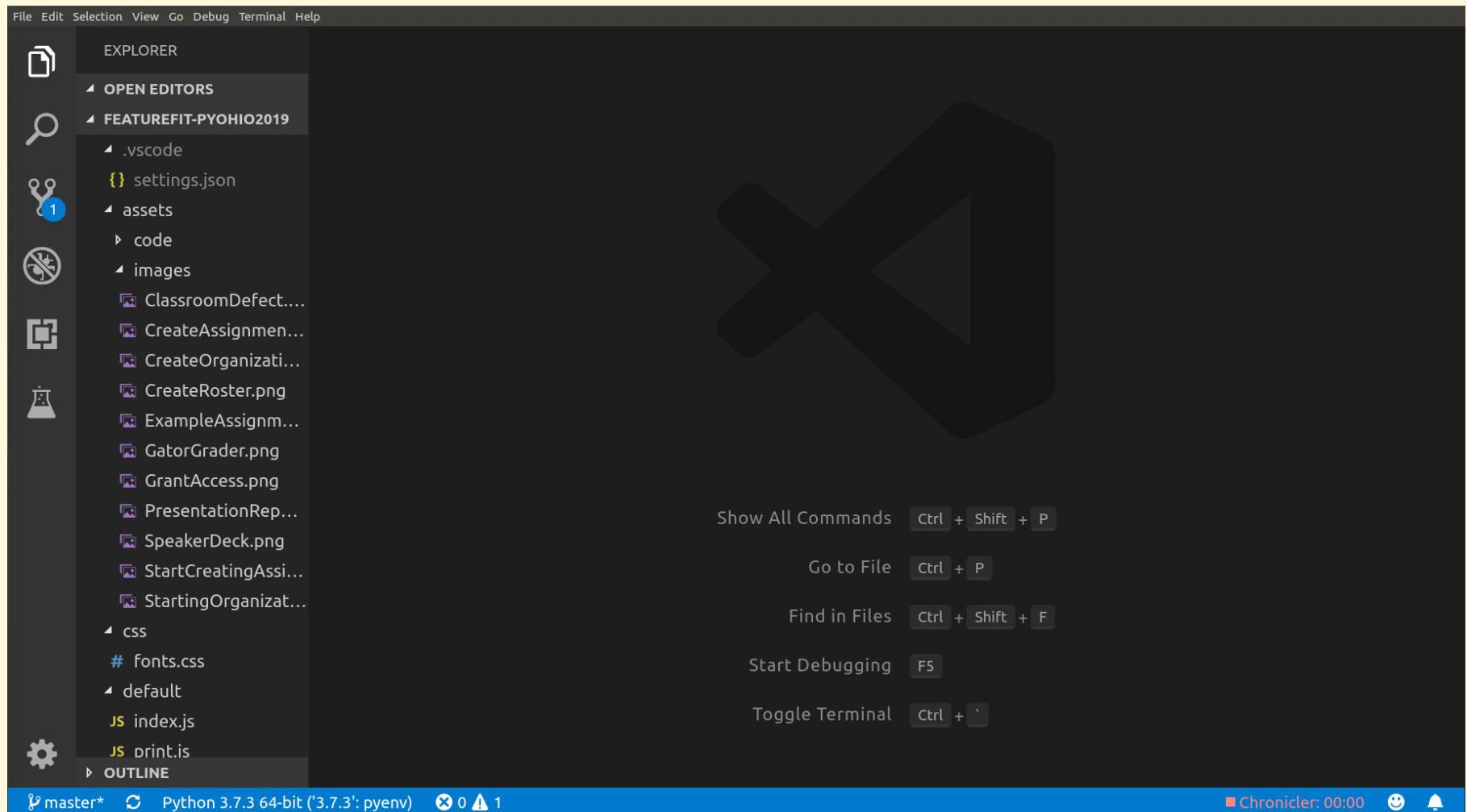
Enable Black Code Formatting



Enable Tab Completion



Enable Flake8 Code Linting

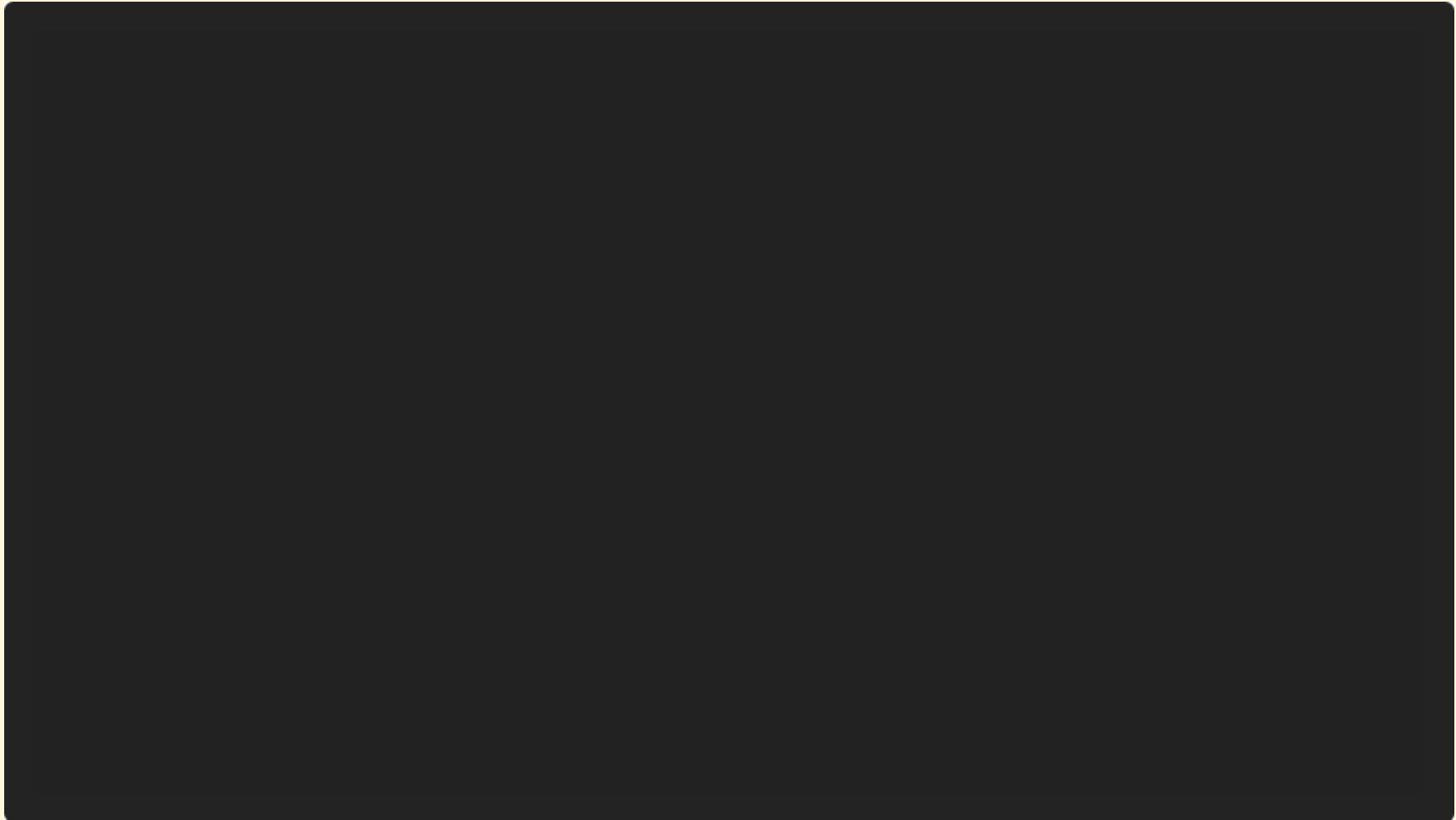


The Basics

Vim or Neovim

- Vim or Neovim in a terminal with tmux
- Configure editor through .vimrc or init.vim
- Install plugins with a plugin manager
- Add configuration options for plugins

Overview of Vim's Configuration File

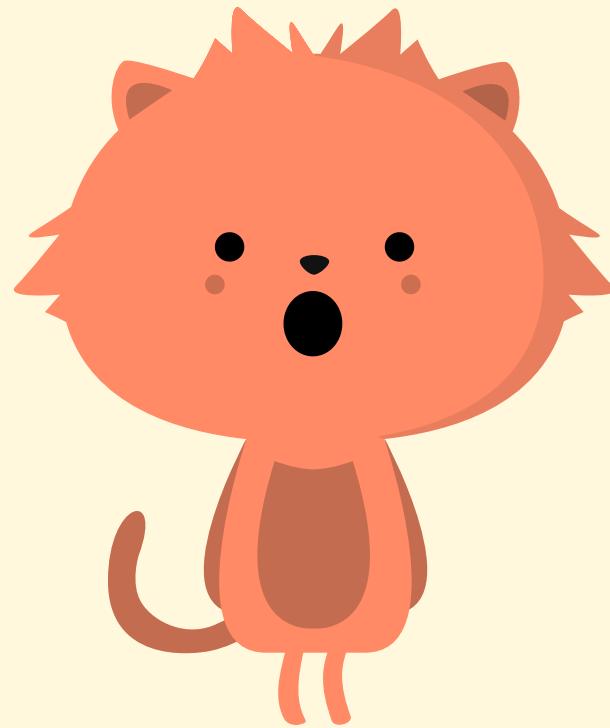


Development Tools

Useful in both VS Code and Vim

- `pyenv`: Download and manage Python versions
- `pipenv`: Application deps and `virtualenvs`
