PYTHON FOR BEGINNER

Prepared by: Celeste Ng

Date: February, 2018

Download the software (1)

(Source URL: https://www.python.org/about/gettingstarted/)

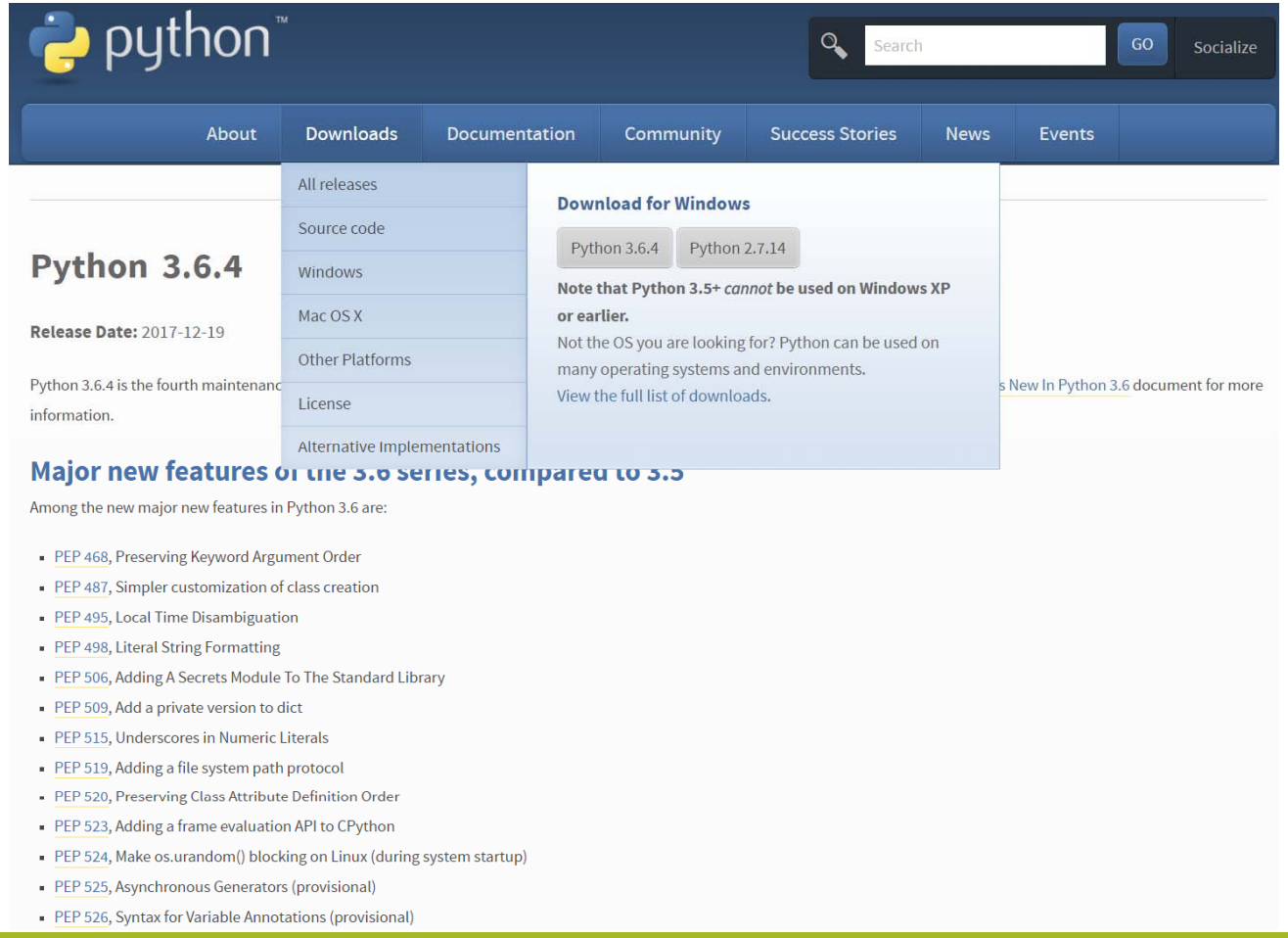
- Installing Python is generally easy, and nowadays many Linux and UNIX distributions include a recent Python. Even some Windows computers (notably those from HP) now come with Python already installed. If you *do* need to install Python and aren't confident about the task you can find a few notes on the BeginnersGuide/Download wiki page, but installation is unremarkable on most platforms.

Downloading Python -1

(Source URL: https://wiki.python.org/moin/BeginnersGuide/Download)

- Before you start, you will need Python on your computer, but you may not need to download it.
- First of all check that you don't already have Python installed by entering **python** in a command line window. If you see a response from a Python interpreter it will include a version number in its initial display. Generally any recent version will do, as Python makes every attempt to maintain backwards compatibility.
- If you need to install Python, you may as well download the most recent stable version. This is the one with the highest number that isn't marked as an alpha or beta release. Please see the Python downloads page for the most up to date versions of Python ...
- **If you're running Windows**: the most stable Windows downloads are available from the Python for Windows page.

Source: https://www.python.org/downloads/release/python-364/



Downloading Python -2

(Source URL: https://wiki.python.org/moin/BeginnersGuide/Download)

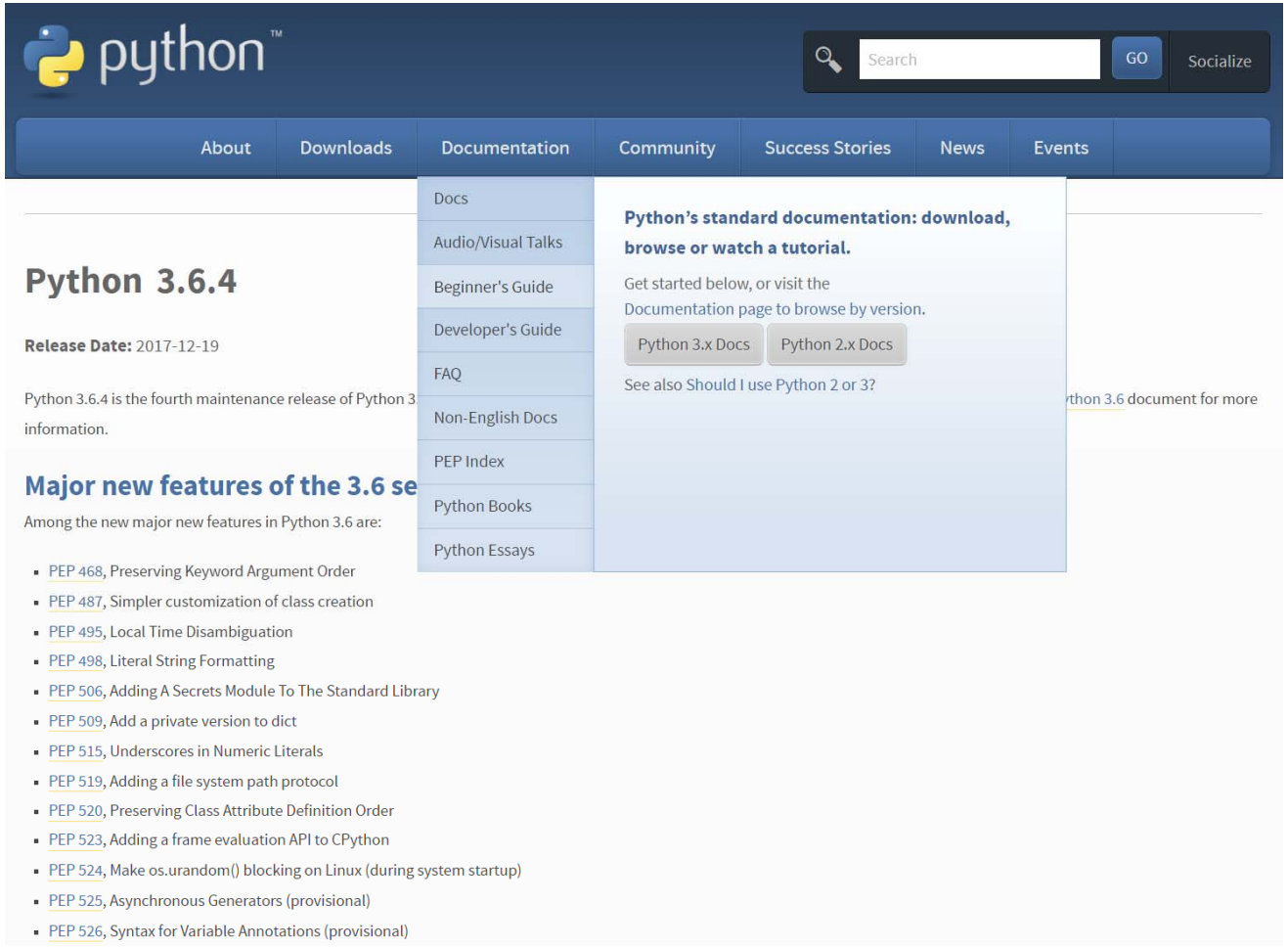
- **If you're running Windows XP**: a complete guide to installing ActivePython is at Python on XP: 7 Minutes To "Hello World!". ShowMeDo has two videos for downloading, installing and getting started with Python on a Windows XP machine - this series talks you through the Python, ActivePython and SciPy distributions.
- **If you are using a Mac**, see the Python for Mac OS X page. MacOS 10.2 (Jaguar), 10.3 (Panther), 10.4 (Tiger) and 10.5 (Leopard) already include various versions of Python.
- **For Red Hat**, install the python2 and python2-devel packages.
- **For Debian or Ubuntu**, install the python2.x and python2.x-dev packages.
- **For Gentoo**, install the '=python-2.x*' ebuild (you may have to unmask it first).
- **For other systems**, or **if you want to install from source**, see the general download page.

Learning

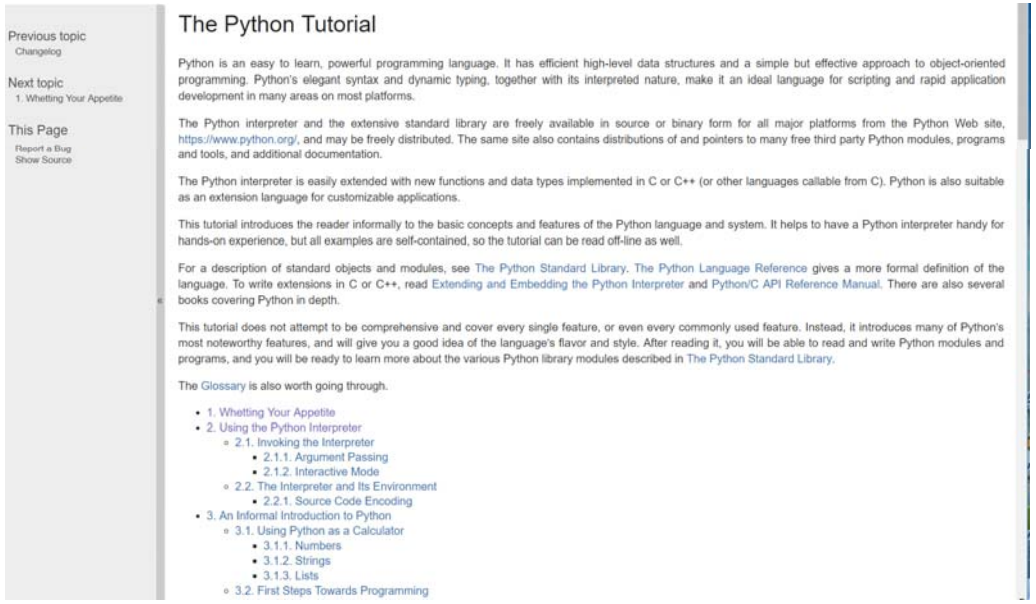
(Source URL: https://www.python.org/about/gettingstarted/)

- Before getting started, you may want to find out which IDEs and text editors are tailored to make Python editing easy, browse the list of introductory books, or look at code samples that you might find helpful.
- There is a list of tutorials suitable for experienced programmers on the BeginnersGuide/Tutorials page. There is also a list of resources in other languages which might be useful if English is not your first language.
- The online documentation is your first port of call for definitive information. There is a fairly brief tutorial that gives you basic information about the language and gets you started. You can follow this by looking at the library reference for a full description of Python's many libraries and the language reference for a complete (though somewhat dry) explanation of Python's syntax. If you are looking for common Python recipes and patterns, you can browse the ActiveState Python Cookbook

Source: https://www.python.org/downloads/release/python-364/



Source: https://docs.python.org/3/tutorial/index.html



Previous topic
Changelog

Next topic
1. Whetting Your Appetite

This Page
Report a Bug
Show Source

The Python Tutorial

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

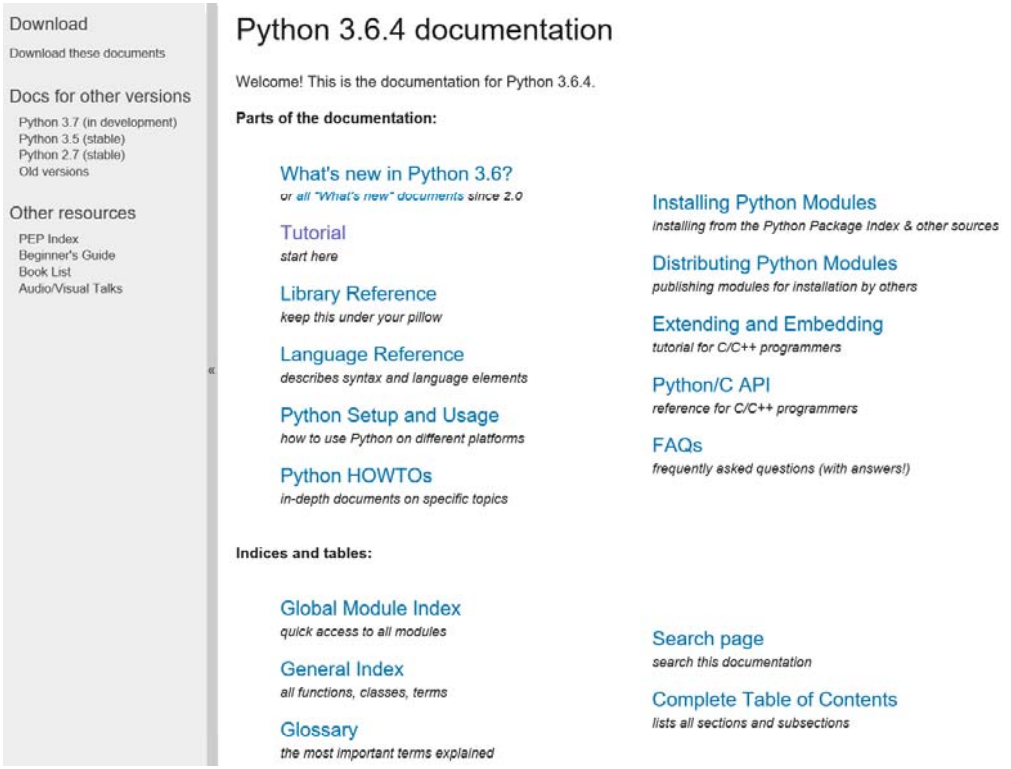
This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.

For a description of standard objects and modules, see The Python Standard Library. The Python Language Reference gives a more formal definition of the language. To write extensions in C or C++, read Extending and Embedding the Python Interpreter and Python/C API Reference Manual. There are also several books covering Python in depth.

This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most noteworthy features, and will give you a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in The Python Standard Library.

The Glossary is also worth going through.

Source: https://docs.python.org/3.6/index.html



Download
Download these documents

Docs for other versions
Python 3.7 (in development)
Python 3.5 (stable)
Python 2.7 (stable)
Old versions

Other resources
PEP Index
Beginner's Guide
Book List
Audio/Visual Talks

Python 3.6.4 documentation

Welcome! This is the documentation for Python 3.6.4.

Parts of the documentation:

What's new in Python 3.6?
or all "What's new" documents since 2.0

Tutorial
start here

Library Reference
keep this under your pillow

Language Reference
describes syntax and language elements

Python Setup and Usage
how to use Python on different platforms

Python HOWTOs
in-depth documents on specific topics

Installing Python Modules
installing from the Python Package Index & other sources

Distributing Python Modules
publishing modules for installation by others

Extending and Embedding
tutorial for C/C++ programmers

Python/C API
reference for C/C++ programmers

FAQs
frequently asked questions (with answers!)

Indices and tables:

Global Module Index
quick access to all modules

General Index
all functions, classes, terms

Glossary
the most important terms explained

Search page
search this documentation

Complete Table of Contents
lists all sections and subsections

Looking for Something Specific?

(Source URL: https://www.python.org/about/gettingstarted/)

- If you want to know whether a particular application, or a library with particular functionality, is available in Python there are a number of possible sources of information.
- The Python web site provides a Python Package Index (also known as the *Cheese Shop*, a reference to the Monty Python script of that name).
- There is also a search page for a number of sources of Python-related information. Failing that, just Google for a phrase including the word "python" and you may well get the result you need.
- If all else fails, ask on the python newsgroup and there's a good chance someone will put you on the right track.

