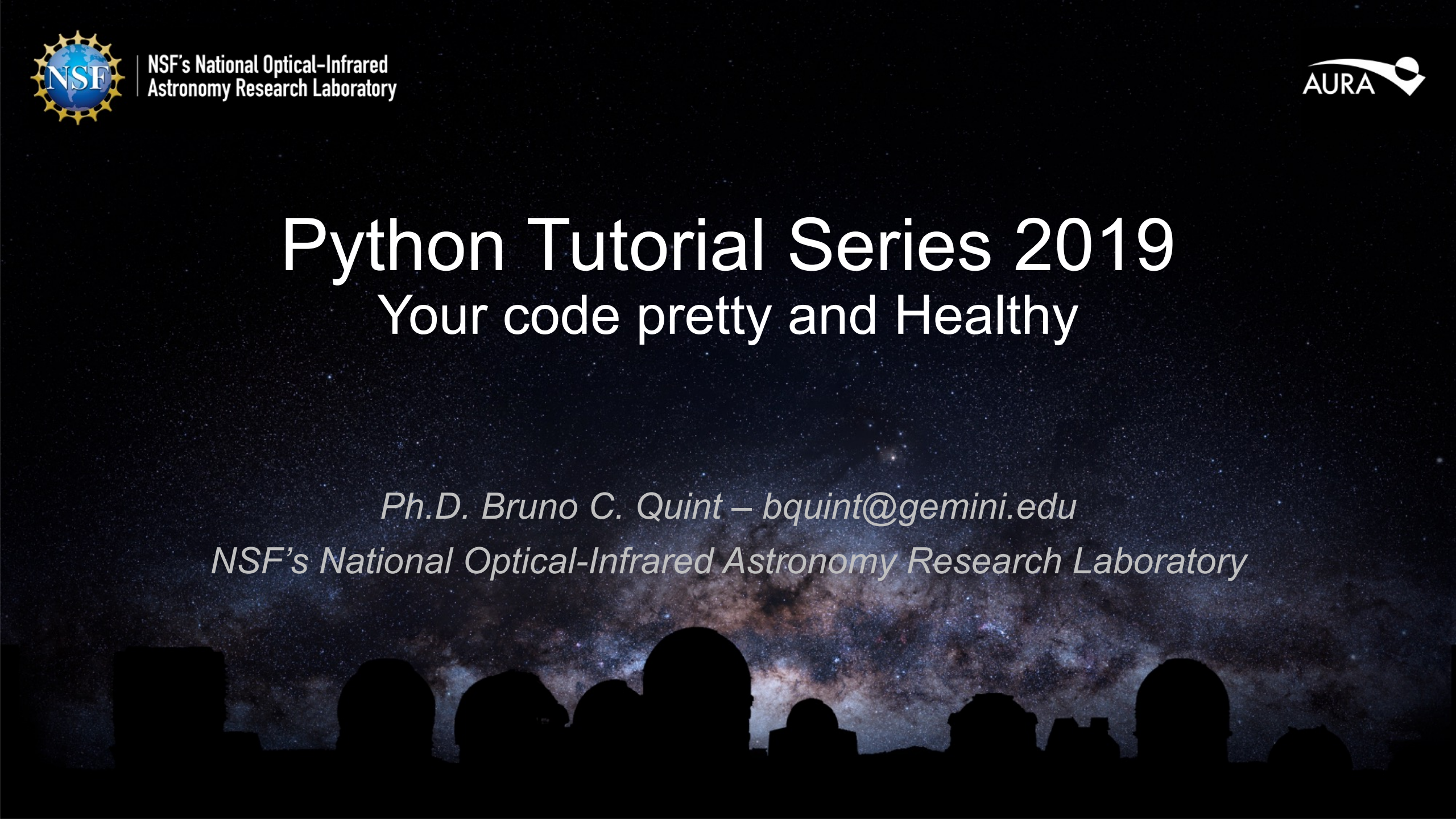


Python Tutorial Series 2019

Your code pretty and Healthy

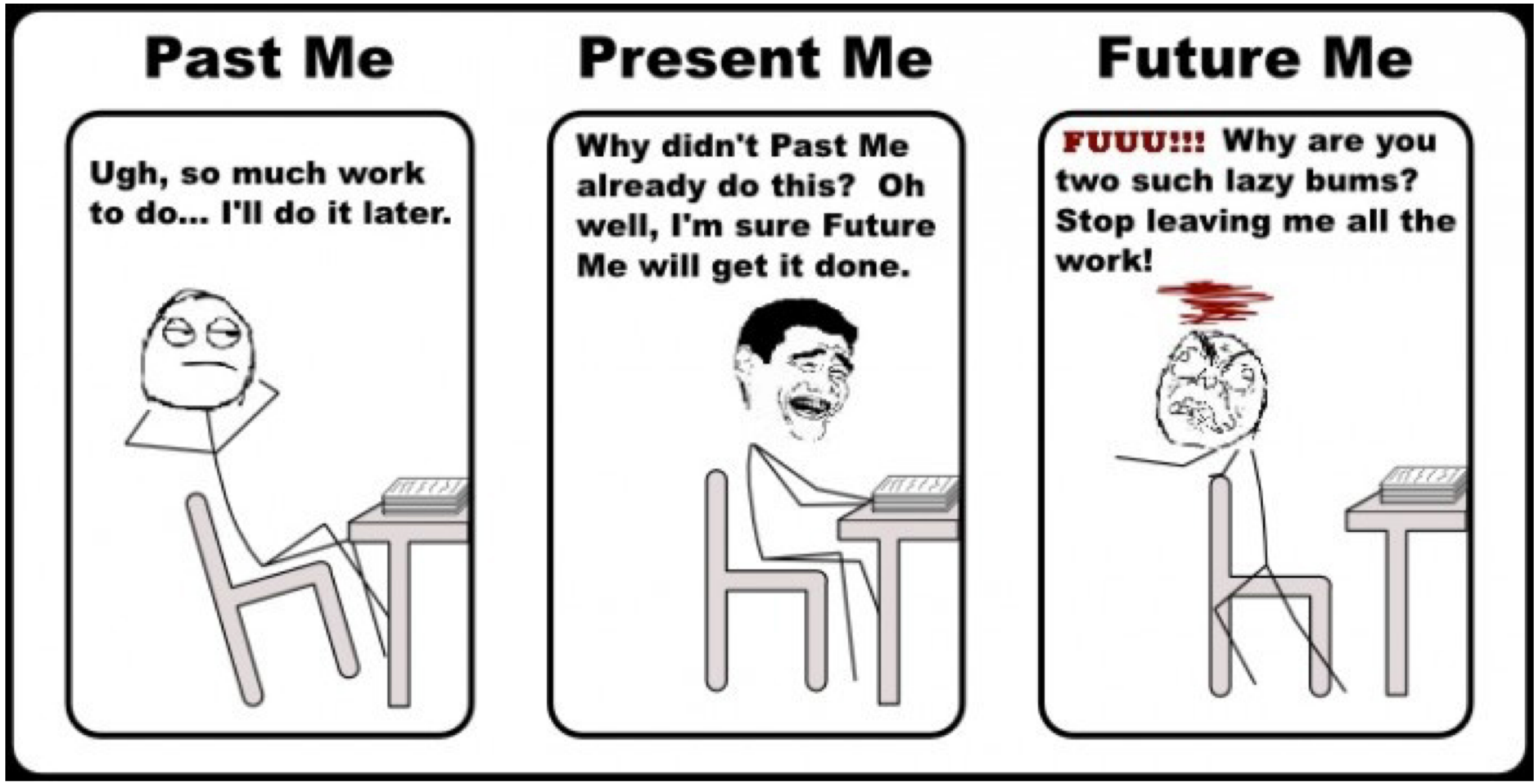
Ph.D. Bruno C. Quint – bquint@gemini.edu

NSF's National Optical-Infrared Astronomy Research Laboratory

Contents

- Code Style
- Documentation
- Testing
- Project Structure
- Versioning and Version Control

Introduction



Code Style

```
>>> import this
```

The Zen Of Python

```
...
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
...
```

Code Style – The PEP8

Why write if others can't read?

Code Style – The PEP8

Why write if others can't read?

Code Style – The PEP8

Why write if others can't read?

