

## UNIT 6: EXPLORING LINTERS TO SUPPORT TESTING IN PYTHON

Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86\_64)

- \* Documentation: <https://help.ubuntu.com>
- \* Management: <https://landscape.canonical.com>
- \* Support: <https://ubuntu.com/advantage>

\* Welcome to the Codio Terminal!

\* <https://docs.codio.com/project/ide/boxes/#overview>

\* Your Codio Box domain is: music-media.codio.io

Last login: Tue Dec 21 04:59:58 2021 from 192.168.11.51  
codio@music-media:~/workspace\$

## Unit 6: Exploring Linters to Support Testing in Python

These activities are to be carried out in line with the content of Unit 6 of the SSD module. These are the Python programs that will support your practical work this Unit and can be seen in the filetree:

- metricTest.py
- pylintTest.py
- styleLint.py
- sums.py
- sums2.py

Run your code using `python3 [filename].py` (or use the rocket icon)

Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86\_64)

- \* Documentation: <https://help.ubuntu.com>
- \* Management: <https://landscape.canonical.com>
- \* Support: <https://ubuntu.com/advantage>

\* Welcome to the Codio Terminal!

\* <https://docs.codio.com/project/ide/boxes/#overview>

\* Your Codio Box domain is: music-media.codio.io

Last Login: Tue Dec 28 03:08:14 2021 from 192.168.10.156  
codio@music-media:~/workspace\$

## 3. Unit 6 - Testing with Python

### Exploring Linters to Support Testing in Python: Question 1

Run `styleLint.py`.

- What happens when the code is run?
- Can you modify this code for a more favourable outcome?
- What amendments have you made to the code?

```
# CODE SOURCE: SOFTWARE ARCHITECTURE WITH PYTHON
```

```
def factorial(n):
    """ Return factorial of n """
    if n == 0:
        return 1
    else:
        return n*factorial(n-1)
```

The screenshot shows a Codio IDE interface. On the left, the file tree displays a project named 'Testing with Python' containing files like .settings, equivalence.py, metricTest.py, pyflintTest.py, README.md, styleLint.py, sums.py, and sums2.py. The main workspace shows a terminal window with the code for 'styleLint.py' and a browser window displaying a course slide titled '3. Unit 6 - Testing with Python'.

**Terminal:**

```
1 # CODE SOURCE: SOFTWARE ARCHITECTURE WITH PYTHON
2
3
4 def factorial(n):
5     """ Return factorial of n """
6     if n == 0:
7         return 1
8     else:
9         return n*factorial(n-1)
```

**Browser (3. Unit 6 - Testing with Python):**

### 3. Unit 6 - Testing with Python

#### Exploring Linters to Support Testing in Python: Question 1

Run `styleLint.py`.

- What happens when the code is run?
- Can you modify this code for a more favourable outcome?
- What amendments have you made to the code?

Next ►

Course: Secure Software Development | Codio - Testing with Python

Filetree

NPIRBHAI-JETHA  
Testing with Python...

styleLint.py

```
1 # CODE SOURCE: SOFTWARE ARCHITECTURE WITH PYTHON
2
3 def factorial(n):
4     """ Return factorial of n """
5     if n == 0:
6         return 1
7     else:
8         return n*factorial(n-1)
```

Terminal

styleLint.py

Run

3. Unit 6 - Testing with Python

Exploring Linters to Support Testing in Python: Question 1

Run `styleLint.py`.

- What happens when the code is run?
- Can you modify this code for a more favourable outcome?
- What amendments have you made to the code?

Next ▶

100% (9:27) Python

ENG UK 07:26 28/12/2021

Course: Secure Software Development | Codio - Testing with Python

Filetree

NPIRBHAI-JETHA  
Testing with Python...

styleLint.py

```
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

*
* Welcome to the Codio Terminal!
*
* https://docs.codio.com/project/ide/boxes/#overview
*
* Your Codio Box domain is: music-media.codio.io
*
Last login: Tue Dec 28 03:18:54 2021 from 192.168.10.156
codio@music-media:~/workspace$ python3 styleLint.py
  File "styleLint.py", line 5
    """ Return factorial of n """
           ^
IndentationError: expected an indented block
codio@music-media:~/workspace$
```

Terminal

styleLint.py

Run

3. Unit 6 - Testing with Python

Exploring Linters to Support Testing in Python: Question 1

Run `styleLint.py`.