Frequently Asked Questions

(Source URL: https://www.python.org/about/gettingstarted/)

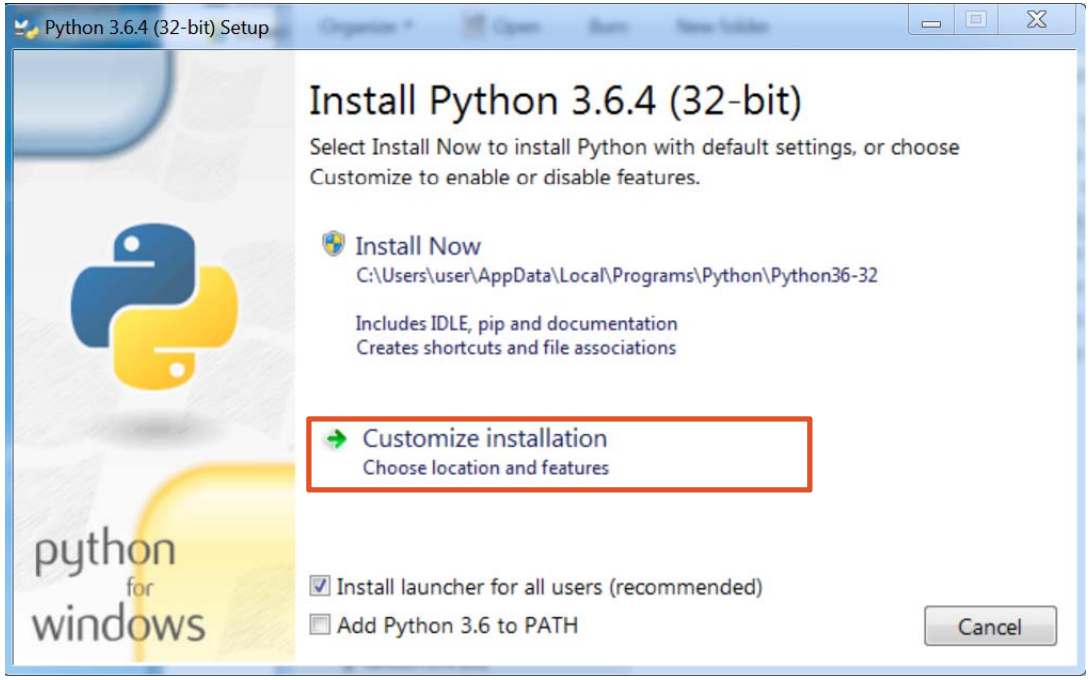
- If you have a question, it's a good idea to try the FAQ, which answers the most commonly asked questions about Python.

Looking to Help?

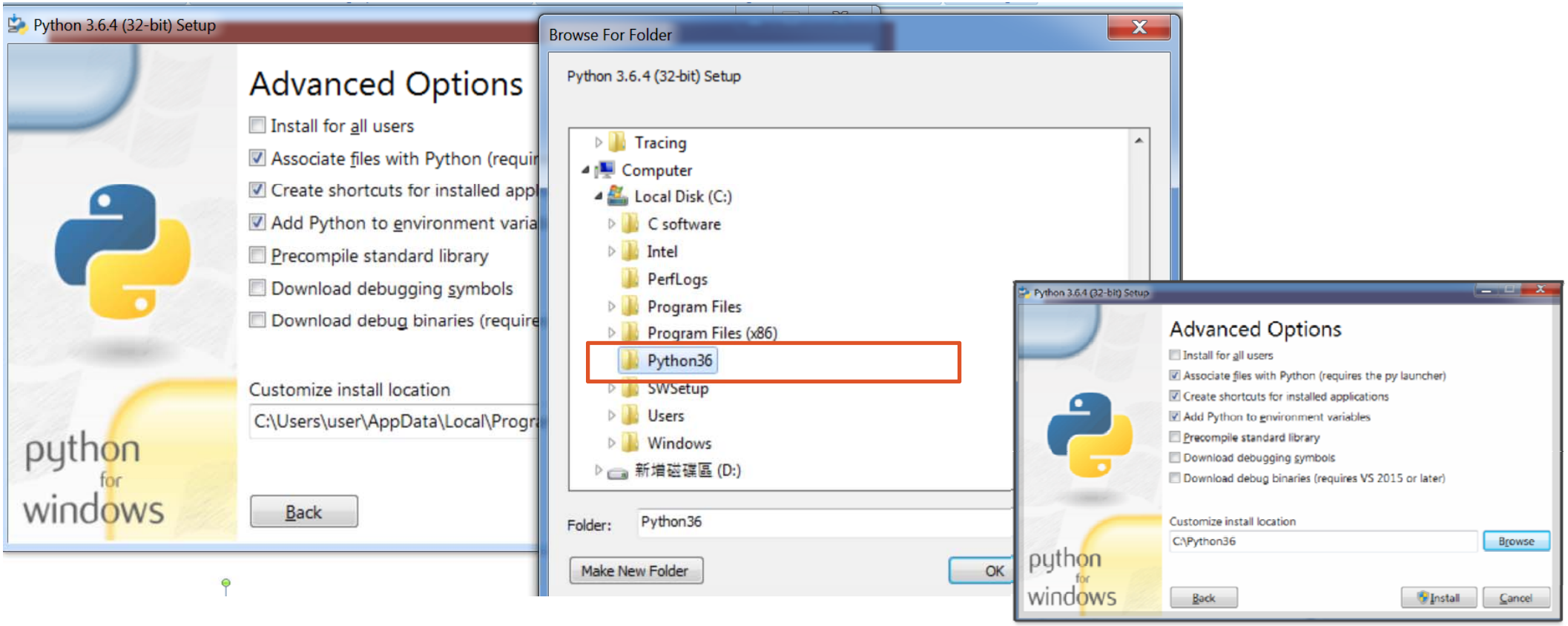
(Source URL: https://www.python.org/about/gettingstarted/)

- If you want to help to develop Python, take a look at the developer area for further information. Please note that you don't have to be an expert programmer to help. The documentation is just as important as the compiler, and still needs plenty of work!

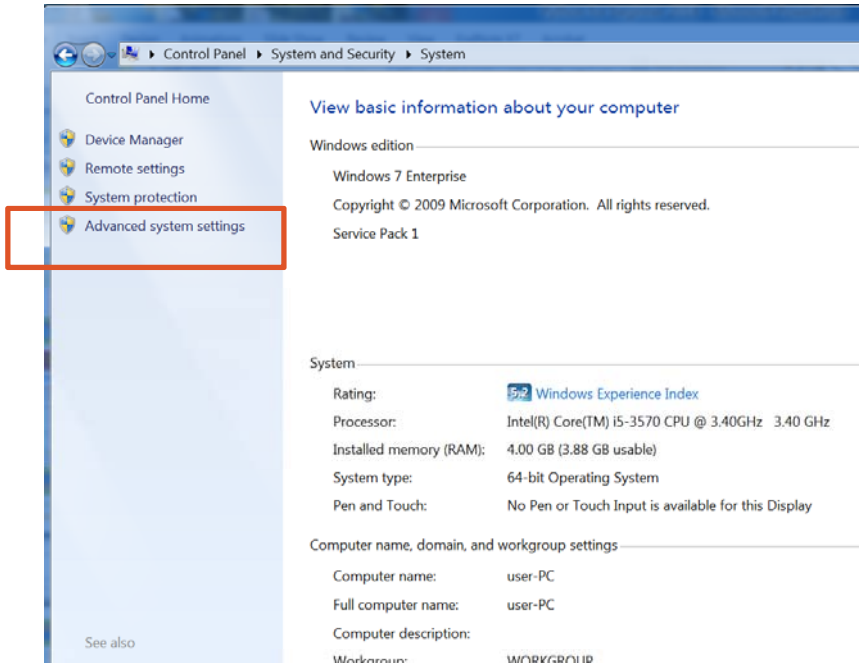
1. Installing Python 3.6.4 (1)

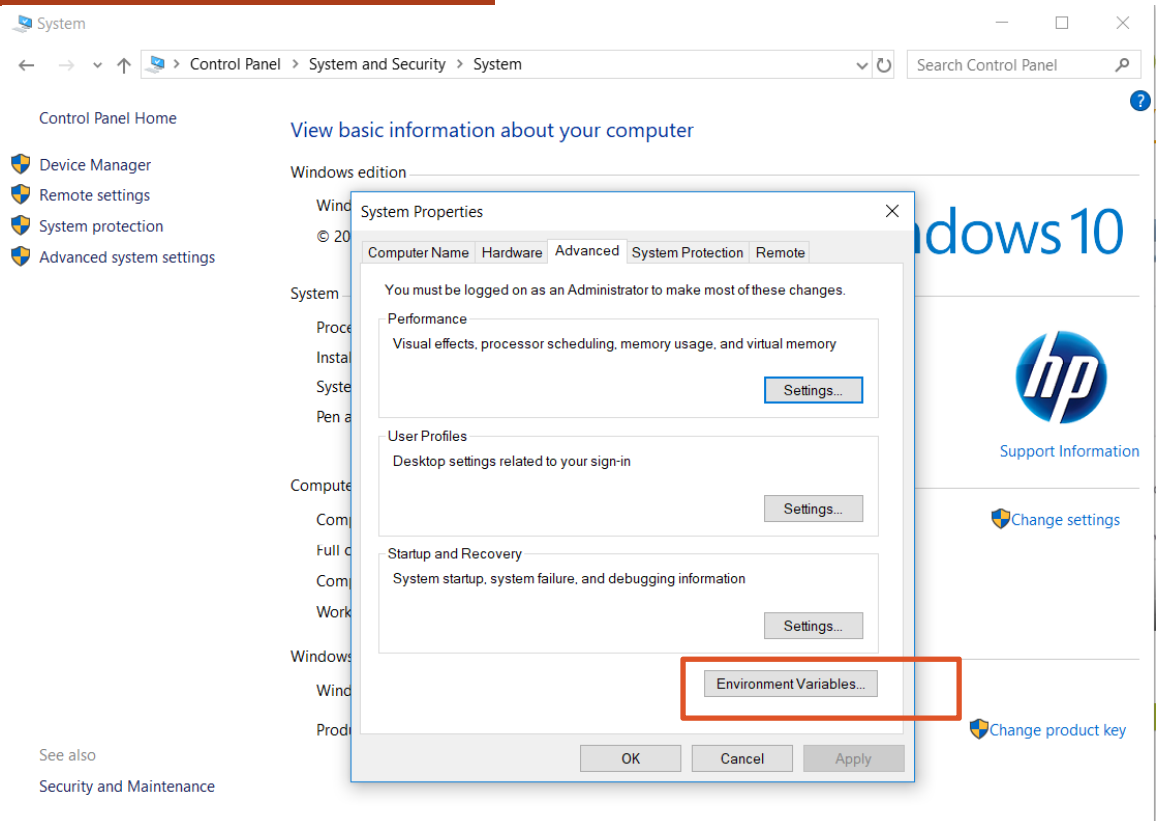


1. Installing Python 3.6.4 (2)

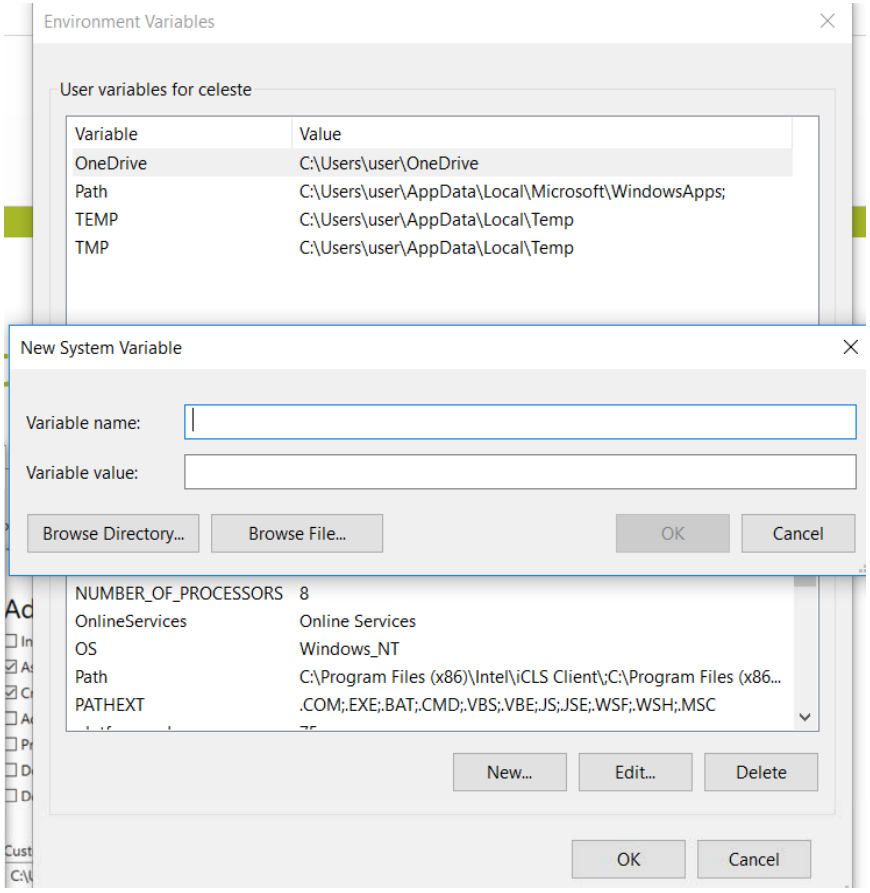


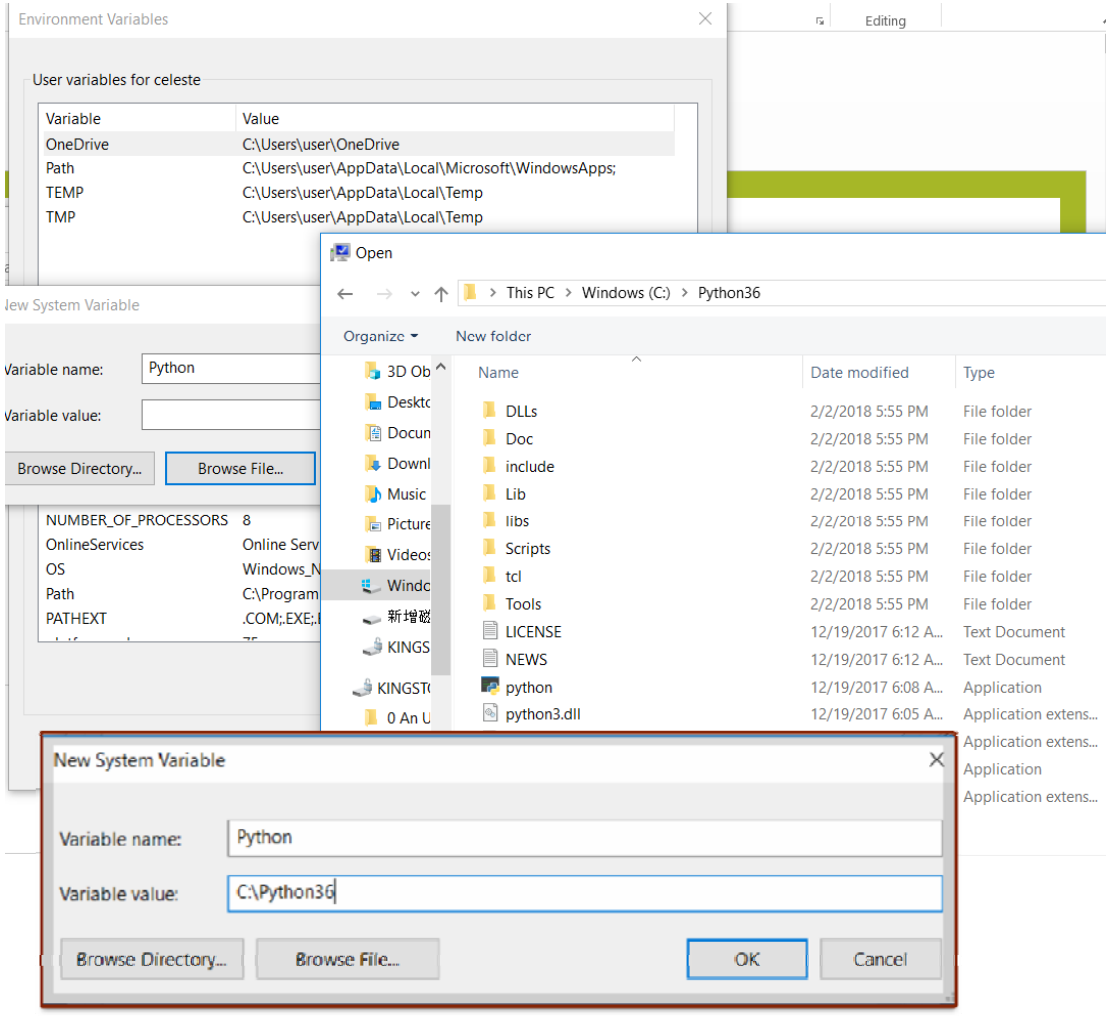
2. Setting environment (1)





