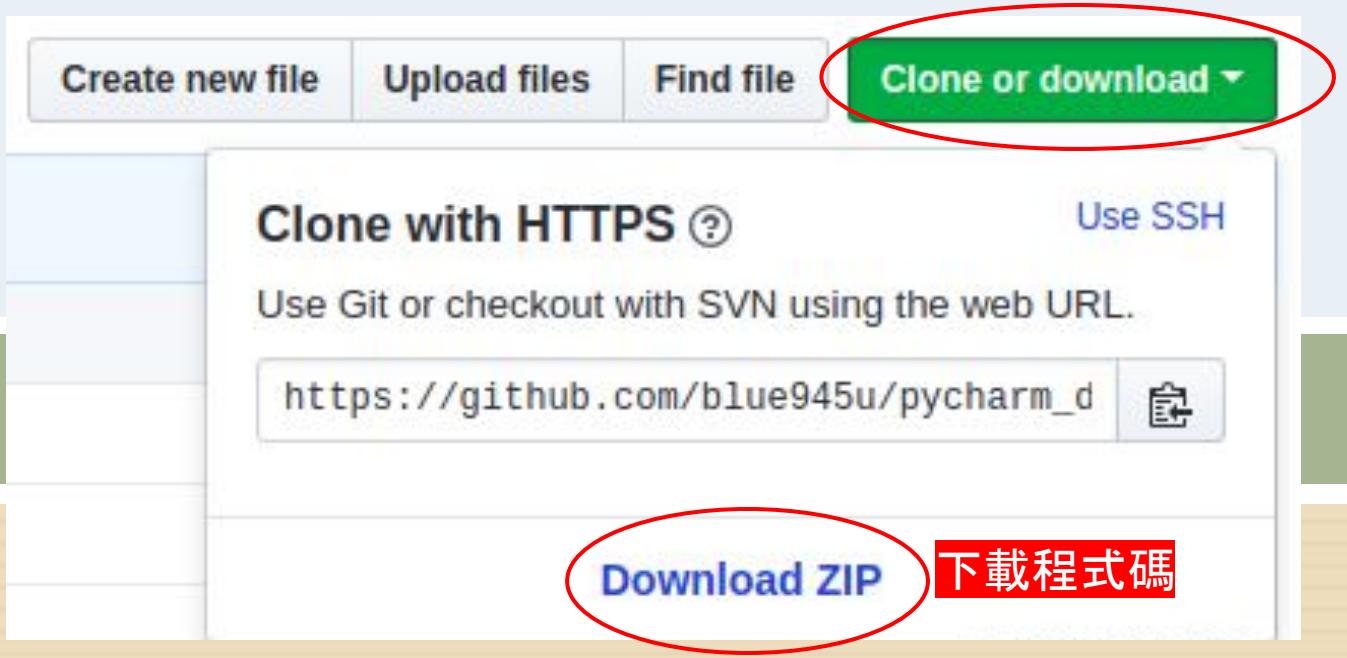
https://github.com/blue945u/pycharm_demo

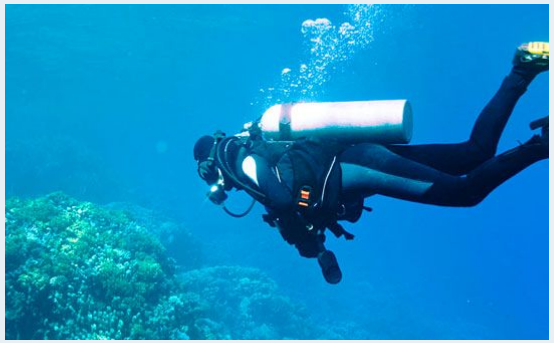


業界常用工具與技術分享
- PyCharm

講師：Coco Lin
Email：blue945u@gmail.com

林佳穎
Coco Lin

infrastructure
private
DATA
DATABASE
backup
MANAGEMENT
ANALYSIS
connections
INFORMATION
web
workload
public
redundant
BIG
management
MINING
resource



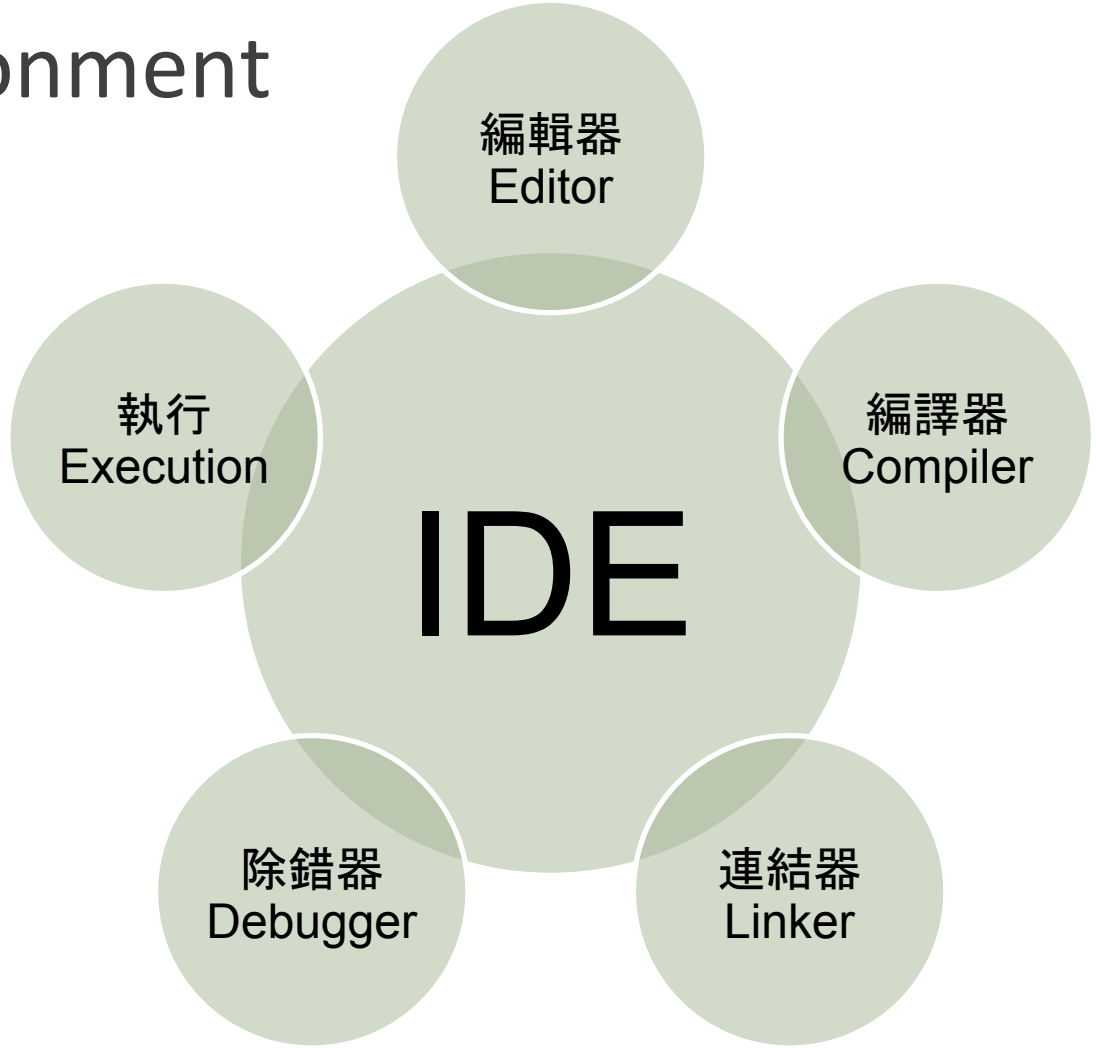
Contents

- 什麼是 IDE？
- Python編寫風格指南
- 程式碼編輯密技
- 測試與除錯
- 多版本開發利器
- 版本控制

什麼是 IDE？

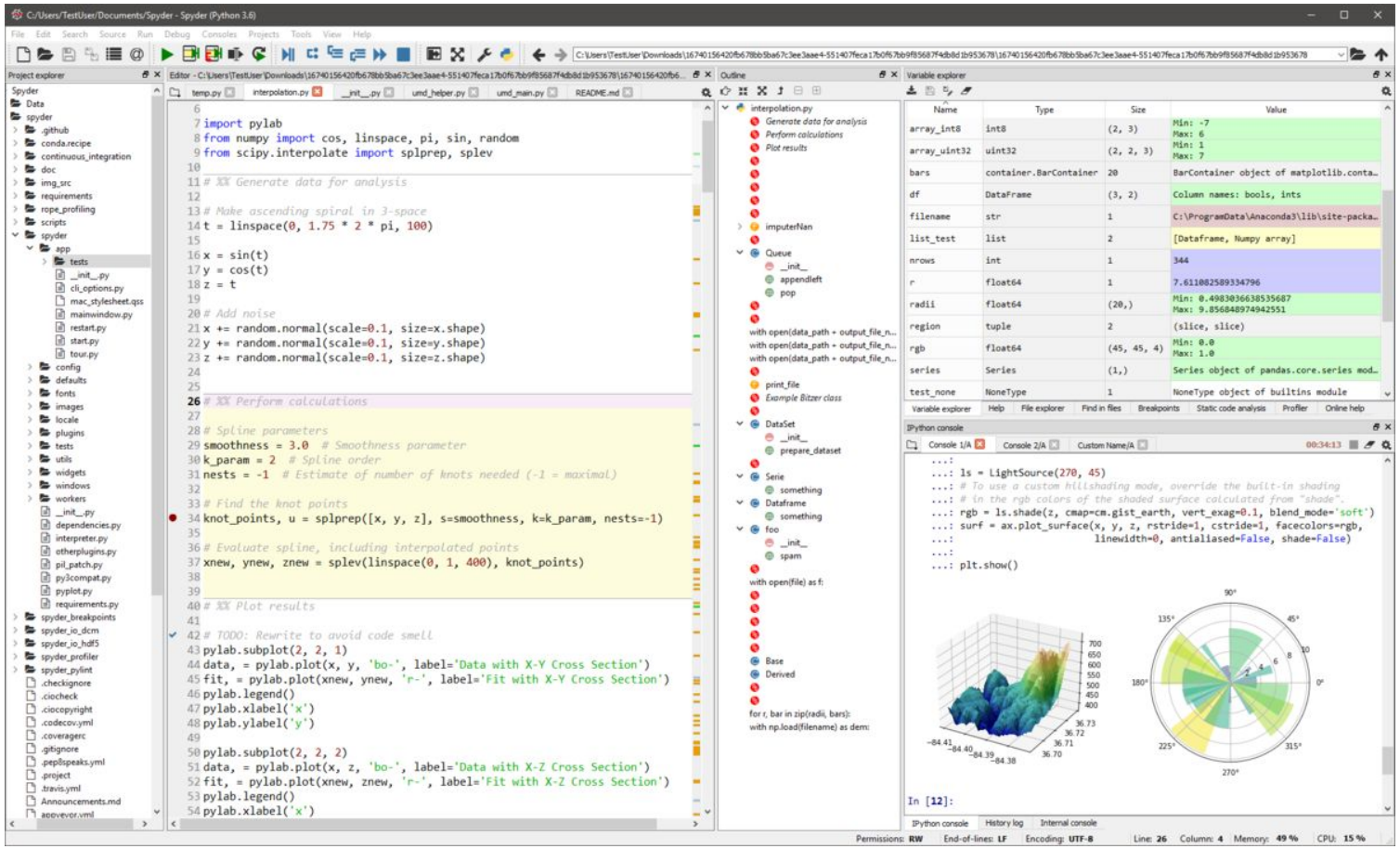
Integrated Development Environment
整合開發環境

