



GOOGLE CLASS ROOM

3g0g2x



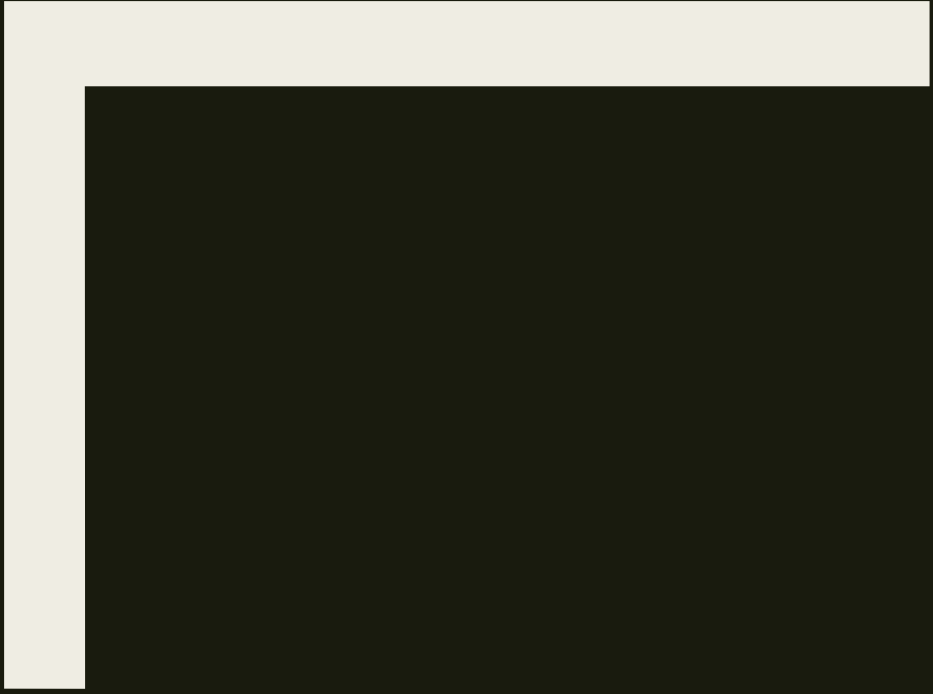


RECOMMENDATION FOR CODING AND LOGICAL



Content

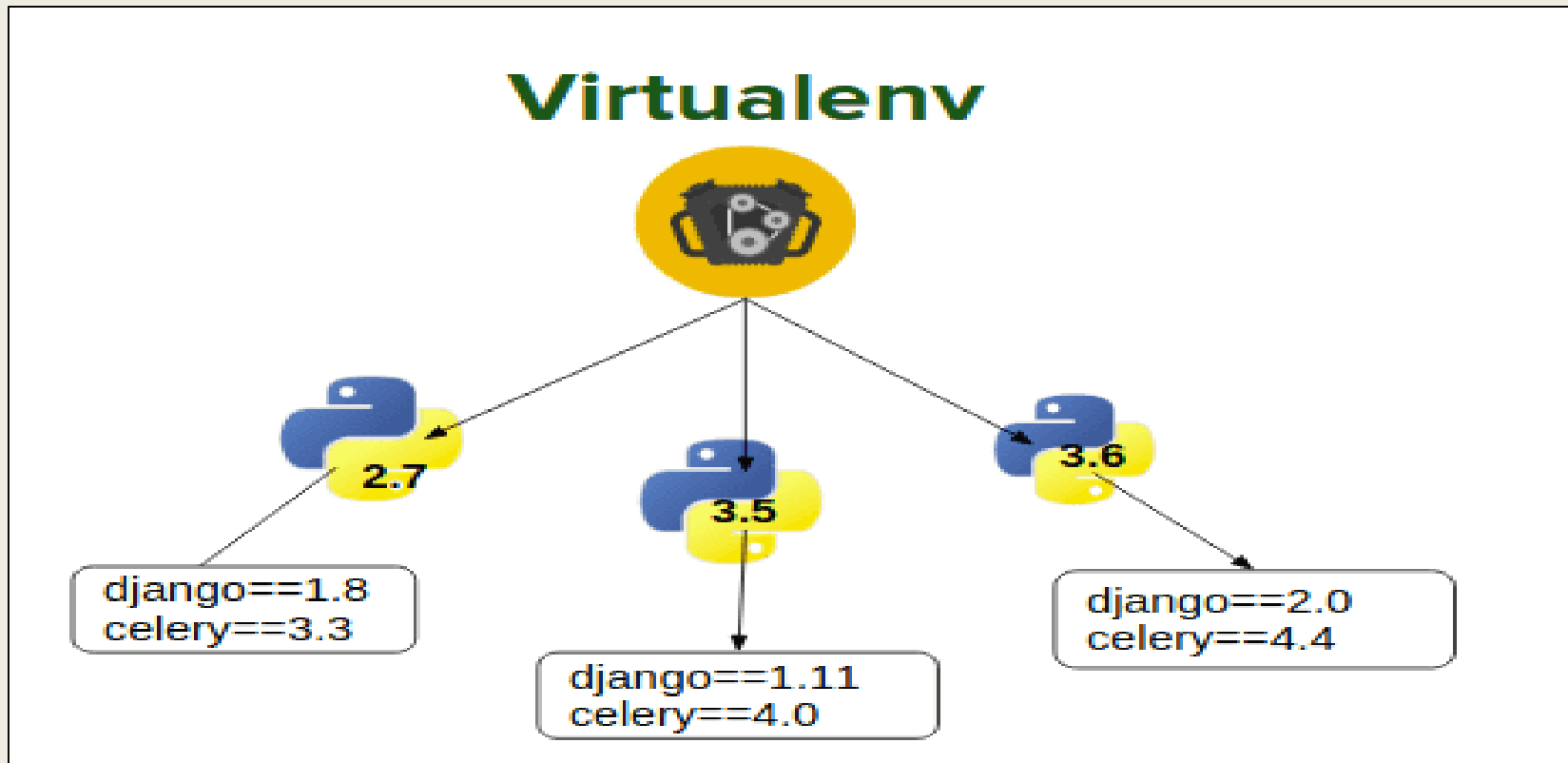
- Virtual-env
- Install lib
 - *Create and use Requirement.txt*
- Run Code on Server(CentOs7)
 - *Python-file*
 - *nohup*
 - *crontab*