Identation – Always use (four) spaces

```
for i in range(10):
....x = i * 2
....print(x)
```

Importing packages

Don't

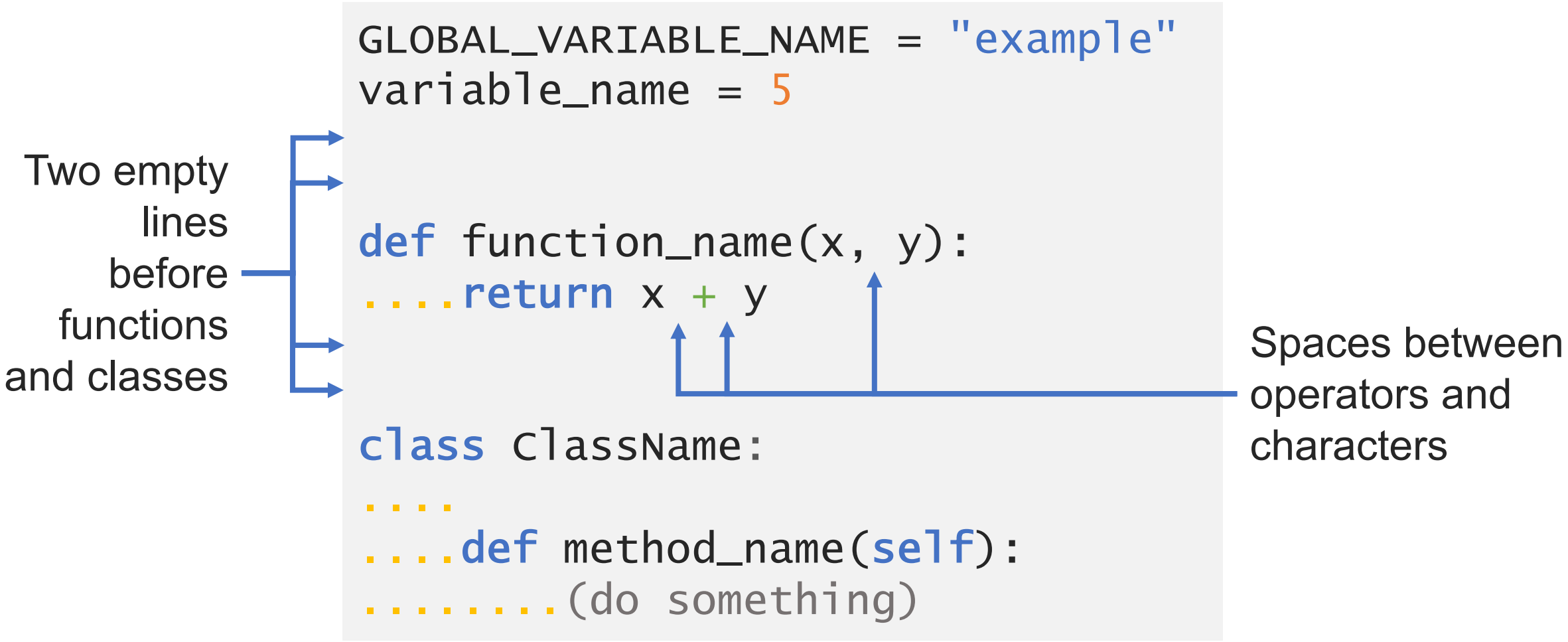
```
from numpy import *
```

Do

```
import numpy as np
```

Code Style – The PEP8

Why write if others can't read?



Code Style – Tools

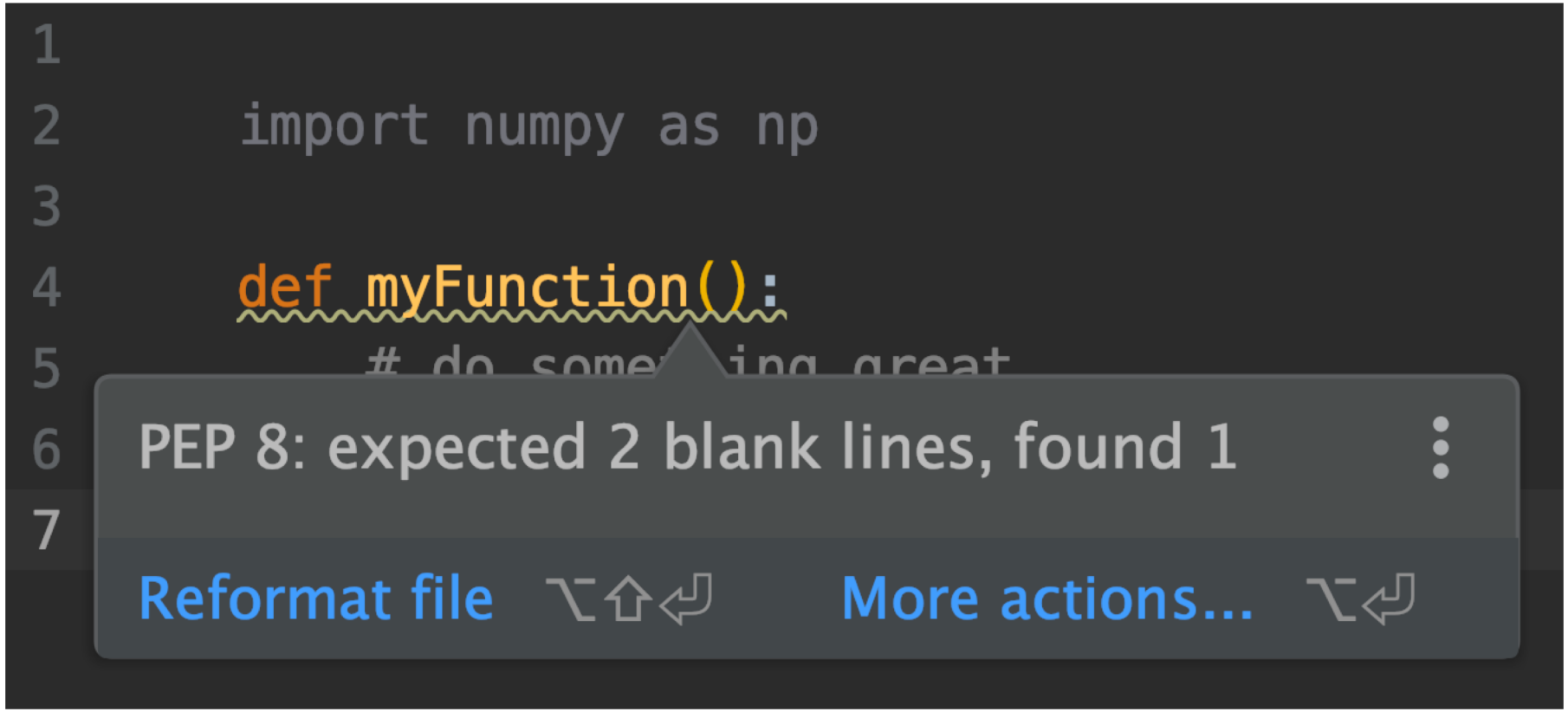
Why write if others can't read?

Tools for code format

- PyLint
- flake8
- Black
- PyCharm CE

Code Style – Tools

Why write if others can't read?



Documentation

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
"""
My Sample Code

This is such a simple documentation! The only
thing I have to do is to add three single quotes
(') or double quotes (") together to start a
docstring block! Then just write down in a few
words what you want to do in your file.

by J. Bond - Feb 31, 202X
"""
```

Documentation

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
"""It is ok if the docstring is just one line"""
```

Documentation

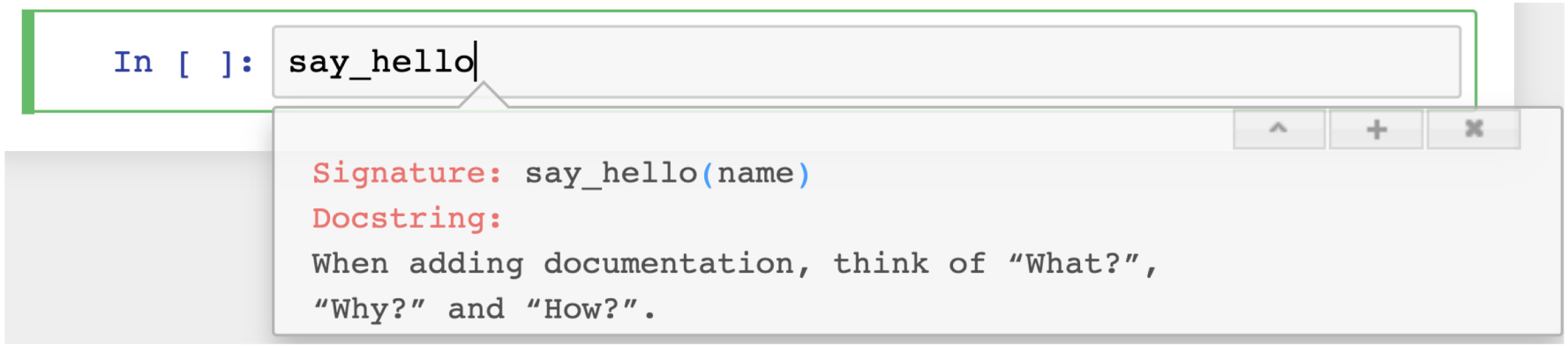
```
def say_hello(name):
    """
    This function says hello to the person whose name
    is given as parameter.
    """
    print("Hello, {0}!".format(name))
```