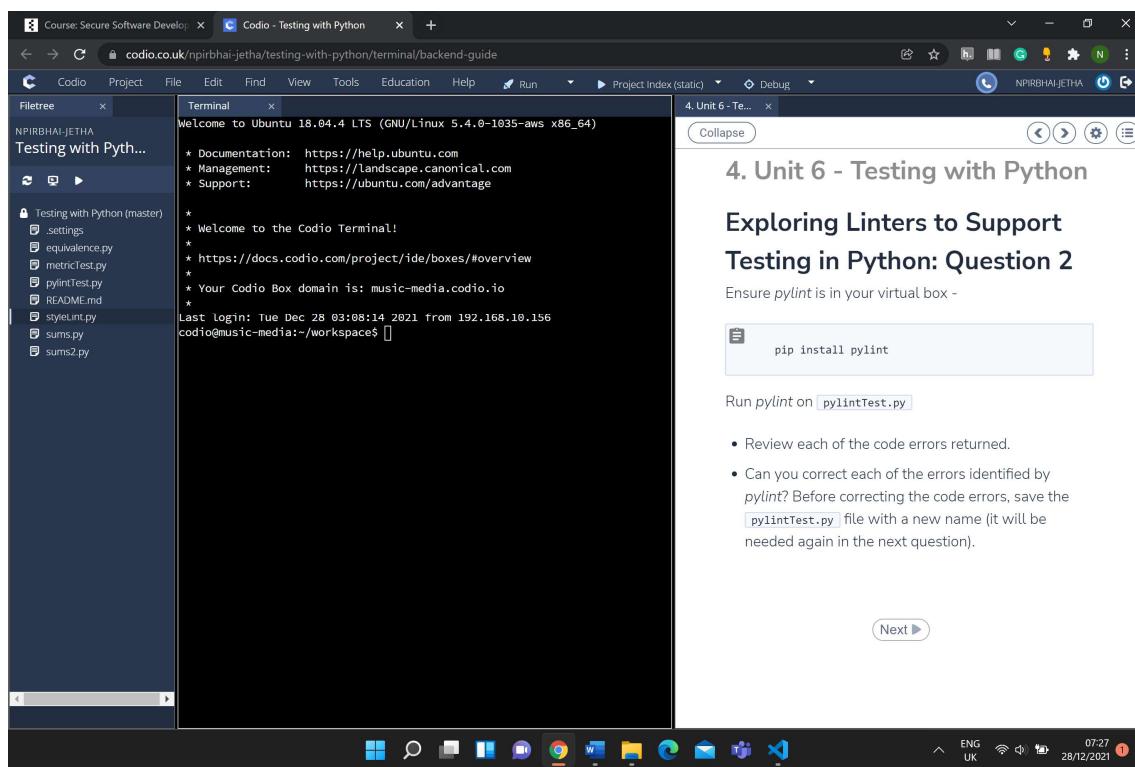
- What happens when the code is run?
- Can you modify this code for a more favourable outcome?
- What amendments have you made to the code?

Add indents:

```
def factorial(n):
    """ Return factorial of n """
    if n == 0:
        return 1
    else:
        return n*factorial(n-1)
```

Indentation Error

07:19 28/12/2021



# SOURCE OF CODE: <https://docs.pylint.org/en/1.6.0/tutorial.html>

```
import string

shift = 3

choice = raw_input("would you like to encode or decode?")

word = (raw_input("Please enter text"))

letters = string.ascii_letters + string.punctuation + string.digits

encoded = ""

if choice == "encode":

    for letter in word:

        if letter == ' ':

            encoded = encoded + ' '

        else:

            x = letters.index(letter) + shift

            encoded=encoded + letters[x]
```

```

if choice == "decode":
    for letter in word:
        if letter == ' ':
            encoded = encoded + ''
        else:
            x = letters.index(letter) - shift
            encoded = encoded + letters[x]

```

print encoded

The screenshot shows the Codio IDE interface. On the left, the 'Filetree' sidebar displays a project structure for 'Testing with Python (master)' containing files like .settings, equivalence.py, metricTest.py, pylintTest.py, README.md, styleUnit.py, sums.py, and sums2.py. The main workspace shows the code for 'pylintTest.py' with line numbers 1 through 26. The code implements a Caesar cipher-like function. The right side features a sidebar titled '4. Unit 6 - Testing with Python' which includes sections for 'Exploring Linters to Support Testing in Python:' and 'Question 2'. It also contains a terminal window with the command 'pip install pylint' and instructions to run pylint on the file.

```

1 # SOURCE OF CODE: https://docs.pylint.org/en/1.6.0/tutorial.html
2
3 import string
4
5 shift = 3
6 choice = raw_input("would you like to encode or decode?")
7 word = raw_input("Please enter text")
8 letters = string.ascii_letters + string.punctuation + string.digits
9 encoded = ''
10
11 if choice == "encode":
12     for letter in word:
13         if letter == ' ':
14             encoded = encoded + ''
15         else:
16             x = letters.index(letter) + shift
17             encoded=encoded + letters[x]
18 if choice == "decode":
19     for letter in word:
20         if letter == ' ':
21             encoded = encoded + ''
22         else:
23             x = letters.index(letter) - shift
24             encoded = encoded + letters[x]
25
26 print encoded

```

The screenshot shows a Codio IDE interface. On the left is a filetree with files: .settings, equivalence.py, metricTest.py, pylintTest.py, README.md, styleLint.py, sums.py, and sums2.py. The main window is a terminal window titled 'pylintTest.py' with the command 'Run'. The output shows:

```
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

*
* Welcome to the Codio Terminal!
*
* https://docs.codio.com/project/ide/boxes/#overview
*
* Your Codio Box domain is: music-media.codio.io
*
Last login: Tue Dec 28 03:28:54 2021 from 192.168.10.156
codio@music-media:~/workspace$ python3 pylintTest.py
  File "pylintTest.py", line 26
    print encoded
           ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print(encoded)?
codio@music-media:~/workspace$
```

A blue oval highlights the line 'print encoded'.