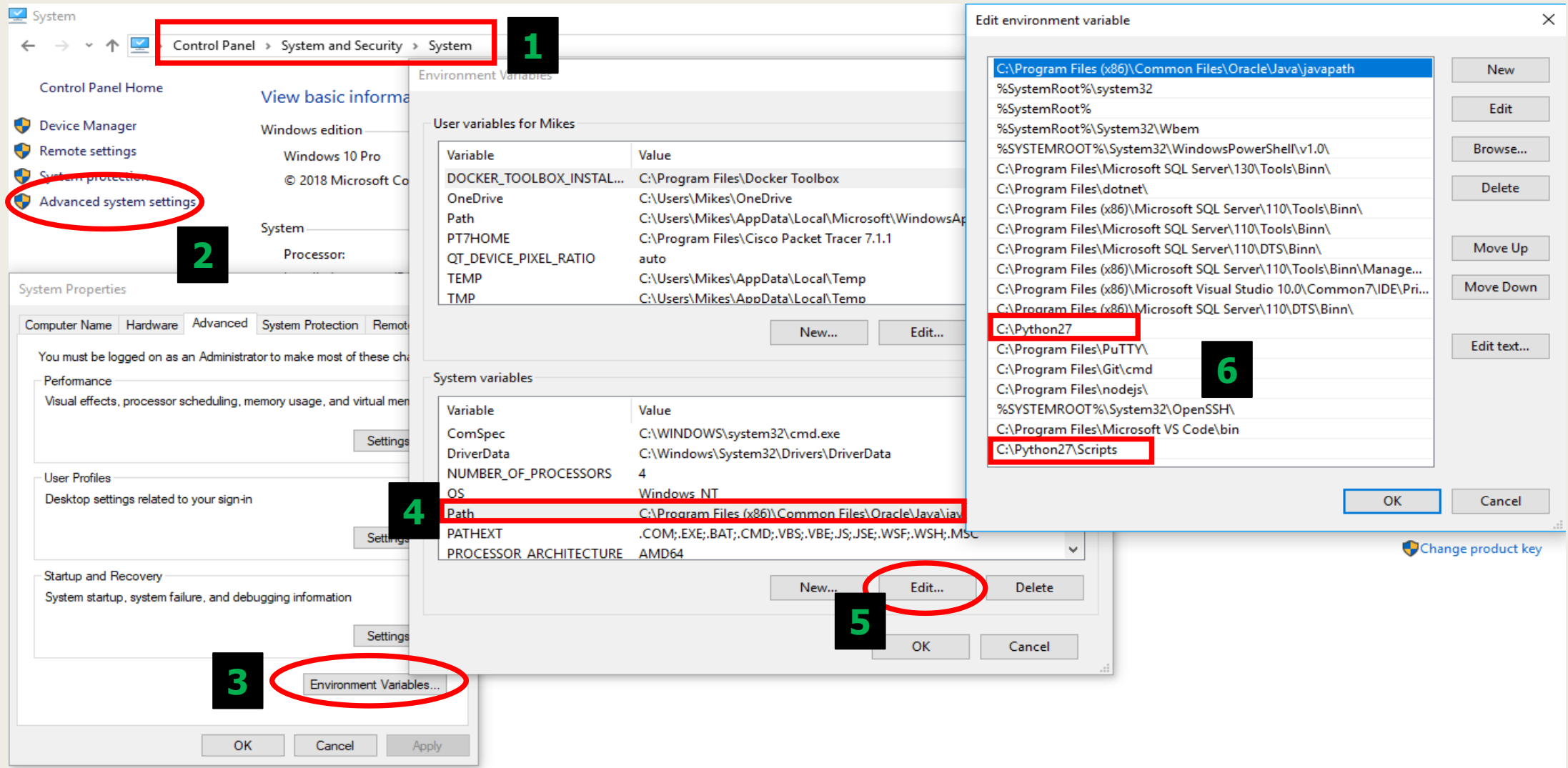
Virtual-env

Virtual-env (Virtual Environment)

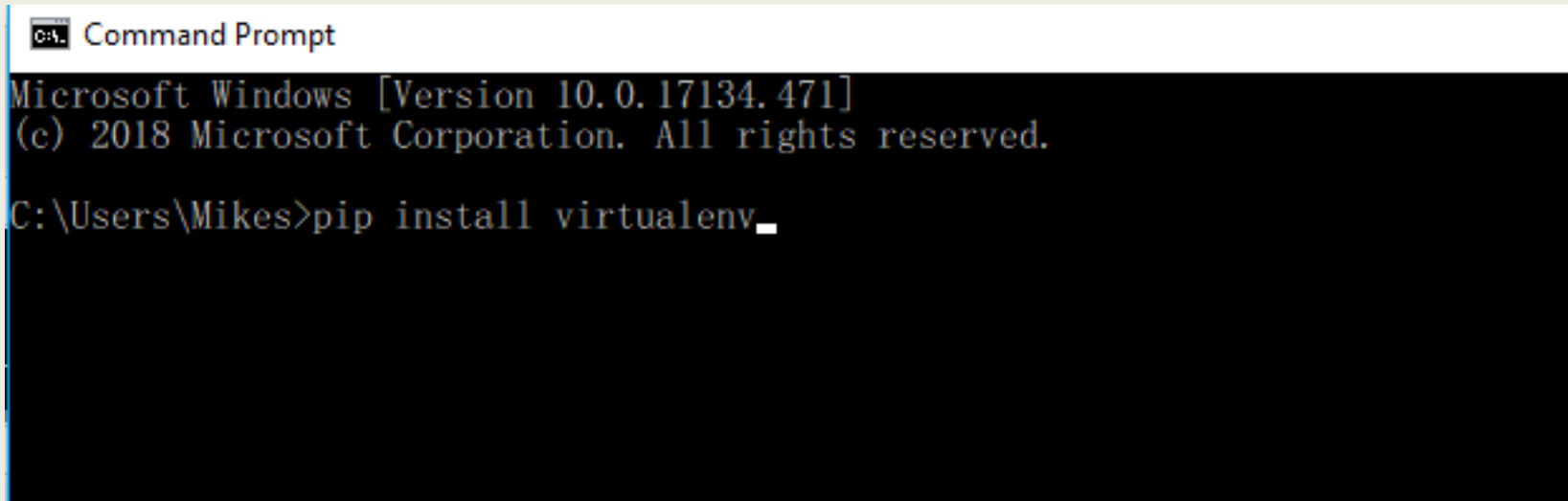
คือ แยก environment ในการทำงาน Python ของแต่ละโปรเจคออกจากกัน



เริ่มต้นด้วยการ Set Path ของ Python



เริ่มลง lib Virtualenv



```
Command Prompt
Microsoft Windows [Version 10.0.17134.471]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Mikes>pip install virtualenv_
```

คำสั่ง pip install virtualenv

ทำการสร้าง virtualenv กับ Project เรา

```
Command Prompt
Microsoft Windows [Version 10.0.17134.471]
(c) 2018 Microsoft Corporation. All rights reserved.

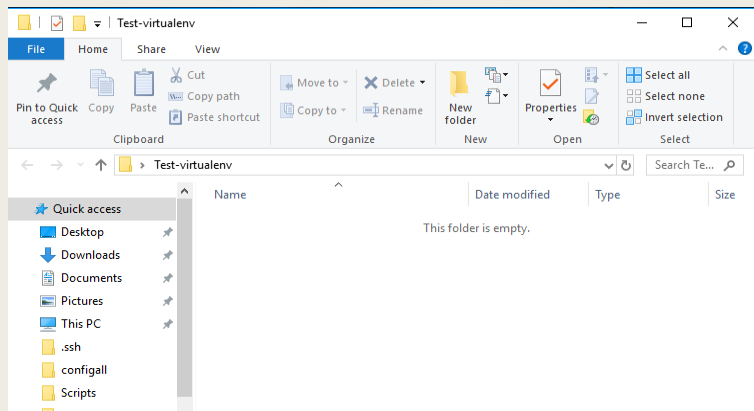
C:\Users\Mikes>cd C:\Users\Mikes\Desktop\Test-virtualenv
C:\Users\Mikes\Desktop\Test-virtualenv>virtualenv ENV
New python executable in C:\Users\Mikes\Desktop\Test-virtualenv\ENV\Scripts\python.exe
Installing setuptools, pip, wheel...
done.
```