Documentation [The NumPy Style]

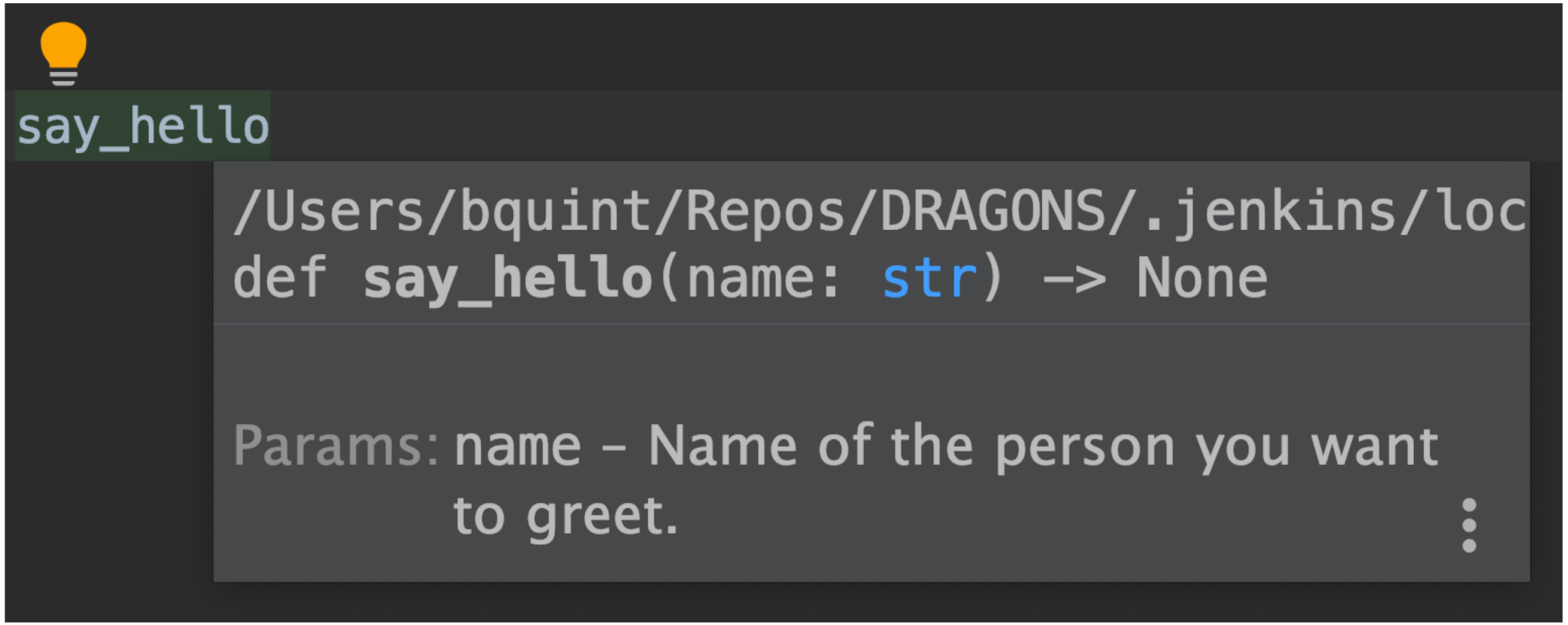
```
def say_hello(name):
        """
        When adding documentation, think of “What?”,
        “Why?” and  “How?”.

        Parameters
        ----------
        name : str
           Name of the person you want to greet.

        """
        print("Hello, {0}!".format(name))
```

Documentation

```
In [2]: say_hello?

Signature: say_hello(name)
Docstring:
When adding documentation, think of “What?”,
“Why?” and “How?”.
```



Documentation



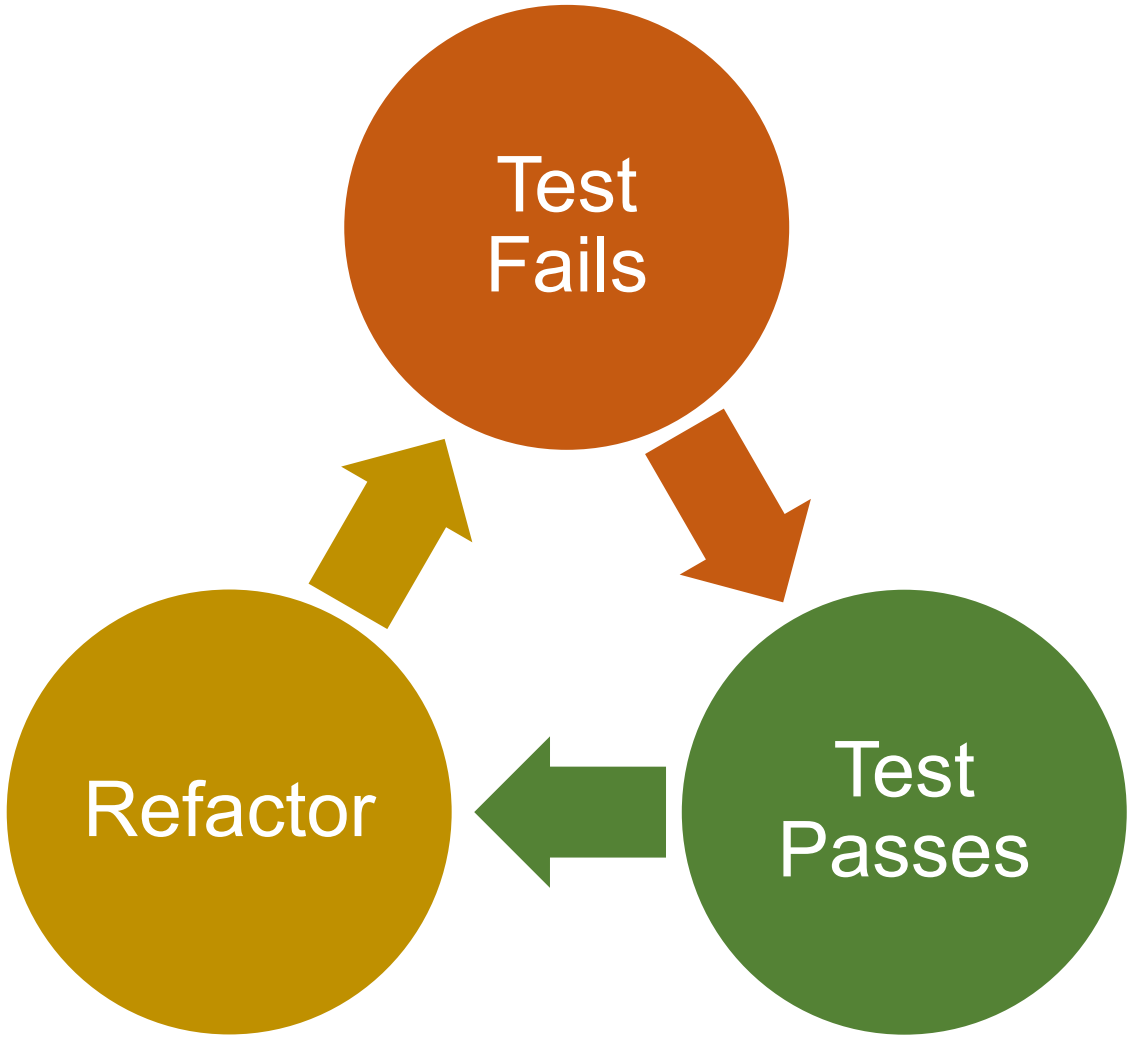
Testing

The refactoring cat

Testing

The refactoring cat

Test-Driven Development



Testing

Tools for code format

- unittest
- nosetest
- pytest
- numpy.testing

- Tests can be top simple top-level functions
- Simple result assertion
- Fixtures provide modularized test suite
- Tests can be easily parametrized