On the right, a sidebar titled '4. Unit 6 - Testing with Python' contains:

## Exploring Linters to Support Testing in Python: Question 2

Ensure pylint is in your virtual box -

```
pip install pylint
```

Run pylint on pylintTest.py

- Review each of the code errors returned.
- Can you correct each of

07:34 28/12/2021

The screenshot shows a Codio IDE interface. On the left is a filetree with files: .settings, equivalence.py, metricTest.py, pylint2.py, pylintTest.py, README.md, styleLint.py, sums.py, and sums2.py. The main window is a terminal window titled 'pylintTest.py' with the command 'Run'. The output shows:

```
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

*
* Welcome to the Codio Terminal!
*
* https://docs.codio.com/project/ide/boxes/#overview
*
* Your Codio Box domain is: music-media.codio.io
*
Last login: Tue Dec 28 03:34:44 2021 from 192.168.10.156
codio@music-media:~/workspace$ python3 pylint2.py
Traceback (most recent call last):
  File "pylint2.py", line 7, in <module>
    choice = raw_input("would you like to encode or decode?")
NameError: name 'raw_input' is not defined
codio@music-media:~/workspace$
```

A blue box highlights the line 'choice = raw\_input("would you like to encode or decode?")' with the text 'raw\_input is creating issues. Just use input'.

On the right, a sidebar titled '4. Unit 6 - Testing with Python' contains:

## Exploring Linters to Support Testing in Python: Question 2

Ensure pylint is in your virtual box -

```
pip install pylint
```

Run pylint on pylintTest.py

- Review each of the code errors returned.
- Can you correct each of

07:37 28/12/2021

Seminar 3 Preparation (and e-po... | Course: Secure Software Develop... | Codio - Testing with Python

Filetree x

NPIRBHAI-JETHA  
Testing with Pyt...

Testing with Python (master)

- settings
- equivalence.py
- metricTest.py
- pylint2.py
- pylintTest.py
- README.md
- styleLint.py
- sums.py
- sums2.py

pylint2.py x Terminal Run

```
1 # SOURCE OF CODE: https://docs pylint.org/en/1.6.0/tutorial.htm
2
3 import string
4
5 shift = 3
6
7 choice = input("would you like to encode or decode?")
8 word = input("Please enter text")
9 letters = string.ascii_letters + string.punctuation + string.digits
10 encoded = ''
11 if choice == "encode":
12     for letter in word:
13         if letter == ' ':
14             encoded = encoded + ' '
15         else:
16             x = letters.index(letter) + shift
17             encoded=encoded + letters[x]
18     if choice == "decode":
19         for letter in word:
20             if letter == ' ':
21                 encoded = encoded + ' '
22             else:
23                 x = letters.index(letter) - shift
24                 encoded = encoded + letters[x]
25
26 print(encoded)
```

4. Unit 6 - Test... x

Collapse

## 4. Unit 6 - Testing with Python

### Exploring Linters to Support Testing in Python: Question 2

Ensure pylint is in your virtual box -

```
pip install pylint
```

Run pylint on pylintTest.py

- Review each of the code errors returned.
- Can you correct each of the errors identified by pylint? Before correcting the code errors, save the pylintTest.py file with a new name (it will be needed again in the next question).

Next ►

ENG US 09:35 28/12/2021

Seminar 3 Preparation (and e-po... | Course: Secure Software Develop... | Codio - Testing with Python

Filetree x

NPIRBHAI-JETHA  
Testing with Pyt...

Testing with Python (master)

- settings
- equivalence.py
- metricTest.py
- pylint2.py
- pylintTest.py
- README.md
- styleLint.py
- sums.py
- sums2.py

pylint2.py Terminal Run

```
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

*
* Welcome to the Codio Terminal!
*
* https://docs.codio.com/project/ide/boxes/#overview
*
* Your Codio Box domain is: music-media.codio.io
*
Last Login: Tue Dec 28 05:29:02 2021 from 192.168.10.156
codio@music-media:~/workspaces$ python3 pylint2.py
would you like to encode or decode?
```

4. Unit 6 - Test... x

Collapse

## 4. Unit 6 - Testing with Python

### Exploring Linters to Support Testing in Python: Question 2

Ensure pylint is in your virtual box -

```
pip install pylint
```

Run pylint on pylintTest.py

- Review each of the code errors returned.
- Can you correct each of the errors identified by pylint? Before correcting the code errors, save the pylintTest.py file with a new name (it will be needed again in the next question).