- `pytest`: Run test cases and report their status
- `coverage`: Track statements and branches

Configuration and Plugins



BOTH SUPPORT PROJECTS AND OFFER MANY PLUGINS, BUT VS CODE IS EASIER TO CONFIGURE

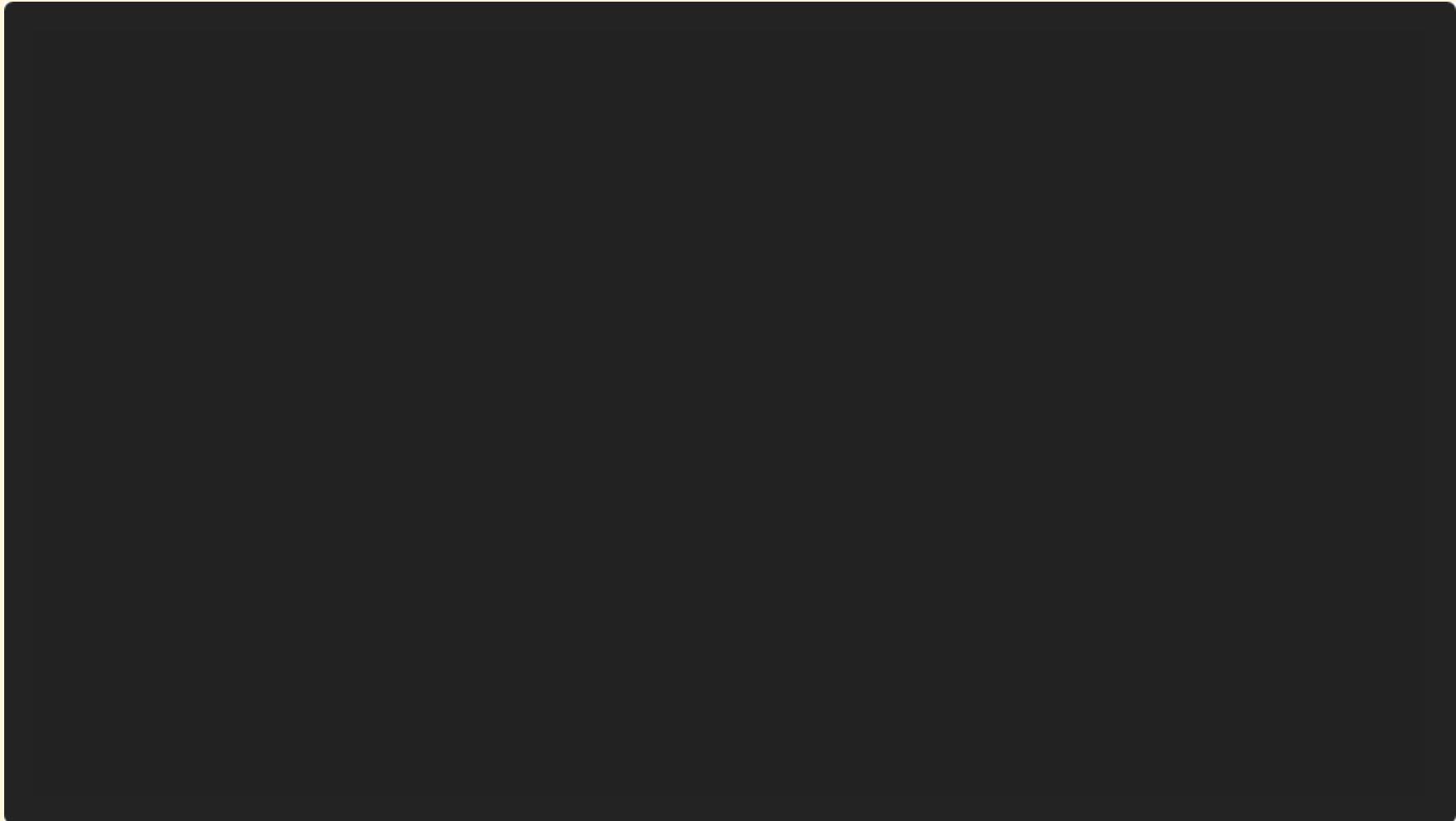
GatorGrader

Open-source tool to check the work of
writers and programmers

- On GitHub at [GatorEducator/gatorgrader](https://github.com/GatorEducator/gatorgrader)
- `pipenv shell`: Enter a virtual environment
- `pipenv run test`: Run the test suite
- `pipenv run cover`: Check for full coverage