Emacs



Python常見IDE：Spyder

Anaconda內建，整合NumPy，SciPy，Matplotlib與IPython等。



Python常見IDE：PyDev

Eclipse的Python插件

Eclipse File Edit Source Refactoring Refactor Navigate Search Project Pydev Run Window Help 100% Mon 1:03 PM

Debug - udemyapp/udemyapp/rest/client.py - Eclipse - /Users/prem/Documents/workspace

Quick Access | Java EE | Debug | PyDev

Debug | Servers

MainThread - pid2218_seq1
- __init__ [client.py:134]
- __init__ [client.py:51]
- setUpClass [test_courses.py:16]
- _handleClassSetUp [suite.py:146]
- run [suite.py:100]
- __call__ [suite.py:70]
- run [suite.py:108]
- __call__ [suite.py:70]
- run [runner.py:151]
- runTests [main.py:232]
- __init__ [main.py:95]

Variables | Breakpoints

- ApiClient.__init__ [client.py:134]
- Courses.get_all [client.py:70]

test_courses | client (udemyapp.rest) | client (clarifai.rest) | test_models | dto | pydevd

```
166         Args:
167             resource:
168             params: parameters passed to the request
169             version: v1 or v2
170             method: GET or POST or DELETE or PATCH
171
172         Returns:
173             JSON from user request
174         """
175
176 #       self._check_token()
177         url = urljoin(self.basev2, resource)
178
179         # only retry under when status_code is non-200, under max-tries
180         # and under some circumstances
181         status_code = 199
```

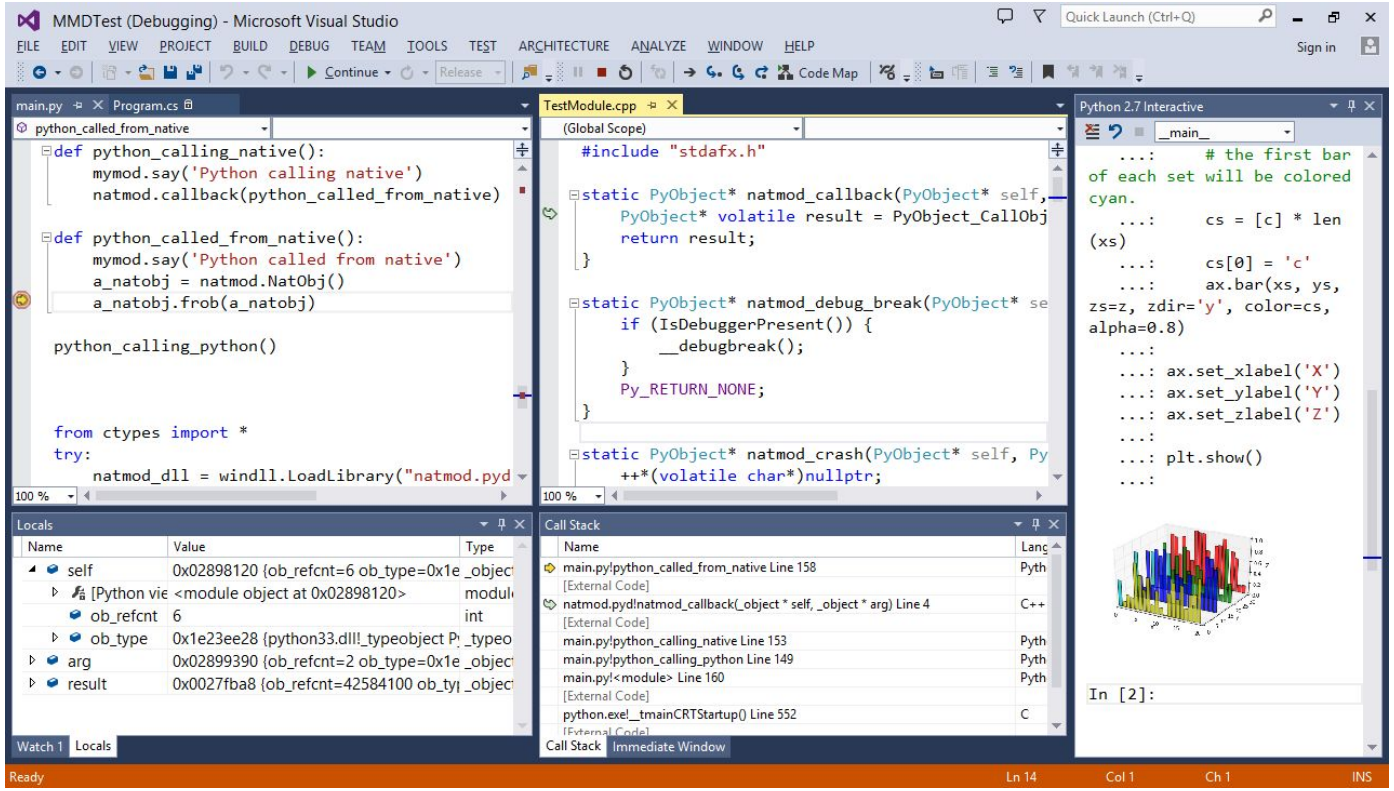
Console | Tasks | Scala Expression Evaluator | Async Stack | PyUnit | Scala Interpreter

test_courses.py

```
pydev debugger: starting (pid: 2218)
```

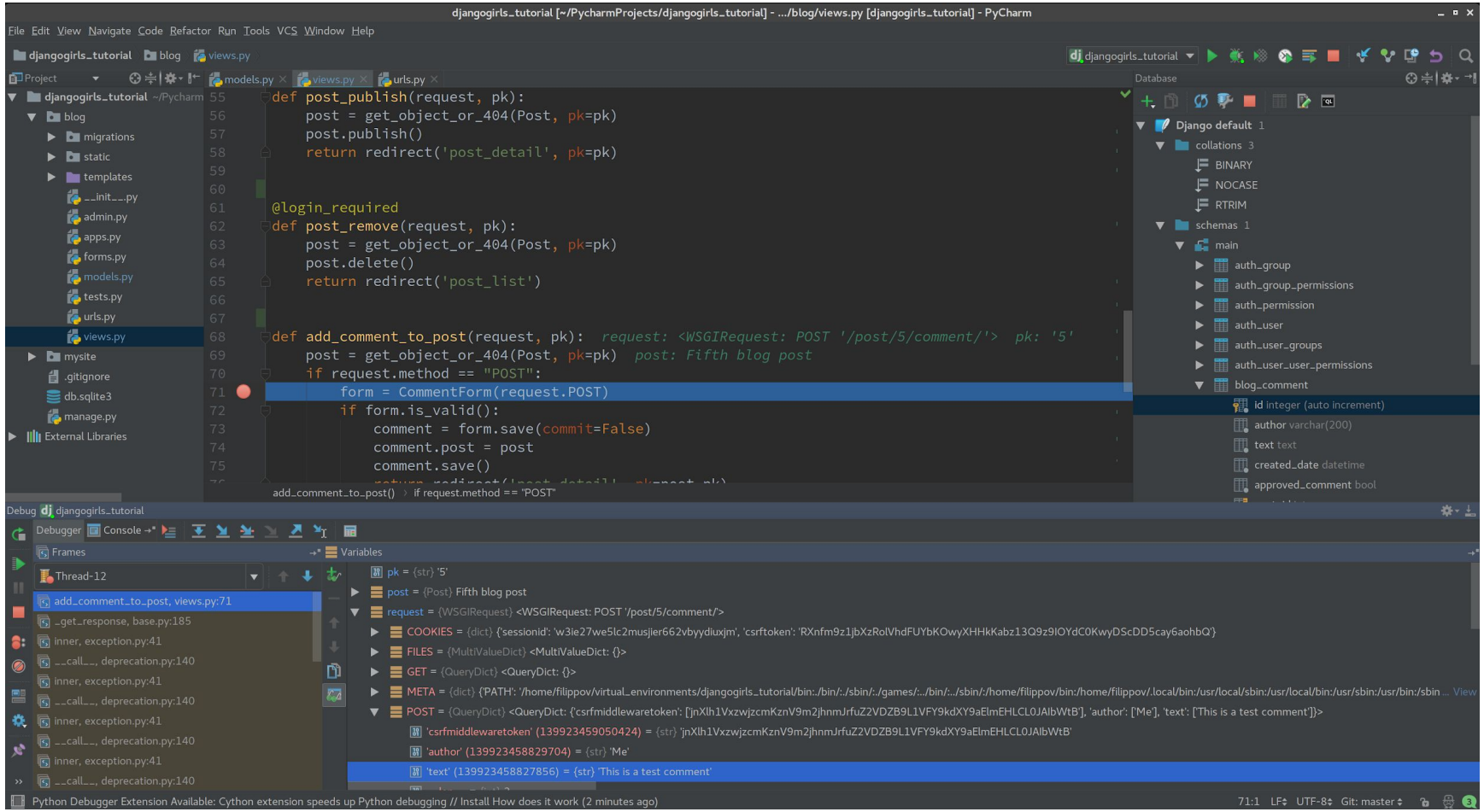
Python常見IDE：Visual Studio

微軟公司的開發與Visual Studio Code的定位不同



Python常見IDE：PyCharm

由JetBrains開發，並支持使用Django進行網頁開發。



Python編寫風格指南

The Zen of Python

PEP

Python Enhancement Proposals

```
> python3
Python 3.6.1
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
```

Python編寫風格指南

```
>>>import this
```

The Zen of Python, by Tim Peters

Beautiful is better than ugly.