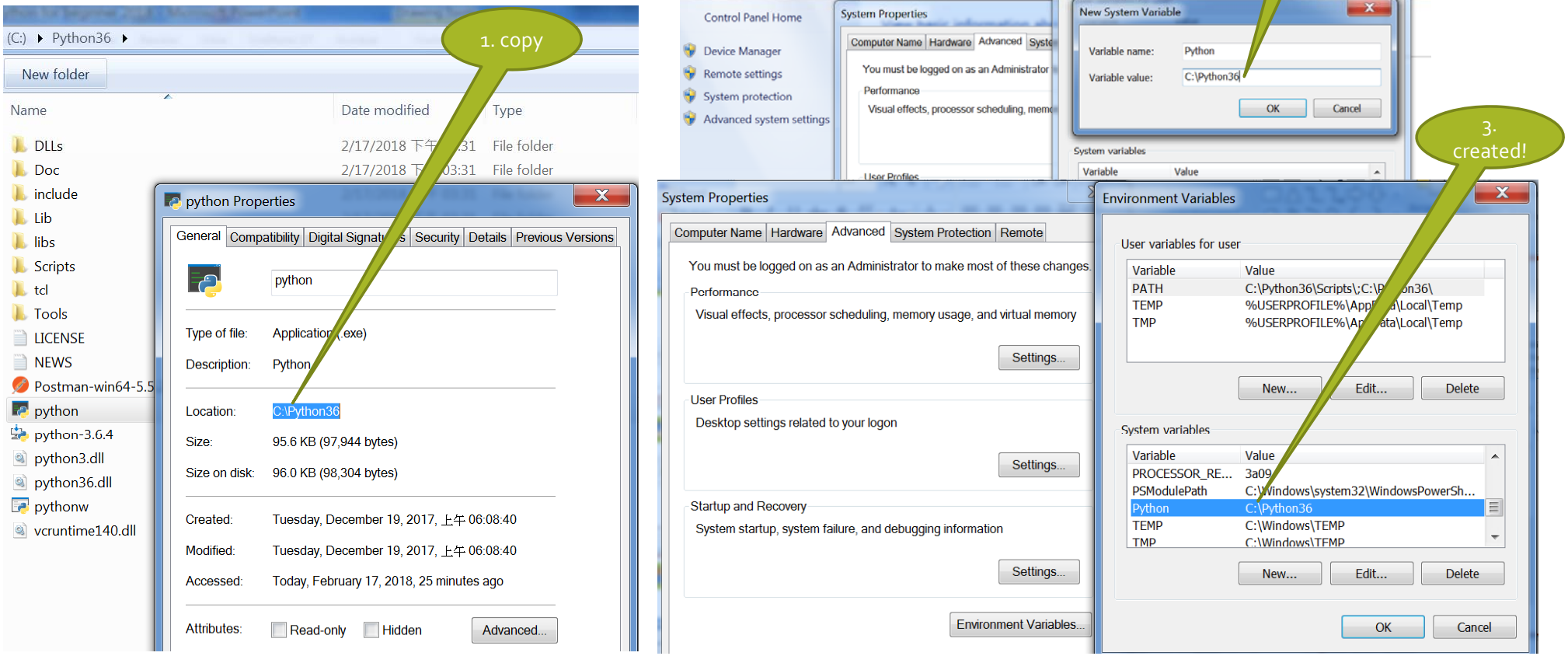
2. Setting environment (2) – my case





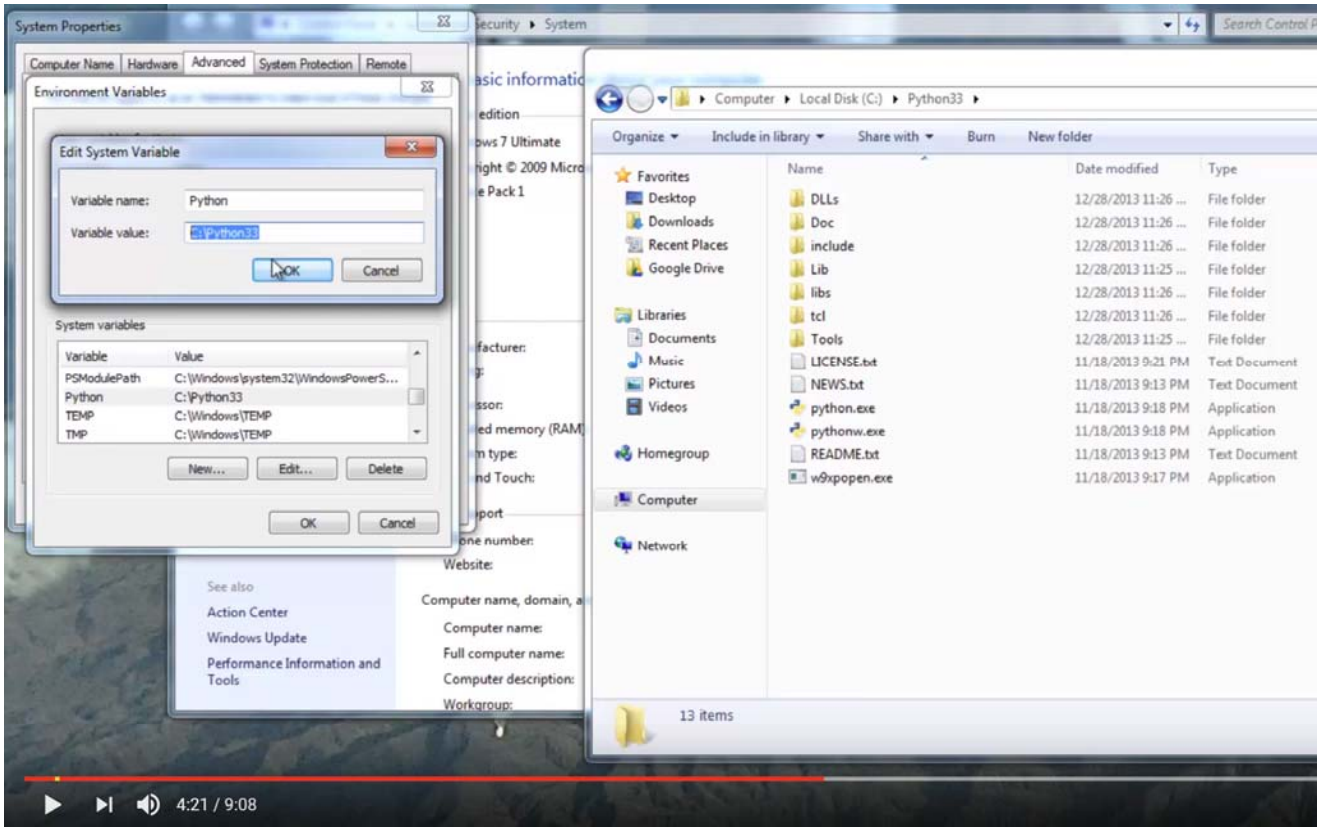
2. Setting environment (3)

Copy the path and paste on “Variable value”

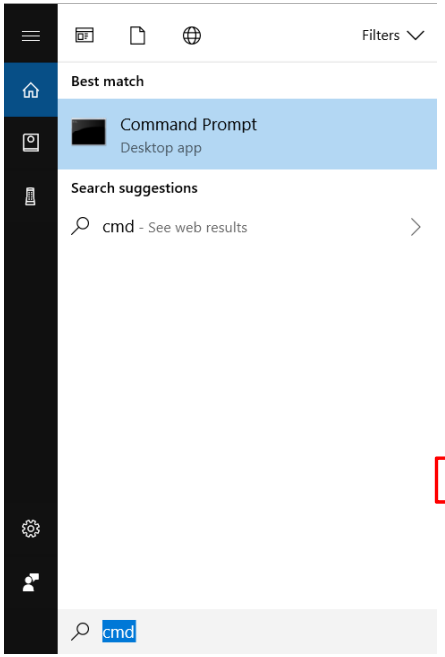


2. Setting environment

(Source URL: https://www.youtube.com/watch?v=cpPGobKHYKc)



3. Type “cmd” in Command Prompt and type “py”



For version 3.6.4, type “**py**” NOT “python”

Command Prompt

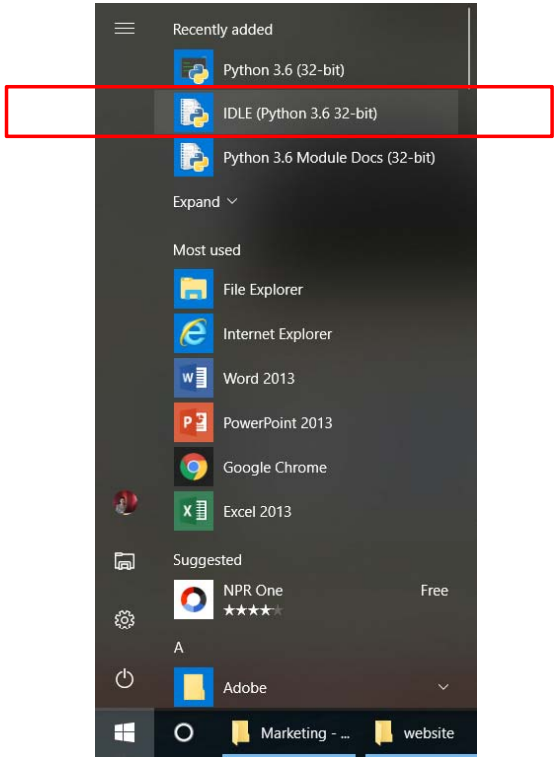
```
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\user>python
'python' is not recognized as an internal or external command,
operable program or batch file.
```

Command Prompt - py

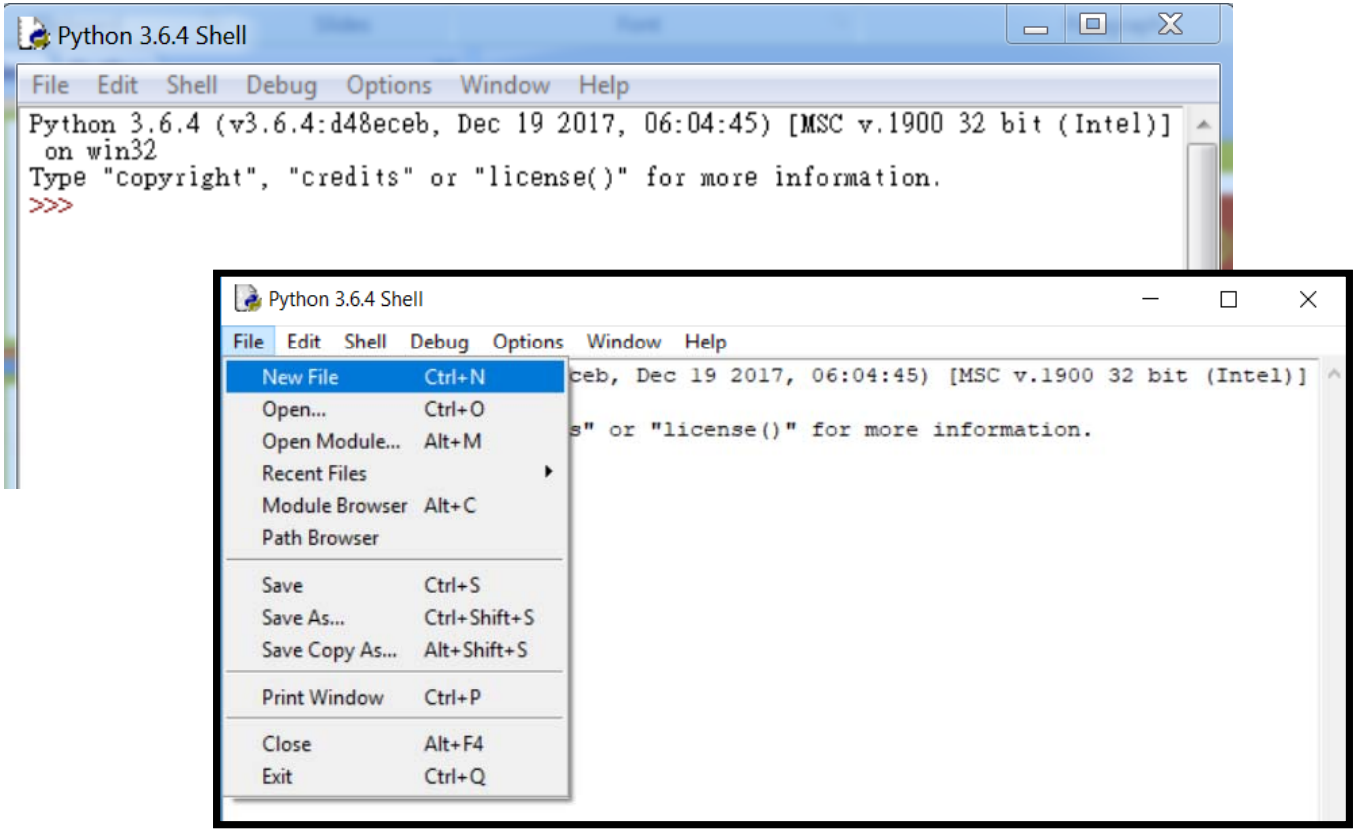
```
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\user>python
'python' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\user>py
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello World')
Hello World
>>>
```

4. Hello world program
Step 1 – Create a python program file (1)

(1) Open "IDLE" from the Start Menu

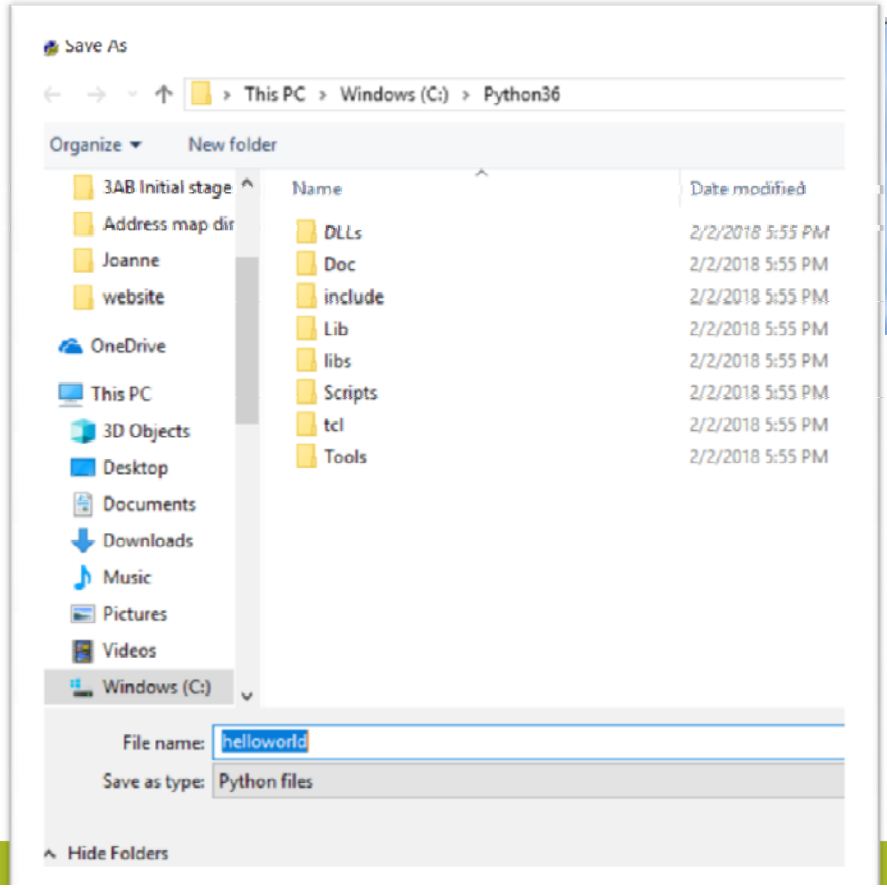


(2) From the "Python Shell, click File → "New File"

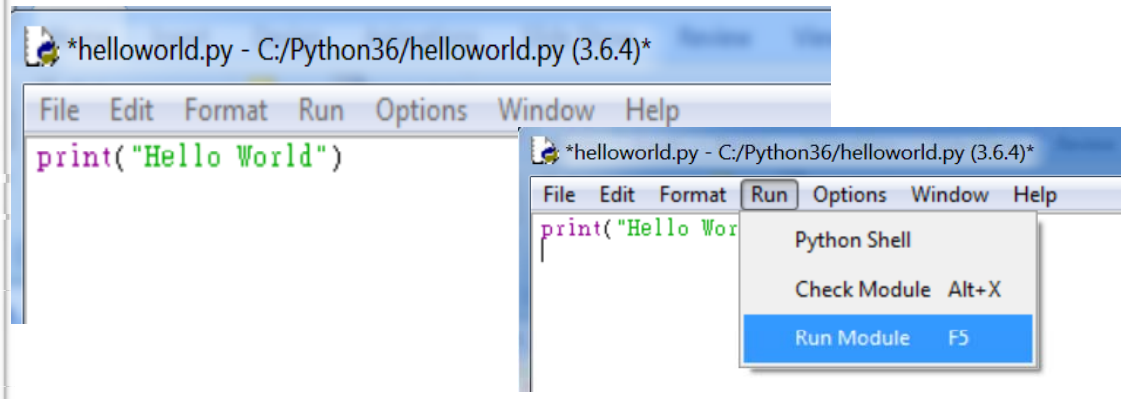


4. Hello world program
Step 1 – Create a python program file (2)

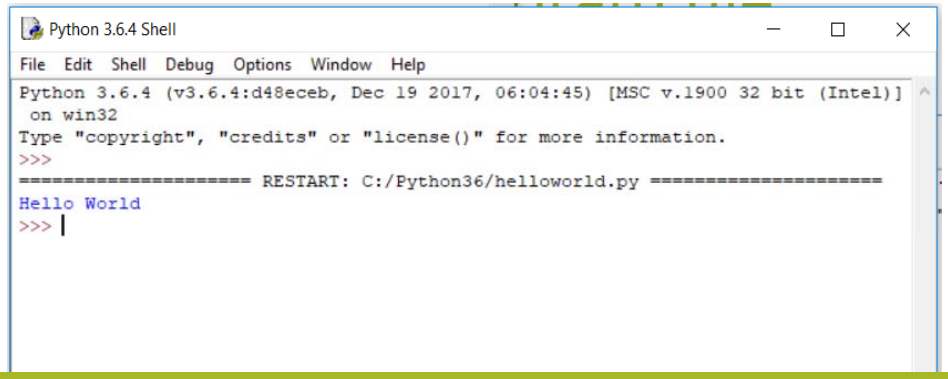
(3) Save the python file as "***.py"