Next ►

ENG US 09:35 28/12/2021

Course: Secure Software Development | Codio - Testing with Python

codio.co.uk/npirbhajetha/testing-with-python/terminal/backend-1640663102644

**Filetree**

NPIRBHAI-JETHA  
Testing with Python...

- Testing with Python (master)
  - .settings
  - equivalence.py
  - metricTest.py
  - pylint2.py
  - pylintTest.py
  - rtAUTM.mdd
  - styleLint.py
  - sums.py
  - sums2.py

**Terminal**

```

pylint2.py
7556e9b7782df118c1f49bdc494da5e5e429c93aa77965f33e81287c8c/z
ipp-1.2.0-py2.py3-none-any.whl
Collecting contextlib2; python_version < "3" (from importlib-
metadata; python_version < "3.8"->flake8)
  Downloading https://files.pythonhosted.org/packages/85/e0/70352f7ef6aa95c52fb001831622f50f923cd575427d021b0ab3311236/c
contextlib2-0.6.0.post1-py2.py3-none-any.whl
Collecting pathlib2; python_version < "3" (from importlib-met
adata; python_version < "3.8"->flake8)
  Downloading https://files.pythonhosted.org/packages/67/d
c02c72177ec79f0176e5bf9921e9c1745a381ed556afbb3b3ec2cbb8bae/p
athlib2-2.3.6-py2.py3-none-any.whl
Collecting six (from pathlib2; python_version < "3"->importli
b-metadata; python_version < "3.8"->flake8)
  Using cached https://files.pythonhosted.org/packages/d9/5a/
e7c31adbe75f2abb91bd84cf2dc52d792b5aa01506781dbcf25c91daf11/
six-1.16.0-py2.py3-none-any.whl
Collecting scandir; python_version < "3.5" (from pathlib2; py
thon_version < "3"->importlib-metadata; python_version < "3.8
"->flake8)
  Downloading https://files.pythonhosted.org/packages/df/f5/9
c052db7bd54d0cbf1bc0bb6554362bb01012d03e5888950a4f5c5dac4e/s
candir-1.10.0.tar.gz
Building wheels for collected packages: functools32, scandir
  Running setup.py bdist_wheel for functools32 ... done
  Stored in directory: /home/codio/.cache/pip/wheels/b5/18/32/
77a1030457155606ba5e3ec3a8a57132b1a64b1c4f765177b2
  Running setup.py bdist_wheel for scandir ... done
  Stored in directory: /home/codio/.cache/pip/wheels/91/95/75/
19c98a91239878abb7c59970abd304e0438a7dd5b61778335
Successfully built functools32 scandir
Installing collected packages: configparser, pyflakes, functo
ols32, contextlib2, zipp, six, scandir, pathlib2, importlib-m
etadata, typing, enum34, mccabe, pycodestyle, flake8
Successfully installed configparser-4.0.2 contextlib2-0.6.0.p
ost1 enum34-1.1.10 flake8-3.9.2 functools32-3.2.3.post2 impor
tlib-metadata-2.1.2 mccabe-0.6.1 pathlib2-2.3.6 pycodestyle-2
.7.0 pyflakes-2.3.1 scandir-1.10.0 six-1.16.0 typing-3.10.0.0
.zipp-1.2.0
codio@music-media:~/workspace$ 
```

5. Unit 6 - Testing with Python

## Exploring Linters to Support Testing in Python: Question 3

Ensure flake8 is in your virtual box -

```
pip install flake8
```

Run flake8 on `pylintTest.py`.

- Review the errors returned. In what way does this error message differ from the error message returned by pylint?

Run flake8 on `metricTest.py`.

- Can you correct each of the errors returned by flake8?
- What amendments have you made to the code?

**Next ▶**

ENG US 07:45 28/12/2021

flake8 pylint2.py

In what way does this error message differ from error message returned by pylint?

More detailed, gives more information about the 'ugly' code lines (such as whitespaces etc.)

- **Pylint(default):** Checks for errors and tries to enforce a coding standard
- **Flake8:** Checks code against style conventions in PEP 8, programming errors and cyclomatic complexity

in Python:  
Question 3

Ensure flake8 is in your virtual box

```
1 # SOURCE OF CODE: https://docs.pylint.org/en/1.6.0/tutorial.html
2
3 import string
4
5 shift = 3
6
7 choice = input("would you like to encode or decode?")
8 word = input("Please enter text")
9 letters = string.ascii_letters + string.punctuation + string.digits
10 encoded = ''
11 if choice == "encode":
12     for letter in word:
13         if letter == ' ':
14             encoded = encoded + ' '
15         else:
16             x = letters.index(letter) + shift
17             encoded=encoded + letters[x]
18     if choice == "decode":
19         for letter in word:
20             if letter == ' ':
21                 encoded = encoded + ' '
22             else:
23                 x = letters.index(letter) - shift
24                 encoded = encoded + letters[x]
25
26 print(encoded)
```

CHECK THIS YOUTUBE VIDEO FOR FLAKE8

<https://www.youtube.com/watch?v=TDUf93vqo3g>

Run `flake8` on `metricTest.py`.