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.

Readability counts.

Special cases aren't special enough to break the rules.

...

...

```
>>>exit()
```

Python編寫風格指南_PEP

python™

Search GO Socialize

About | Downloads | Documentation | Community | Success Stories | News | Events

Tweets by @ThePSF

Python Software @ThePSF

Help us raise $30,000 by November 21st! Donate to our Recurring Giving Campaign - it starts in 2 days. #idonatedtothepsf #ijoinedthepsf

8h

Python Software @ThePSF

Boston Python Meetup Oct. 23 -

Python >>> Python Developer's Guide >>> PEP Index >>> PEP 0 -- Index of Python Enhancement Proposals (PEPs)

PEP 0 -- Index of Python Enhancement Proposals (PEPs)

PEP:	0
Title:	Index of Python Enhancement Proposals (PEPs)
Last-Modified:	2018-10-20

P	1	PEP Purpose and Guidelines	Warsaw, Hylton, Goodger, Coghlan
P	4	Deprecation of Standard Modules	Cannon, von Löwis
P	5	Guidelines for Language Evolution	Prescod
P	6	Bug Fix Releases	Aahz, Baxter
P	7	Style Guide for C Code	GvR, Warsaw
P	8	Style Guide for Python Code	GvR, Warsaw, Coghlan
P	10	Voting Guidelines	Warsaw
P	11	Removing support for little used platforms	von Löwis, Cannon
P	12	Sample reStructuredText PEP Template	Goodger, Warsaw
P	573	Module State Access from C Extension Methods	Viktorin, Coghlan, Snow, Plch

Other Informational PEPs

	PEP	PEP Title	PEP Author(s)
I	13	Python Language Governance	Warsaw
I	20	The Zen of Python	Peters

Python編寫風格指南_PEP 8

PEP 8 -- Style Guide for Python Code

縮排 (Indentation)
- 每層縮排使用 4 個空格

使用Tab還是空格縮排
- 用空格

每行最長長度
- 所有行都不超過 80 個字符

斷行風格
- 垂直縮排，首行不能有參數或
- 斷行首字母與開頭的括號垂直對齊

多行的括號
- 括號結束符與**最後行的首字符**對齊或與**首行的首字符**對齊

```
def a_function(
        var_one, var_two, var_three,
        var_four):
    print(var_one)

foo = a_function(var_one,
                 var_two)
```

```
my_list = [
    1, 2, 3,
    4, 5, 6,
    ]
```

```
my_list = [
    1, 2, 3,
    4, 5, 6,
]
```

PEP 8 -- Style Guide for Python Code
https://www.python.org/dev/peps/pep-0008/

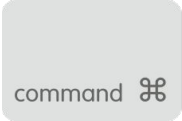
實作時間

解除範例程式main.py中9-22行的註解,
調整程式碼風格

必學PyCharm快捷鍵

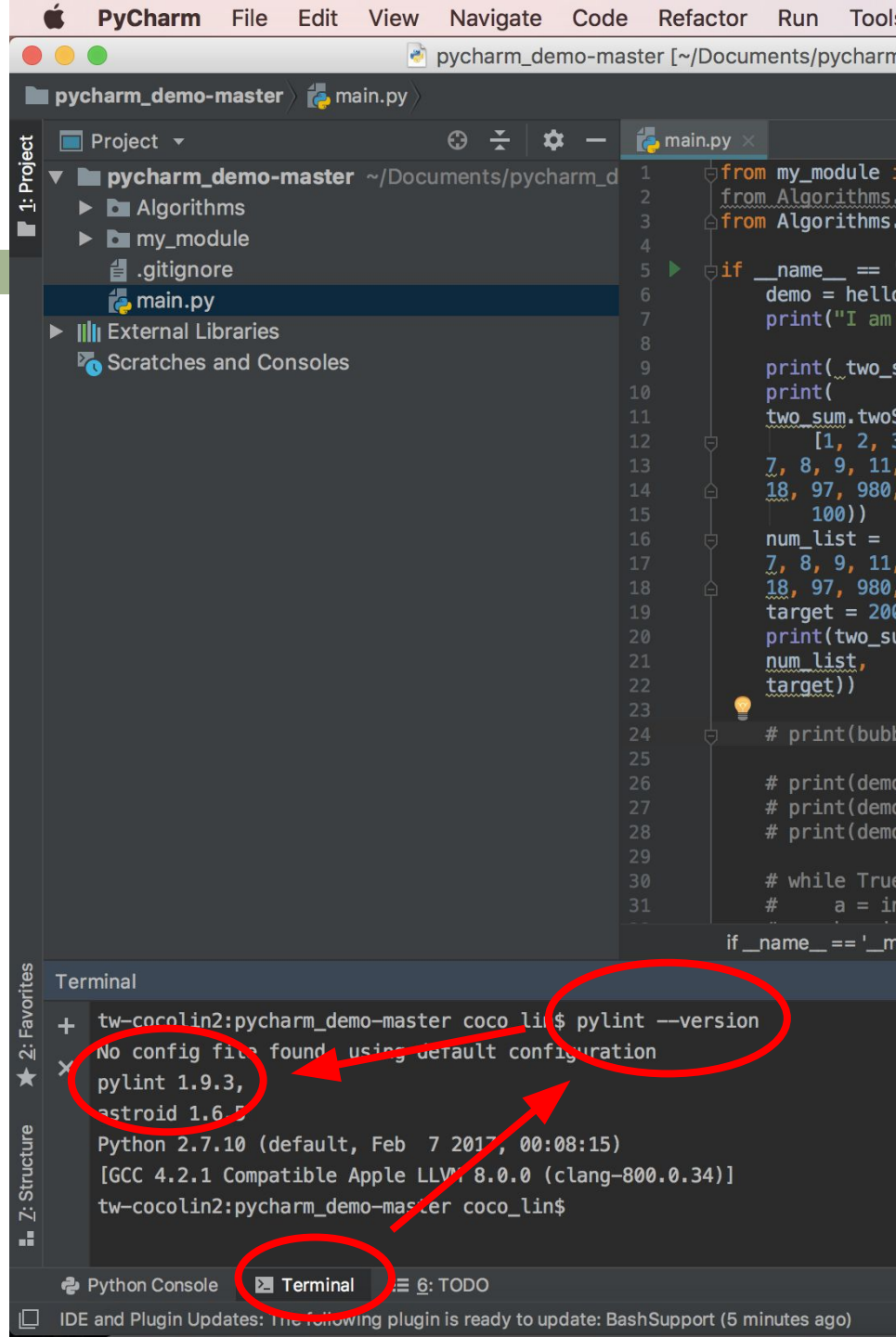
- 選取到上/下一行：Shift + Up/Down
- 註解/取消註解選取的內容，也可不選取：Ctrl + /
- 將選取的內容縮排：Tab

  將選取的內容逆向縮排：Shift + Tab



Python編寫風格指南_Pylint

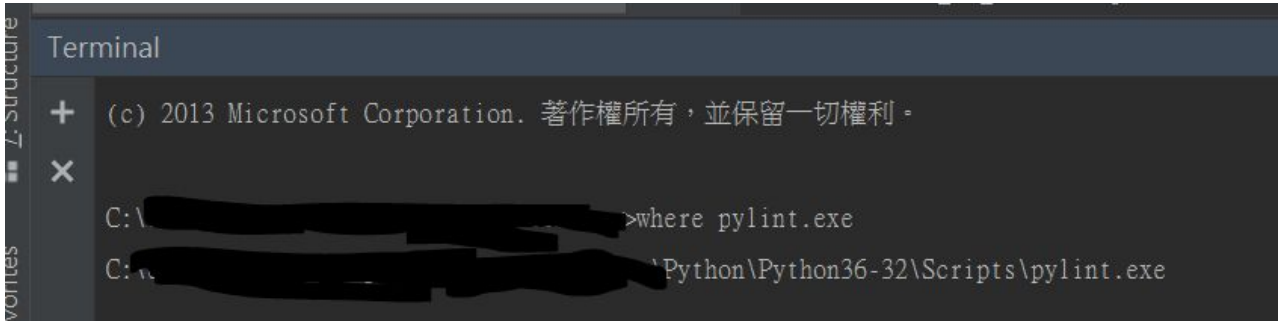
- ✓ Python代碼風格的檢查工具，依據Guido van Rossum的PEP8標準
- ✓ 一般Pylint會隨著Python安裝好，可以用指令**pylint --version**看版本
- ✓ 若還沒安裝，可用**pip install pylint**安裝