(4) Type the coding in the NEW-File → select "Run" → "Run Module"



(5) Result will be shown in the "**Python Shell**"

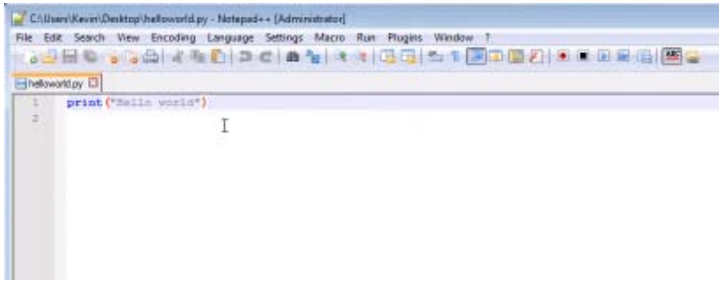


4. Hello world program

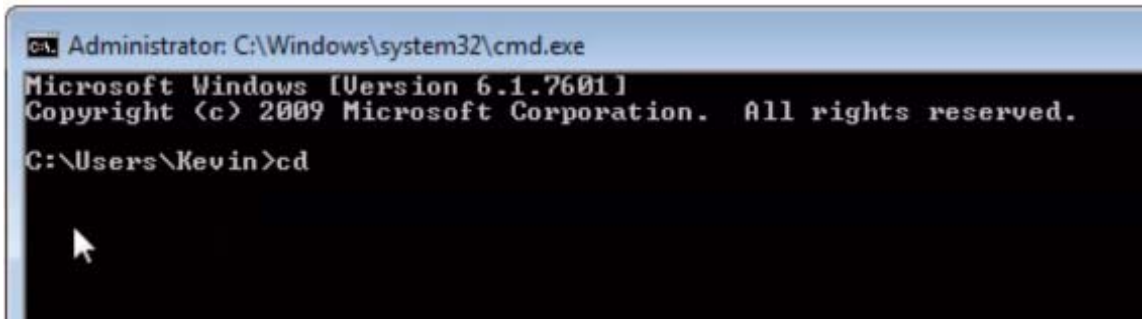
(URL: https://www.youtube.com/watch?v=hFhiV5X5QM4)

Step 2 – Determine the path of a python program file

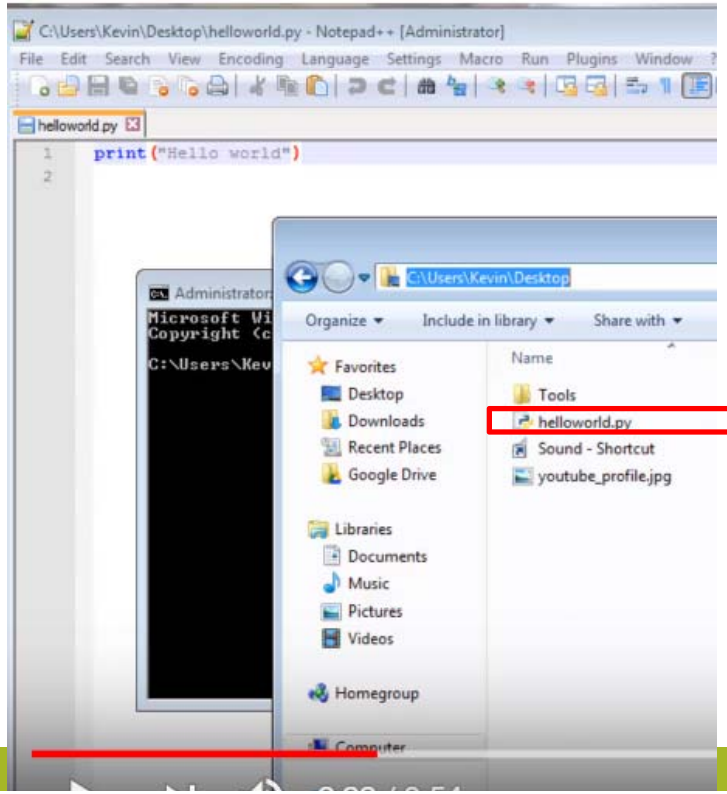
It is better type code & edit in "Notepad ++"



Call out the "command prompt" n type cmd



Type "cmd" to change to the directory where you save python program file….
Copy the file directory PATH

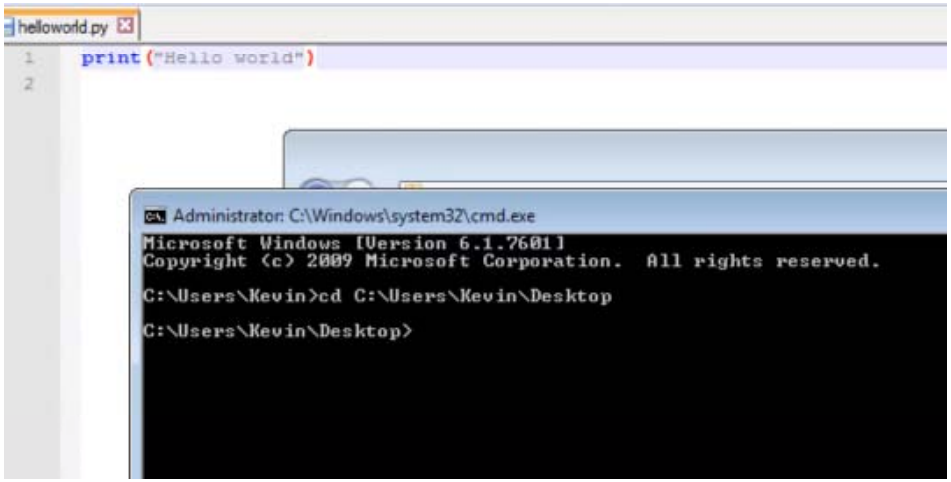


4. Hello world program

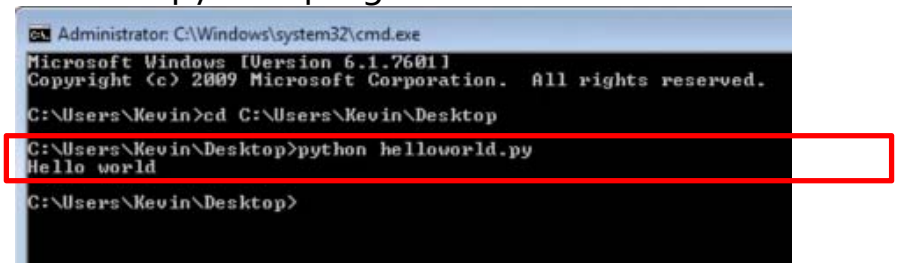
(Adapted from URL: https://www.youtube.com/watch?v=hFhiV5X5QM4)

Step 3 – Run a python program file in cmd.exe

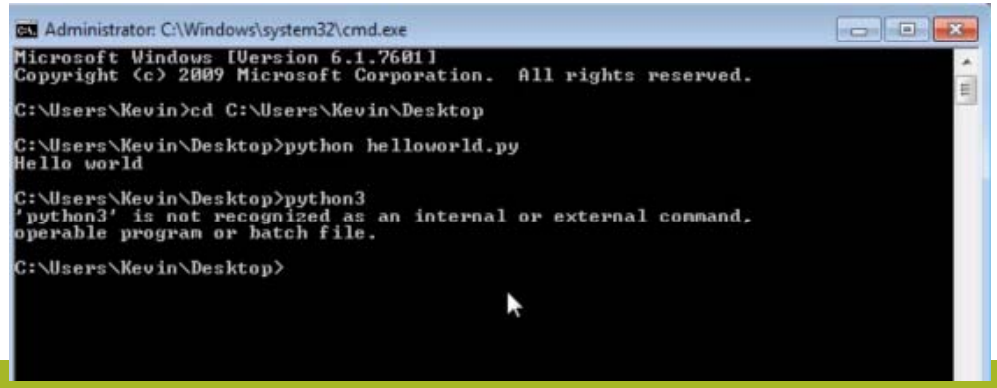
Copy and paste to the cmd.exe



Run the python program



If error like this, then go back to the "variable environment setting" OR make sure that you type "**py**" instead of "python" **for Python 3.6.4**

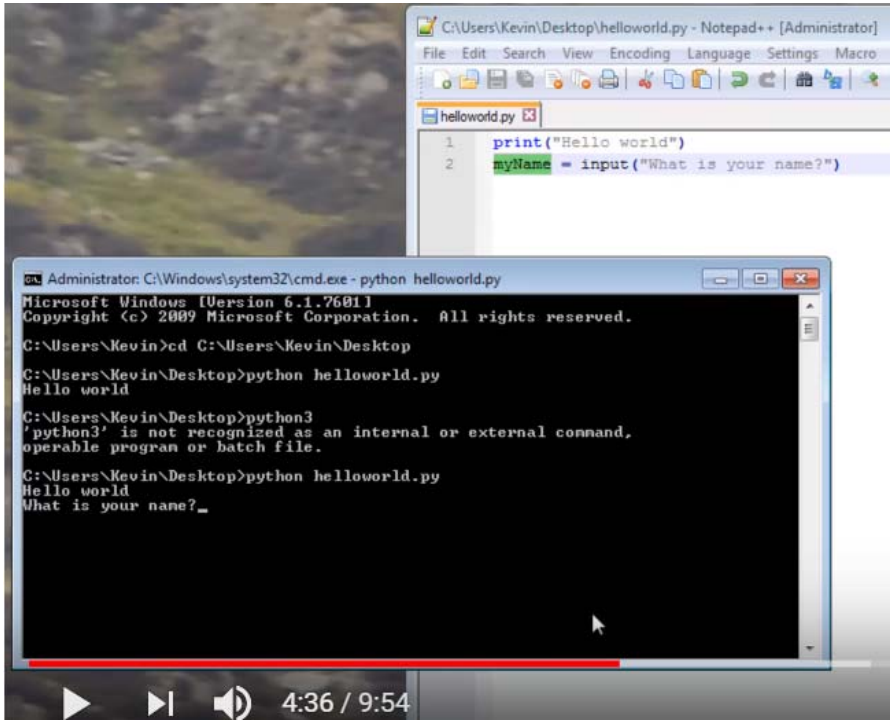


4. Hello world program – Adding Inputs

(Adapted from URL: https://www.youtube.com/watch?v=hFhiV5X5QM4)

Step 4 – Add inputs in a python program file

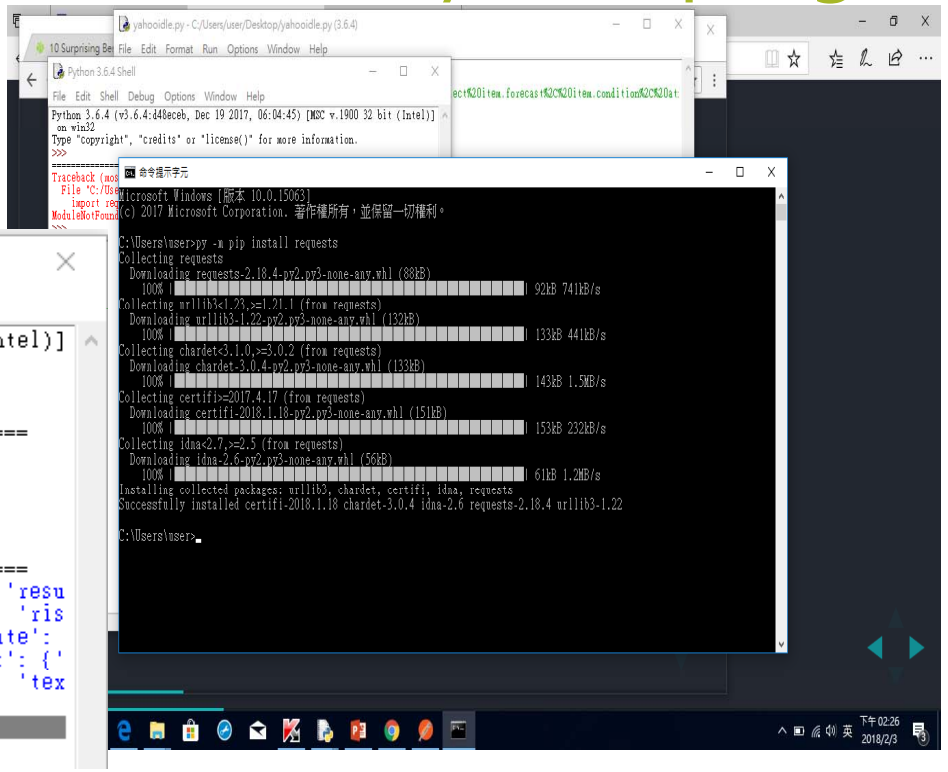
Adding "input"



Determine the content to crawl: Open a webpage in Chrome → right click and select "check (N)" --- to see its HTML code

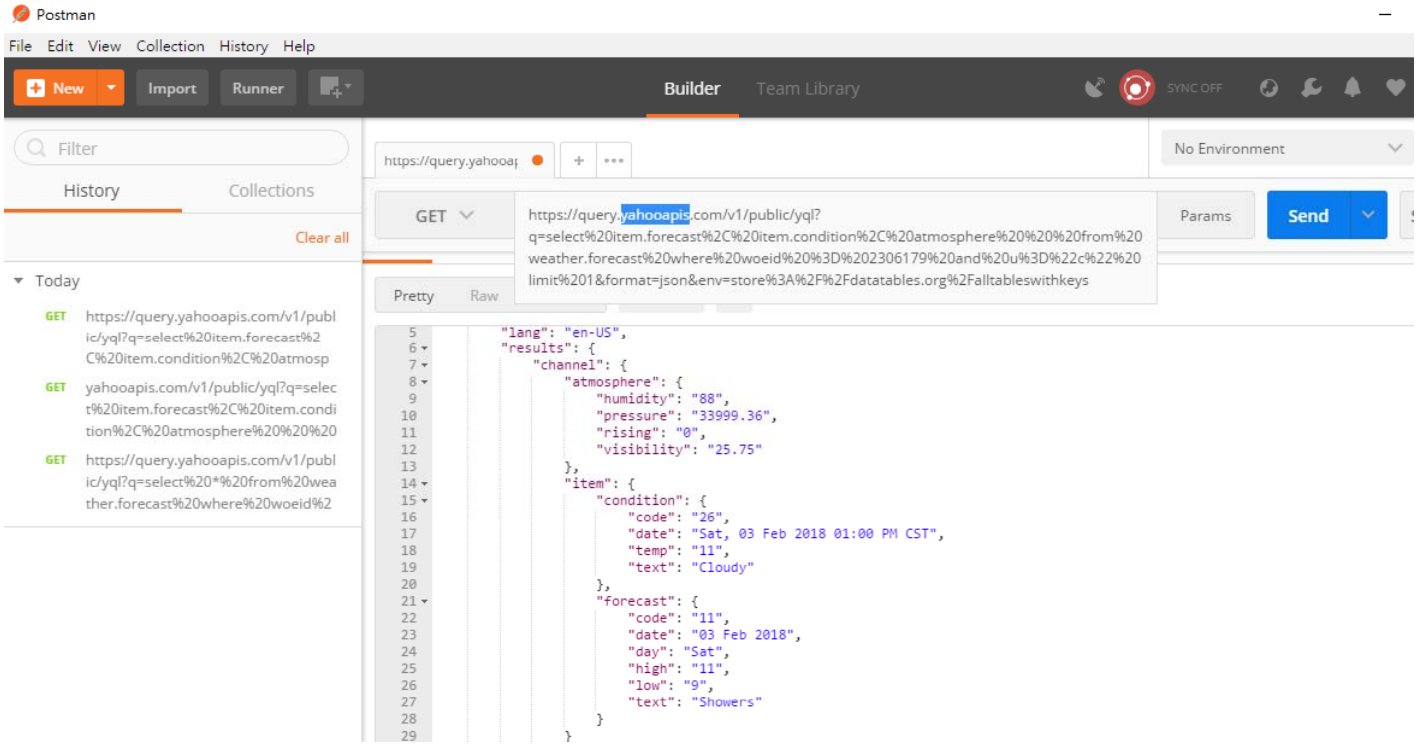
Install "requests-module" in the cmd in order to call the "requests" module in Python program

(1) Install "requests-module" in the cmd
(2) In the cmd type: py –m pip install requests