Testing

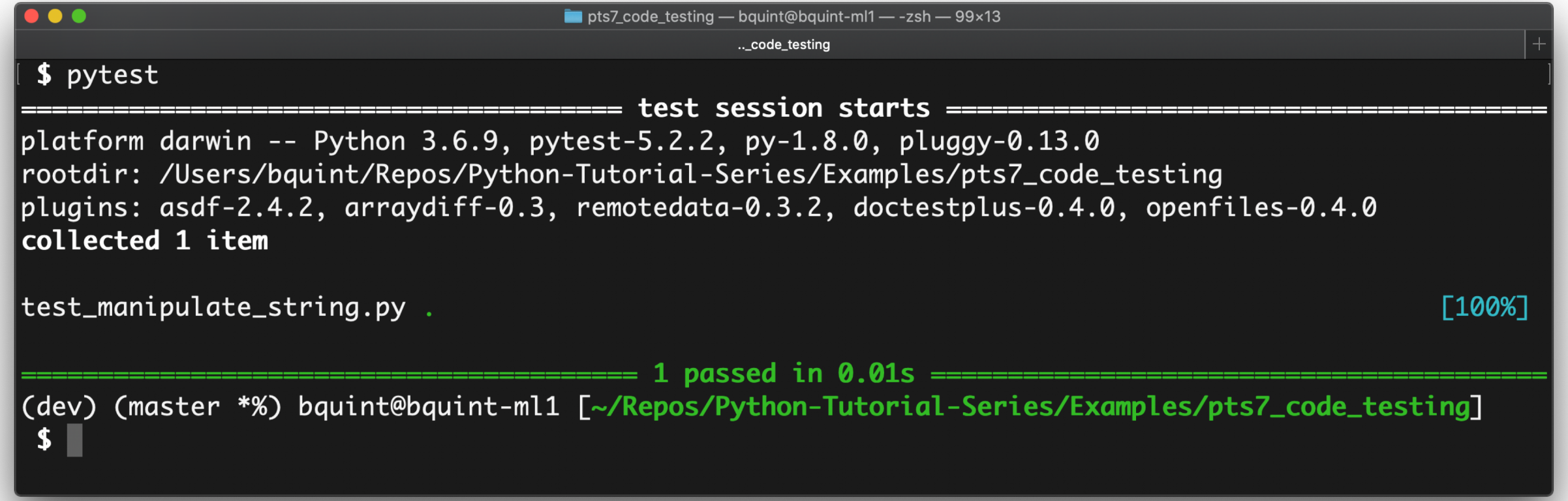
```
# contents of test_capital_case.py
from manipulate_strings import capital_case

def test_capital_case():
    assert capital_case('semaphore') == 'Semaphore'
```

Testing

```
# contents of test_capital_case.py
from manipulate_strings import capital_case

def test_capital_case():
    assert capital_case('semaphore') == 'Semaphore'
```

```
# contents of manipulate_strings.py
def capital_case(my_str):
    return my_str.capitalize()
```

Testing


pts7_code_testing — bquint@bquint-ml1 — -zsh — 99×13
../_code_testing
$ pytest
========== test session starts ==========
platform darwin -- Python 3.6.9, pytest-5.2.2, py-1.8.0, pluggy-0.13.0
rootdir: /Users/bquint/Repos/Python-Tutorial-Series/Examples/pts7_code_testing
plugins: asdf-2.4.2, arraydiff-0.3, remotedata-0.3.2, doctestplus-0.4.0, openfiles-0.4.0
collected 1 item
test_manipulate_string.py . [100%]
========== 1 passed in 0.01s ==========
(dev) (master *%) bquint@bquint-ml1 [~/Repos/Python-Tutorial-Series/Examples/pts7_code_testing]
$

Testing

```
# contents of test_capital_case.py
...
def test_raises_exception_on_non_string_arguments():
    with pytest.raises(TypeError):
        capital_case(9)
```

Testing

pts7_code_testing — bquint@bquint-ml1 — -zsh — 99×22

..._code_testing

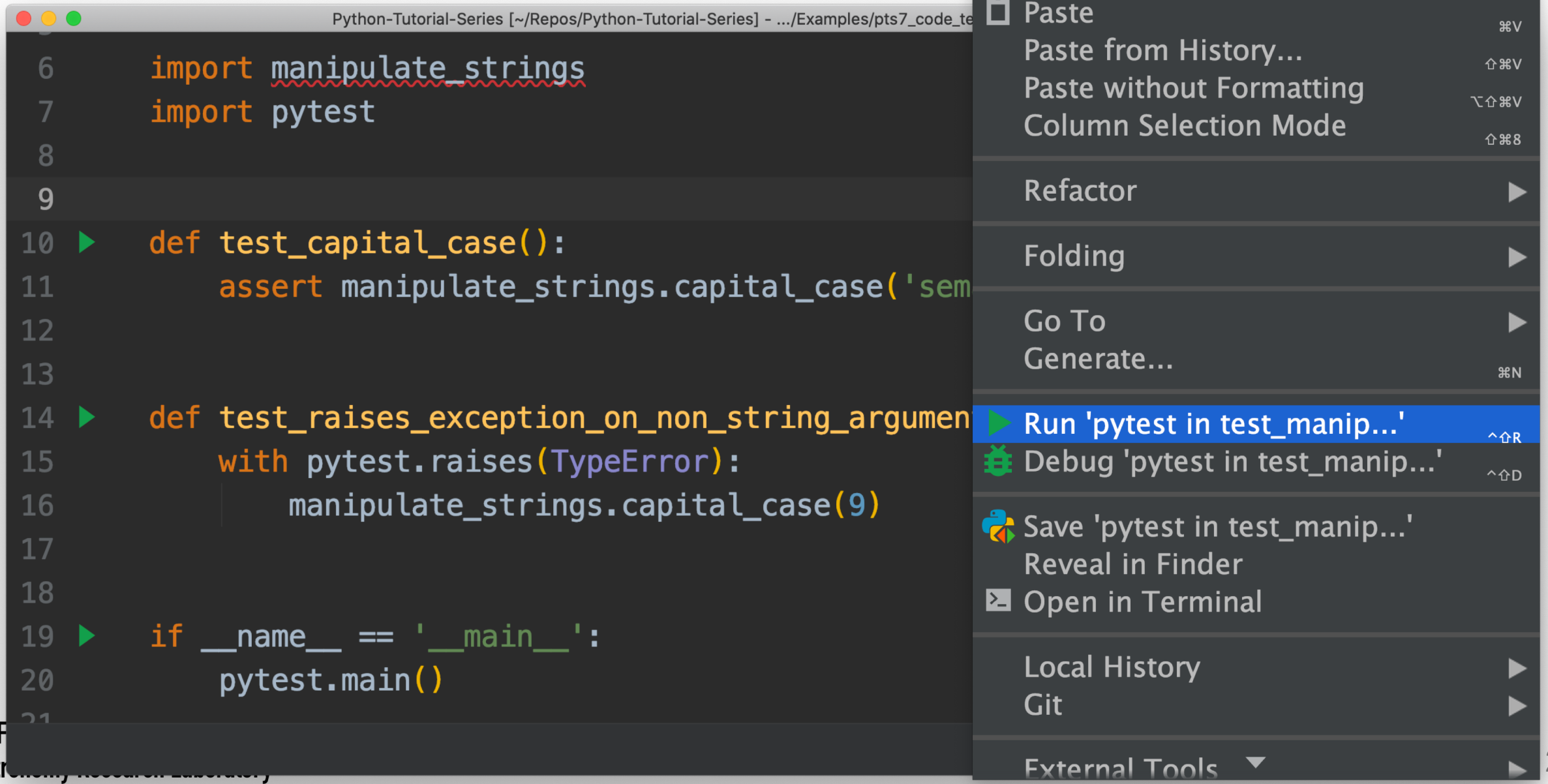
```
test_manipulate_string.py .F                                             [100%]

================================== FAILURES ===================================
_____________________ test_raises_exception_on_non_string_arguments _____________________

    def test_raises_exception_on_non_string_arguments():
        with pytest.raises(TypeError):
>           manipulate_strings.capital_case(9)

test_manipulate_string.py:16:
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

my_str = 9

    def capital_case(my_str):
>       return my_str.capitalize()
E       AttributeError: 'int' object has no attribute 'capitalize'

manipulate_strings.py:8: AttributeError
============================== 1 failed, 1 passed in 0.07s ===============================
(dev) (master *%) bquint@bquint-ml1 [~/Repos/Python-Tutorial-Series/Examples/pts7_code_testing]
$
```