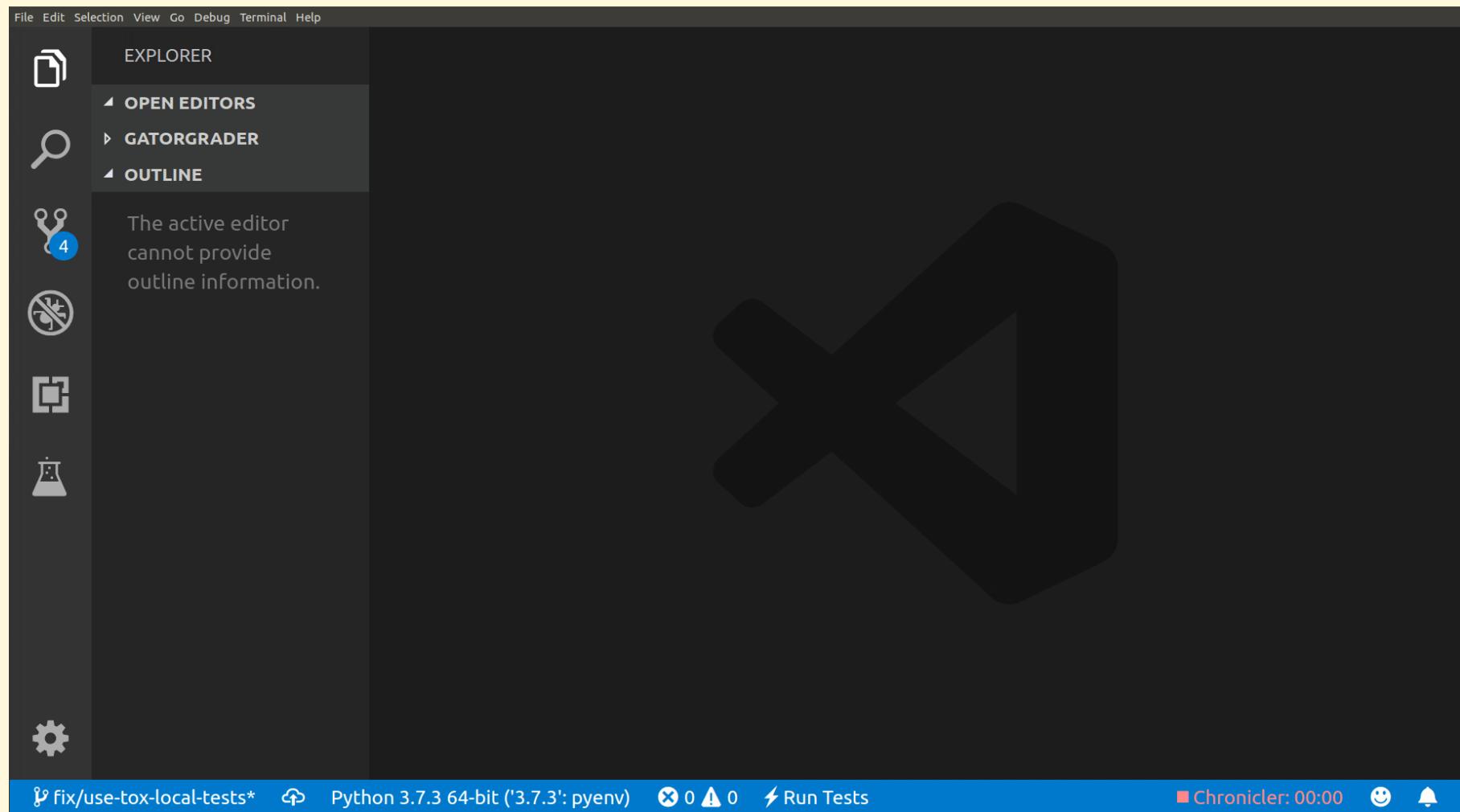
Install the
Application Dependencies

Install Application Dependencies

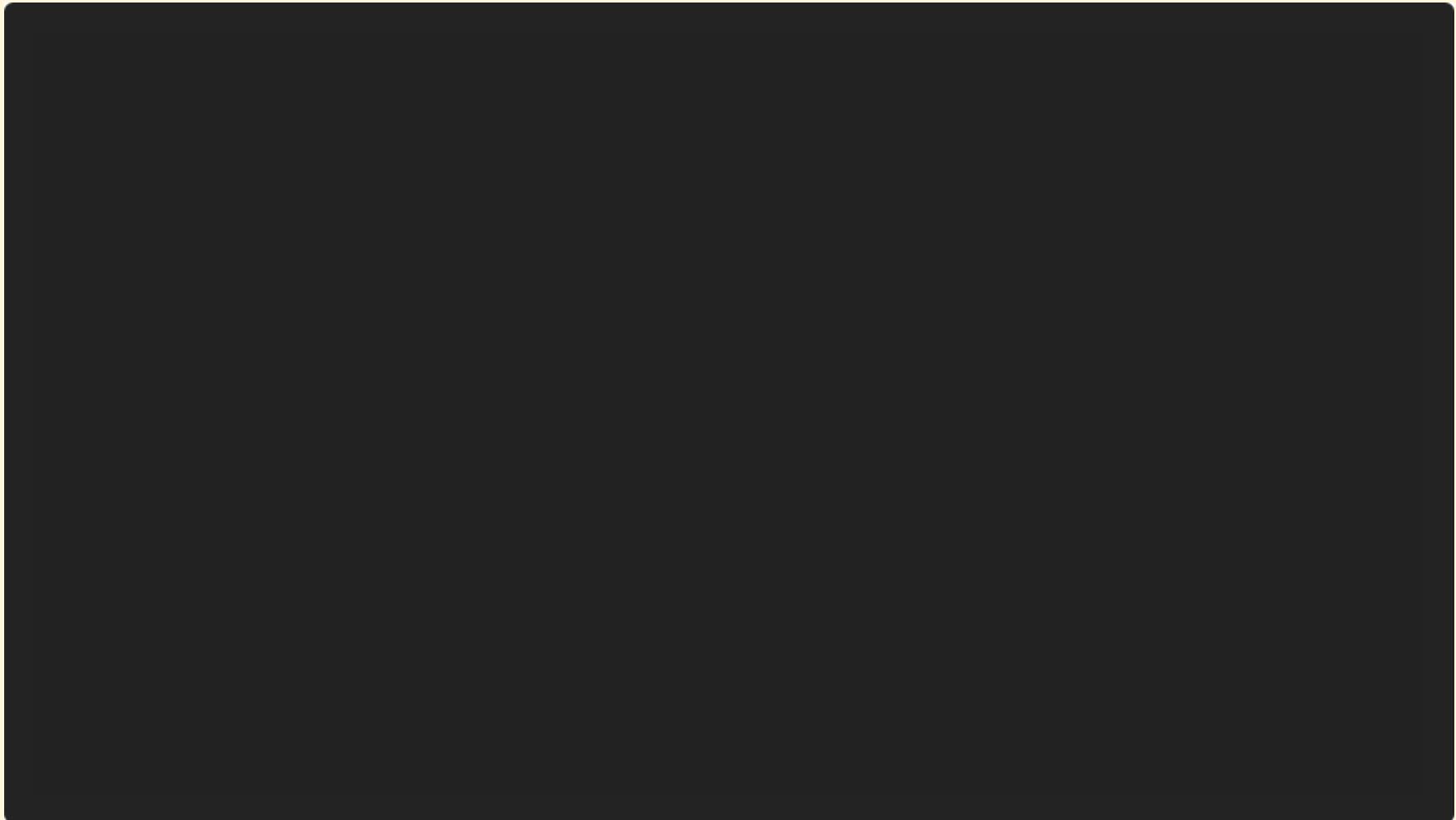


Start Your
Virtual Environment

Selecting a Virtual Environment



Start Project in a Virtual Environment



Virtual Environments and Package Installation



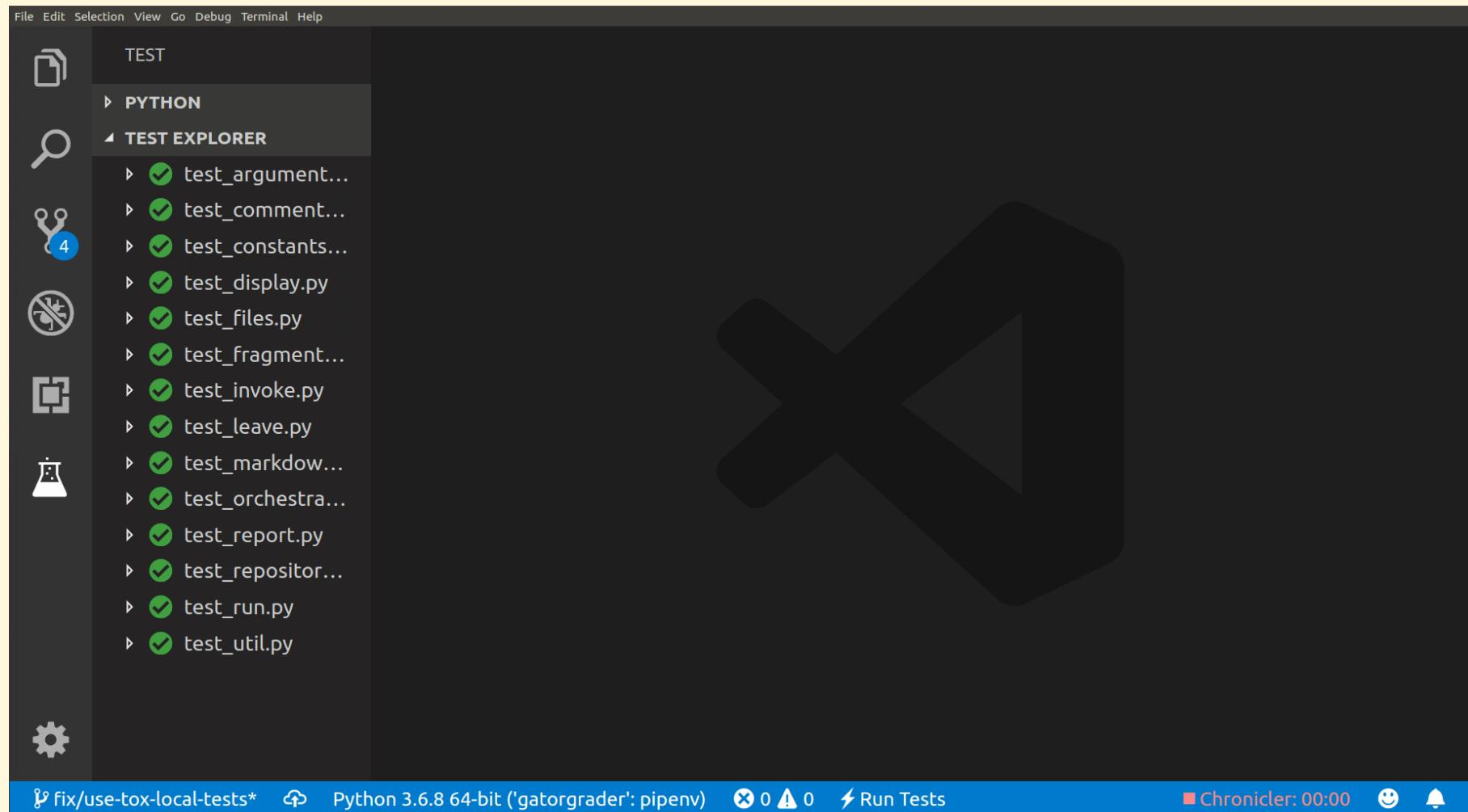
BOTH EFFECTIVELY USE PIPENV TO MANAGE VIRTUALENVS AND APP DEPENDENCIES

**Let's Start
Programming in Python!**

Calculate Test Suite Coverage

Work on an existing code base

Terminal command: pipenv run cover





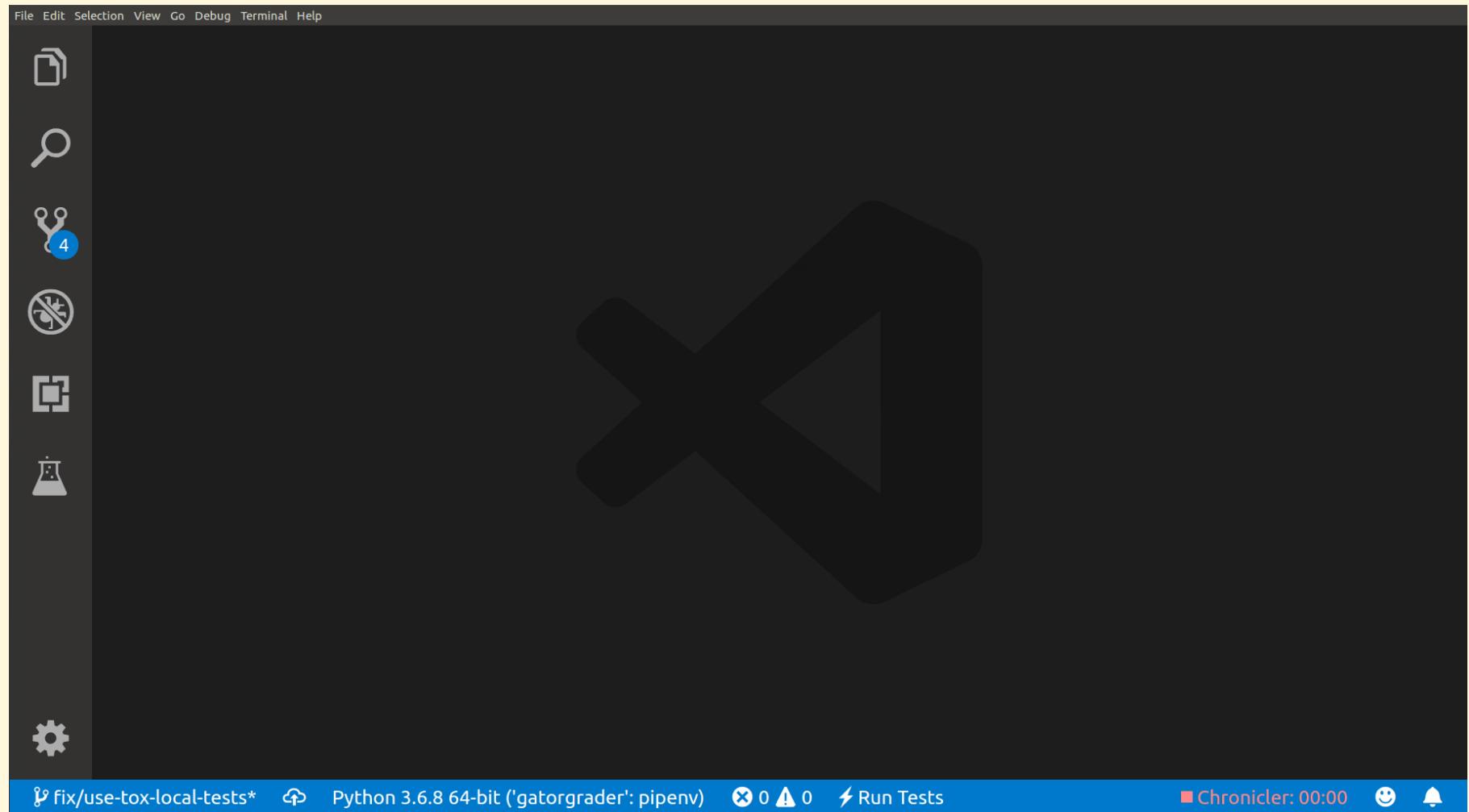
99% total
coverage

98%
coverage
`test_util.py`

I thought
coverage was 100%?

"It is 100%! Wait, what?"

Fuzzy File Finding for the Test





What's the Problem?

Hardcoded Test Variable

```
51      Run Test | Debug Test | Run | Debug | Show in Test Explorer
52  ✓ def test_gatorgrader_home_is_set_after_os_dictionary_set_example():
53      """Ensure that the GATORGRADER_HOME environment variable is set."""
54      os.environ["GATORGRADER_HOME"] = "/home/gkapfham/source/gatorgrader"
55      gatorgrader_home = util.get_gatorgrader_home()
56      assert gatorgrader_home is not None
57      assert "gatorgrader" in gatorgrader_home