The screenshot shows the Codio IDE interface. On the left is a filetree showing files like .settings, equivalence.py, metricTest.py, pylint2.py, pylintTest.py, README.md, stylelint.py, sums.py, and sums2.py. The main area has two panes: a terminal pane on the left and a content pane on the right. The terminal pane shows the command "codio@music-media:~/workspace\$ flake8 metricTest.py" and its output:

```
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86_64)
 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch
*
* Welcome to the Codio Terminal!
*
* https://docs.codio.com/project/ide/boxes/#overview
*
* Your Codio Box domain is: music-media.codio.io
*
Last login: Tue Dec 28 06:49:16 2021 from 192.168.10.156
codio@music-media:~/workspace$ flake8 metricTest.py
```

The content pane on the right is titled "5. Unit 6 - Testing with Python" and contains a section titled "Python: Question 3". It asks to ensure `flake8` is installed and provides a command to do so:

```
pip install flake8
```

It also lists several questions for review:

- Review the errors returned. In what way does this error message differ from the error message returned by `pylint`?
- Run `flake8` on `metricTest.py`.
  - Can you correct each of the errors returned by `flake8`?
  - What amendments have you made to the code?

A "Next" button is visible at the bottom right.

This screenshot shows the same Codio IDE setup as the previous one, but the terminal output for `flake8 metricTest.py` now includes many errors:

```
* Your Codio Box domain is: music-media.codio.io
Last login: Tue Dec 28 06:49:16 2021 from 192.168.10.156
codio@music-media:~/workspace$ flake8 metricTest.py
metricTest.py:21:1: E265 block comment should start with '#'
metricTest.py:24:8: W291 trailing whitespace
metricTest.py:13:8: E999 SyntaxError: invalid syntax
metricTest.py:20:1: E112 expected an indented block
metricTest.py:21:1: E126 continuation line under-indented for visual indent
metricTest.py:22:1: E126 continuation line under-indented for visual indent
metricTest.py:23:1: E112 expected an indented block
metricTest.py:27:8: E225 missing whitespace around operator
metricTest.py:28:1: E112 expected an indented block
metricTest.py:30:3: E261 at least two spaces before inline comment
metricTest.py:31:8: E225 missing whitespace around operator
metricTest.py:31:17: E225 missing whitespace around operator
metricTest.py:32:1: E112 expected an indented block
metricTest.py:34:3: E261 at least two spaces before inline comment
metricTest.py:34:8: E501 line too long (83 > 79 characters)
metricTest.py:35:2: E201 Whitespace after '['
metricTest.py:35:5: E202 Whitespace before ']'
metricTest.py:36:8: E225 missing whitespace around operator
metricTest.py:37:1: E112 expected an indented block
metricTest.py:37:8: E225 missing whitespace around operator
metricTest.py:38:1: E112 expected an indented block
metricTest.py:38:3: E261 at least two spaces before inline comment
metricTest.py:39:22: E231 missing whitespace after ','
metricTest.py:40:18: E225 missing whitespace around operator
metricTest.py:41:1: E112 expected an indented block
metricTest.py:41:3: E261 at least two spaces before inline comment
metricTest.py:42:35: E231 missing whitespace after ','
metricTest.py:42:45: E231 missing whitespace after ','
metricTest.py:43:1: E128 continuation line under-indented for visual indent
metricTest.py:44:1: E128 continuation line under-indented for visual indent
metricTest.py:45:10: E225 missing whitespace around operator
metricTest.py:46:1: E112 expected an indented block
metricTest.py:46:3: E261 at least two spaces before inline comment
metricTest.py:48:1: E126 continuation line under-indented for visual indent
metricTest.py:52:1: E112 expected an indented block
metricTest.py:54:24: E231 missing whitespace after ','
metricTest.py:55:1: E112 expected an indented block
metricTest.py:59:1: E112 expected an indented block
```

The content pane remains the same as in the first screenshot, asking to ensure `flake8` is installed and providing a command to do so:

```
pip install flake8
```

It also lists several questions for review:

- Review the errors returned. In what way does this error message differ from the error message returned by `pylint`?
- Run `flake8` on `metricTest.py`.
  - Can you correct each of the errors returned by `flake8`?
  - What amendments have you made to the code?

A "Next" button is visible at the bottom right.