1.เข้าไปที่ path project ของเรา

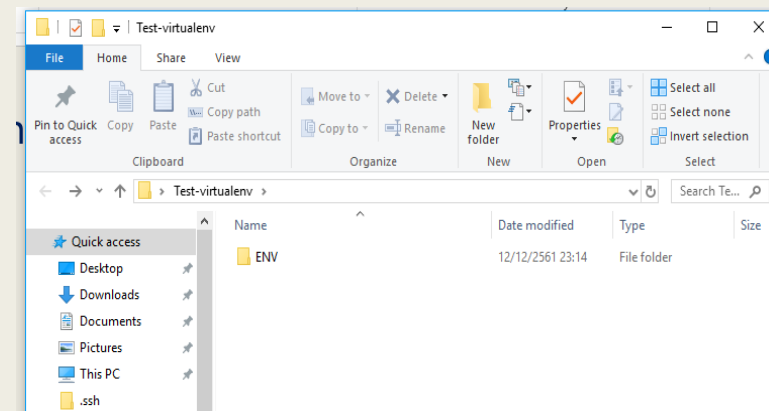
2.ทำการสั่งสร้าง virtualenv ชื่อ ENV

คำสั่ง virtualenv <ชื่อ>

Before



After



การลง lib ใน virtualenv ของเรา

```
C:\Users\Mikes\Desktop\Test-virtualenv>cd ENV
C:\Users\Mikes\Desktop\Test-virtualenv\ENV>cd Scripts
C:\Users\Mikes\Desktop\Test-virtualenv\ENV\Scripts>activate.bat
(ENV) C:\Users\Mikes\Desktop\Test-virtualenv\ENV\Scripts>pip install flask
```

ทำการเข้าไปใน Path ของ virtualenv

คำสั่งใช้งาน virtualenv

ลง lib ชื่อ Flask

คำสั่ง

activate.bat เพื่อเรียกใช้งาน virtualenv

pip install flask เพื่อ install lib

deactivate เพื่อออกจากการใช้งาน virtualenv

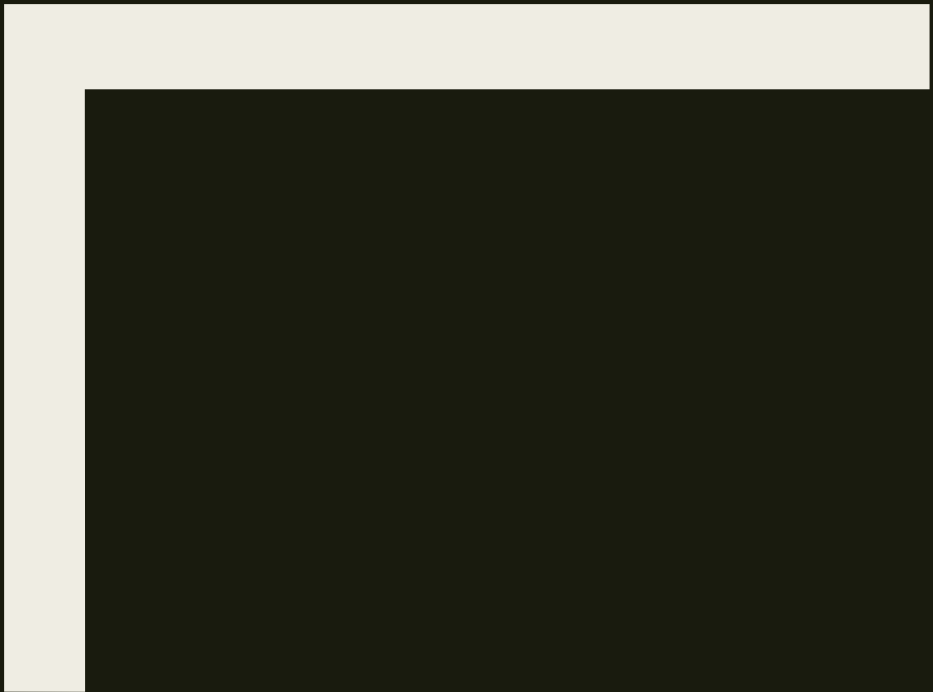
Hint หากเป็น Linux จะใช้คำสั่ง source [virtualenv name]/bin/activate ในการ activate

Lab 1

ให้ทำการสร้าง Virtual Environment ภายใน Project ที่สร้างขึ้นมาใหม่และทำการ install lib ที่ชื่อว่า Flask

ขั้นตอน

1. ทำการ New Folder Project
2. เข้า cmd และทำการ cd ไปที่ Path Project
3. ใช้คำสั่งสร้าง Virtual Environment
4. เข้าไปยัง Path ของ virtualenv แล้วเรียกใช้งาน virtualenv และจึงทำการสั่ง pip install <lib>