58
```

Let's Fix the Test!

Use Pytest's tmpdir test fixture

Use Pytest's tmpdir fixture

```
File Edit Selection View Go Debug Terminal Help

tests › test_util.py › ...
45     current_working_directory = files.create_cwd_path()
46     assert gatorgrader_home is not None
47     assert files.create_path(home=gatorgrader_home) == files.create_path(
48         home=current_working_directory
49     )
50
51
52 def test_gatorgrader_home_is_set_after_os_dictionary_set_example():
53     """Ensure that the GATORGRADER_HOME environment variable is set."""
54     os.environ["GATORGRADER_HOME"] = "/home/gkapfham/source/gatorgrader"
55     gatorgrader_home = util.get_gatorgrader_home()
56     assert gatorgrader_home is not None
57     assert "gatorgrader" in gatorgrader_home
58
59
60 def test_gatorgrader_home_is_set_after_os_dictionary_set_wrong_directory(tmpdir):
61     """Ensure that the GATORGRADER_HOME environment variable is set."""
62     os.environ["GATORGRADER_HOME"] = str(tmpdir) + "/" + "INCORRECT"
63     gatorgrader_home = util.get_gatorgrader_home()

Run Test | Debug Test | Run | Debug | Show in Test Explorer
```

**Wait, this fix
is still incomplete!**

View and Act on Flake8 Warning

```
File Edit Selection View Go Debug Terminal Help