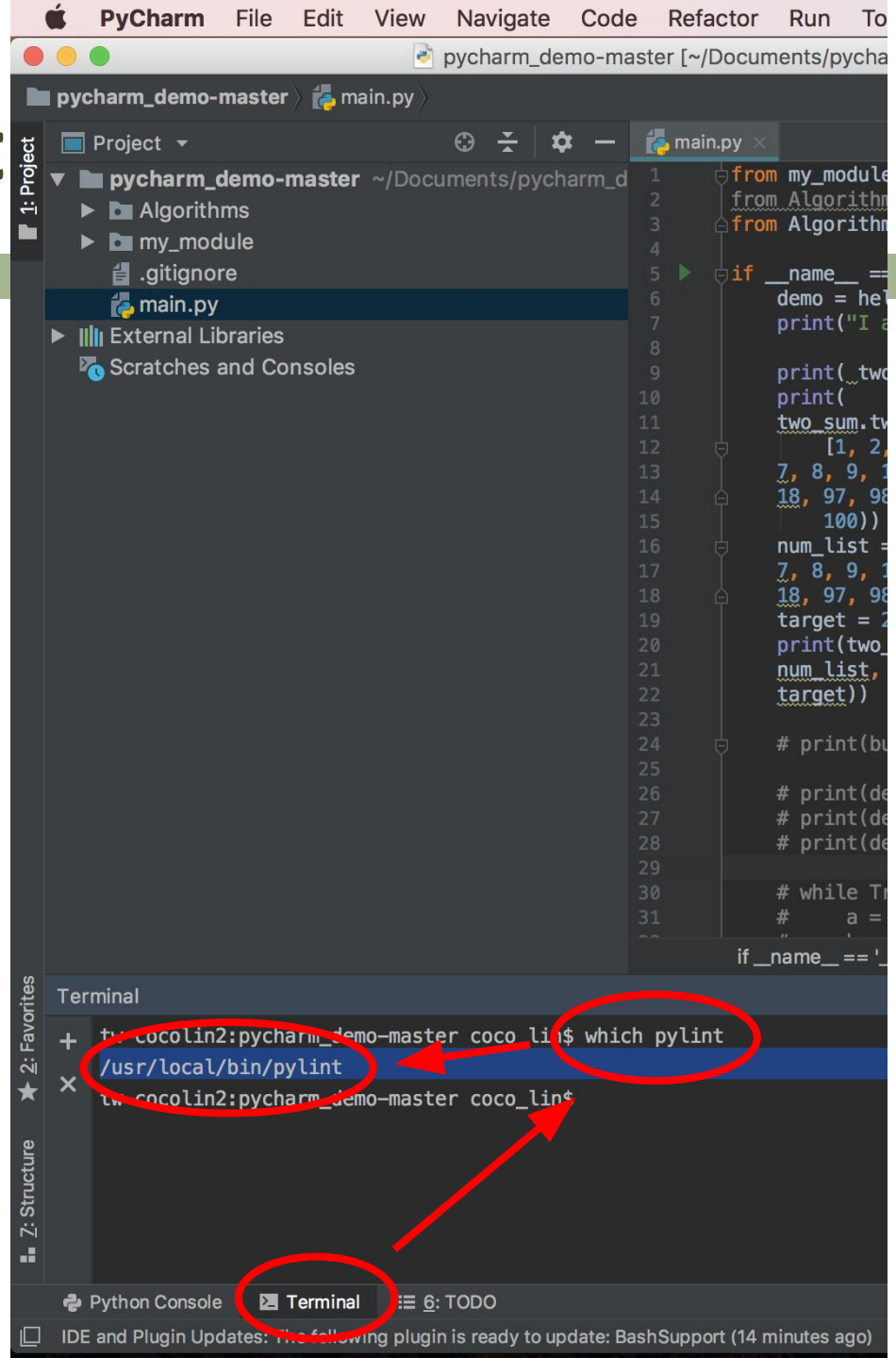


Python編寫風格指南_Pylint

使用指令查詢檔案位置

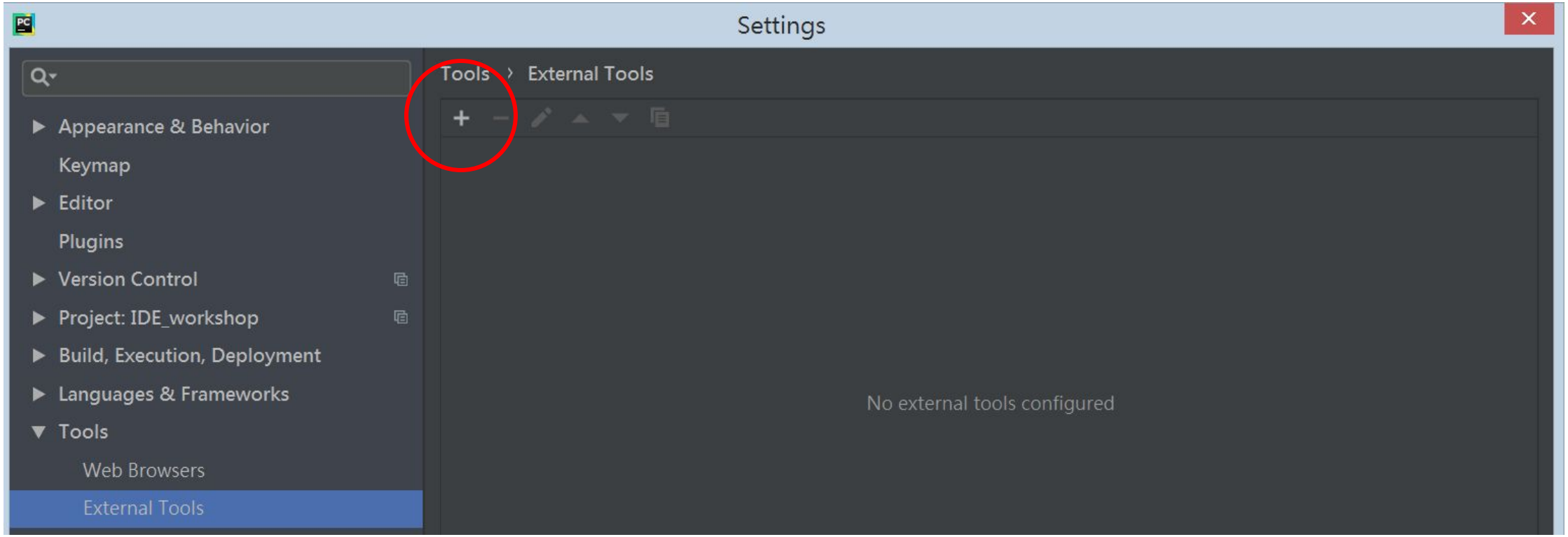
- Windows：where pylint.exe
- MAC：which pylint
- Linux：which pylint



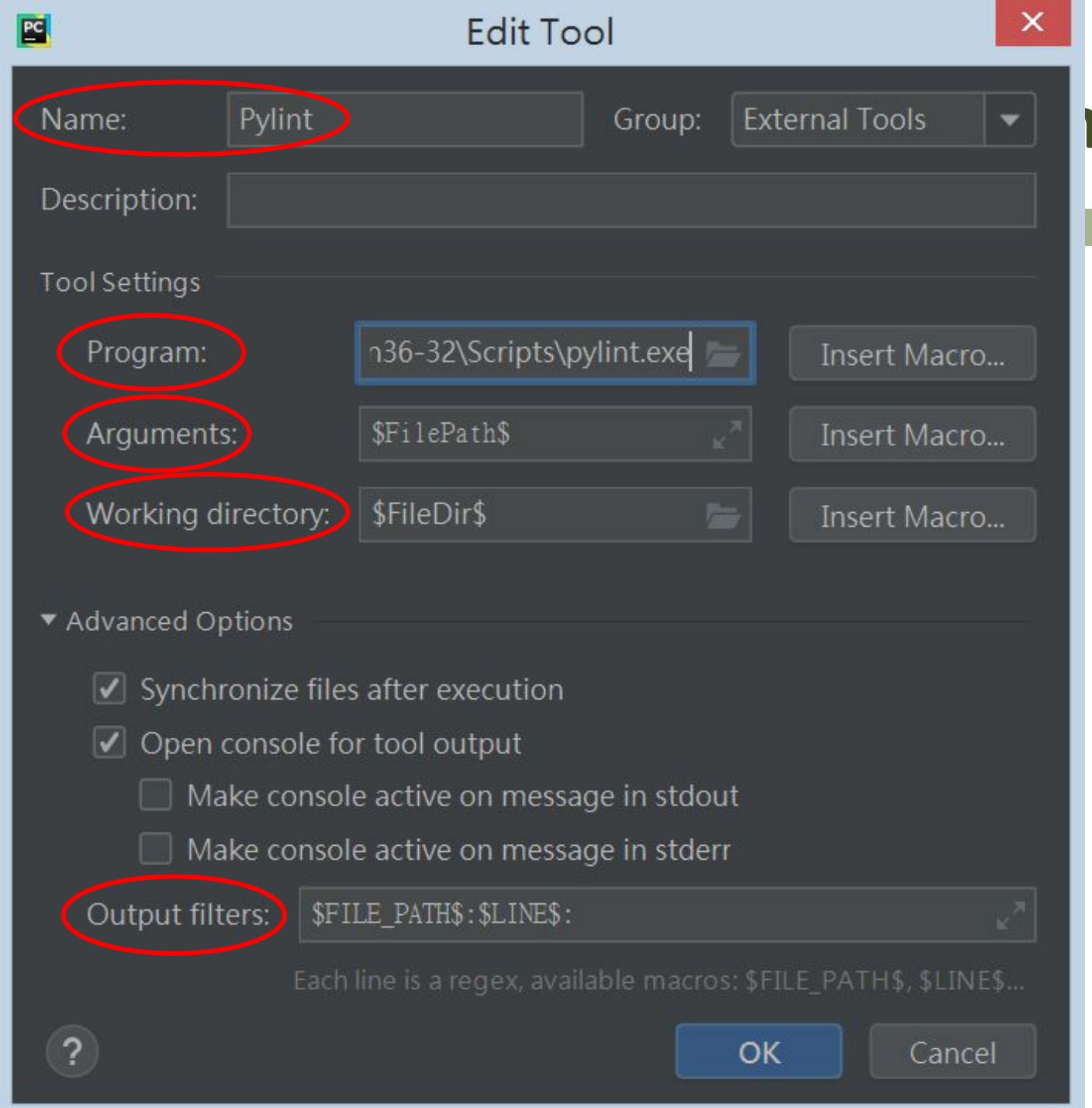


Python編寫風格指南_Pylint in PyCharm

Windows & Linux：File > Settings > Tools > External Tools
MacOS : Pycharm > Preferences > Tools > External Tools