The screenshot shows the Codio IDE interface. On the left, the filetree shows a project named "Testing with Python" containing several files like settings, equivalence.py, metricTest.py, pylint2.py, pylintTest.py, styleLint.py, sums.py, and sums2.py. The terminal window on the left displays numerous flake8 errors from metricTest.py, such as E202 whitespace before ']' and E112 expected an indented block. To the right, a browser window displays a course page titled "5. Unit 6 - Testing with Python". The main content is "Python: Question 3" with the instruction "Ensure flake8 is in your virtual box -". It includes a code editor with the command "pip install flake8", a note to run flake8 on pylintTest.py, and a list of questions:

- Review the errors returned. In what way does this error message differ from the error message returned by pylint?
- Can you correct each of the errors returned by flake8?
- What amendments have you made to the code?

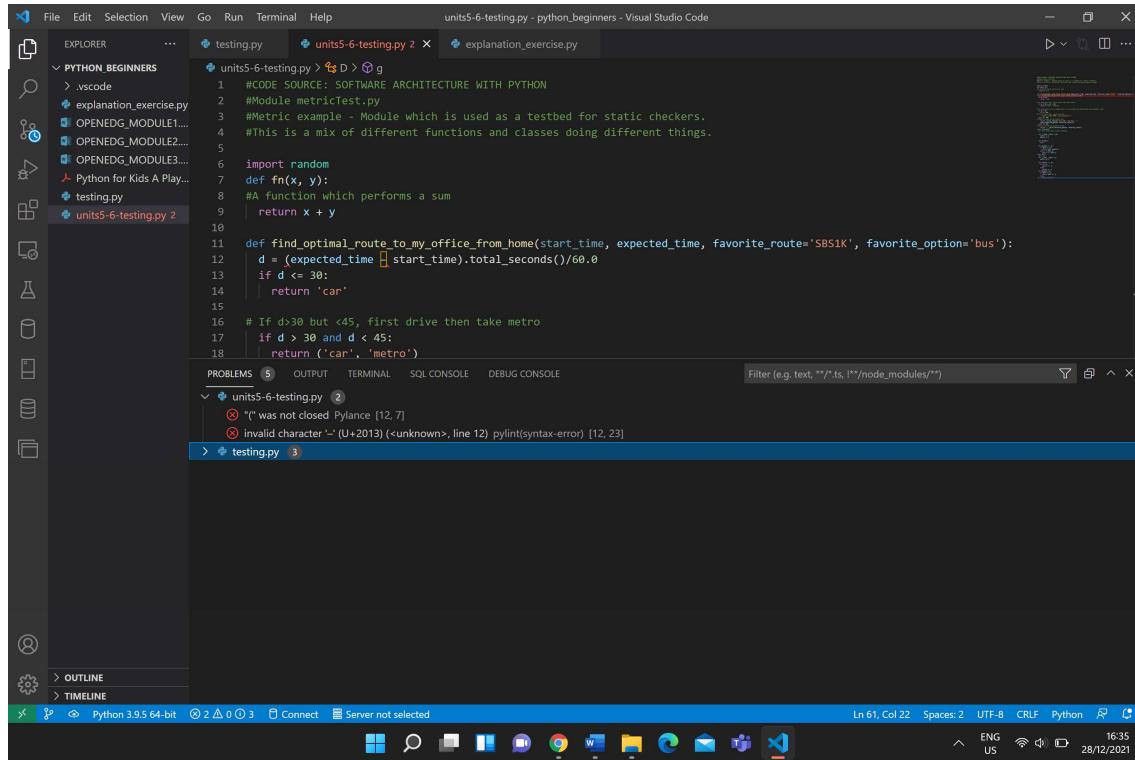
At the bottom right of the browser window, there is a "Next" button.

## Amendments made:

- Remove all numberings
- Check all indents

For "**Indentation is not a multiple of four (E111)**", PEP8 recommends that Python code indentation be a multiple of four. [<https://www.python.org/dev/peps/pep-0008/#indentation> or <https://www.flake8rules.com/rules/E111.html>]

- Minus sign checked



```

#Module metricTest.py
#Metric example - Module which is used as a testbed for static checkers.
#This is a mix of different functions and classes doing different things.

import random
def fn(x, y):
    #A function which performs a sum
    return x + y

def find_optimal_route_to_my_office_from_home(start_time, expected_time,
favorite_route='SBS1K', favorite_option='bus'):
    d = (expected_time-start_time).total_seconds()/60.0
    if d <= 30:
        return 'car'

    # If d>30 but <45, first drive then take metro
    if d > 30 and d < 45:
        return ('car', 'metro')

    # If d>45 there are a combination of optionsWriting Modifiable and Readable
    # Code
    if d > 45:
        if d < 60:
            # First volvo,then connecting bus
            return ('bus:335E', 'bus:connector')