test_util.py x
tests > test_util.py > test_gatorgrader_home_is_set_after_os_dictionary_set_example
45     current_working_directory = files.create_cwd_path()
46     assert gatorgrader_home is not None
47     assert files.create_path(home=gatorgrader_home) == files.create_path(
48         home=current_working_directory
49     )
50
51
52     Run Test | Debug Test | Run | Debug | Show in Test Explorer
53     def test_gatorgrader_home_is_set_after_os_dictionary_set_example():
54         """Ensure that the GATORGRADER_HOME environment variable is set."""
55         os.environ["GATORGRADER_HOME"] = tmpdir + "/" "gatorgrader/"
56         gatorgrader_home = util.get_gatorgrader_home()
57         assert gatorgrader_home is not None
58         assert "gatorgrader" in gatorgrader_home
59
59
60     Run Test | Debug Test | Run | Debug | Show in Test Explorer
60     def test_gatorgrader_home_is_set_after_os_dictionary_set_wrong_directory(tmpdir):
61         """Ensure that the GATORGRADER_HOME environment variable is set."""
62         os.environ["GATORGRADER_HOME"] = str(tmpdir) + "/" + "INCORRECT"
63         gatorgrader_home = util.get_gatorgrader_home()

Spaces: 4  UTF-8  LF  Python  Chronicer: 00:00  ☺  🔔
```