nt in PyCharm



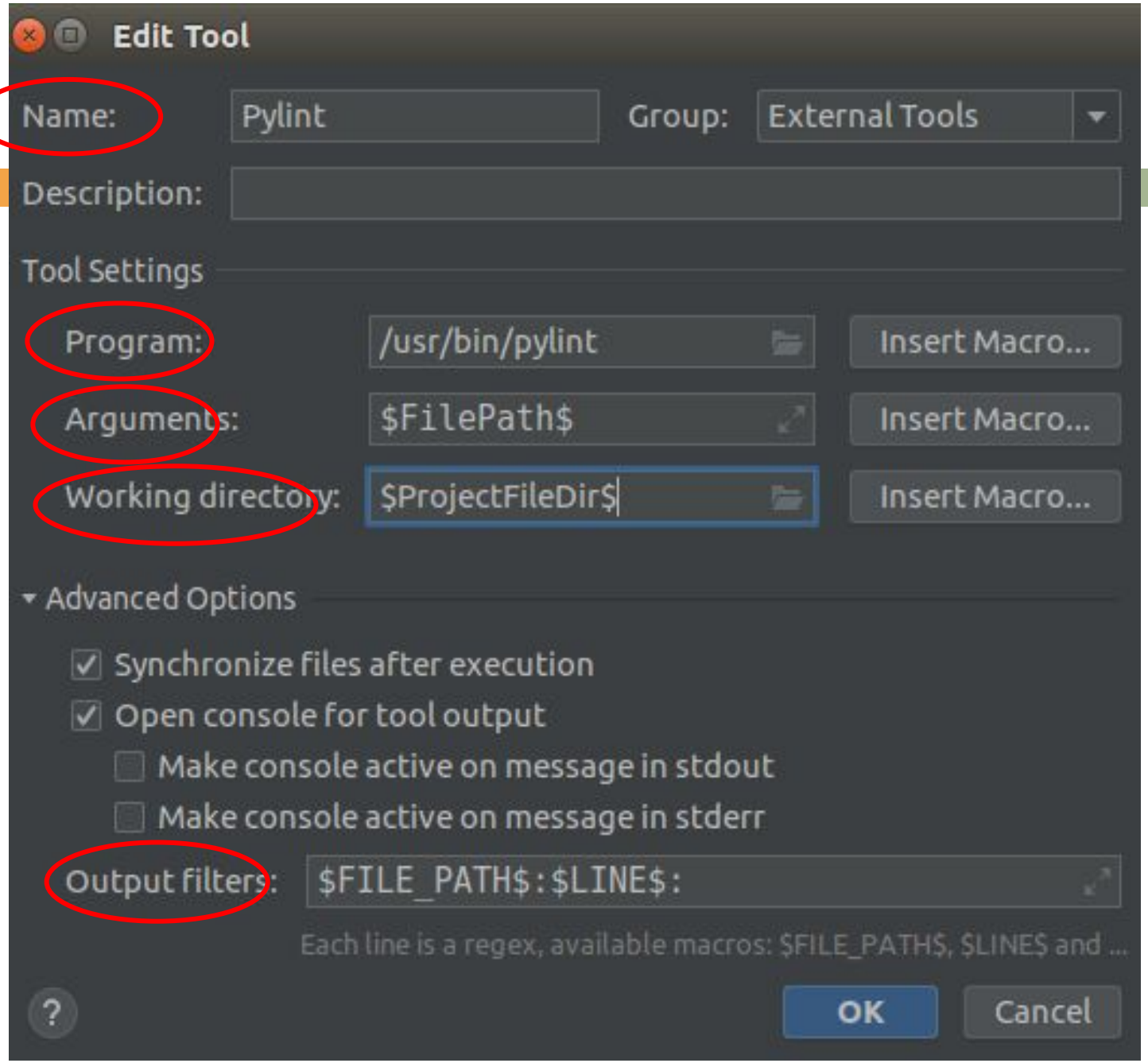
For Windows

Name：Pylint

Tool Settings

- Programs：pylint的路徑
- Arguments：$FilePath$
- Working directory：$FileDir

Output filters：$FILE_PATH$:$LINE$:



in PyCharm

For MacOS/Linux

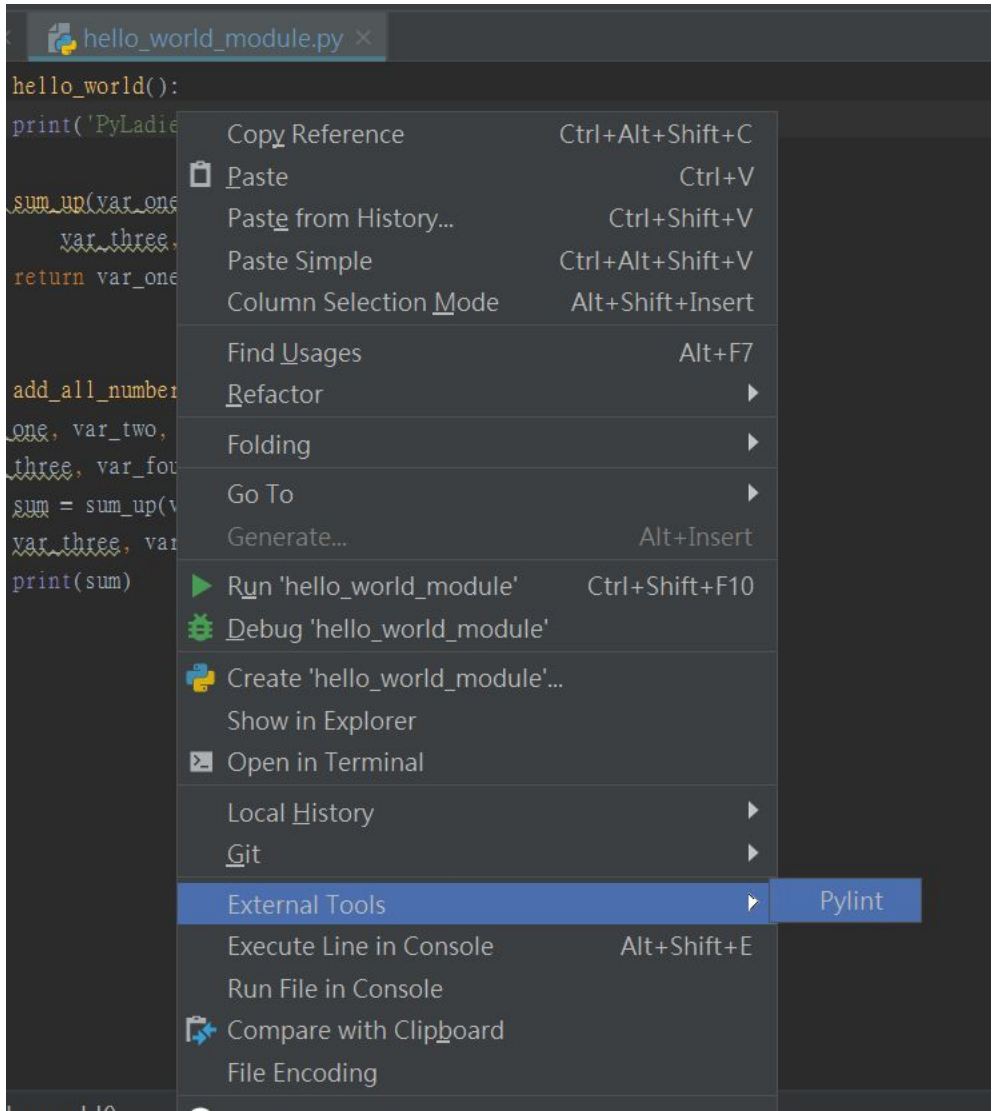
Name：Pylint

Tool Settings

- Programs：pylint的路徑
- Arguments：$FilePath$
- Working directory：$ProjectFileDir$

Output filters：$FILE_PATH$:$LINE$:

Python編寫風格指南_Pylint in PyCharm



在檔案中，點擊右鍵即可執行Pylint

必學PyCharm快捷鍵

自動排版：Code > Reformat Code

- 快捷鍵：Ctrl + Alt + L

24

實作時間

程式碼編輯密技_必學PyCharm快捷鍵

- Ctrl + Z 復原

  Ctrl + Shift + Z 重做
- Shift + Up/Down 選取到上/下一行
- Ctrl + Shift + Left/Right 往前/後選取
- Ctrl + Shift + Up/Down 上下移動選取的內容
- Ctrl + Shift +]/[往前/後選取”整塊”程式碼

程式碼編輯密技_必學PyCharm快捷鍵

- Ctrl + Delete/Backspace 往後/前刪除一個單字
- Ctrl + X 刪除一整行(不用選取)
- Ctrl + / 註解/取消註解選取的內容(也可不選取)
- Tab 將選取的內容縮排

  Shift + Tab 將選取的內容逆向縮排
- Ctrl + Alt + L 自動排版

Ctrl command ⌘ Alt alt option

找到實作 bubble_sort 的檔案

小挑戰

程式碼編輯密技

Finding Text in Project

1. Edit
2. Find
3. Find in Path
4. InProject

快捷鍵：Ctrl+Shift+F
Mac：Control^+Shift+F

Search Everywhere

快捷鍵：Shift
(Double-press)

程式碼編輯密技

Replacing Text in Project

1. Edit
2. Find
3. Replace in Path
4. InProject

快捷鍵：Ctrl+Shift+R
Mac：Control^+Shift+R

找到使用 primeFactors 的所有檔案,
並以 prime_factors 取代

小挑戰

如何說服主管我的程式完成並且是正確的？

Unit Test 單元測試

針對程式函式或方法(最小單位)
進行程式邏輯正確性檢驗的測試工作。

特點:

1. Fast:快速
2. Independent:獨立
3. Repeatable:可重複
4. Self-Validating:可反應驗證結果
5. Timely:及時

```
def test_sum_correct():

    # GIVEN
    a = 1
    b = 2
    ans = 3

    # WHEN
    result = sum(a, b)

    # THEN
    assert result == ans
```