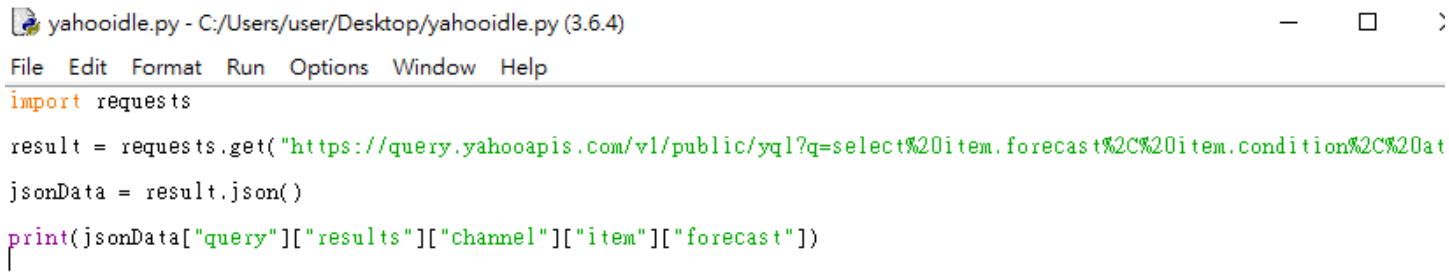


Run the Yahoo Weather API program in Python

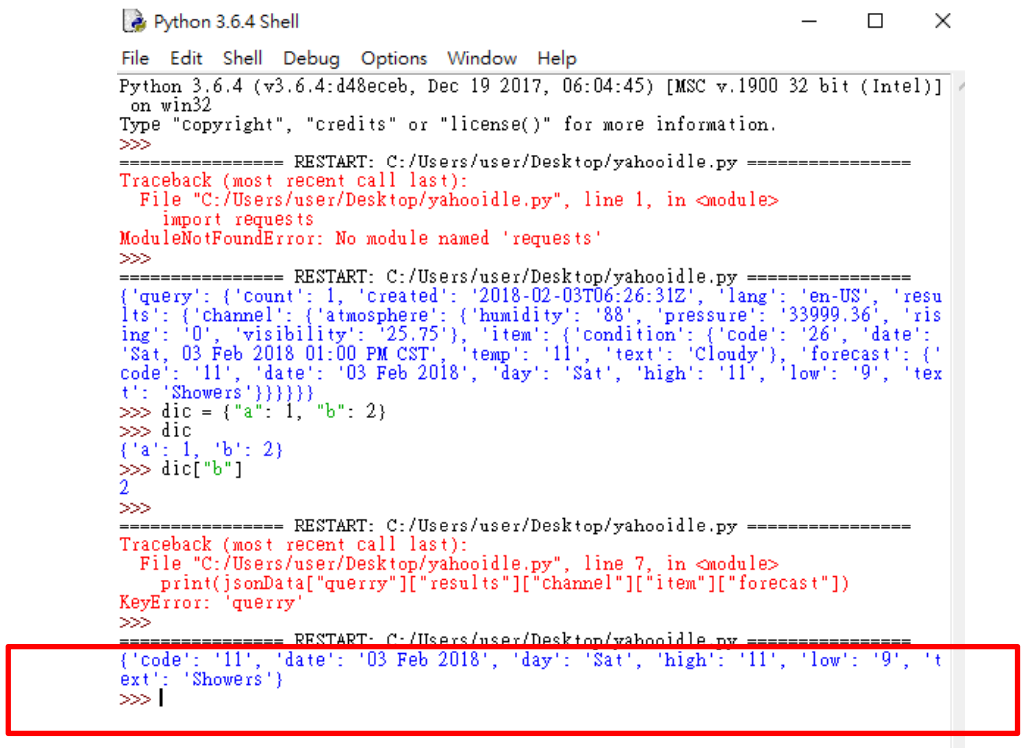
Run the Yahoo Weather API program in Postman



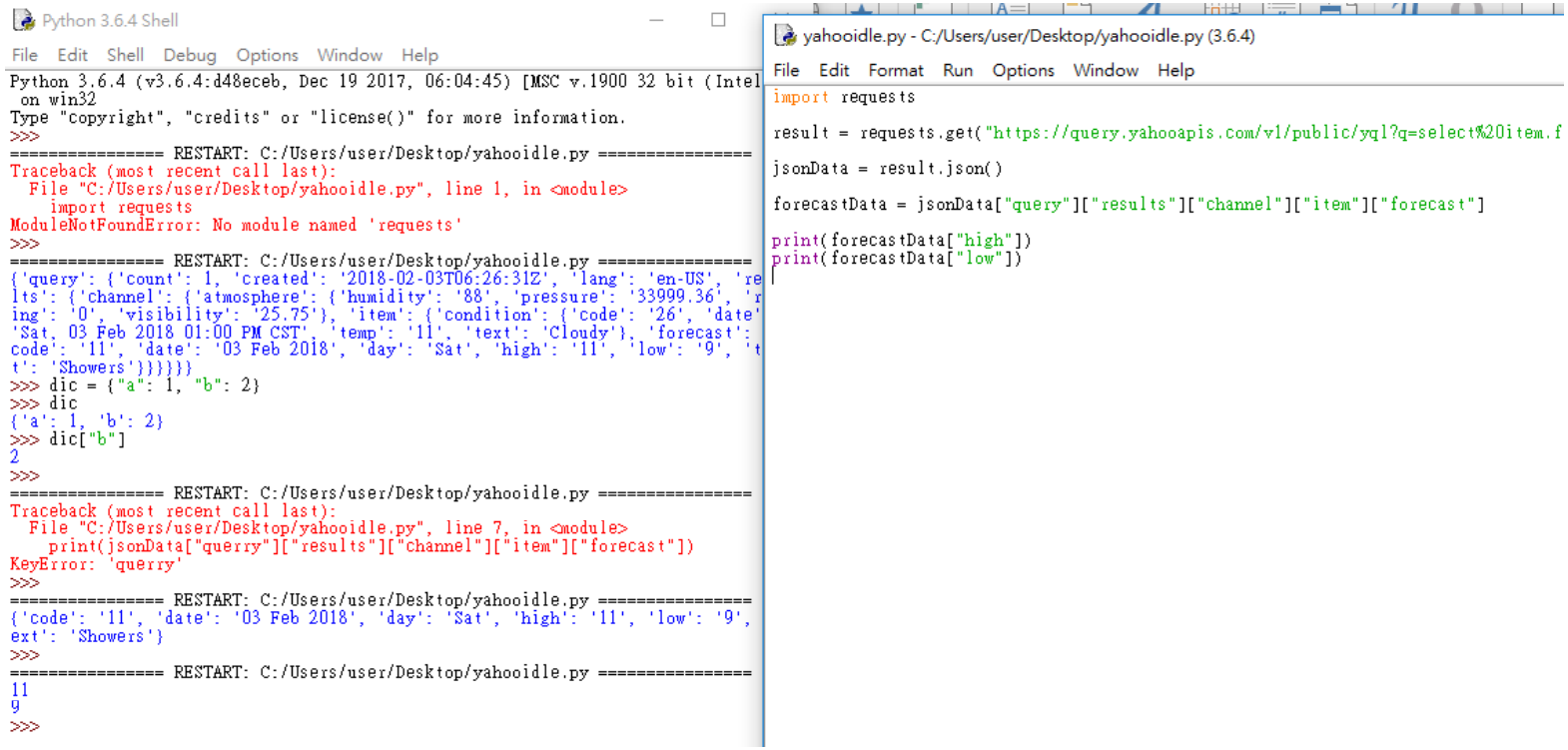
Type more code in IDLE (e.g. yahooidle.py) and press "Run"



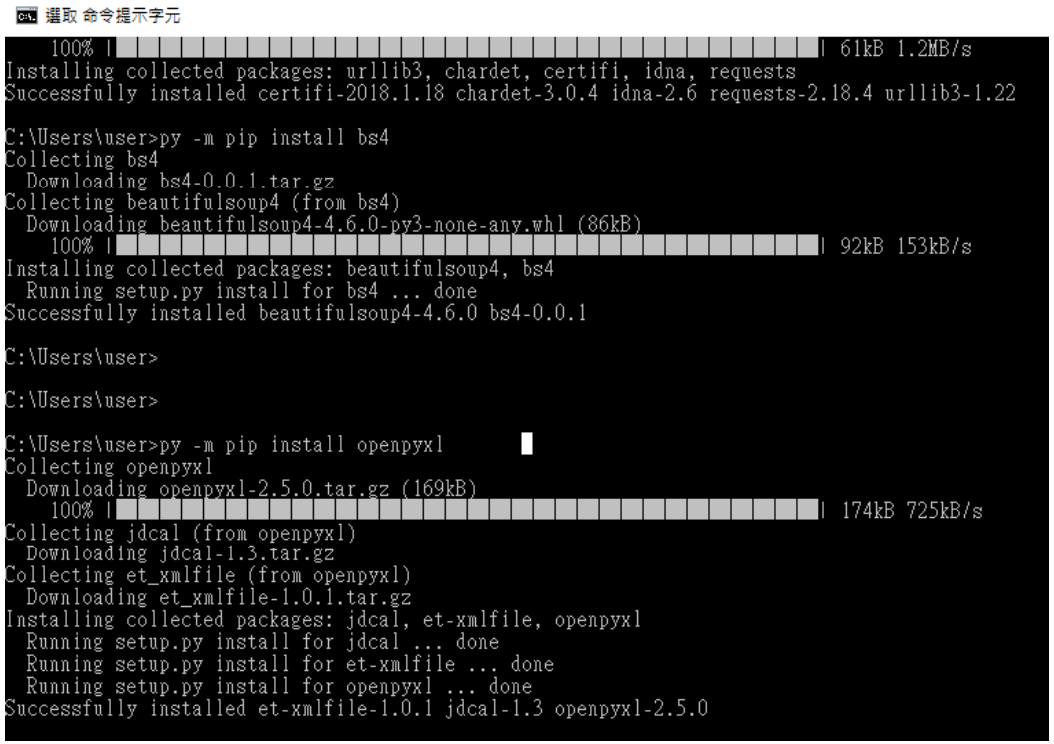
Result is shown in Python Shell



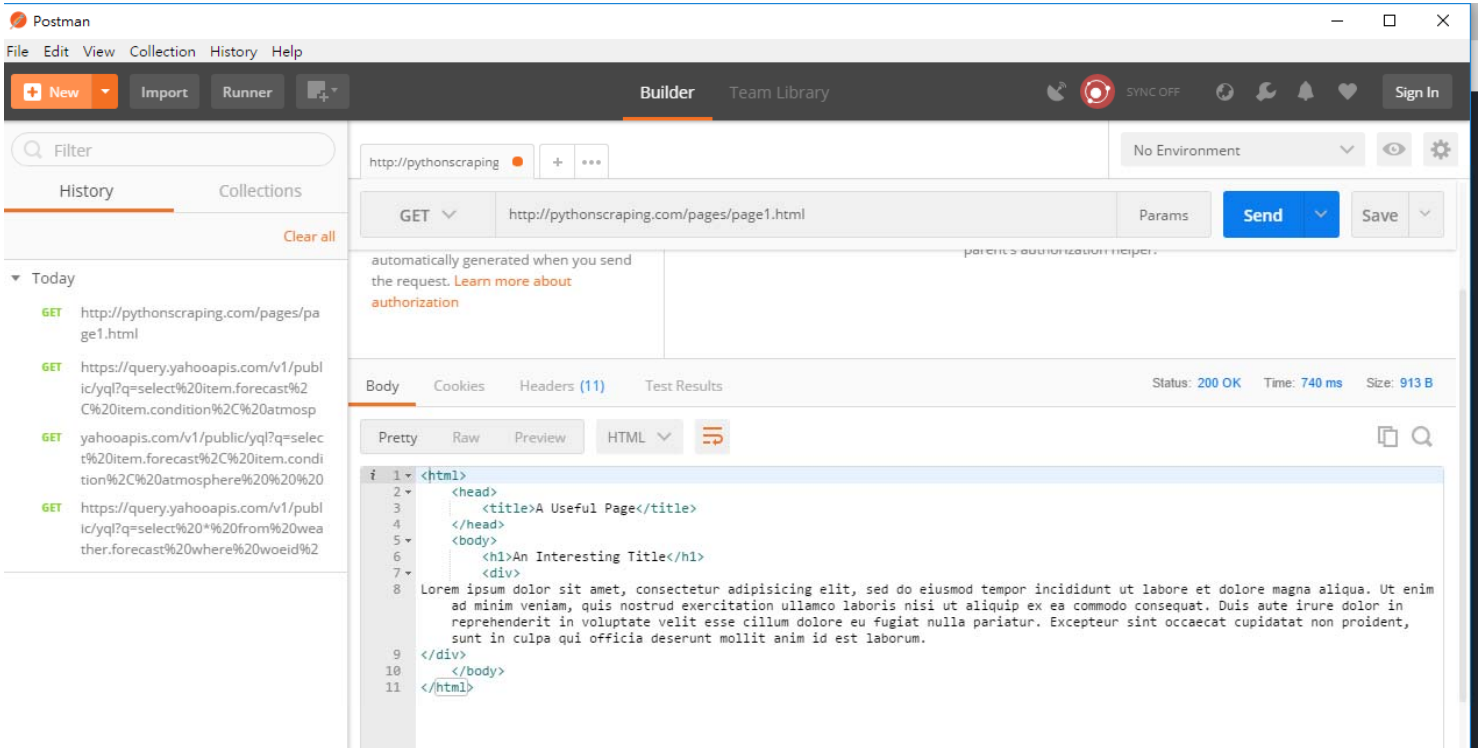
Simplify result in Shell



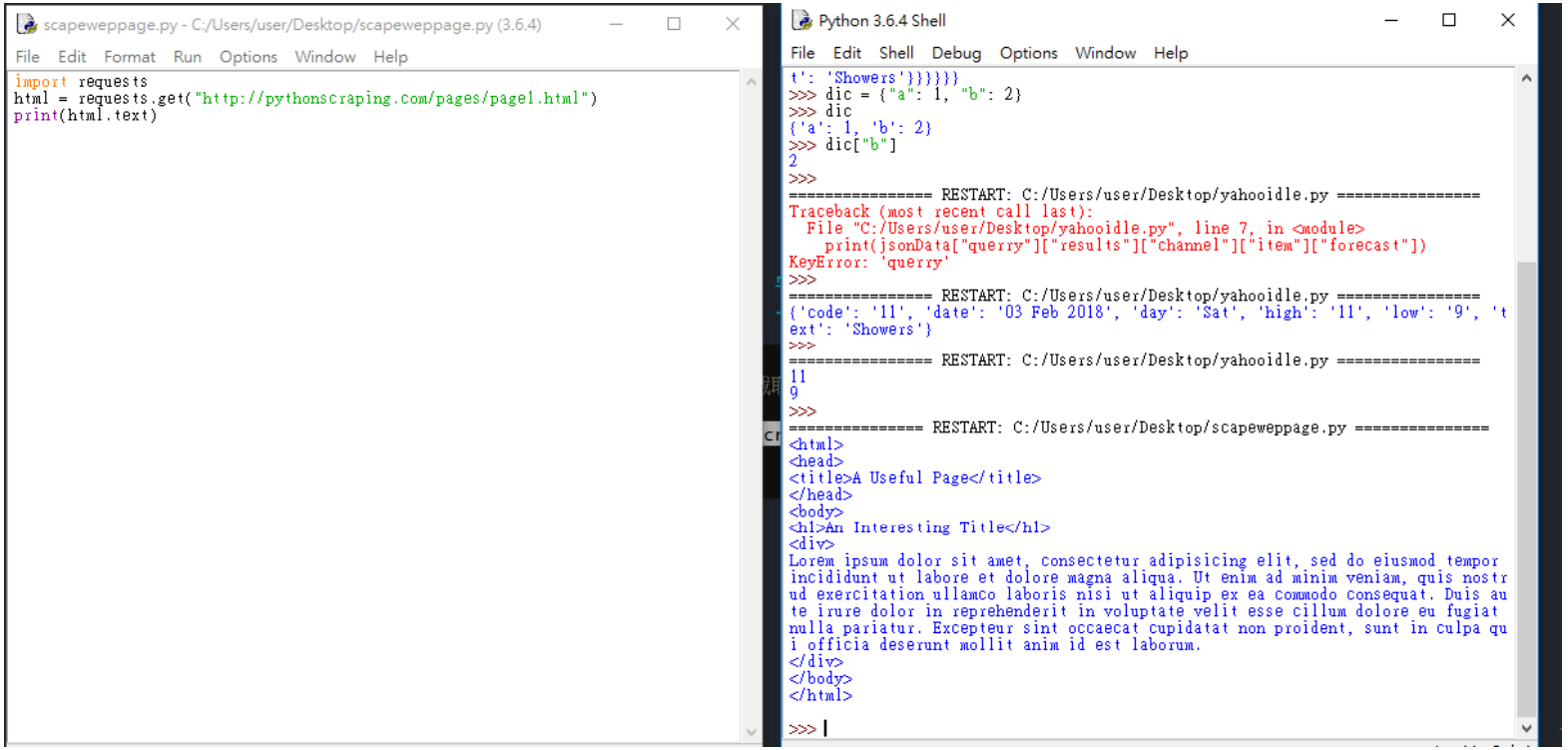
Install “BeautifulSoup” and “Openpyxl” in cmd