Automated Code Linting



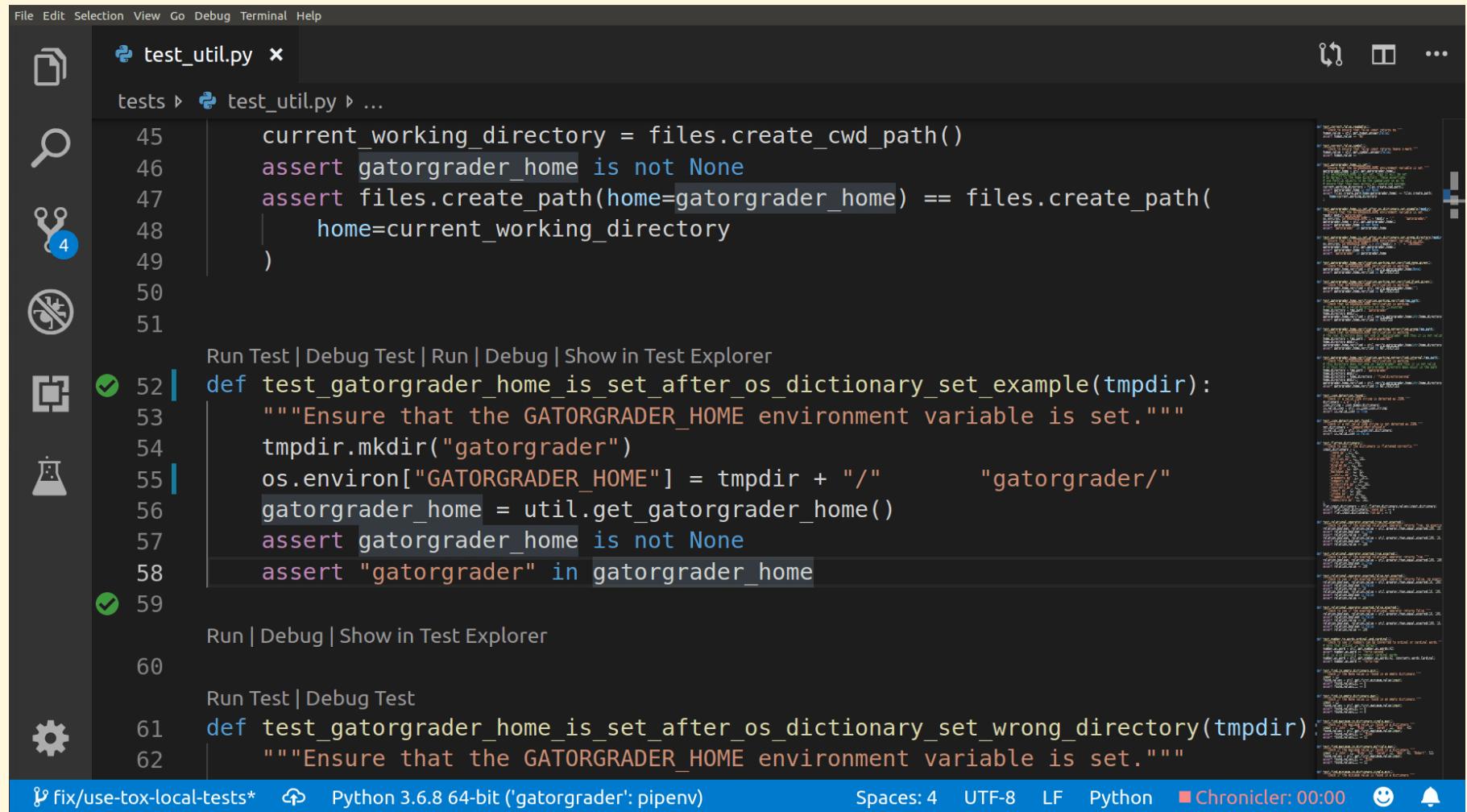
BOTH ALLOW MULTIPLE BACKGROUND LINTERS,
WITH CUSTOMIZATION DIFFERENCES

Code Autocompletion



BOTH EDITORS OFFER SUPPORT, WITH TRADE-OFFS
IN RELEVANCE AND PERFORMANCE

Source Code Formatting



The screenshot shows a dark-themed code editor interface with a sidebar containing various icons. The main area displays a Python file named `test_util.py`. The code is a test suite for the `gatorgrader` project, specifically testing the handling of the `GATORGRADER_HOME` environment variable. The code includes several assertions and comments explaining the purpose of each test. The editor's status bar at the bottom shows the file name, Python version, encoding, and other settings.

```
File Edit Selection View Go Debug Terminal Help
File test_util.py x
tests > test_util.py ...
45     current_working_directory = files.create_cwd_path()
46     assert gatorgrader_home is not None
47     assert files.create_path(home=gatorgrader_home) == files.create_path(
48         home=current_working_directory
49     )
50
51
Run Test | Debug Test | Run | Debug | Show in Test Explorer
52 def test_gatorgrader_home_is_set_after_os_dictionary_set_example(tmpdir):
53     """Ensure that the GATORGRADER_HOME environment variable is set."""
54     tmpdir.mkdir("gatorgrader")
55     os.environ["GATORGRADER_HOME"] = tmpdir + "/"           "gatorgrader/"
56     gatorgrader_home = util.get_gatorgrader_home()
57     assert gatorgrader_home is not None
58     assert "gatorgrader" in gatorgrader_home
59
Run | Debug | Show in Test Explorer
60
Run Test | Debug Test
61 def test_gatorgrader_home_is_set_after_os_dictionary_set_wrong_directory(tmpdir):
62     """Ensure that the GATORGRADER_HOME environment variable is set."""
Fix/use-tox-local-tests* Python 3.6.8 64-bit ('gatorgrader': pipenv)
Spaces: 4 UTF-8 LF Python Chronicler: 00:00
```

Code Formatting



BOTH EDITORS CAN LEVERAGE BLACK, BUT VS CODE
IS MUCH EASIER TO CONFIGURE

Run Test Suites

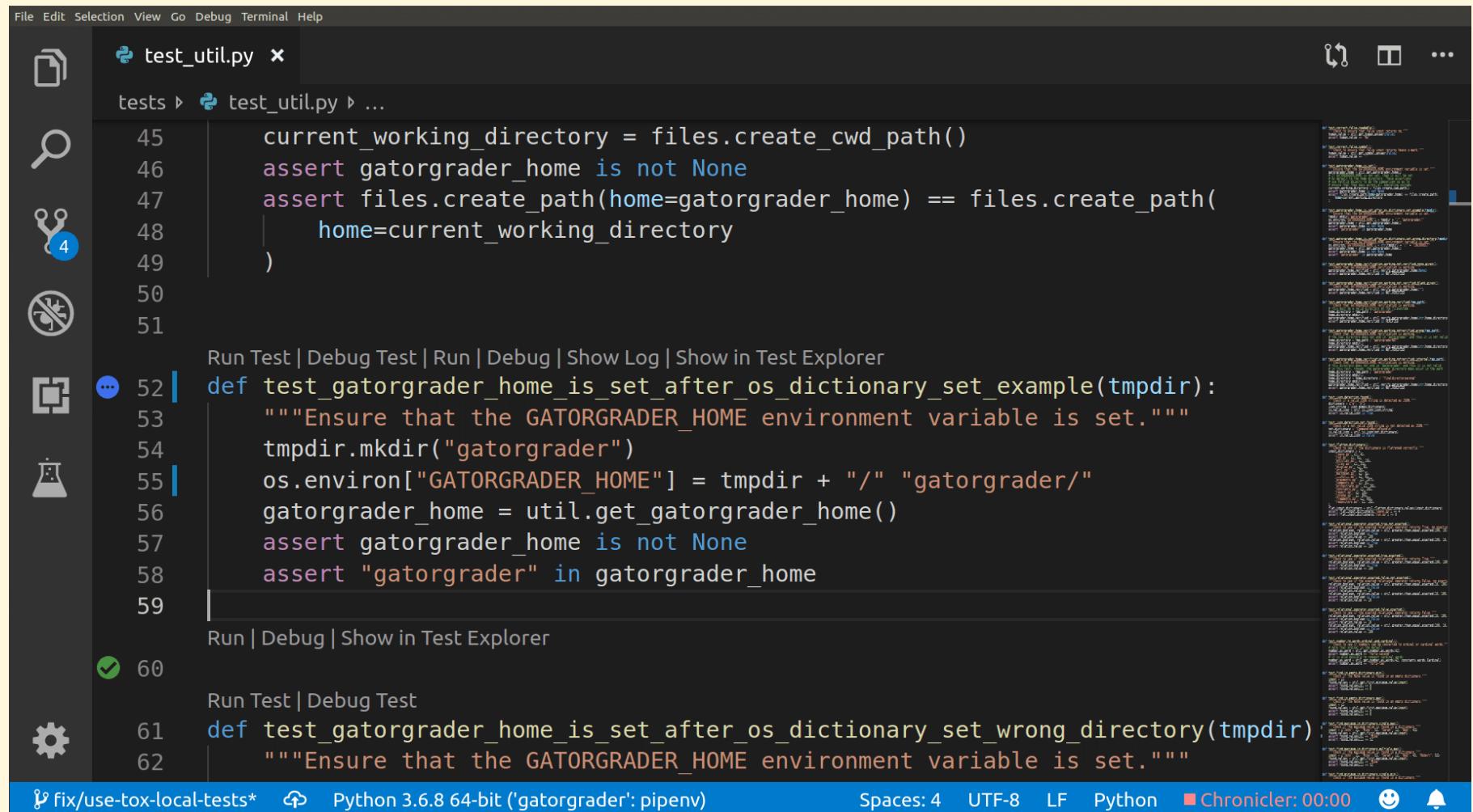
Did we fix the hard-coded test?