測試篇

PyCharm File Edit View Navigate Code Refactor Run Tools VCS Window Help

pycharm_demo-master [~/Documents/pycharm_demo-master] - .../my_module/test_dem

pycharm_demo-master > my_module > test_demo.py

Project

- pycharm_demo-master ~/Documents/pycharm_d
  - Algorithms
  - my_module
    - __init__.py
    - hello_module.py
    - test_demo.py
    - .gitignore
    - main.py
  - External Libraries
  - Scratches and Consoles

main.py | test_demo.py

```
from unittest import TestCase
from my_module import hello_module


class TestDemo(TestCase):

    def test_quadratic_formula_correct(self):
        demo = hello_module.Demo()
        self.assertEqual((0.0, -2.0), demo.q

    def test_quadratic_formula_exception(sel
        demo = hello_module.Demo()
        self.assertRaises(Exception, demo.qu
```

Run 'Unittests for test_d...' ^⇧F10

Debug 'Unittests for test_d...' ^⇧F9

Run: Unittests for test_demo.TestDemo

Tests passed: 3 of 3 tests – 0 ms

All Tests Passed 0 ms

```
Testing started at 1:11 AM ...
/usr/local/bin/python3.7 "/Applications/PyCharm CE.app/Contents/hel
Launching unittests with arguments python -m unittest test_demo.Tes

Ran 3 tests in 0.002s

OK

Process finished with exit code 0
```

Python Console | Terminal | 4: Run | 6: TODO

Tests passed: 3 (moments ago)

測試篇

1. 選取Class名稱按右鍵
2. Go to
3. Test

   快捷鍵：Ctrl+Shift+T

   Mac：command ⌘+Shift+T

34

實作時間

當你遇到Bug的時候會如何解決?

- print大法
- Logging (高級版的print?
- Debugger

Bram Cohen
@bramcohen

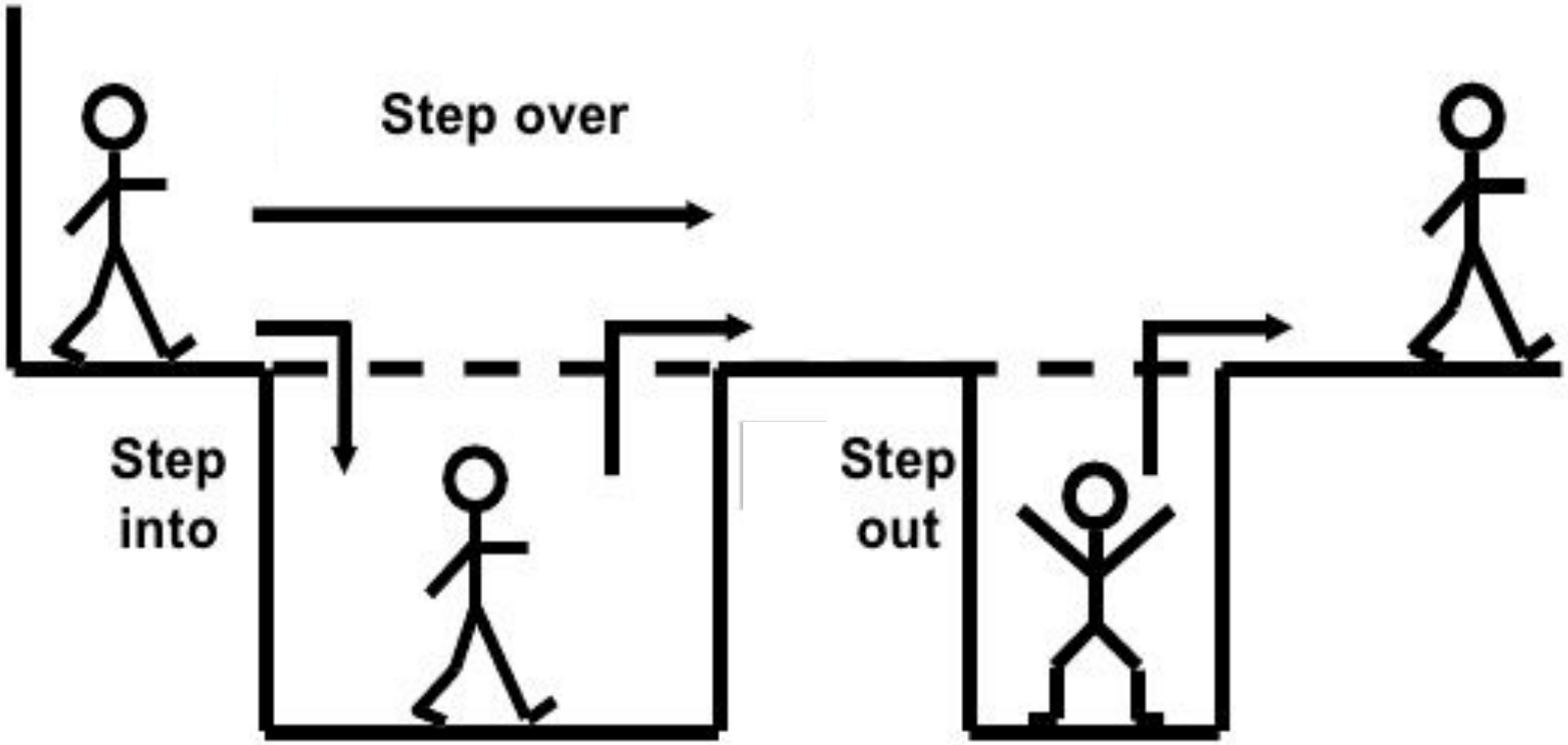
"90% of coding is debugging. The other 10% is writing bugs"

12:06 AM · 27 Mar 11

4,707 RETWEETS **3,192** LIKES

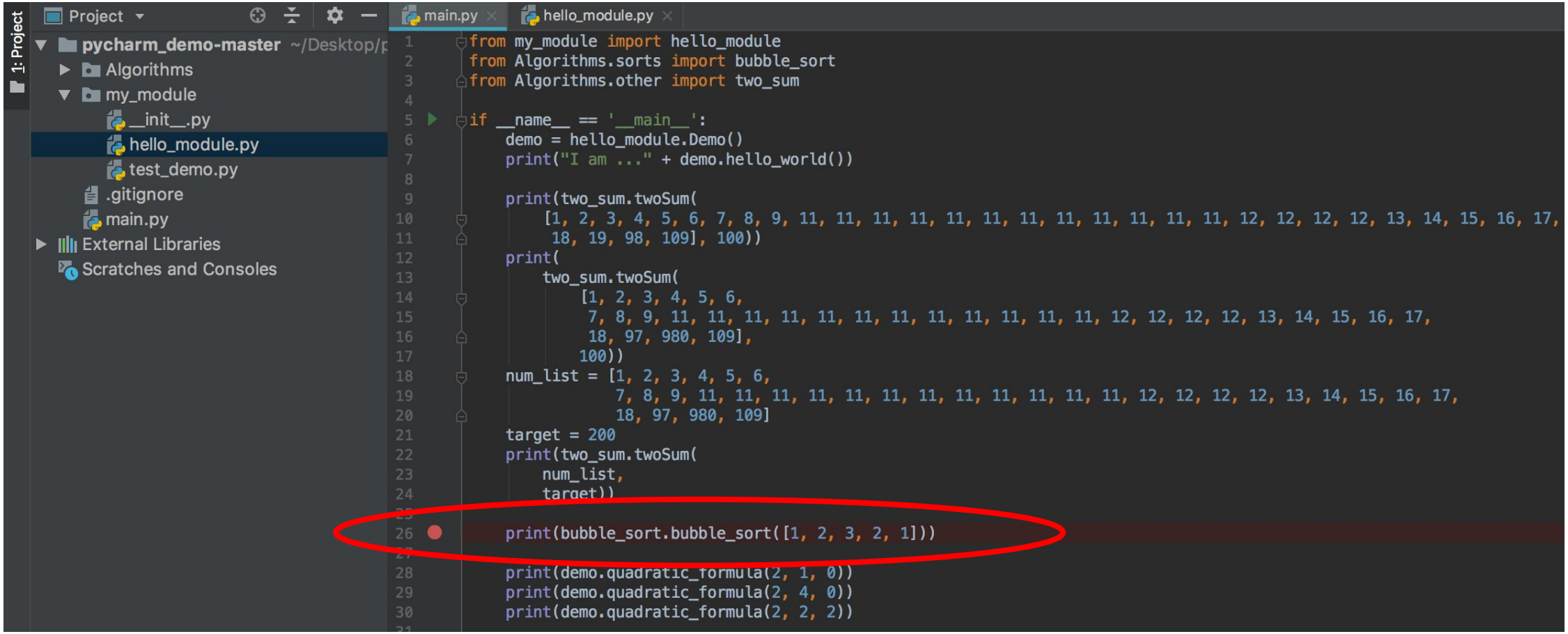
除錯篇

Debugger心法：Step Over、Step Into、Step Out



除錯篇

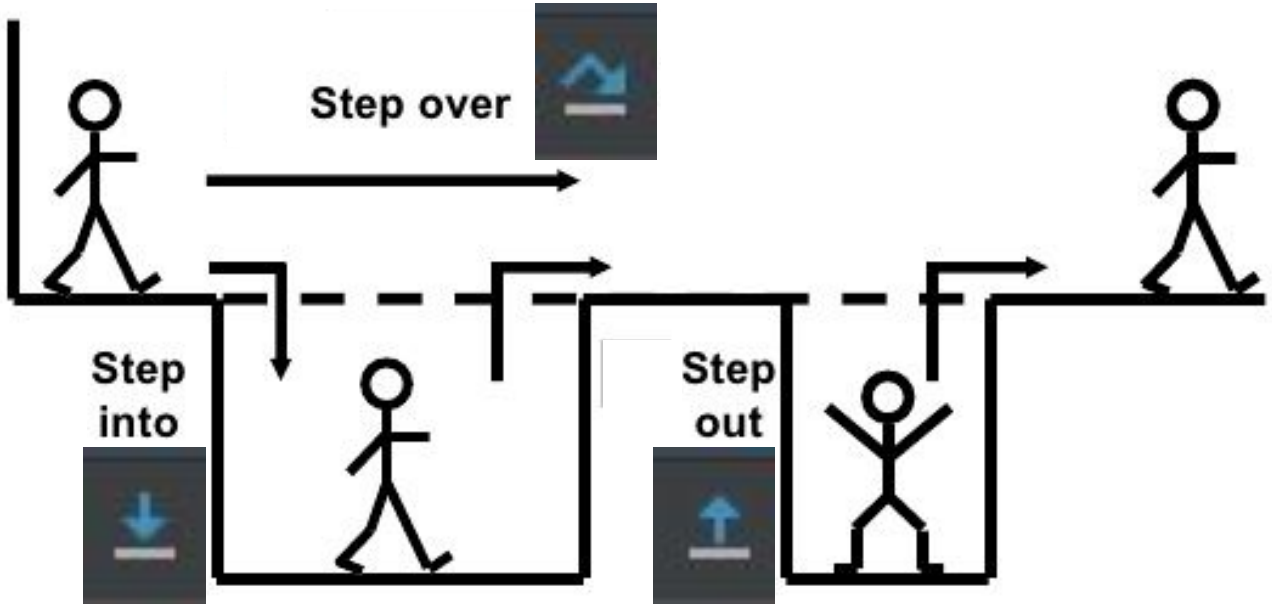
Step1 設置斷點(break points)