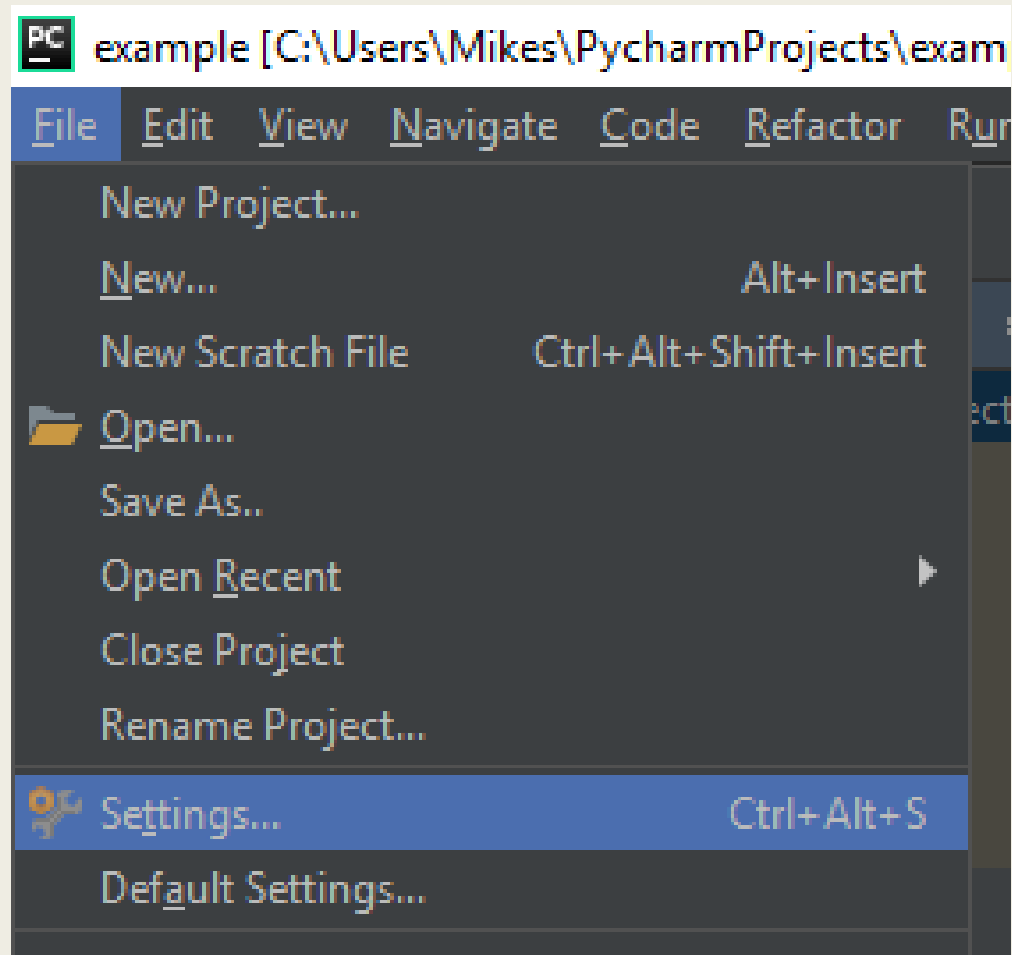
Install Libs

Install Libs

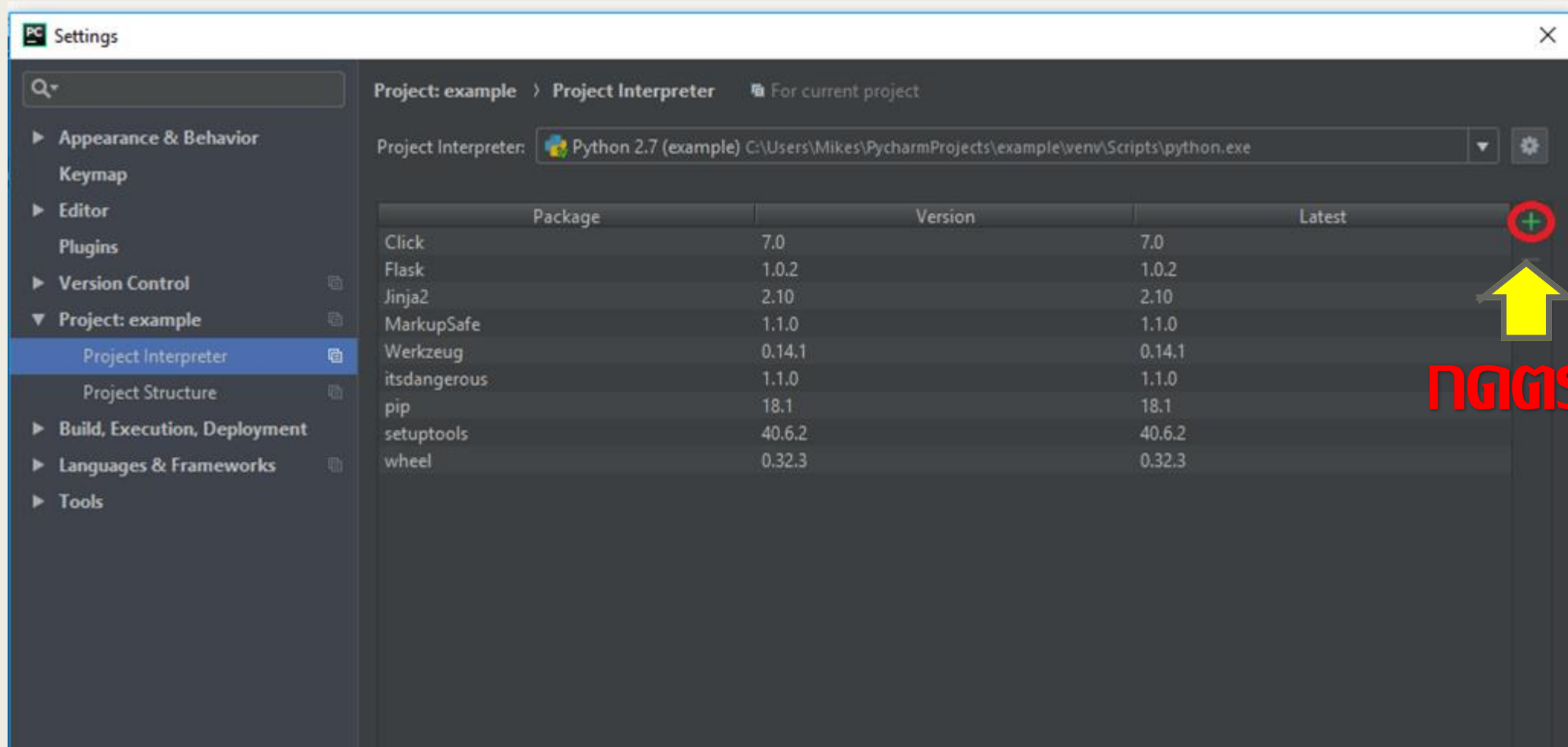
คำสั่ง **command**

`pip install <lib name>`

ลงโดย **Pycharm**



Install Libs ลงโดย Pycharm

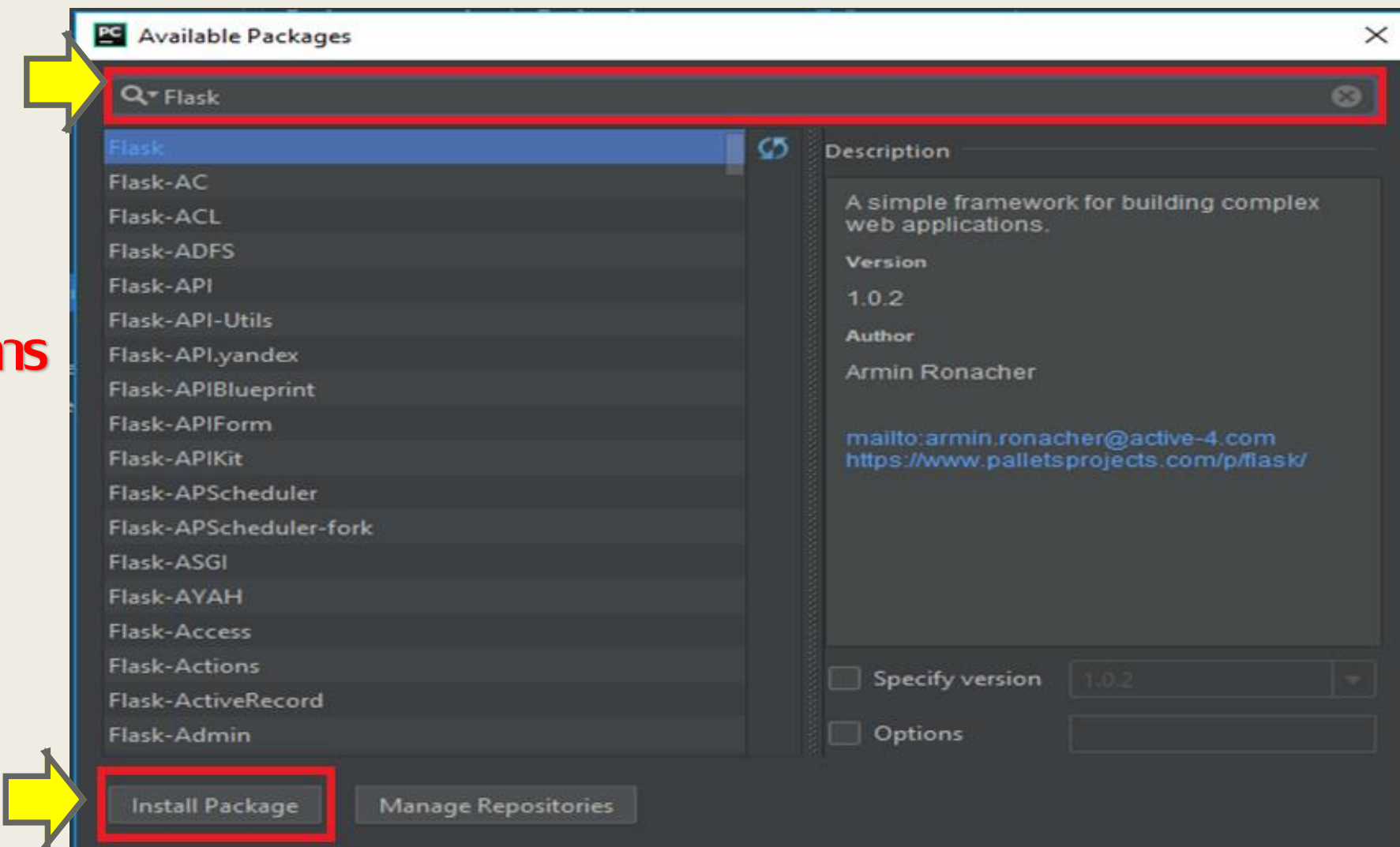


Install Libs ลงโดย Pycharm

ใส่ชื่อ lib เพื่อค้นหา

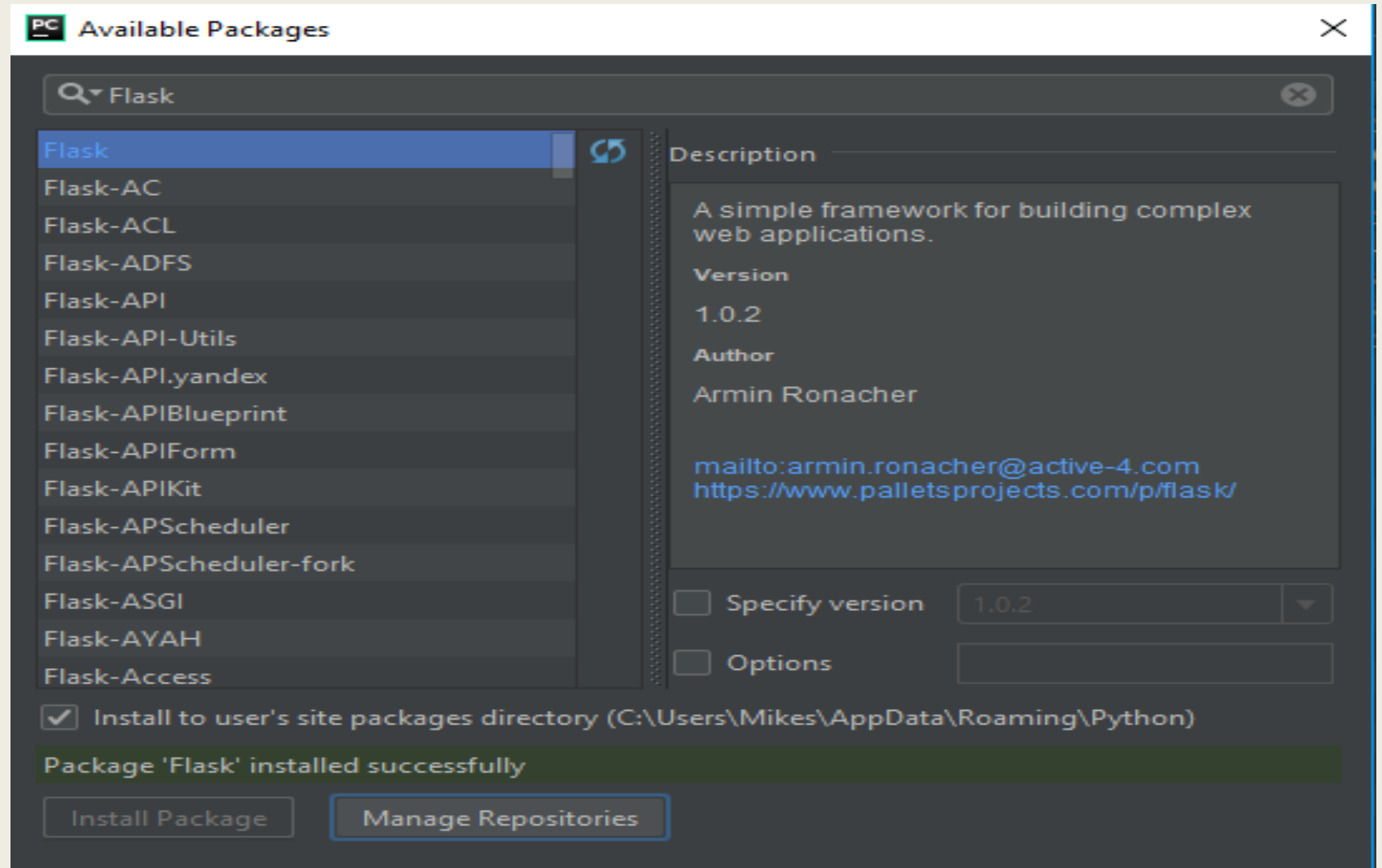
กดเลือก lib ที่ต้องการ
ติดตั้ง