```

```

        elif d > 80:
# Might as well go by normal bus
    return random.choice(('bus:330','bus:331',':'.
        join((favorite_option, favorite_route))))
elif d > 90:
# Relax and choose favorite route
    return ':' .join((favorite_option, favorite_route))
class C(object):
    #A class which does almost nothing

    def __init__(self, x,y):
        self.x = x
        self.y = y

    def f(self):
        pass

    def g(self, x, y):
        if self.x > x:
            return self.x+self.y
        elif x > self.x:
            return x + self.y
class D(C):
    #D class
    def __init__(self, x):
        self.x = x

    def f(self, x, y):
        if x > y:
            return x - y
        else:
            return x + y
    def g(self, y):
        if self.x > y:
            return self.x + y
        else:
            return y-self.x

```

Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86\_64)

```
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch
*
* Welcome to the Codio Terminal!
*
* https://docs.codio.com/project/ide/boxes/#overview
*
* Your Codio Box domain is: music-media.codio.io
*
Last login: Tue Dec 28 11:30:25 2021 from 192.168.10.156
codio@music-media:~/workspace$
```

6. Unit 6 - Testing with Python

Exploring Linters to Support Testing in Python: Question 4

Ensure mccabe is in your virtual box -

```
pip install mccabe
```

Run mccabe on `sums.py`. What is the result?

Run mccabe on `sums2.py`. What is the result?

- What are the contributors to the cyclomatic complexity in each piece of code?

[Next ▶](#)

Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1035-aws x86\_64)

```
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch
*
* Welcome to the Codio Terminal!
*
* https://docs.codio.com/project/ide/boxes/#overview
*
* Your Codio Box domain is: music-media.codio.io
*
Last login: Tue Dec 28 11:30:25 2021 from 192.168.10.156
codio@music-media:~/workspace$ pip install mccabe
Collecting mccabe
  Using cached https://files.pythonhosted.org/packages/87/89/479dc97e18549e21354893e4e4ef36db1d237534982482c36b1ee6eb57/mccabe-0.6.1-py2.py3-none-any.whl
Installing collected packages: mccabe
Successfully installed mccabe-0.6.1
codio@music-media:~/workspace$
```

6. Unit 6 - Testing with Python

Exploring Linters to Support Testing in Python: Question 4

Ensure mccabe is in your virtual box -

```
pip install mccabe
```

Run mccabe on `sums.py`. What is the result?

Run mccabe on `sums2.py`. What is the result?

- What are the contributors to the cyclomatic complexity in each piece of code?

[Next ▶](#)

The screenshot shows the Codio IDE interface. On the left, there's a filetree with a project named "Testing with Python". In the center, a terminal window displays the following Python code:

```

1
2 # SOURCE OF CODE: https://realpython.com/python-testing/
3
4 def test_sum():
5     assert sum([1, 2, 3]) == 6, "Should be 6"
6
7 if __name__ == "__main__":
8     test_sum()
9     print("Everything passed")

```

On the right, a panel titled "6. Unit 6 - Testing with Python" contains the following text:

## Exploring Linters to Support Testing in Python: Question 4

Ensure mccabe is in your virtual box -

```
pip install mccabe
```

Run mccabe on sums.py. What is the result?

Run mccabe on sums2.py. What is the result?

- What are the contributors to the cyclomatic complexity in each piece of code?

[Next ▶](#)

The screenshot shows the Codio IDE interface. On the left, there's a filetree with a project named "Testing with Python". In the center, a terminal window displays the following command and its output:

```
Terminal x sums.py
codio@music-media:~/workspace$ flake8 sums.py
sums.py:7:1: E305 expected 2 blank lines after class or function definition, found 1
sums.py:9:31: W292 no newline at end of file
codio@music-media:~/workspace$ mccabe sums.py
-bash: mccabe: command not found
codio@music-media:~/workspace$ flake8 --version
3.9.2 (mccabe: 0.6.1, pycodestyle: 2.7.0, pyflakes: 2.3.1) CPython 2.7.17 on Linux
codio@music-media:~/workspace$ lake8 --max-complexity 10 sums.py
-bash: lake8: command not found
codio@music-media:~/workspace$ flake8 --max-complexity 10 sums.py
sums.py:7:1: E305 expected 2 blank lines after class or function definition, found 1
sums.py:9:31: W292 no newline at end of file
codio@music-media:~/workspace$
```

On the right, a panel titled "6. Unit 6 - Testing with Python" contains the following text:

## Exploring Linters to Support Testing in Python: Question 4

Ensure mccabe is in your virtual box -

```
pip install mccabe
```

Run mccabe on sums.py. What is the result?

Run mccabe on sums2.py. What is the result?

- What are the contributors to the cyclomatic complexity in each piece of code?

