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
A FAST PACED 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

Show All Commands: Ctrl + Shift + P
Go to File: Ctrl + P
Find in Files: Ctrl + Shift + F
Start Debugging: F5
Toggle Terminal: Ctrl + `
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
- 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

EXPLORER
- OPEN EDITORS
- GATORGRADER
- OUTLINE

The active editor cannot provide outline information.
Start Project in a Virtual Environment
Virtual Environments and Package Installation

Both effectively use pipenv to manage virtual environments 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

```python
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
```
Let's Fix the Test!
Use Pytest's tmpdir test fixture
Use Pytest's tmpdir fixture
Wait, this fix is still incomplete!
View and Act on Flake8 Warning
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

def test_gatorgrader_home_is_set_after_os_dictionary_set_example(tmpdir):
    """Ensure that the GATORGRADER_HOME environment variable is set."""
    tmpdir.mkdir("gatorgrader")
    os.environ["GATORGRADER_HOME"] = tmpdir + "/gatorgrader/
    gatorgrader_home = util.get_gatorgrader_home()
    assert gatorgrader_home is not None
    assert "gatorgrader" in gatorgrader_home

def test_gatorgrader_home_is_set_after_os_dictionary_set_wrong_directory(tmpdir):
    """Ensure that the GATORGRADER_HOME environment variable is set."""
    tmpdir.mkdir("gatorgrader")
    os.environ["GATORGRADER_HOME"] = tmpdir + "/WrongDir"
    gatorgrader_home = util.get_gatorgrader_home()
    assert gatorgrader_home is not None
    assert "gatorgrader" in gatorgrader_home
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

def test_gatorgrader_home_is_set_after_os_dictionary_set_example(tmpdir):
    """Ensure that the GATORGRADER_HOME environment variable is set."""
    tmpdir.mkdir("gatorgrader")
    os.environ["GATORGRADER_HOME"] = tmpdir + "/" "gatorgrader/"
gatorgrader_home = util.get_gatorgrader_home()
    assert gatorgrader_home is not None
    assert "gatorgrader" in gatorgrader_home


def test_gatorgrader_home_is_set_after_os_dictionary_set_wrong_directory(tmpdir):
    """Ensure that the GATORGRADER_HOME environment variable is set."""

Convert tmpdir to a String

```python
def test_gatorgrader_home_is_set_after_os_dictionary_set_example(tmpdir):
    """Ensure that the GATORGRADER_HOME environment variable is set."""
    tmpdir.mkdir("gatorgrader")
    os.environ["GATORGRADER_HOME"] = tmpdir + "/" "gatorgrader/"
    gatorgrader_home = util.get_gatorgrader_home()
    assert gatorgrader_home is not None
    assert "gatorgrader" in gatorgrader_home
```
Run Tests and Coverage
Check Test Suite Coverage

```python
def test_gatorgrader_home_is_set_after_os_dictionary_set_example(tmpdir):
    """Ensure that the GATORGRADER_HOME environment variable is set."""
    tmpdir.mkdir("gatorgrader")
    os.environ["GATORGRADER_HOME"] = str(tmpdir) + "/" "gatorgrader/"
    gatorgrader_home = util.get_gatorgrader_home()
    assert gatorgrader_home is not None
    assert "gatorgrader" in gatorgrader_home

def test_gatorgrader_home_is_set_after_os_dictionary_set_wrong_directory(tmpdir):
    """Ensure that the GATORGRADER_HOME environment variable is set."""
    os.environ["GATORGRADER_HOME"] = str(tmpdir) + "/" + "INCORRECT"
    gatorgrader_home = util.get_gatorgrader_home()
    assert gatorgrader_home is not None
    assert "gatorgrader" in gatorgrader_home
```
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?
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
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