กดเพื่อติดตั้ง lib



Install Libs

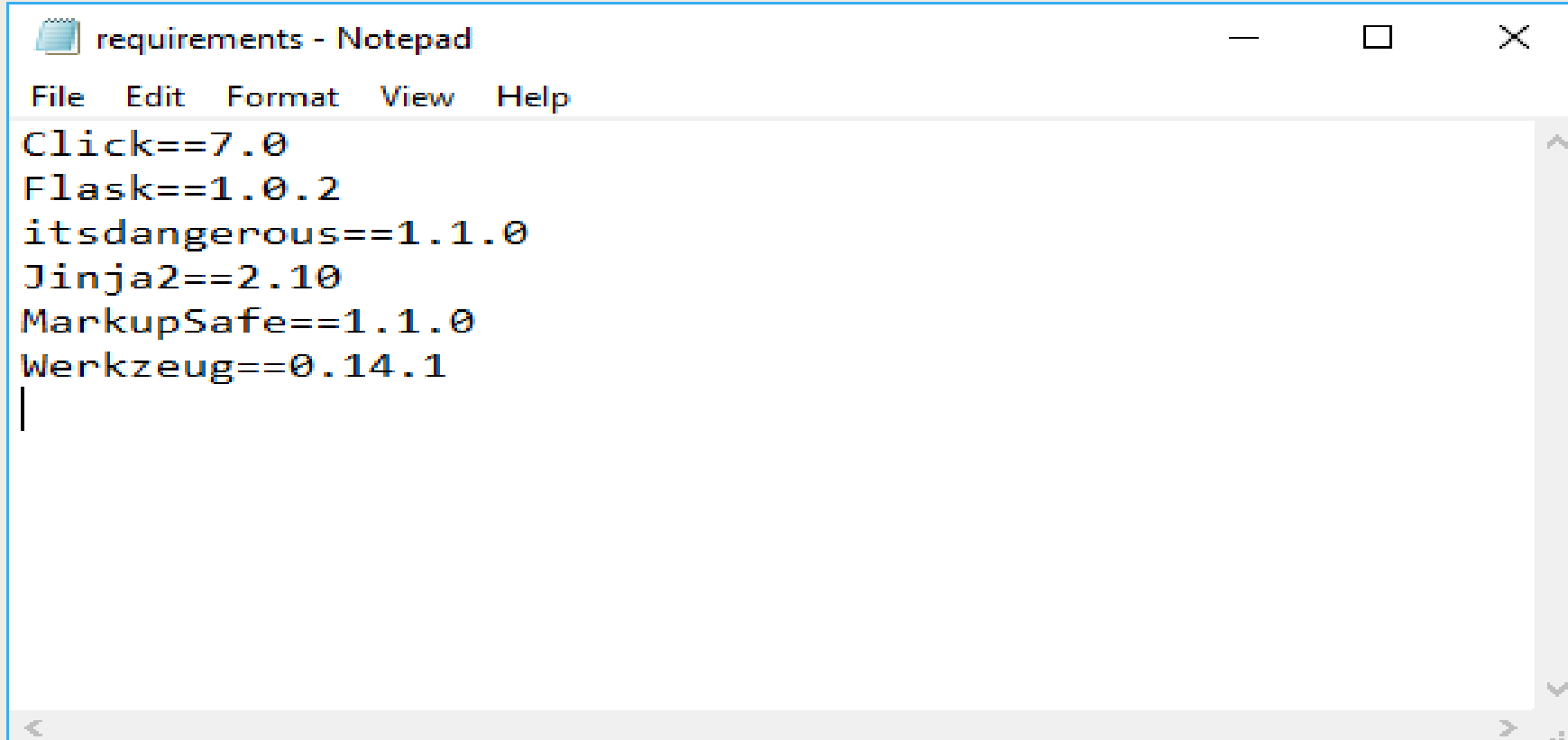
ติดตั้งสำเร็จเรียบร้อย



Requirements.txt

คือ ไฟล์ที่เก็บรายชื่อ lib ต่างๆที่เราใช้นั่นเอง

ตัวอย่างไฟล์



```
requirements - Notepad
File Edit Format View Help
Click==7.0
Flask==1.0.2
itsdangerous==1.1.0
Jinja2==2.10
MarkupSafe==1.1.0
Werkzeug==0.14.1
|
```


Requirements.txt ดียังไง ?