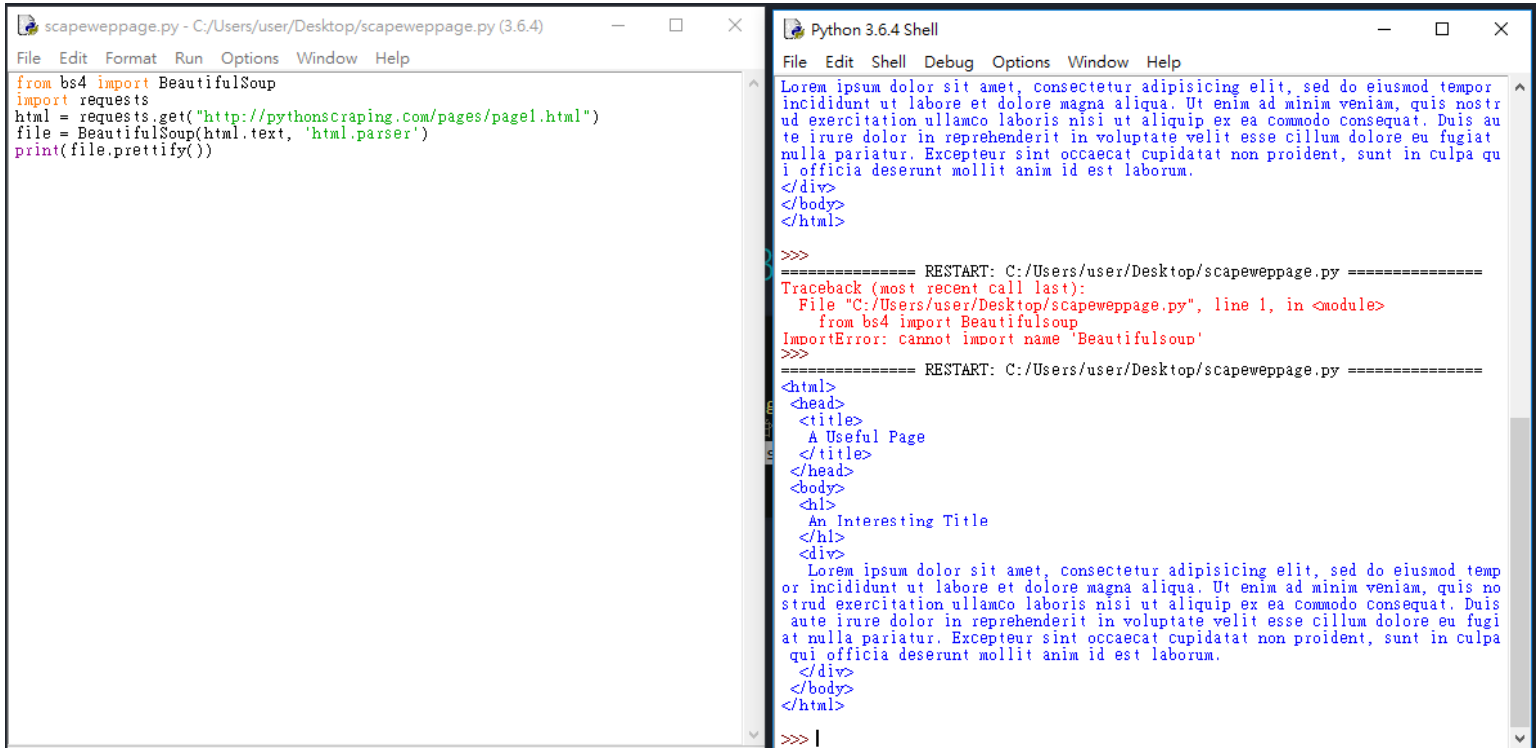
Crawling a webpage – using Postman



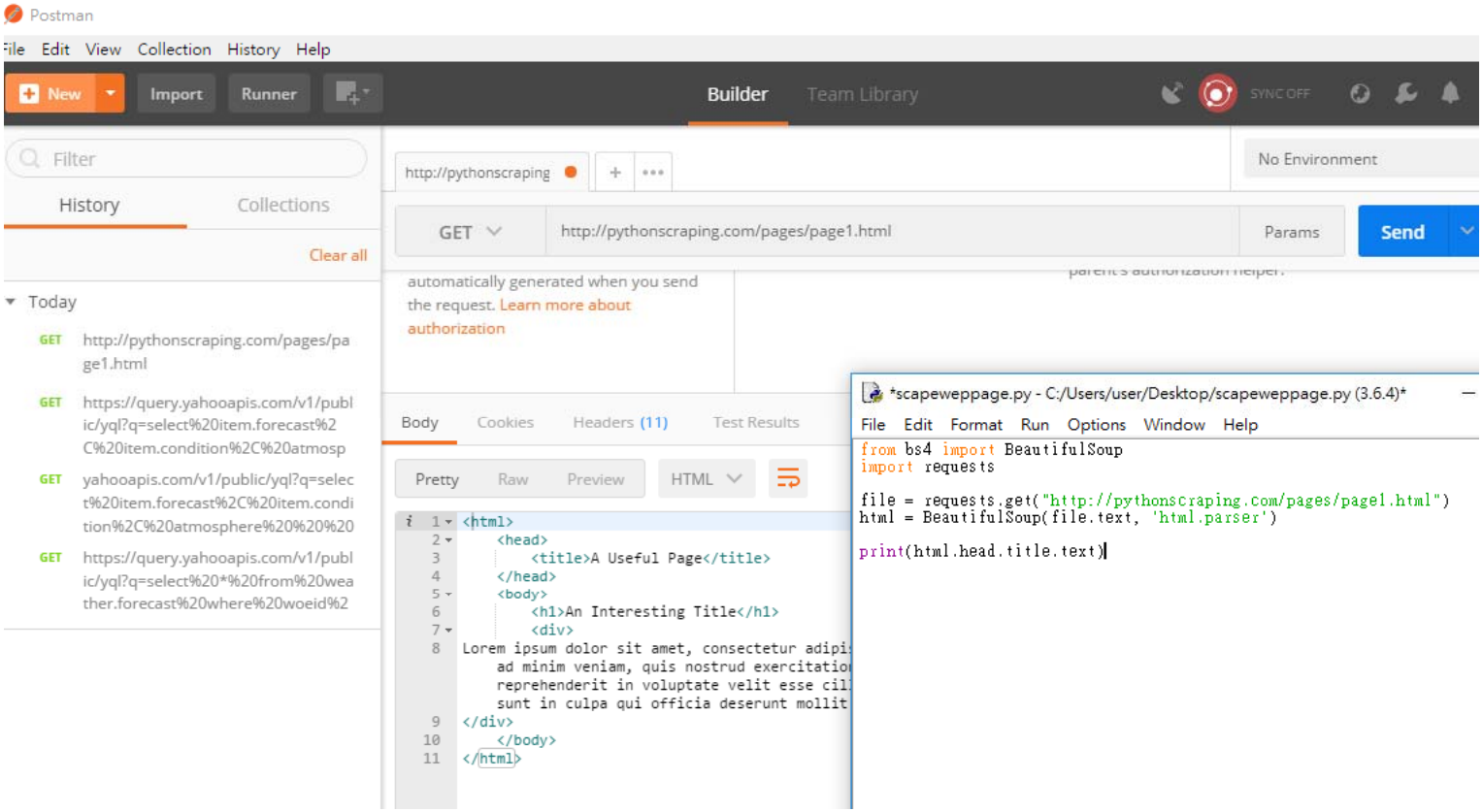
Crawling a webpage – using Python



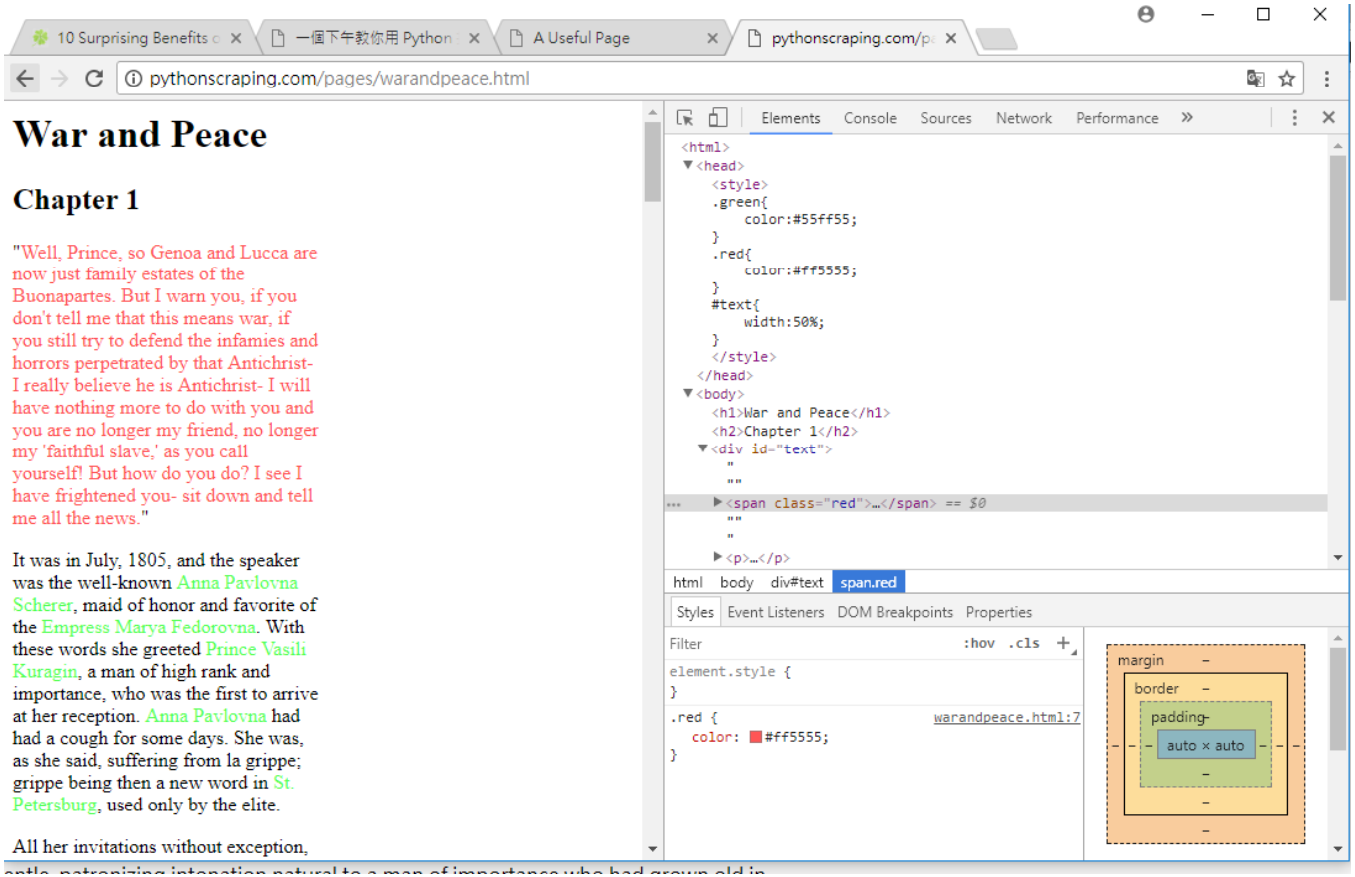
Beautifying output using "BeautifulSoup – in Python Program



Extract text – using Python

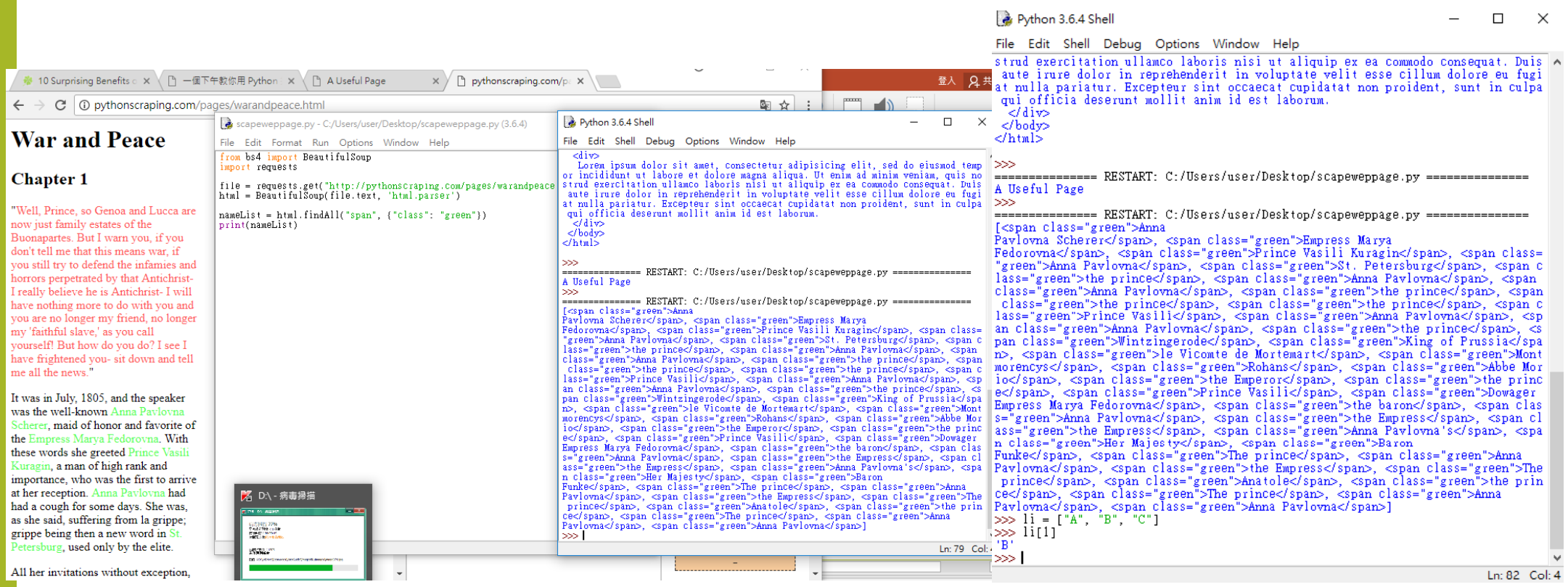


Css

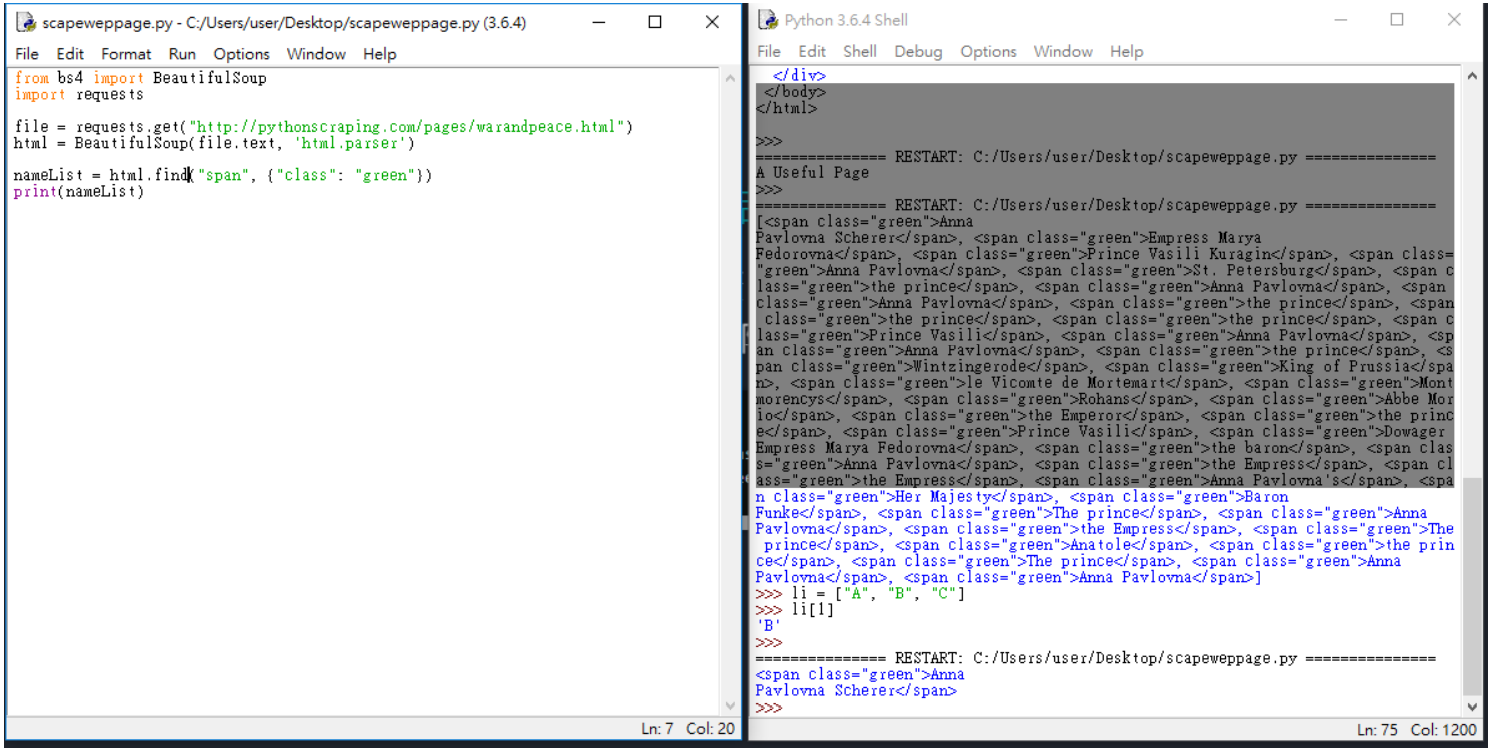

10 Surprising Benefits
一個下午教你用 Python
A Useful Page
pythonscraping.com/
pythonscraping.com/pages/warandpeace.html
War and Peace
Chapter 1
"Well, Prince, so Genoa and Lucca are now just family estates of the Buonapartes. But I warn you, if you don't tell me that this means war, if you still try to defend the infamies and horrors perpetrated by that Antichrist- I really believe he is Antichrist- I will have nothing more to do with you and you are no longer my friend, no longer my 'faithful slave,' as you call yourself! But how do you do? I see I have frightened you- sit down and tell me all the news."
It was in July, 1805, and the speaker was the well-known Anna Pavlovna Scherer, maid of honor and favorite of the Empress Marya Fedorovna. With these words she greeted Prince Vasili Kuragin, a man of high rank and importance, who was the first to arrive at her reception. Anna Pavlovna had had a cough for some days. She was, as she said, suffering from la grippe; grippe being then a new word in St. Petersburg, used only by the elite.
All her invitations without exception,
Elements
Console
Sources
Network
Performance
<html>
<head>
<style>
.green{
color:#55ff55;
}
.red{
color:#ff5555;
}
#text{
width:50%;
}
</style>
</head>
<body>
<h1>War and Peace</h1>
<h2>Chapter 1</h2>