Testing

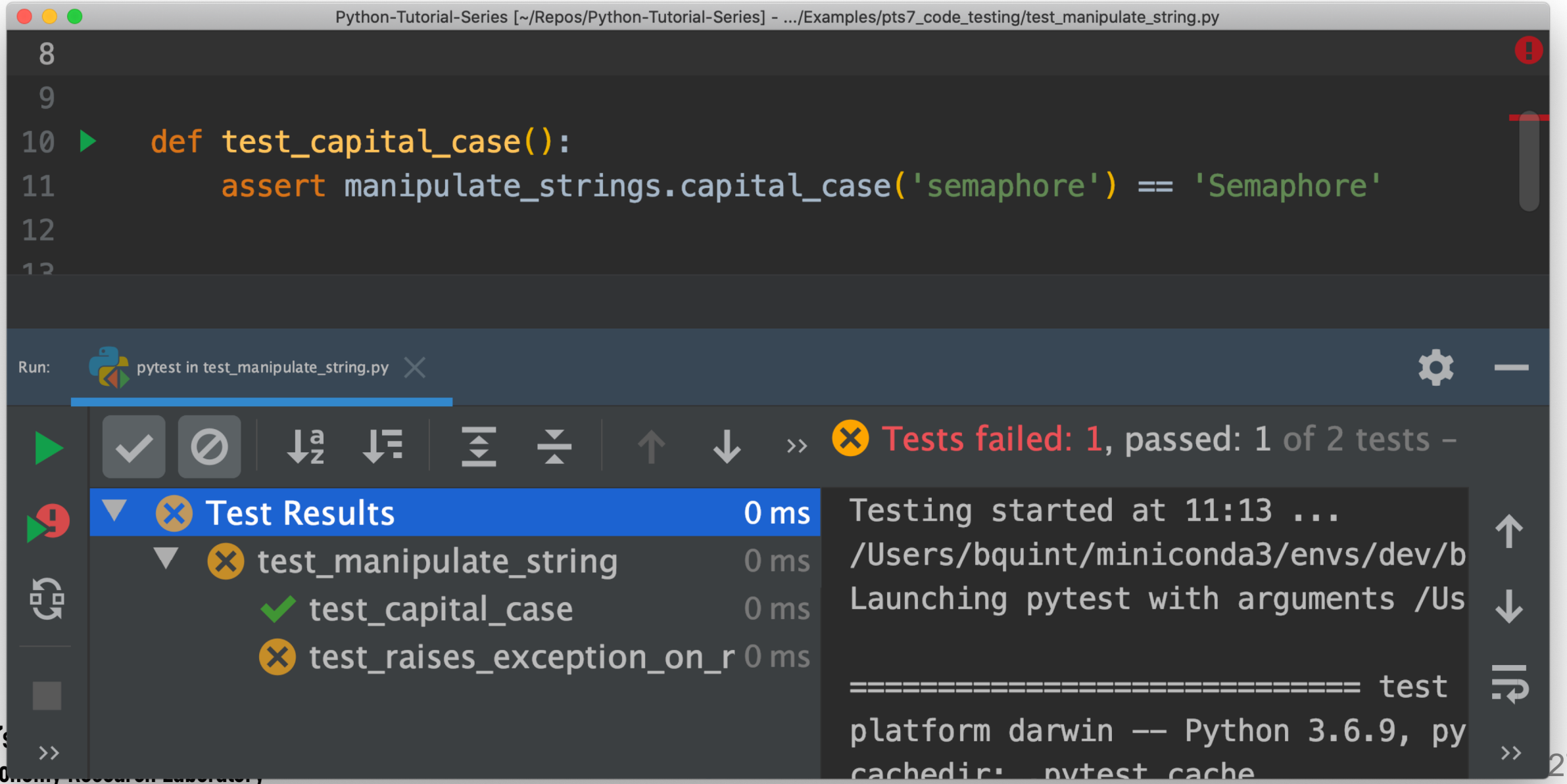
Python-Tutorial-Series [~/Repos/Python-Tutorial-Series] - .../Examples/pts7_code_testing/test_manipulate_string.py

```
6   import manipulate_strings
7   import pytest
8
9
10  def test_capital_case():
11      assert manipulate_strings.capital_case('semaphore') == 'Semaphore'
12
13
14  def test_raises_exception_on_non_string_arguments():
15      with pytest.raises(TypeError):
16          manipulate_strings.capital_case(9)
17
18
19  if __name__ == '__main__':
20      pytest.main()
```

Testing


Python-Tutorial-Series [~/Repos/Python-Tutorial-Series] - .../Examples/pts7_code_te
6 import manipulate_strings
7 import pytest
8
9
10 def test_capital_case():
11 assert manipulate_strings.capital_case('sem
12
13
14 def test_raises_exception_on_non_string_argumen
15 with pytest.raises(TypeError):
16 manipulate_strings.capital_case(9)
17
18
19 if __name__ == '__main__':
20 pytest.main()
Show Context Actions ⌥⏎
Copy Reference ⌥⇧⌘C
Paste ⌘V
Paste from History... ⇧⌘V
Paste without Formatting ⌥⇧⌘V
Column Selection Mode ⇧⌘8
Refactor
Folding
Go To
Generate... ⌘N
Run 'pytest in test_manip...' ^⇧R
Debug 'pytest in test_manip...' ^⇧D
Save 'pytest in test_manip...'
Reveal in Finder
Open in Terminal
Local History
Git
External Tools

Testing


Python-Tutorial-Series [~/Repos/Python-Tutorial-Series] - .../Examples/pts7_code_testing/test_manipulate_string.py
8
9
10 def test_capital_case():
11 assert manipulate_strings.capital_case('semaphore') == 'Semaphore'
12
Run:
pytest in test_manipulate_string.py
Tests failed: 1, passed: 1 of 2 tests –
Test Results 0 ms
test_manipulate_string 0 ms
test_capital_case 0 ms
test_raises_exception_on_r 0 ms
Testing started at 11:13 ...
/Users/bquint/miniconda3/envs/dev/b
Launching pytest with arguments /Us
========== test
platform darwin -- Python 3.6.9, py
cachedir: .pytest_cache

Testing

```
def test_can_perform_task(_):
    ...
    assert test_was_performed()
```

Yes

```
def test_cpt(_):
    ...
    assert test_was_performed()
```

No

```
def test_1(_):
    ...
    assert test_was_performed()
```

NEVER!!!

Project Structure

```
+--- README.rst
+--- LICENSE
+--- setup.py
+--- requirements.txt
```

```
+--- sample/
|   +--- __init__.py
|   +--- core.py
|   +--- helpers.py
...
```

```
+--- docs/
|   +--- conf.py
|   +--- index.rst
...
```

```
+--- tests/
|   +--- __init__.py
|   +--- conftest.py
|   +--- test_core.py
...
```

Project Structure

```
+--- README.rst
+--- LICENSE
+--- setup.py
+--- requirements.txt



+--- docs/
|    +--- conf.py
|    +--- index.rst
...
```

```
    +--- sample/
    |    +--- __init__.py
    |    +--- core.py
    |    +--- helpers.py
    |    +--- tests/
    |    |    +--- __init__.py
    |    |    +--- conftest.py
    |    |    +--- test_core.py
    ...
```

Project Structure

```
+--- README.rst
+--- LICENSE
+--- setup.py
+--- requirements.txt
+--- .gitignore
+--- bin/
+--- scrips/
+--- docs/
|    +--- conf.py
|    +--- index.rst
...
```

```
+--- sample/
|    +--- __init__.py
|    +--- core.py
|    +--- helpers.py
|    +--- tests/
|    |    +--- __init__.py
|    |    +--- conftest.py
|    |    +--- test_core.py
...
```

Project Structure setup.py

Using setuptools

```
# contents of setup.py
from setuptools import setup, find_packages

setup(
    name="HelloWorld",
    version="0.1",
    packages=find_packages(),
)
```