Python Test Explorer

```
File Edit Selection View Go Debug Terminal Help
test_util.py x
tests > test_util.py > ...
45     current_working_directory = files.create_cwd_path()
46     assert gatorgrader_home is not None
47     assert files.create_path(home=gatorgrader_home) == files.create_path(
48         home=current_working_directory
49     )
50
51
Run Test | Debug Test | Run | Debug | Show Log | Show in Test Explorer
52 def test_gatorgrader_home_is_set_after_os_dictionary_set_example(tmpdir):
53     """Ensure that the GATORGRADER_HOME environment variable is set."""
54     tmpdir.mkdir("gatorgrader")
55     os.environ["GATORGRADER_HOME"] = tmpdir + "/gatorgrader/"
56     gatorgrader_home = util.get_gatorgrader_home()
57     assert gatorgrader_home is not None
58     assert "gatorgrader" in gatorgrader_home
59
Run | Debug | Show in Test Explorer
60
Run Test | Debug Test
61 def test_gatorgrader_home_is_set_after_os_dictionary_set_wrong_directory(tmpdir):
62     """Ensure that the GATORGRADER_HOME environment variable is set."""

```

Convert tmpdir to a String

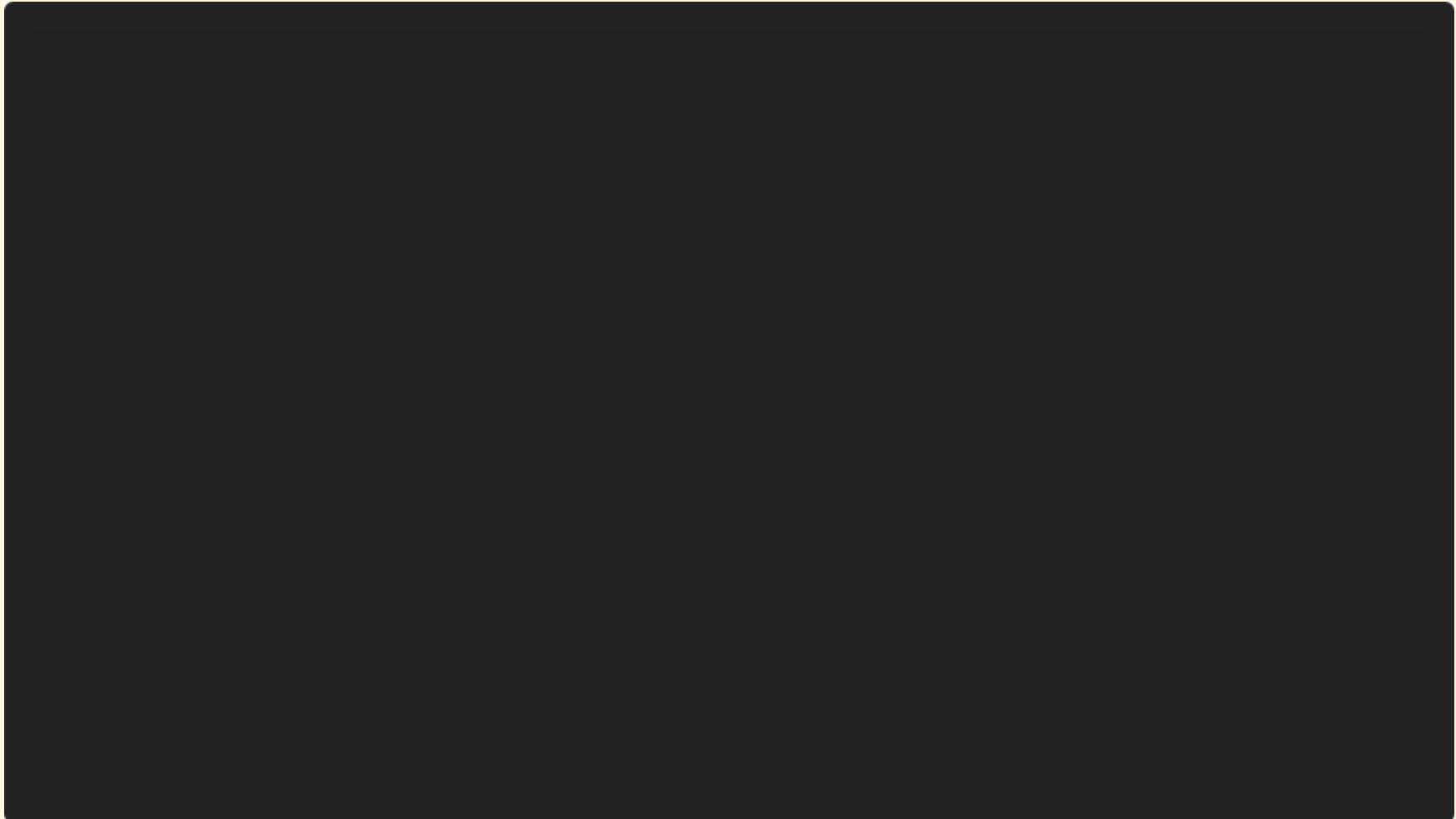


A screenshot of a code editor (VS Code) displaying a Python test file named `test_util.py`. The code is part of a test suite for `gatorgrader`, specifically testing the handling of the `GATORGRADER_HOME` environment variable.

```
File Edit Selection View Go Debug Terminal Help
File test_util.py x
tests > test_util.py ...
45     current_working_directory = files.create_cwd_path()
46     assert gatorgrader_home is not None
47     assert files.create_path(home=gatorgrader_home) == files.create_path(
48         home=current_working_directory
49     )
50
51
52 Run Test | Debug Test | Run | Debug | Show Log | Show in Test Explorer
53 def test_gatorgrader_home_is_set_after_os_dictionary_set_example(tmpdir):
54     """Ensure that the GATORGRADER_HOME environment variable is set."""
55     tmpdir.mkdir("gatorgrader")
56     os.environ["GATORGRADER_HOME"] = tmpdir + "/" "gatorgrader/"
57     gatorgrader_home = util.get_gatorgrader_home()
58     assert gatorgrader_home is not None
59     assert "gatorgrader" in gatorgrader_home
60
61 Run | Debug | Show in Test Explorer
62
63 Run Test | Debug Test
64 def test_gatorgrader_home_is_set_after_os_dictionary_set_wrong_directory(tmpdir):
65     """Ensure that the GATORGRADER_HOME environment variable is set."""
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
99
```