"
""
… == $0
""
"
…
html
body
div#text
span.red
Styles
Event Listeners
DOM Breakpoints
Properties
Filter
:hov
.cls
element.style {
}
.red {
color: #ff5555;
}
warandpeace.html:7
margin
border
padding
auto × auto

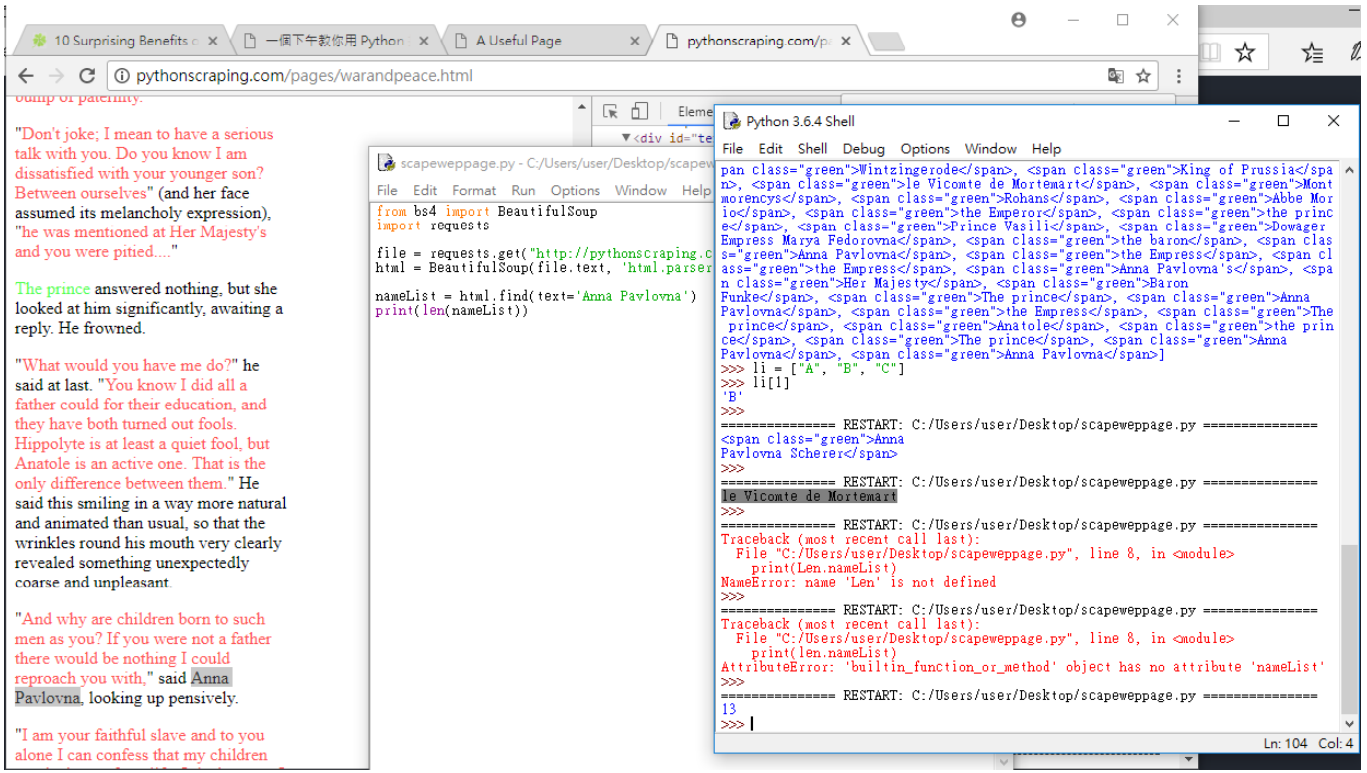
Scrape green text only & array



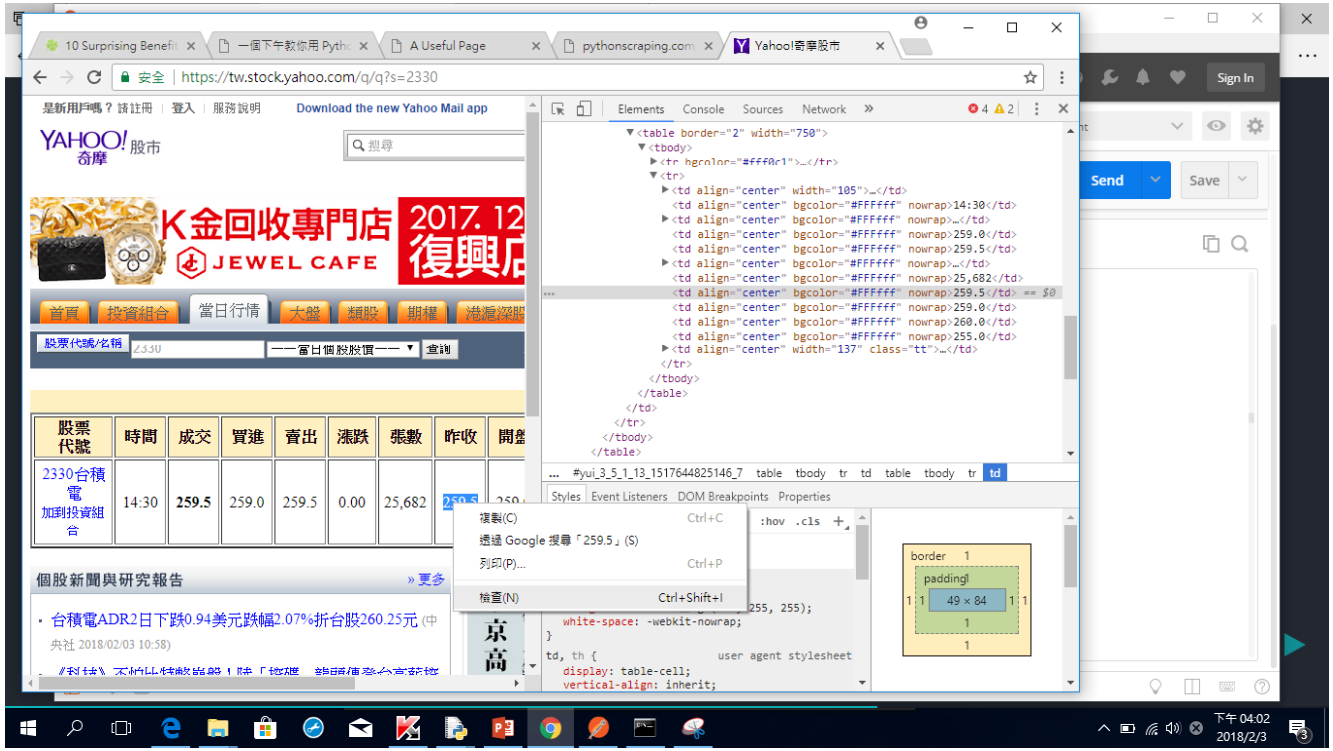
Get 1st set of data & last (-1)



Len of a phrase (number of occurrence)

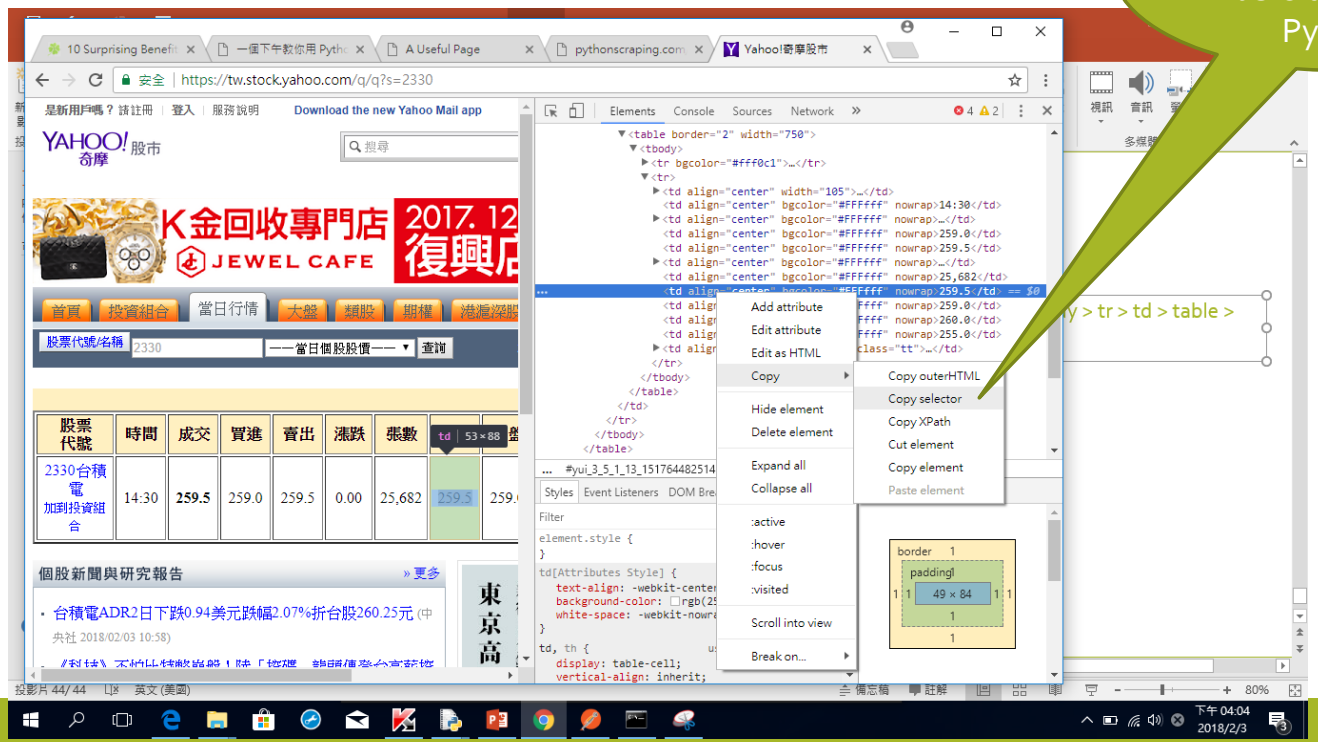


Stock closing price (1)



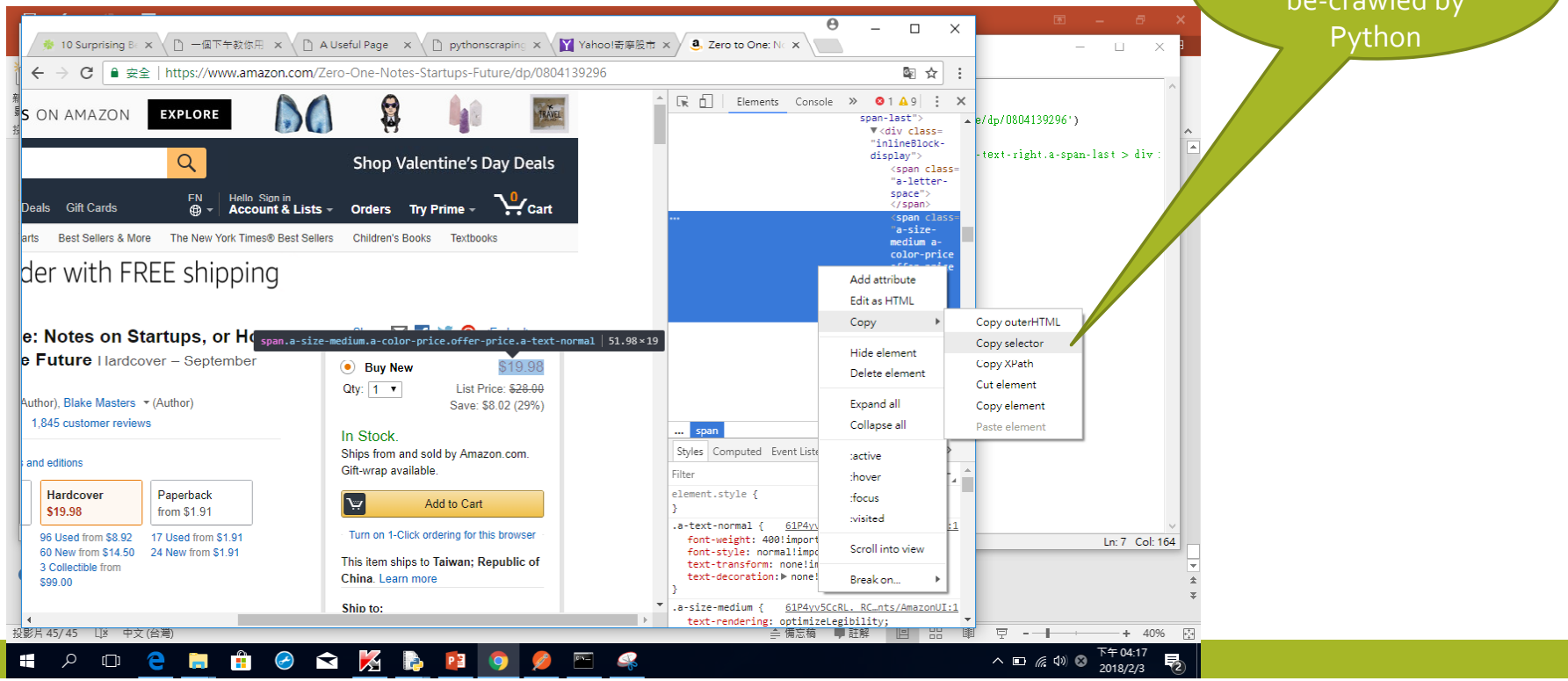
Stock closing price (2)

- #yui_3_5_1_13_1517644825146_7 > table:nth-child(22) > tbody > tr > td > table > tbody > tr:nth-child(2) > td:nth-child(8)

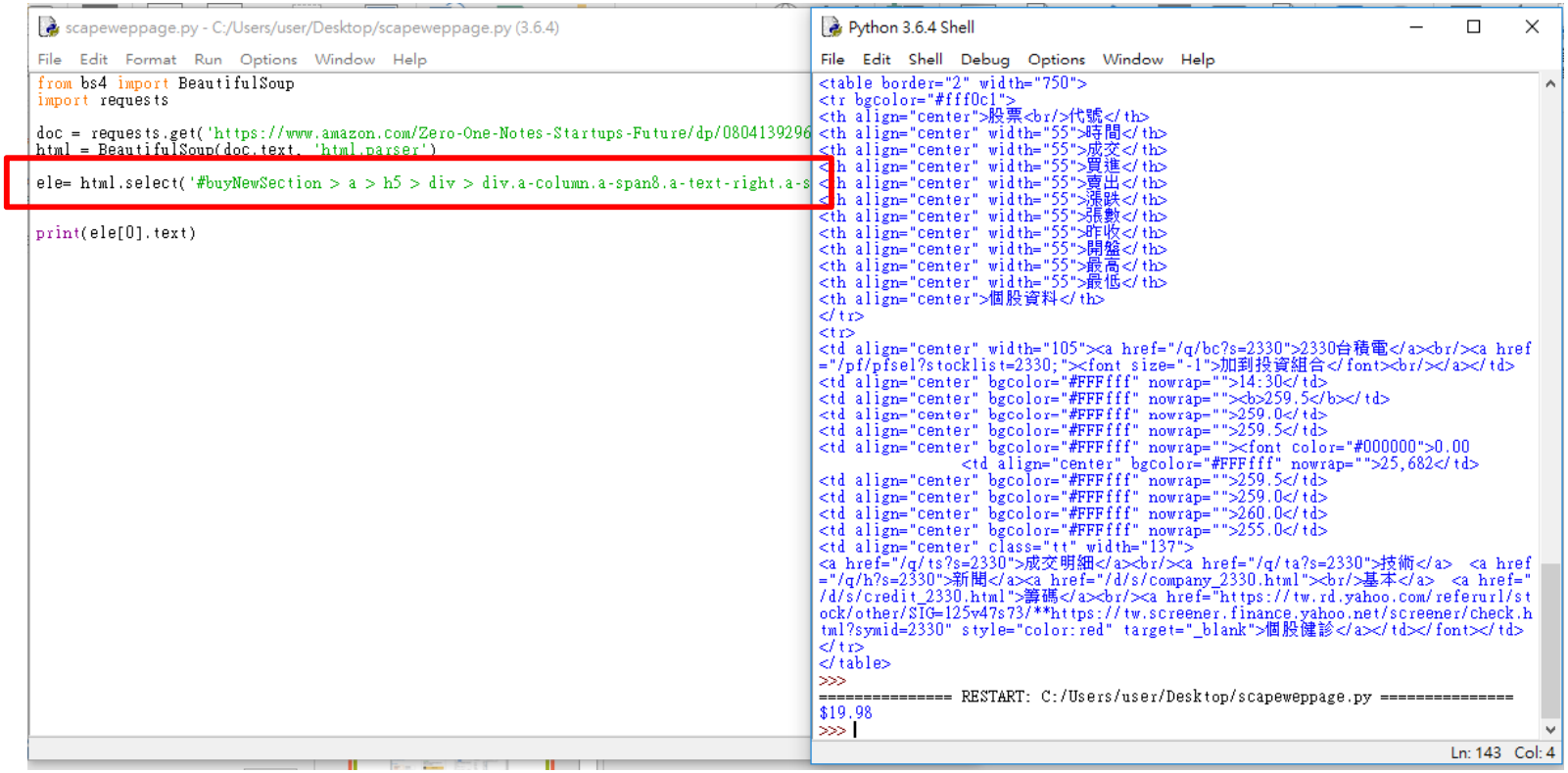


Book price (1)