ถ้า Lib ที่ต้องลงมี
หลายตัว เช่น

- flask
- netmiko
- redis
- requests



pip install แบบปกติ

**pip install flask & netmiko &
redis & requests**

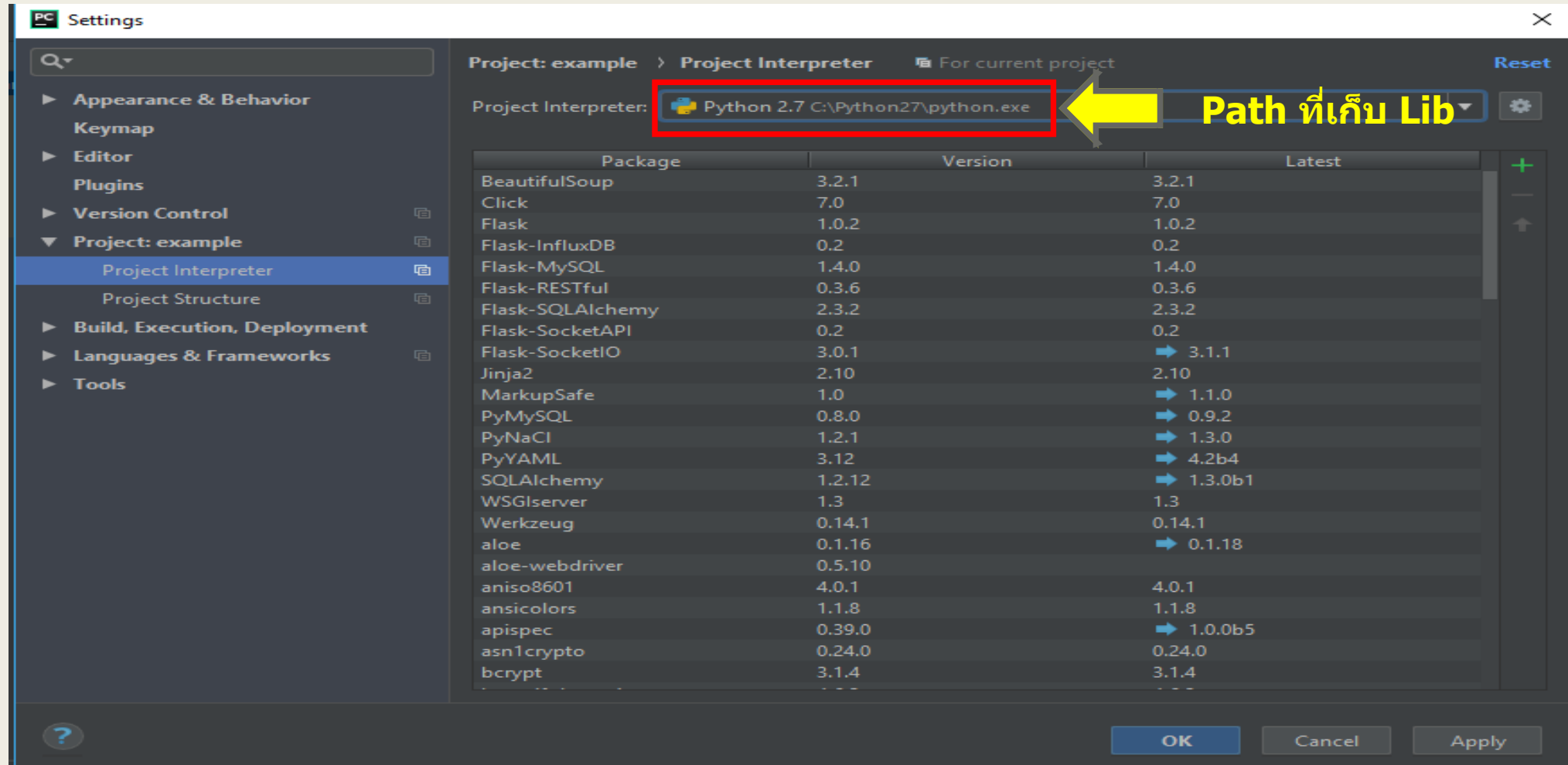


pip install แบบใช้ requirements

pip install -r requirements.txt

Requirements.txt สร้างยังไง?

ตัวอย่าง Lib



Requirements.txt สร้างยังไง?

คำสั่ง

pip freeze > requirements.txt

```
Command Prompt
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Mikes>cd C:\Python27\Scripts

C:\Python27\Scripts>pip freeze > requirements.txt
You are using pip version 8.1.1, however version 18.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Python27\Scripts>
```

เข้าไป Path ของ python เพื่อ
เข้าไปหา pip

ใช้คำสั่ง pip freeze เพื่อสร้าง
requirement.txt

Requirements.txt นำไปใช้อย่างไร?

คำสั่ง

```
pip install -r requirements.txt
```

```
requirements - Notepad
File Edit Format View Help
Click==7.0
Flask==1.0.2
itsdangerous==1.1.0
Jinja2==2.10
MarkupSafe==1.1.0
Werkzeug==0.14.1
|
```

```
1. Home 4. 35.243.183.158
os.remove(fullname)
OSError: [Errno 13] Permission denied: '/usr/lib/python2.7/site-packages/urllib3/__init__.py'
You are using pip version 8.1.2, however version 18.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
[mike@instance-1 ~]$ sudo pip install -r requirements.txt
[sudo] password for mike:
Collecting elasticsearch==6.2.0 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/c3/e3/146b675e6d0138a49c4b817b4e68170eb9b75cee7e71fa3ec69624c4f467/elasticsearch-6.2.0-py2.py3-none-any.whl (69kB)
    100% |#####| 71kB 2.0MB/s
Collecting elasticsearch-dsl==6.1.0 (from -r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/2a/cf/374708bbddf8b5ac750a68190ac459b2fa0b9505333f357c98f12c12d754/elasticsearch_dsl-6.1.0-py2.py3-none-any.whl (47kB)
    100% |#####| 51kB 3.9MB/s
Collecting enum34==1.1.6 (from -r requirements.txt (line 3))
  Downloading https://files.pythonhosted.org/packages/c5/db/e56e6b4bbac7c4a06de1c50de6fe1ef3810018ae11732a50f15f62c7d050/enum34-1.1.6-py2-none-any.whl
Requirement already satisfied (use --upgrade to upgrade): Flask==1.0.2 in /usr/lib64/python2.7/site-packages (from -r requirements.txt (line 4))
Collecting flask-apispec==0.7.0 (from -r requirements.txt (line 5))
  Downloading https://files.pythonhosted.org/packages/39/94/181cd383519bc2b078a95deaf1a11ee792fc572a85ed5c0aa198195fecad/flask_apispec-0.7.0-py2.py3-none-any.whl (2.9MB)
    100% |#####| 2.9MB 366kB/s
```

สรุป Requirements.txt

Host1



pip freeze > requirements.txt

```
requirements - Notepad
File Edit Format View Help
Click==7.0
Flask==1.0.2
itsdangerous==1.1.0
Jinja2==2.10
MarkupSafe==1.1.0
Werkzeug==0.14.1
```

Host2



pip install -r requirements.txt

```
requirements - Notepad
File Edit Format View Help
Click==7.0
Flask==1.0.2
itsdangerous==1.1.0
Jinja2==2.10
MarkupSafe==1.1.0
Werkzeug==0.14.1
```