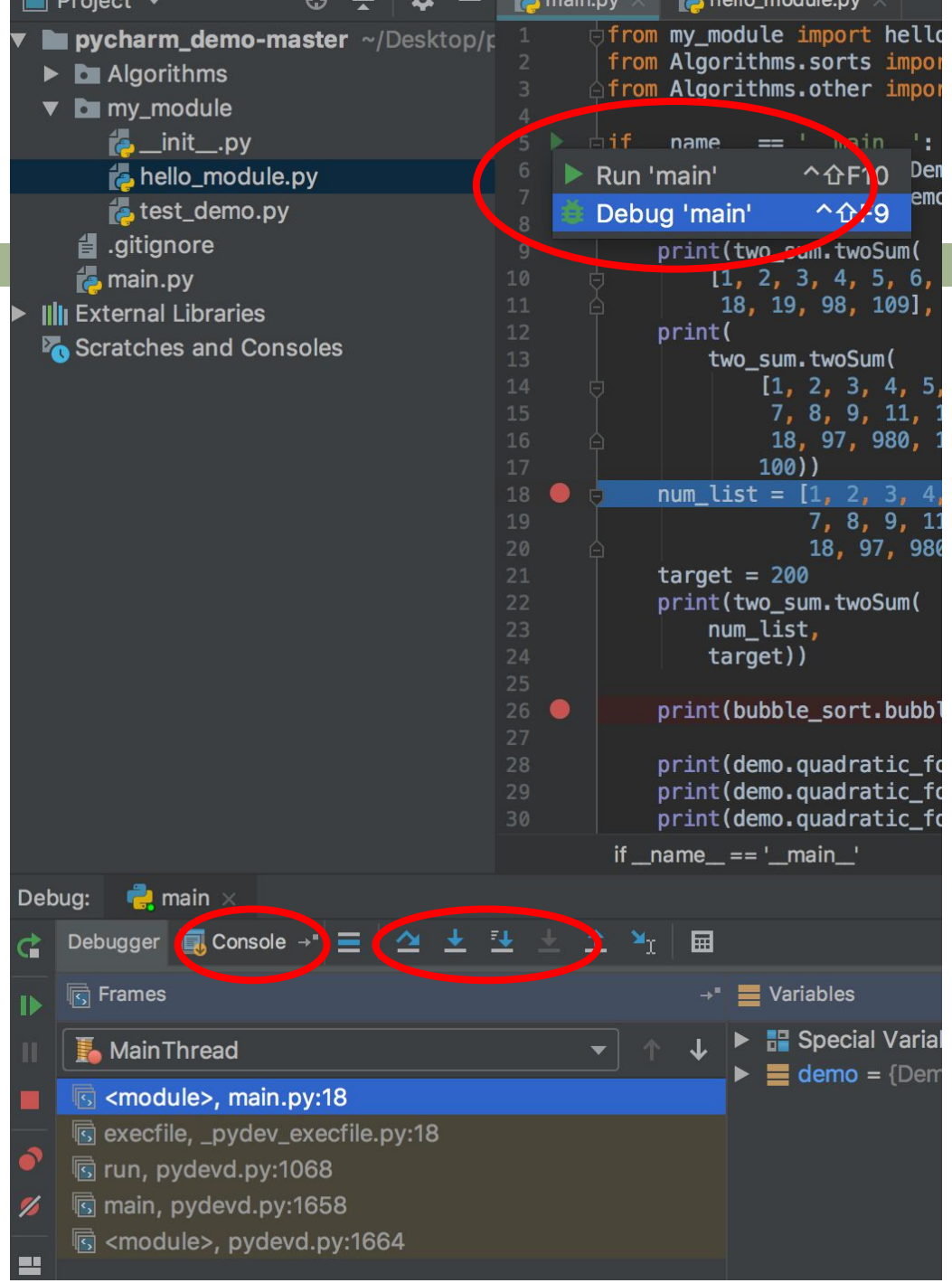


除錯篇

Step2 Run -> Debug

逐步觀察程式運行，縮小範圍



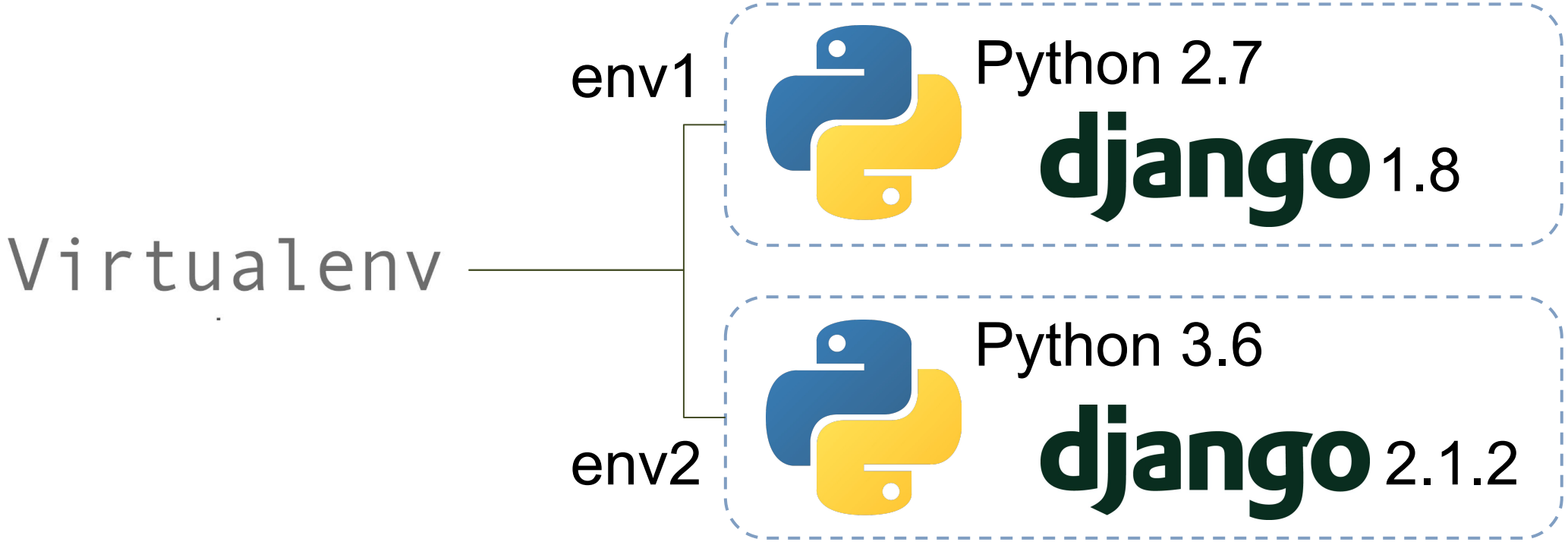


實作時間

解除範例程式main.py中剩下的註解，練習使用Debug功能觀察程式行為

多版本開發利器_Virtualenv

Virtualenv is a tool that lets you create an **isolated Python environment** for your project.



多版本開發利器

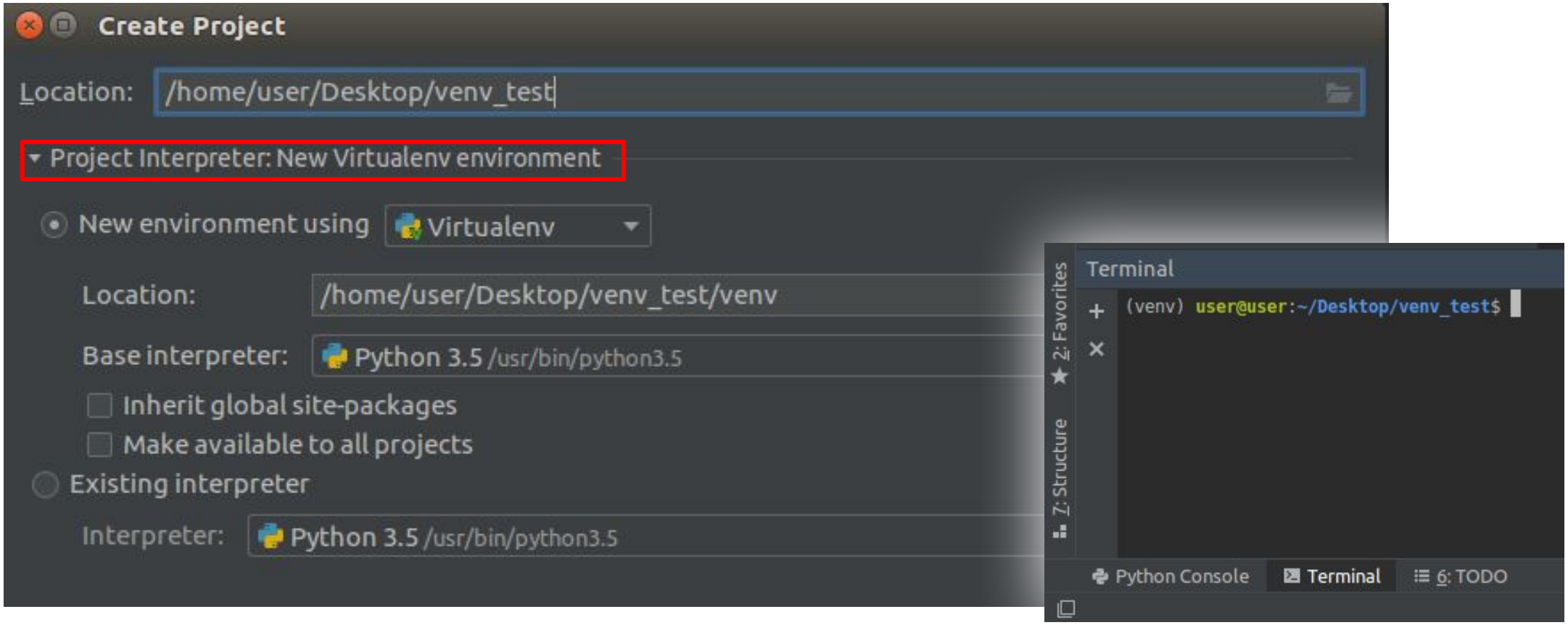
安裝Virtualenv

```
sudo pip install virtualenv
virtualenv --version
```