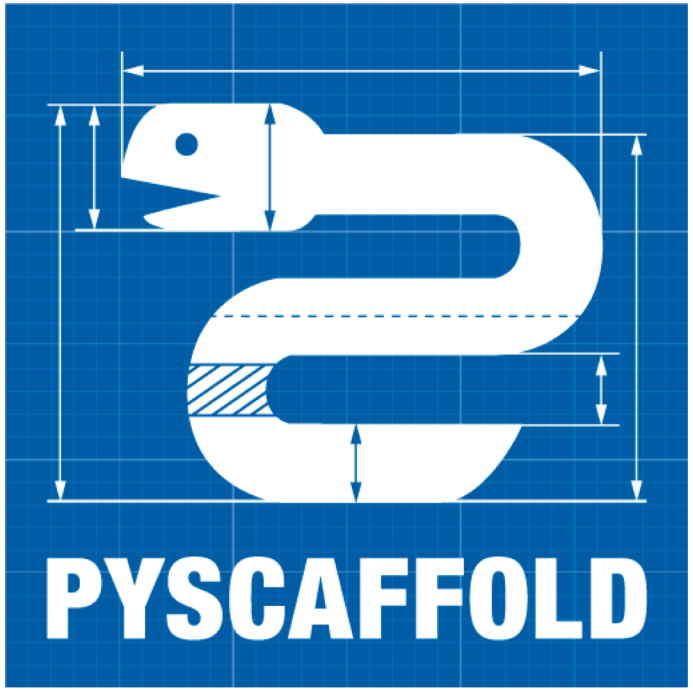
Project Structure setup.py and setup.cfg

Using setuptools

```
# contents of setup.cfg
[metadata]
name = HelloWorld
version = 0.1

[options]
packages = find:
```

```
# contents of setup.py
from setuptools import setup, find_packages
setup()
```

Project Structure

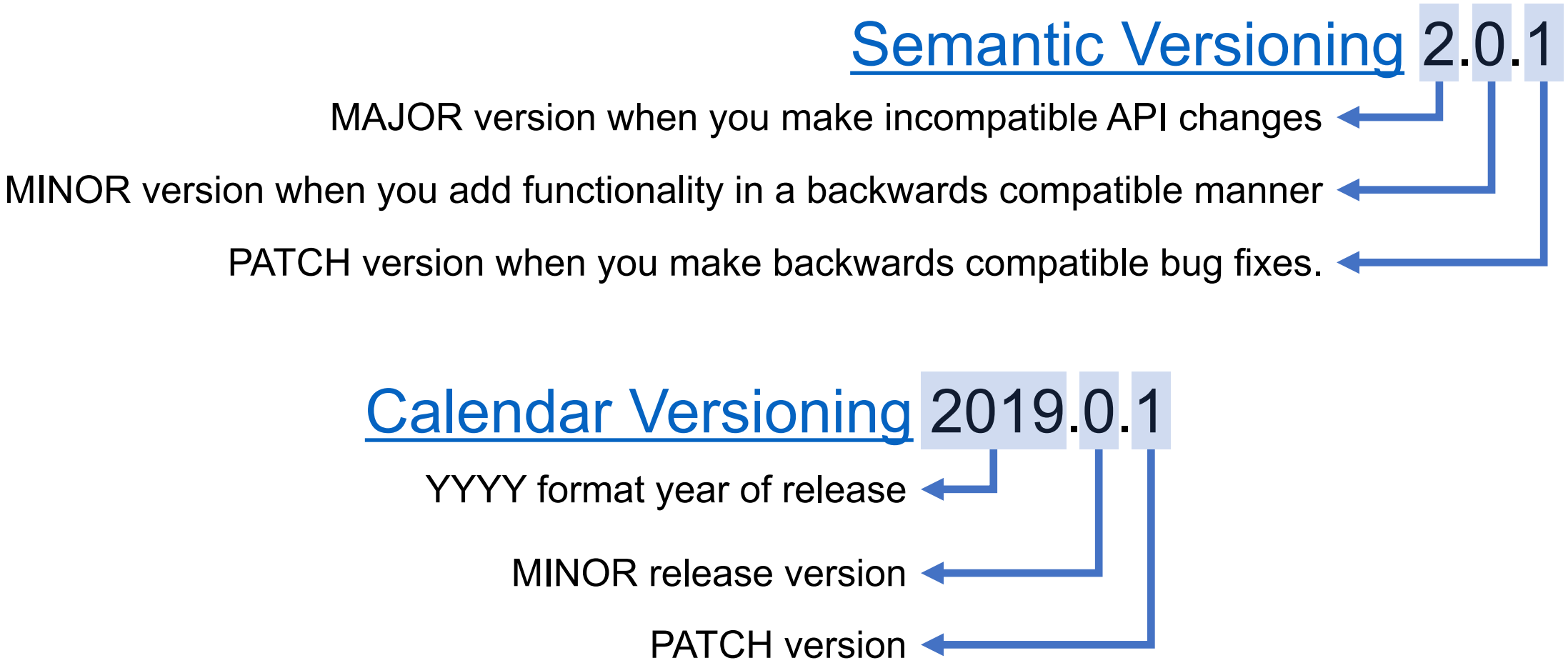



```
$ putup my_project
```

```
$ ls my_project

AUTHORS.rst          requirements.txt
CHANGELOG.rst        setup.cfg
LICENSE.txt          setup.py
README.rst           src
docs                 tests
```

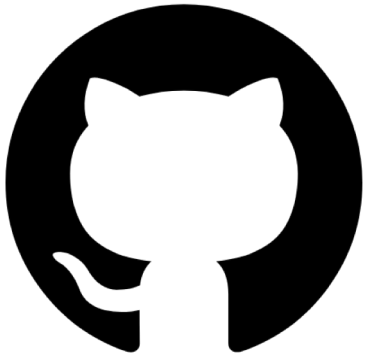
Link to PyScaffold documentation

Versioning and Version Control



Versioning and Version Control

Git Online Book (free)



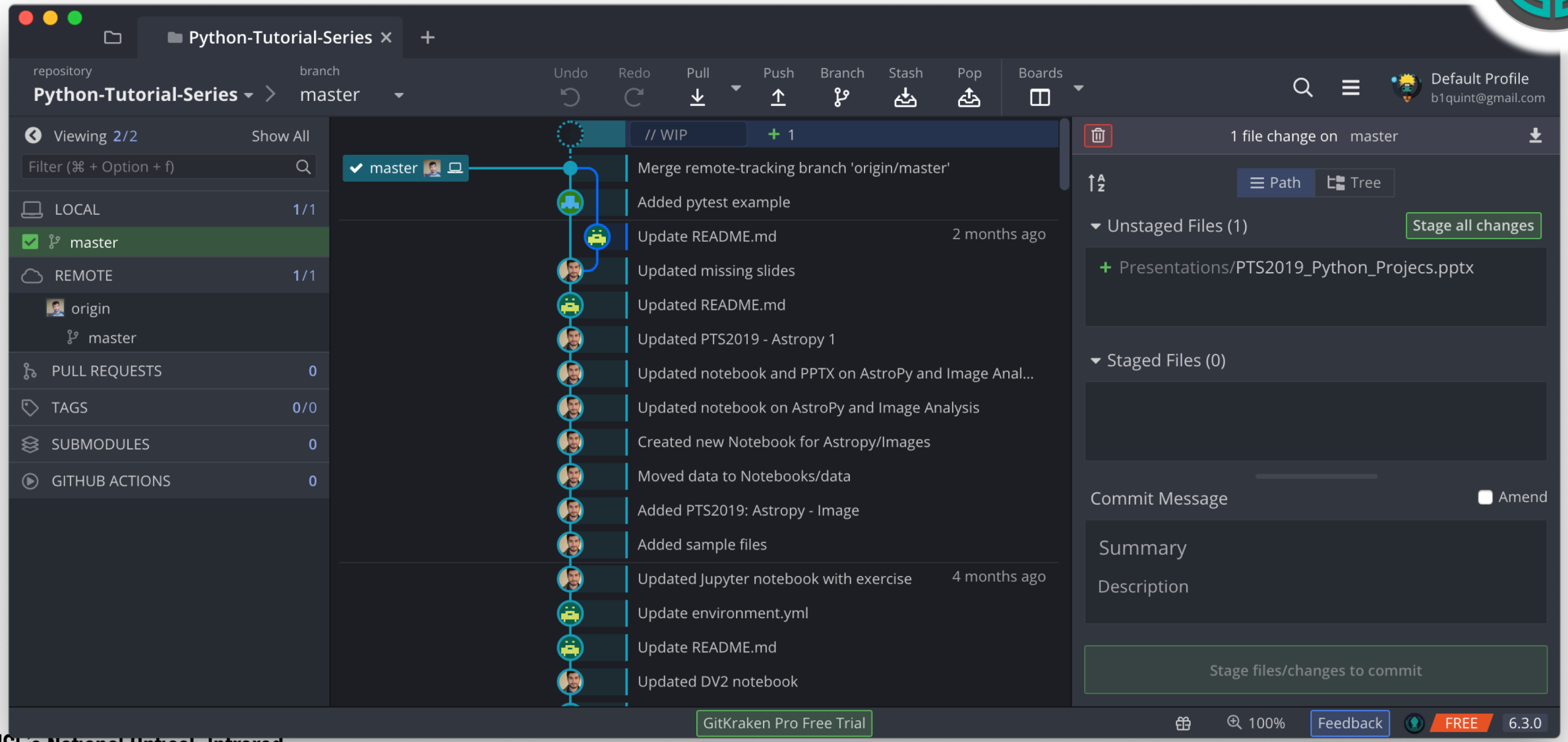
GitHub (free)