ย้ายไฟล์ requirement.txt

Lab 2

ให้นำเอา lib จาก virtualenv ของ Lab1 ให้อยู่ในรูปแบบ requirements.txt และนำไปลงใน Vm เพื่อทำการลง lib

ขั้นตอน

1. เข้า cmd และทำการ cd ไปที่ Path Project
2. ใช้คำสั่งเพื่อนำ lib ออกมาในรูปแบบ requirements.txt
3. นำไฟล์ requirements.txt ไปยังเครื่อง Vm ที่ต้องการจะลง
4. ใช้คำสั่งเพื่อลง lib โดยใช้ requirements.txt



Run Code on
Server(CentOs7)

RUN PYTHON ON SERVER (CentOS7)

คำสั่ง

```
python <filename>.py
```

ไฟล์ตัวอย่าง example.py

```
from datetime import datetime  
print datetime.now()
```

ตัวอย่างการสั่งรัน

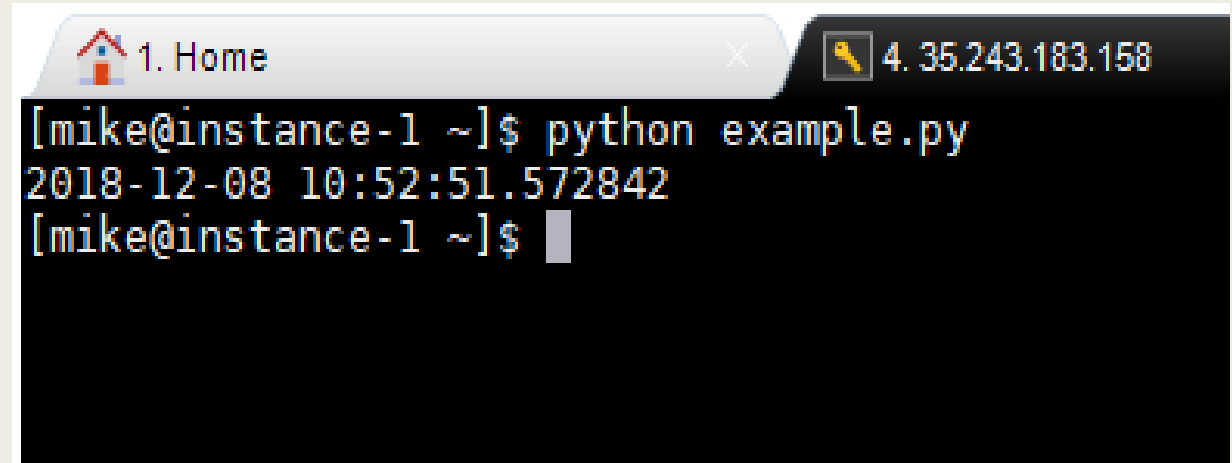
```
python example.py
```

การย้ายไฟล์สำหรับ Mac

```
sudo scp <path file source> <user>@<ip host>:<pathfile destination>
```

```
ตัวอย่าง sudo scp -i <keypath> ~/Documents/INET/Authen.py centos@203.154.58.132:python-training-lab
```

ผลลัพธ์

A screenshot of a terminal window titled "1. Home" with a close button. The terminal shows the command "python example.py" being executed. The output is "2018-12-08 10:52:51.572842". The prompt is "[mike@instance-1 ~]\$". The IP address "4. 35.243.183.158" is visible in the top right corner.

```
1. Home 4. 35.243.183.158  
[mike@instance-1 ~]$ python example.py  
2018-12-08 10:52:51.572842  
[mike@instance-1 ~]$
```


nohup (no hangup)

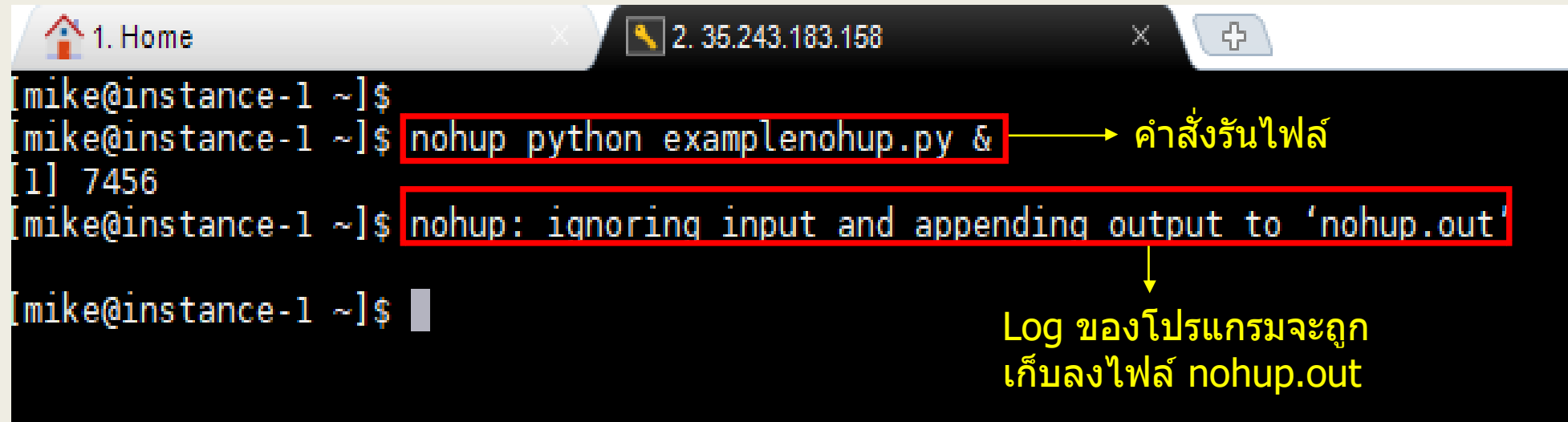
คือ คำสั่งทาง Linux ที่ใช้ในการป้องกันการหยุดของ background process

การรัน background process ปกติ

```
python <filename>.py &
```

การรัน background process โดยใช้ nohup

```
nohup python <filename>.py &
```



The image shows a terminal window with two tabs: '1. Home' and '2. 35.243.183.158'. The terminal output is as follows:

```
[mike@instance-1 ~]$  
[mike@instance-1 ~]$ nohup python examplenohup.py &  
[1] 7456  
[mike@instance-1 ~]$ nohup: ignoring input and appending output to 'nohup.out'  
[mike@instance-1 ~]$
```

Annotations in the image:

- A red box highlights the command `nohup python examplenohup.py &` in the second line. A yellow arrow points from this box to the text "คำสั่งรันไฟล์" (Run file command).
- A red box highlights the message `nohup: ignoring input and appending output to 'nohup.out'` in the third line. A yellow arrow points from this box to the text "Log ของโปรแกรมจะถูกเก็บลงไฟล์ nohup.out" (Program log will be saved to file nohup.out).