多版本開發利器

在PyCharm中進行設置

- 新增Project時，展開Project Interpreter建立虛擬環境：



多版本開發利器

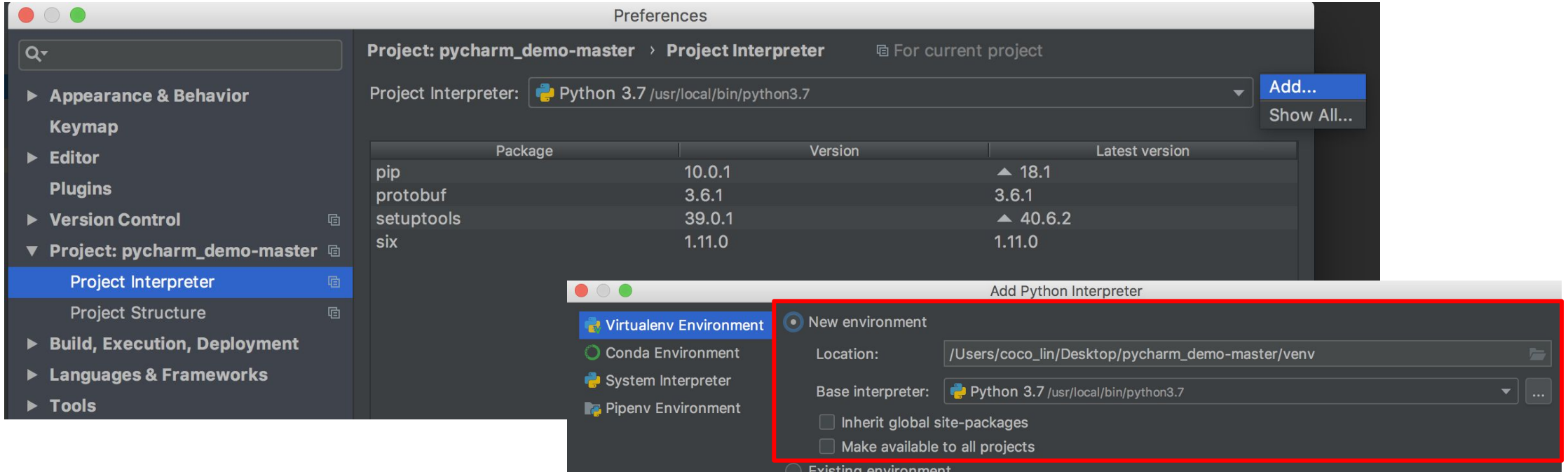
在PyCharm中進行設置

- 在已建立的Project，設置虛擬環境：

1. 開啟Project Interpreter

Windows and Linux：File > Settings > Project Interpreter
macOS：PyCharm > Preferences > Project Interpreter or ⌘,

2.



版本控制_建立GitHub Repository

PyCharm File Edit View Navigate Code Refactor Run Tools VCS Window Help

pycharm_demo-master [~/Documents/pycharm_d

pycharm_demo-master > main.py

Project

pycharm_demo-master ~/D
- Algorithms
- my_module
  - __init__.py
  - hello_module.py
  - test_demo.py
- .gitignore

main.py | hello_module.py

```
from Algorithms.sorts import bubble_s
from Algorithms.other import two_sum

if __name__ == '__main__':
    demo = hello_module.Demo()
    print("I am ..." + demo.hello_wor

    print(two_sum.twoSum(
        [1, 2, 3, 4, 5, 6, 7, 8, 9, 1
```

VCS
- Local History
- Enable Version Control Integration...
- VCS Operations Popup... ^V
- Apply Patch...
- Apply Patch from Clipboard...
- Checkout from Version Control
- Import into Version Control
- Browse VCS Repository
- Sync Settings

Import into Version Control
- Import into CVS...
- Create Git Repository...
- Import into Subversion...
- Share Project (Subversion)...
- Create Mercurial Repository
- Share Project on GitHub

Share Project On GitHub

Repository name: pycharm_code Private

Remote: origin

Description:

Cancel Share

[Log In to Github]
輸入Github帳號/密碼

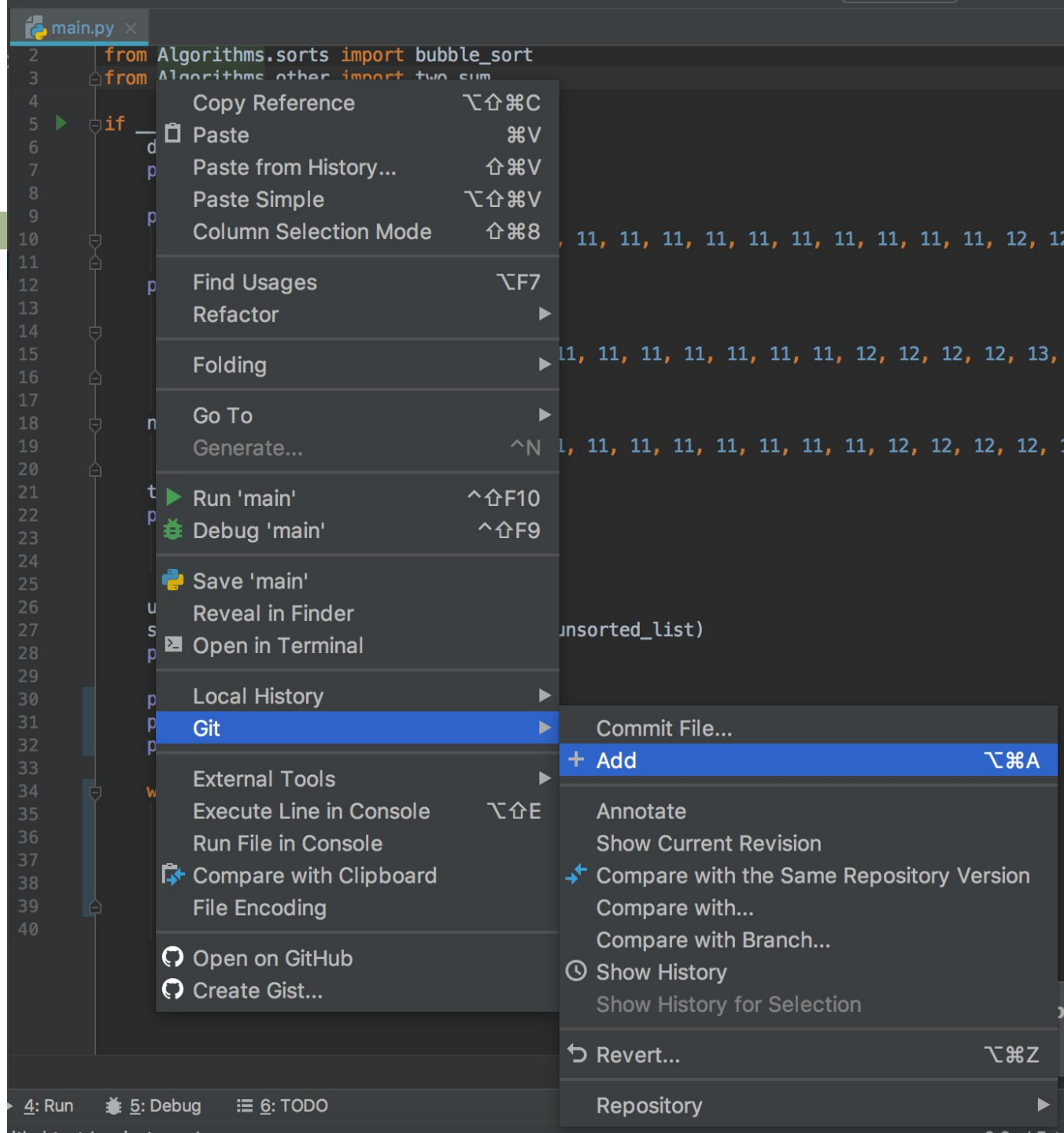
版本控制

Commit

在檔案上點擊右鍵 -> git -> Add -> Commit file

Push

在檔案上點擊右鍵 -> git -> Repository -> Push



PyLadies 主辦・系列活動

業界常用工具與技術分享

- PyCharm
  2018-11-18 (日)
  臺北市婦女館
  詳細路線說明
- Version control(日期時間未定)
  2018-12-23 (日)
  未定
- Unit Test(日期時間未定)
  2019-01-05 (六)
  未定
- Devops CI/CD
  2019-03-24 (日)
  未定
- Business Intelligence(日期時間未定)
  2019-04-06 (六)
  未定
- Distributed Task Queue/Scheduler/Monitor(日期時間未定)
  2019-05-04 (六)
  未定

Thanks for your attention!

林佳穎 Coco Lin
blue945u@gmail.com
https://www.linkedin.com/in/jiaying-lin-coco/