GitLab
(paid – available for Gemini Users)

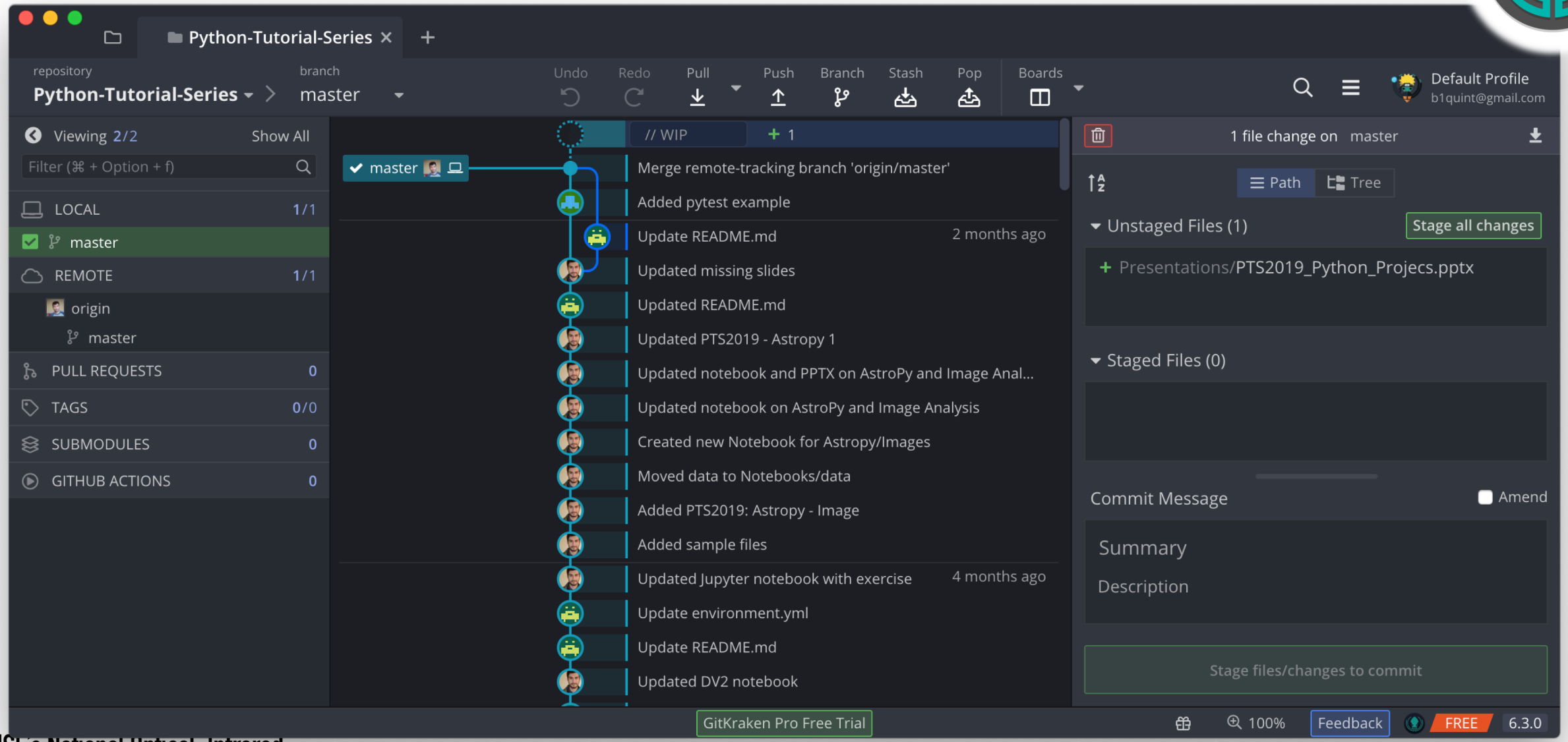
PyCharm CE (free)

GitKraken (free)