The code includes two test functions: `test_gatorgrader_home_is_set_after_os_dictionary_set_example` and `test_gatorgrader_home_is_set_after_os_dictionary_set_wrong_directory`. The first test creates a temporary directory and sets the `GATORGRADER_HOME` environment variable to its path, then checks if the variable is correctly set. The second test is currently commented out.

Run Tests and Coverage



Check Test Suite Coverage

File Edit Selection View Go Debug Terminal Help

test_util.py x

tests > test_util.py > ...

49 |)
50
51
Run Test | Debug Test | Run | Debug | Show in Test Explorer
52 | def test_gatorgrader_home_is_set_after_os_dictionary_set_example(tmpdir):
53 | """Ensure that the GATORGRADER_HOME environment variable is set."""
54 | tmpdir.mkdir("gatorgrader")
55 | os.environ["GATORGRADER_HOME"] = str(tmpdir) + "/" "gatorgrader/"
56 | gatorgrader_home = util.get_gatorgrader_home()
57 | assert gatorgrader_home is not None
58 | assert "gatorgrader" in gatorgrader_home
59
Run | Debug | Show in Test Explorer
60
Run Test | Debug Test
61 | def test_gatorgrader_home_is_set_after_os_dictionary_set_wrong_directory(tmpdir):
62 | """Ensure that the GATORGRADER_HOME environment variable is set."""
63 | os.environ["GATORGRADER_HOME"] = str(tmpdir) + "/" + "INCORRECT"
64 | gatorgrader_home = util.get_gatorgrader_home()
65 | assert gatorgrader_home is not None
66 | assert "gatorgrader" in gatorgrader_home
67

Fix/use-tox-local-tests* Python 3.6.8 64-bit ('gatorgrader': pipenv) Spaces: 4 UTF-8 LF Python Chronicler: 00:00



100%
total
coverage!

Automated Testing



BOTH SUPPORT TESTING WELL, BUT OFFER A VERY
DIFFERENT USER EXPERIENCE

Source Code Highlighting



BOTH SUPPORT FAST SYNTAX HIGHLIGHTING, BUT
VIM REQUIRES AN EXTRA PLUGIN

Text Snippets

Save time by using a template for text

Autocompletion trigger inserts text

Snippet Expansion



BOTH EDITORS SUPPORT SNIPPETS, BUT VIM WORKS
WELL IN MORE SCENARIOS

Try multiple editors!
Which one did you
like best? Why?

VS CODE



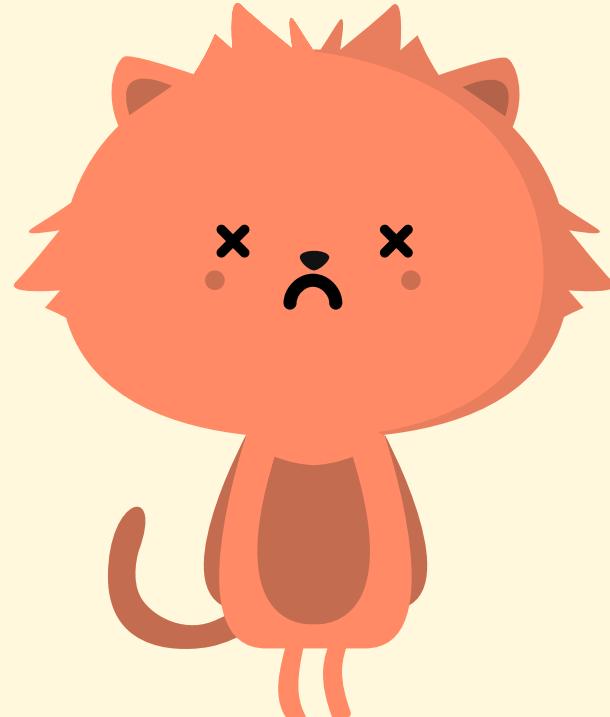
VIM



VS CODE



VIM



Learning Opportunities

- Editing Documentation in Markdown
- Integration with Git and GitHub
- Automated source code refactoring
- Adding and removing dependencies
- Releasing a Python application to PyPI

WANT MORE?

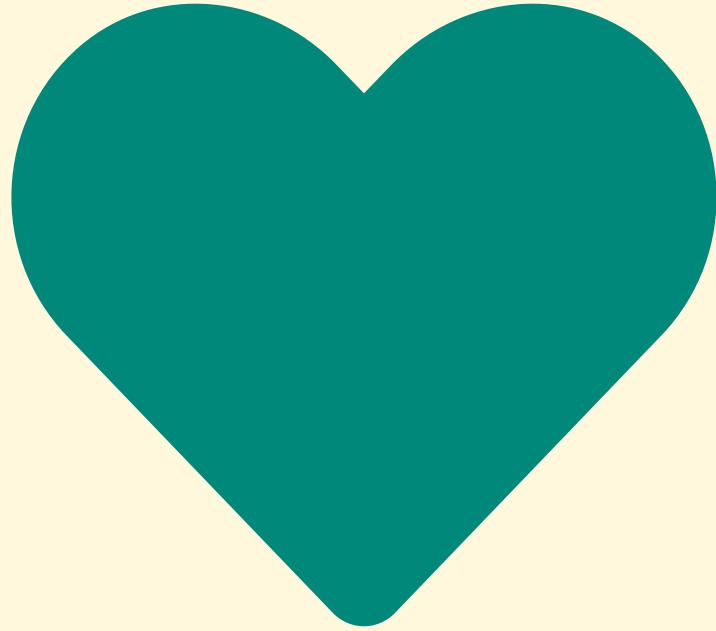
100 likes for a FREE blog post!

On all things VS Code and Vim ...

... and what it's like to
work as a "Father Daughter Duo"!

@GregKapfhammer

@MaddyKapfhammer



Just 100 Likes

come on, you can do it
it's not that hard!

Key Points

Two awesome text editors and
a great learning experience!

Remember ... 100 likes for a blog post!

@GregKapfhammer

@MaddyKapfhammer