[Next ▶](#)

The screenshot shows the Codio IDE interface. On the left, there's a filetree with a project named 'Testing with Python' containing several files like .settings, equivalence.py, metricTest.py, pylint2.py, pylintTest.py, styleLint.md, sums.py, and sums2.py. The main area has two tabs: 'Terminal' and 'File'. The 'Terminal' tab is active, displaying the output of a command-line session:

```

codio@music-media:~/workspace$ flake8 sums.py
sums.py:7:1: E305 expected 2 blank lines after class or function definition, found 1
sums.py:9:31: W292 no newline at end of file
codio@music-media:~/workspace$ mccabe sums.py
-mccabe: command not found
codio@music-media:~/workspace$ flake8 --version
3.9.2 (mccabe: 0.6.1, pycodestyle: 2.7.0, pyflakes: 2.3.1) CPython 2.7.17 on
Linux
codio@music-media:~/workspace$ lake8 --max-complexity 10 sums.py
-lake8: command not found
codio@music-media:~/workspace$ flake8 --max-complexity 10 sums.py
sums.py:7:1: E305 expected 2 blank lines after class or function definition, found 1
sums.py:9:31: W292 no newline at end of file
codio@music-media:~/workspace$ mccabe sums2.py
sums2.py:7:1: E302 expected 2 blank lines, found 1
sums2.py:10:1: E305 expected 2 blank lines after class or function definition, found 1
sums2.py:13:31: W292 no newline at end of file
codio@music-media:~/workspace$ 

```

To the right of the terminal, there's a sidebar titled '6. Unit 6 - Testing with Python' with the heading 'Exploring Linters to Support Testing in Python: Question 4'. It contains a note about ensuring McCabe is in the virtual box, a code input field with 'pip install mccabe', and two questions asking what happens when McCabe is run on sums.py and sums2.py respectively. A 'Next' button is at the bottom right.

## 7. Unit 6 - Testing with Python

### Exploring Linters to Support Testing in Python: Question 5 (e-Portfolio Entry)

From Section 5 of the Firdaus et al (2014) reading, select a test technique from the following categories:

- Specification-based techniques
- Structure-based techniques
- Experience-based techniques

Discuss the scenario(s) in which each technique would be important to be used

<ul style="list-style-type: none"> <li>• Specification-based techniques</li> </ul>	<ul style="list-style-type: none"> <li>• Structure-based techniques</li> </ul>	<ul style="list-style-type: none"> <li>• Experience-based techniques</li> </ul>
<p><b>There are four specification-based or black-box technique:</b></p> <ul style="list-style-type: none"> <li>• Equivalence partitioning.</li> <li>• Boundary value analysis.</li> <li>• Decision tables.</li> <li>• State transition testing.</li> </ul> <ul style="list-style-type: none"> <li>• The testers have no knowledge of how the system or component is structured inside the box. In black-box testing the tester is concentrating on what the software does, not how it does it.</li> <li>• The definition mentions both functional and non-functional testing. Functional testing is concerned with what the system does its features or</li> </ul>	<p>White-box testing e.g path testing</p> <p>Structure-based techniques serve two purposes: test coverage measurement and structural test case design</p> <ul style="list-style-type: none"> <li>• They are often used first to assess the amount of testing performed by tests derived from specification-based techniques, i.e. to assess coverage.</li> <li>• They are then used to design additional tests with the aim of increasing the test coverage.</li> <li>• Structure-based test design techniques are a good way of generating additional test cases that are</li> </ul>	<p>In <b>experience-based techniques</b>, people's knowledge, skills and background are of prime importance to the test conditions and test cases. The experience of both technical and business people is required, as they bring different perspectives to the test analysis and design process. Because of the previous experience with similar systems, they may have an idea as what could go wrong, which is very useful for testing.</p> <ul style="list-style-type: none"> <li>• Experience-based techniques go together with specification-based and structure-based techniques, and are also used when there is</li> </ul>

<p>functions. Non-functional testing is concerned with examining how well the system does. Non-functional testing like performance, usability, portability, maintainability, etc.</p> <ul style="list-style-type: none"> <li>• Specification-based techniques are appropriate at all levels of testing (component testing through to acceptance testing) where a specification exists. For example, when performing system or acceptance testing, the requirements specification or functional specification may form the</li> </ul>	<p>different from existing tests.</p> <ul style="list-style-type: none"> <li>• They can help ensure more breadth of testing, in the sense that test cases that achieve 100% <b>coverage</b> in any measure will be exercising all parts of the software from the point of view of the items being covered.</li> </ul> <p>Source: <a href="http://tryqa.com/what-is-structure-based-technique-in-software-testing/">http://tryqa.com/what-is-structure-based-technique-in-software-testing/</a></p>	<p>no specification, or if the specification is inadequate or out of date.</p> <ul style="list-style-type: none"> <li>• This may be the only type of technique used for low-risk systems, but this approach may be particularly useful under extreme time pressure - in fact this is one of the factors leading to exploratory testing.</li> </ul> <p>Source: <a href="http://tryqa.com/what-is-experience-based-testing-technique/">http://tryqa.com/what-is-experience-based-testing-technique/</a></p>
--	--	---

basis of the tests.  Source: <a href="http://tryqa.com/what-is-black-box-specification-based-also-known-as-behavioral-testing-techniques/">http://tryqa.com/what-is-black-box-specification-based-also-known-as-behavioral-testing-techniques/</a>		
---	--	--