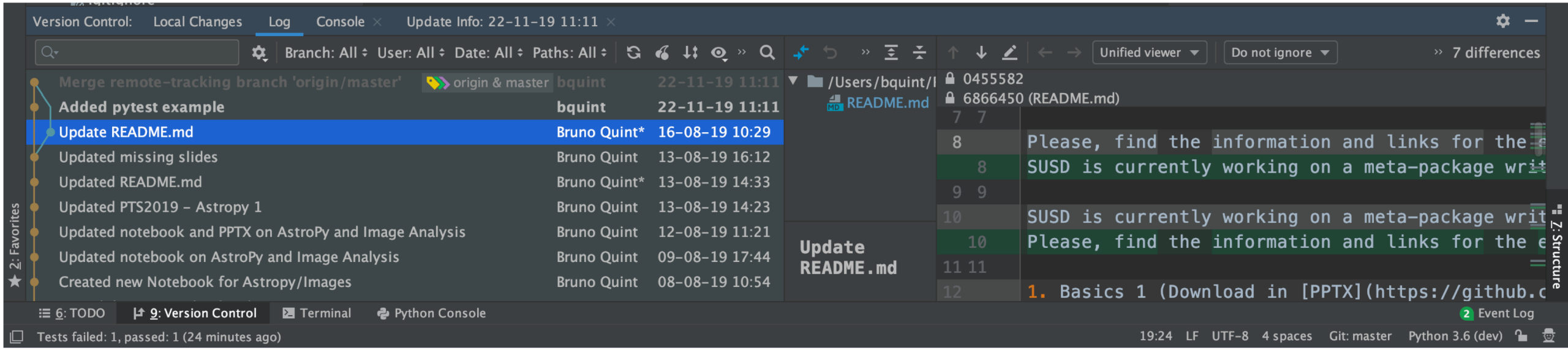
Versioning and Version Control

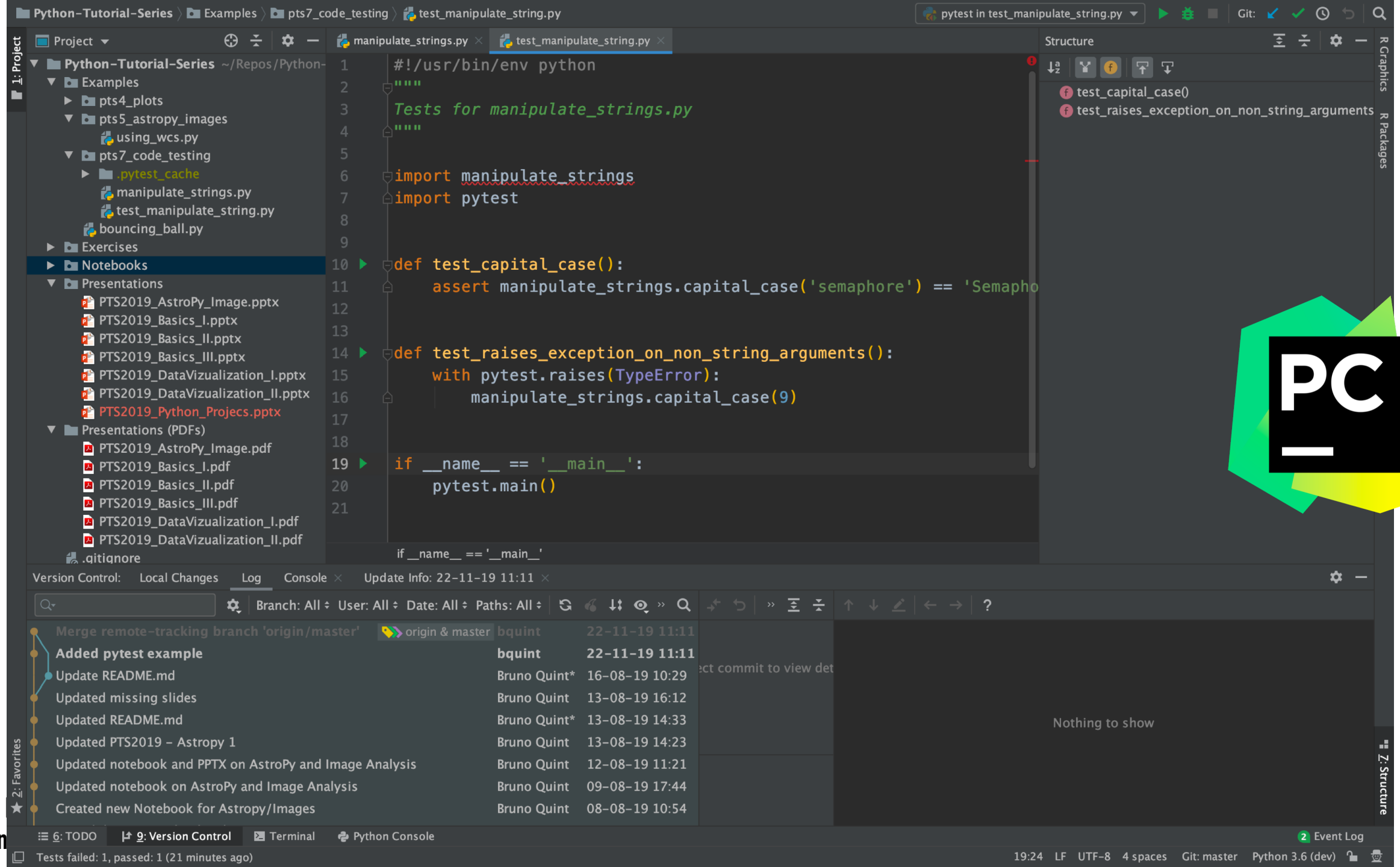


Versioning and Version Control



Versioning and Version Control




Python-Tutorial-Series
Examples
pts7_code_testing
test_manipulate_string.py
pytest in test_manipulate_string.py
Git:
Project
manipulate_strings.py
test_manipulate_string.py
Structure
test_capital_case()
test_raises_exception_on_non_string_arguments
Python-Tutorial-Series ~/Repos/Python-
Examples
pts4_plots
pts5_astropy_images
using_wcs.py
pts7_code_testing
.pytest_cache
manipulate_strings.py
test_manipulate_string.py
bouncing_ball.py
Exercises
Notebooks
Presentations
PTS2019_AstroPy_Image.pptx
PTS2019_Basics_I.pptx
PTS2019_Basics_II.pptx
PTS2019_Basics_III.pptx
PTS2019_DataVizualization_I.pptx
PTS2019_DataVizualization_II.pptx
PTS2019_Python_Projecs.pptx
Presentations (PDFs)
PTS2019_AstroPy_Image.pdf
PTS2019_Basics_I.pdf
PTS2019_Basics_II.pdf
PTS2019_Basics_III.pdf
PTS2019_DataVizualization_I.pdf
PTS2019_DataVizualization_II.pdf
.gitignore
#!/usr/bin/env python
"""
Tests for manipulate_strings.py
"""
import manipulate_strings
import pytest
def test_capital_case():
assert manipulate_strings.capital_case('semaphore') == 'Semapho
def test_raises_exception_on_non_string_arguments():
with pytest.raises(TypeError):
manipulate_strings.capital_case(9)
if __name__ == '__main__':
pytest.main()
if __name__ == '__main__'
Version Control:
Local Changes
Log
Console
Update Info: 22-11-19 11:11
Branch: All
User: All
Date: All
Paths: All
Merge remote-tracking branch 'origin/master' origin & master bquint 22-11-19 11:11
Added pytest example bquint 22-11-19 11:11
Update README.md Bruno Quint* 16-08-19 10:29
Updated missing slides Bruno Quint 13-08-19 16:12
Updated README.md Bruno Quint* 13-08-19 14:33
Updated PTS2019 - Astropy 1 Bruno Quint 13-08-19 14:23
Updated notebook and PPTX on AstroPy and Image Analysis Bruno Quint 12-08-19 11:21
Updated notebook on AstroPy and Image Analysis Bruno Quint 09-08-19 17:44
Created new Notebook for Astropy/Images Bruno Quint 08-08-19 10:54
ct commit to view det
Nothing to show
6: TODO
9: Version Control
Terminal
Python Console
Event Log
Tests failed: 1, passed: 1 (21 minutes ago)
19:24 LF UTF-8 4 spaces Git: master Python 3.6 (dev)
1: Project
2: Favorites
R Graphics
R Packages
Z: Structure

Versioning and Version Control

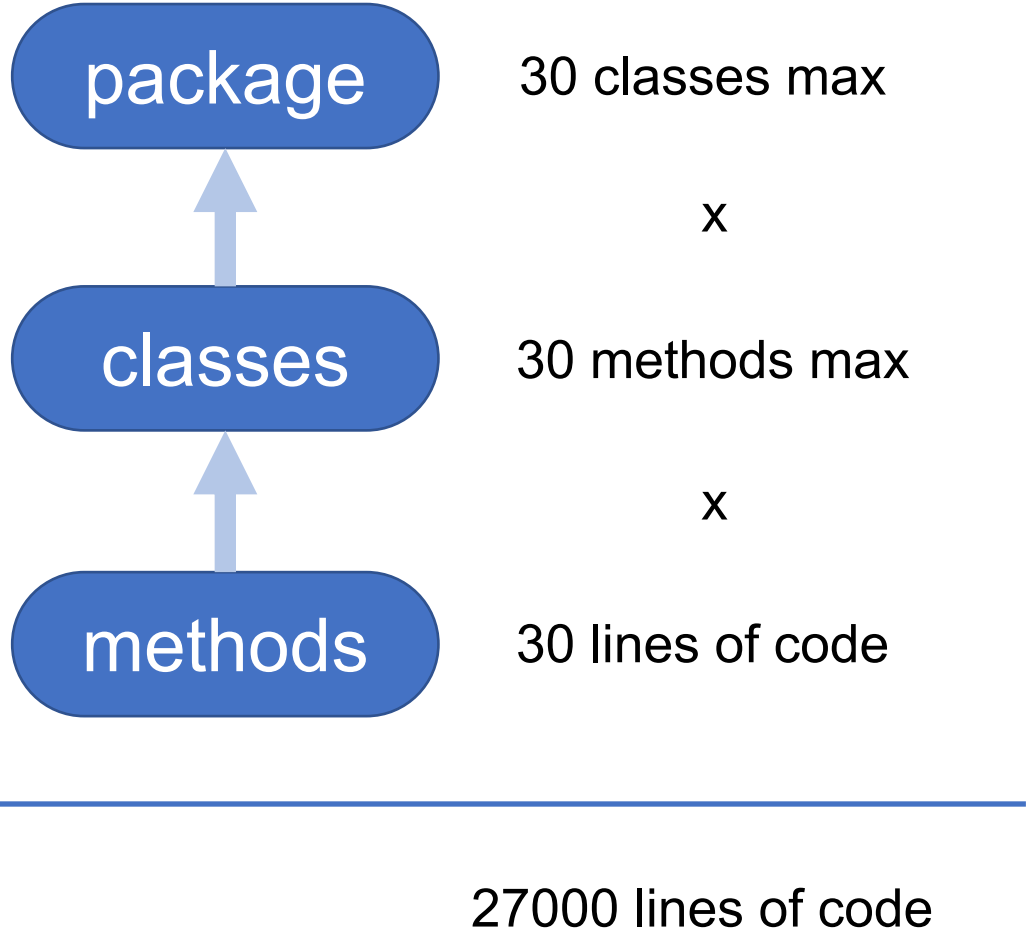
Write meaningful commit messages

https://xkcd.com/1296/

Before you go...

Rule of 3

- Three strikes and you refactor
- Duplication: hard to maintain
- Seeing some cases helps code design



Rule of 30

If an element consists of more than 30 subelements, it is highly probable that there is a serious problem.

Before you go...

```
def snellsLaw(n, t):
    return np.rad2deg(np.arcsin(np.sin(np.deg2rad(t))/n))
```

Before you go...

```
def snells_law(incident_angle, refractive_index):
    """ ... """
    theta = np.deg2rad(indicent_angle)
    n = refractive_index

    temp = np.sin(theta) / n
    temp = np.arcsin(theta)
    output_angle = np.rad2deg(temp)

    return output_angle
```

Questions?