การใช้การทำงาน nohup

คำสั่ง

```
ps -aux | grep <filename>
```

มีไฟล์ที่ทำการรัน nohup อยู่ซึ่ง PID คือ 7456

```
[mike@instance-1 ~]$ ps aux|grep examplenohup.py
mike    7456  0.0  0.2 127564  4660 pts/1    S   11:10   0:00 python examplenohup.py
mike    7526  0.0  0.0 112708   992 pts/1    R+  11:16   0:00 grep --color=auto examplenohup.py
```

เลข PID

การสั่งปิดการทำงานของ nohup

คำสั่ง

```
kill -9 <เลข PID>
```

มีไฟล์ที่ทำการรัน nohup อยู่ซึ่ง PID คือ 7456

```
[mike@instance-1 ~]$ ps aux|grep examplenohup.py
mike      7456  0.0  0.2 127564  4660 pts/1    S   11:10   0:00 python examplenohup.py
mike      7526  0.0  0.0 112708   992 pts/1    R+  11:16   0:00 grep --color=auto examplenohup.py
[mike@instance-1 ~]$ kill -9 7456 → สั่งปิดการทำงาน
[mike@instance-1 ~]$ ps aux|grep examplenohup.py
mike      7578  0.0  0.0 112708   992 pts/1    S+  11:22   0:00 grep --color=auto examplenohup.py
[1]+  Killed                  nohup python examplenohup.py
[mike@instance-1 ~]$
```

ไฟล์ที่ทำการรันอยู่ถูกปิดการทำงานเรียบร้อยแล้ว

การเอา nohup ไปใช้งาน (ลองทำดู)

ส่วนมากใช้ nohup กับ Code ที่ต้องการให้ Run ตลอดเวลา เช่น Flask, Web Server

ตัวอย่าง Code ของ Flask

flasktest.py

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"
app.run(host="0.0.0.0",port=5000)
```

ตัวอย่างการ Run

```
nohup python flasktest.py &
```

ตัวอย่าง While True ที่รันตลอดเวลา

example_w_file.py

```
from datetime import datetime
import time
i=0
while True:
    i=i+1
    print i
    time.sleep(1)
```

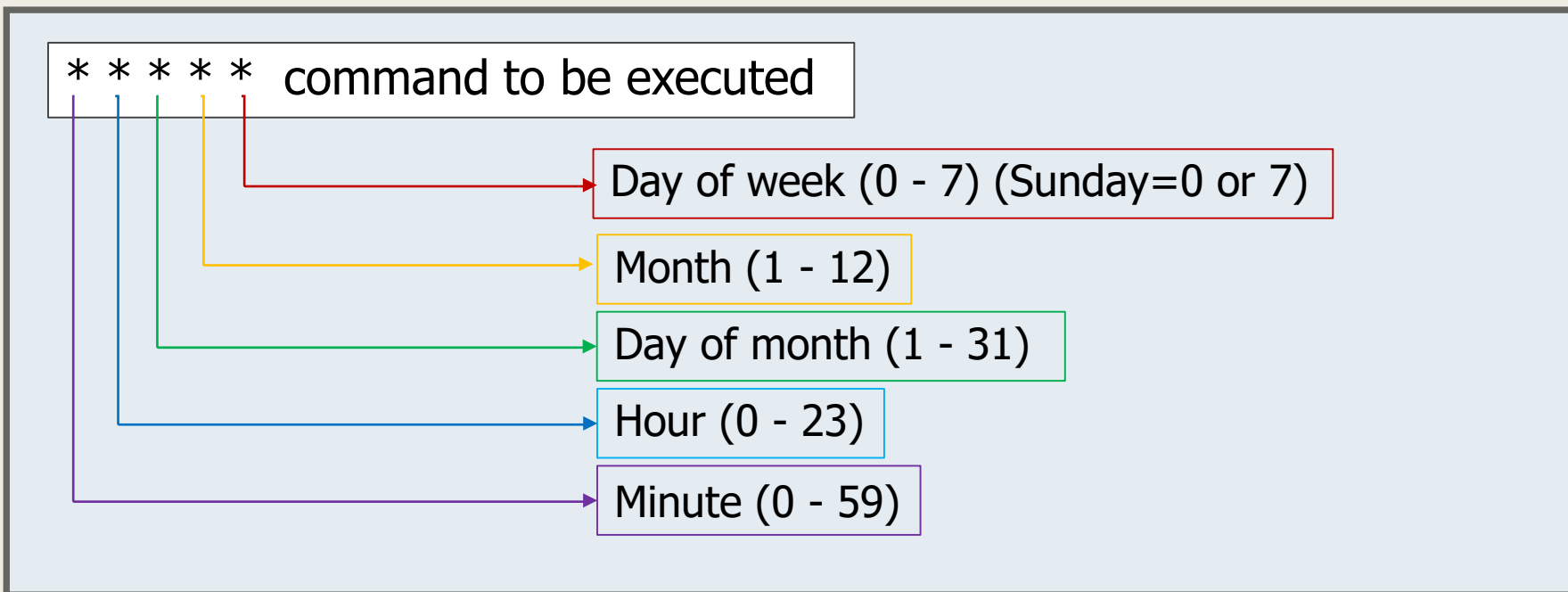
ตัวอย่างการ Run

```
nohup python -u example_w_file.py > output.txt &
```

Crontab (Cron Jobs)

คือ คำสั่งที่จะทำงานตามเวลาที่กำหนด

รูปแบบ Crontab ใน File config



เริ่มต้นใช้ Crontab

1. เริ่มด้วยคำสั่งเพื่อเข้า File Config ของ Crontab

```
crontab -e
```

2. จะเข้าไปใน File Config ของ Crontab และเริ่มการตั้ง Config

```
*/1 * * * * python examplecrontab.py
```

ตัวอย่างค่าที่ Config นี้จะทำการ Run ไฟล์ Python ที่ชื่อ examplecrontab.py ทุกๆ 1 นาที

ตัวอย่างรูปแบบการตั้งเวลาของ Crontab

คำสั่ง	ความหมาย
1 * * * * <คำสั่ง>	จะ Run ต่อเมื่อเวลาเป็น 1นาทื เช่น 1:01,2:01,3:01
* * * * * <คำสั่ง>	จะ Run ทุกๆนาทื
0 */1 * * * <คำสั่ง>	จะ Run ทุกๆชั่วโมง เช่น 12:00,1:00,7:00
0 0 1 * * <คำสั่ง>	จะ Run ทุกๆวันที่ 1 ของเดือนนั้นๆในเวลา 00:00 เช่น วันที่ 1/1/2019 เวลา 00:00 วันที่ 1/2/2019 เวลา 00:00

ปล. เว็บไซต์การแปลงเวลาของ Crontab https://crontab.guru/#*/1_*_*_*_*

ตัวอย่างการใช้ Crontab (ลองทำดู)

1. Code Python ทำหน้าที่เขียนเวลาลง File txt_testcrontab.txt

```
from datetime import datetime  
file = open("txt_testcrontab.txt","w")  
file.write(str(datetime.now()))
```

2. File Config ของ Crontab (เข้าถึงโดยคำสั่ง crontab -e)

```
*/1 * * * * python examplecrontab.py
```

ผลลัพธ์ ไฟล์ **txt_testcrontab.txt** จะต้องมีการ
เขียนค่าของเวลาตลอดทุกๆ 1 นาที

3. หลังจากเซฟไฟล์ที่ทำการ Config แล้วให้ทำการ restart service Crontab โดยคำสั่ง

```
sudo systemctl restart crond
```


nohup VS Crontab

nohup	Crontab
<p>จุดประสงค์</p> <ul style="list-style-type: none">• ใช้กับ Code ที่ต้องการรันตลอดเวลาไม่มีวันจบ <p>ข้อเสีย</p> <ul style="list-style-type: none">• หาก Code เกิด error จะไม่ทราบ• หาก Vm down จำเป็นต้องมาสั่งรันใหม่	<p>จุดประสงค์</p> <ul style="list-style-type: none">• ใช้สำหรับ Code ที่ทำแล้วจบ• ใช้สำหรับ Code ที่ต้องการรันเป็นช่วงเวลา <p>ข้อเสีย</p> <ul style="list-style-type: none">• Config เข้าใจยาก• หากตั้งค่า Config ผิดจะไม่ทราบ ต้องตรวจเช็คผลลัพธ์ถึงจะรู้ว่า Contrab ไม่ทำงาน