- #buyNewSection > a > h5 > div > div.a-column.a-span8.a-text-right.a-span-last > div > span.a-size-medium.a-color-price.offer-price.a-text-normal



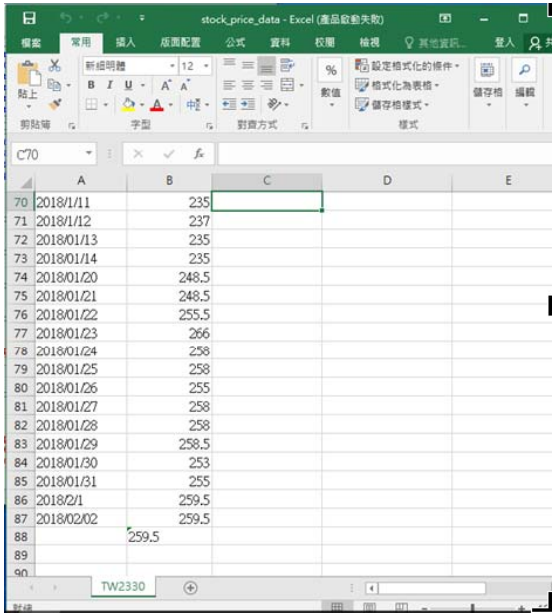
Book price (2)


scapeweppage.py - C:/Users/user/Desktop/scapeweppage.py (3.6.4)
File Edit Format Run Options Window Help
from bs4 import BeautifulSoup
import requests
doc = requests.get('https://www.amazon.com/Zero-One-Notes-Startups-Future/dp/0804139296
html = BeautifulSoup(doc.text, 'html.parser')
ele= html.select('#buyNewSection > a > h5 > div > div.a-column.a-span8.a-text-right.a-s
print(ele[0].text)
Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
<table border="2" width="750">
<tr bgcolor="#fff0c1">
<th align="center">股票
代號</th>
<th align="center" width="55">時間</th>
<th align="center" width="55">成交</th>
<th align="center" width="55">買進</th>
<th align="center" width="55">賣出</th>
<th align="center" width="55">漲跌</th>
<th align="center" width="55">張數</th>
<th align="center" width="55">昨收</th>
<th align="center" width="55">開盤</th>
<th align="center" width="55">最高</th>
<th align="center" width="55">最低</th>
<th align="center">個股資料</th>
</tr>
<tr>
<td align="center" width="105">2330台積電
<a href
="/pf/pfsel?stocklist=2330;">加到投資組合
</td>
<td align="center" bgcolor="#FFFfff" nowrap="">14:30</td>
<td align="center" bgcolor="#FFFfff" nowrap="">259.5</td>
<td align="center" bgcolor="#FFFfff" nowrap="">259.0</td>
<td align="center" bgcolor="#FFFfff" nowrap="">259.5</td>
<td align="center" bgcolor="#FFFfff" nowrap="">0.00
<td align="center" bgcolor="#FFFfff" nowrap="">25,682</td>
<td align="center" bgcolor="#FFFfff" nowrap="">259.5</td>
<td align="center" bgcolor="#FFFfff" nowrap="">259.0</td>
<td align="center" bgcolor="#FFFfff" nowrap="">260.0</td>
<td align="center" bgcolor="#FFFfff" nowrap="">255.0</td>
<td align="center" class="tt" width="137">
成交明細
技術 <a href
="/q/h?s=2330">新聞
基本 <a href="
/d/s/credit_2330.html">籌碼
<a href="https://tw.rd.yahoo.com/referurl/st
ock/other/SIG=125v47s73/**https://tw.screener.finance.yahoo.net/screener/check.h
tml?symid=2330" style="color:red" target="_blank">個股健診</td></td>
</tr>
</table>
>>>
========== RESTART: C:/Users/user/Desktop/scapeweppage.py ==========
$19.98
>>>
Ln: 143 Col: 4

Export data to Excel

```
File  Edit  Format  Run  Options  Window  Help
from bs4 import BeautifulSoup

import requests

import openpyxl
import os


doc = requests.get('https://tw.stock.yahoo.com/q/q?s=2330')

html = BeautifulSoup(doc.text, 'html.parser')

table = html.findAll(text='個股資料')[0].parent.parent.parent

dataRow = table.select('tr')[1].select('td')

closingPrice = dataRow[7].text

print("台積電今日收盤價：${}".format(closingPrice))

os.chdir(r"C:\Users\user\Desktop")
workbook = openpyxl.load_workbook('stock_price_data.xlsx')
sheet = workbook.get_sheet_by_name('TW2330')

sheet.cell(row=88, column=2).value = closingPrice

workbook.save('stock_price_data.xlsx')
```

```
Python 3.6.4 Shell
File  Edit  Shell  Debug  Options  Windo
<td align="center" bgcolor="#FFFFFF"
                  <td align="center" bg
<td align="center" bgcolor="#FFFFFF"
<td align="center" bgcolor="#FFFFFF"
<td align="center" bgcolor="#FFFFFF"
<td align="center" bgcolor="#FFFFFF"
<td align="center" class="tt" width="
<a href="/q/ts?s=2330">成交明細</a><b
="/q/h?s=2330">新聞</a><a href="/d/s/
/d/s/credit_2330.html">籌碼</a><br/><
ock/other/SIG=125w47s73/**https://tw.
tml?symid=2330" style="color:red" tar
</tr>
</table>
>>>
========== RESTART: C:/Users/use
$19.98
>>>
========== RESTART: C:/Users/use
台積電今日收盤價：$259.0
>>>
========== RESTART: C:/Users/use
台積電今日收盤價：$259.5
>>>
========== RESTART: C:/Users/use
台積電今日收盤價：$259.5
Traceback (most recent call last):
  File "C:/Users/user/Desktop/scapewe
    os.chdir(r"C:\Users\user\Desktop"
NameError: name 'os' is not defined
>>>
========== RESTART: C:/Users/use
台積電今日收盤價：$259.5

Warning (from warnings module):
  File "C:/Users/user/Desktop/scapewe
    sheet = workbook.get_sheet_by_nam
DeprecationWarning: Call to deprecate
ame]).
>>>
```



	A	B	C	D	E
70	2018/1/11	235			
71	2018/1/12	237			
72	2018/01/13	235			
73	2018/01/14	235			
74	2018/01/20	248.5			
75	2018/01/21	248.5			
76	2018/01/22	255.5			
77	2018/01/23	266			
78	2018/01/24	258			
79	2018/01/25	258			
80	2018/01/26	255			
81	2018/01/27	258			
82	2018/01/28	258			
83	2018/01/29	258.5			
84	2018/01/30	253			
85	2018/01/31	255			
86	2018/2/1	259.5			
87	2018/02/02	259.5			
88		259.5			
89					
90					

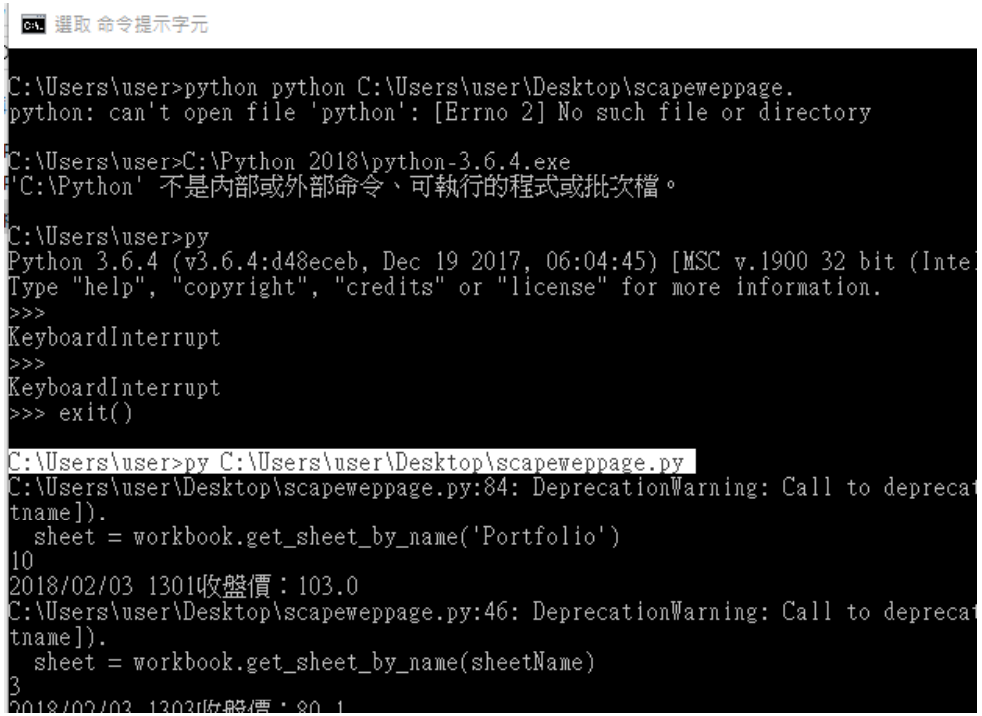
Input data to excel --- for all stock

```
import time
import os
import openpyxl

def getStockPriceById(stockID, workbook):
   url = 'https://tw.stock.yahoo.com/q/q?s={}'.format(stockID)
   doc = requests.get(url)
   html = BeautifulSoup(doc.text, 'html.parser')
   table = html.findAll(text='個股資料')[0].parent.parent.parent
   dataRow = table.select('tr')[1].select('td')
   date = time.strftime("%Y/%m/%d")
closingPrice = dataRow[7].text

   print("{} {}收盤價：{}".format(date, stockID, closingPrice))

   # 找出試算表名稱
   sheetName = "TW{}".format(stockID)
   sheet = workbook.get_sheet_by_name(sheetName)
   # 找出目前該試算表有多少筆資料
   numOfRows = len(sheet['A'])
   print(numOfRows)

# 將新日期與收盤價印在下一列
   sheet.cell(row=numOfRows+1, column=1).value = date
   sheet.cell(row=numOfRows+1, column=2).value = float(closingPrice)

# 主程式從這裏開始
stocks = [2330, 2311]
os.chdir(r"C:\Users\user\Desktop")
workbook = openpyxl.load_workbook('stock_price_data.xlsx')

# 用回圈把每一個 stockID 都跑一次
for stockID in stocks:
   getStockPriceById(stockID, workbook)
workbook.save("stock_price_data.xlsx")
```

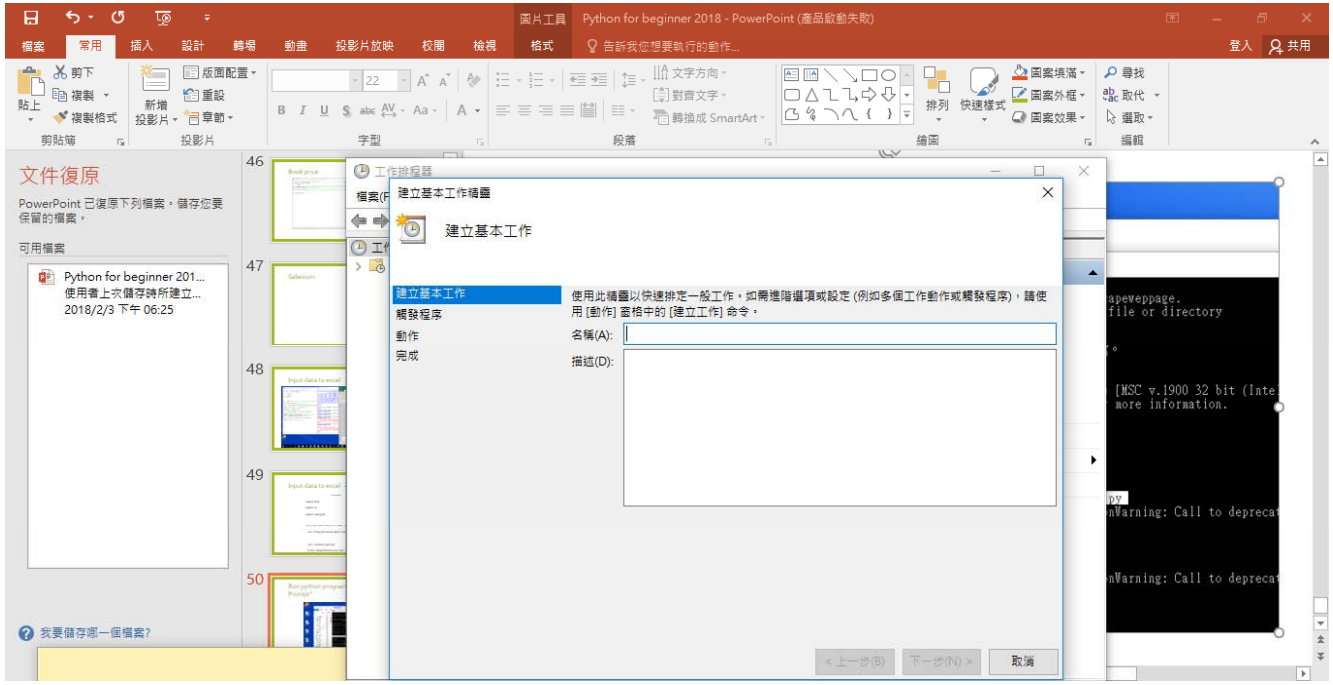
Source, **URL:** https://gist.github.com/yuyueugene84/80bb61ebb3762c97d642ecb935e3ca84

```
from bs4 import BeautifulSoup
import requests
import time
import os
import openpyxl

def getStockPriceById(stockID, workbook):

url = 'https://tw.stock.yahoo.com/q/q?s={}'.format(stockID)


doc = requests.get(url)
html = BeautifulSoup(doc.text, 'html.parser')

table = html.findAll(text='個股資料')[0].parent.parent.parent

dataRow = table.select('tr')[1].select('td')
date = time.strftime("%Y/%m/%d")
closingPrice = dataRow[7].text


print("{} {}收盤價：{}".format(date, stockID, closingPrice))


# 找出試算表名稱
sheetName = "TW{}".format(stockID)

sheet = workbook.get_sheet_by_name(sheetName)

# 找出目前該試算表有多少筆資料
numOfRows = len(sheet['A'])
print(numOfRows)
# 將新日期與收盤價印在下一列

sheet.cell(row=numOfRows+1, column=1).value = date

sheet.cell(row=numOfRows+1, column=2).value = float(closingPrice)


# 主程式從這裏開始
stocks = [2330, 2311]


os.chdir(r"你的Excel檔案所在目錄")

workbook = openpyxl.load_workbook('stock_price_data.xlsx')

# 用回圈把每一個 stockID 都跑一次
for stockID in stocks:
getStockPriceById(stockID, workbook)
```

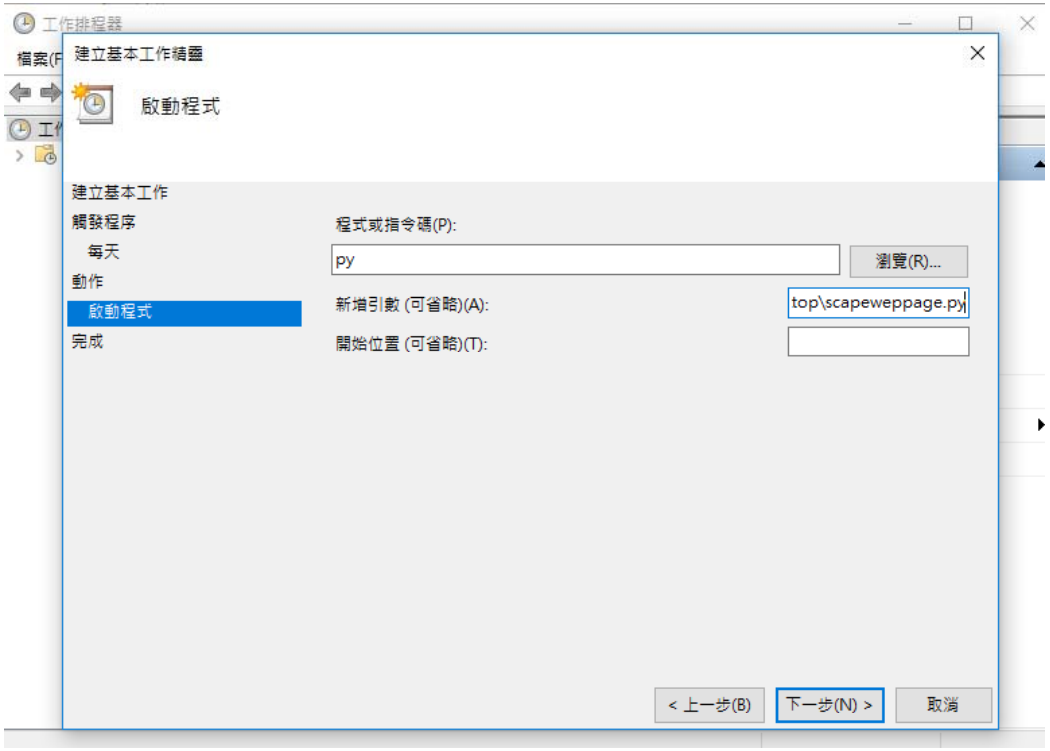
Run Python program from “Command Prompt”



Scheduling (1)

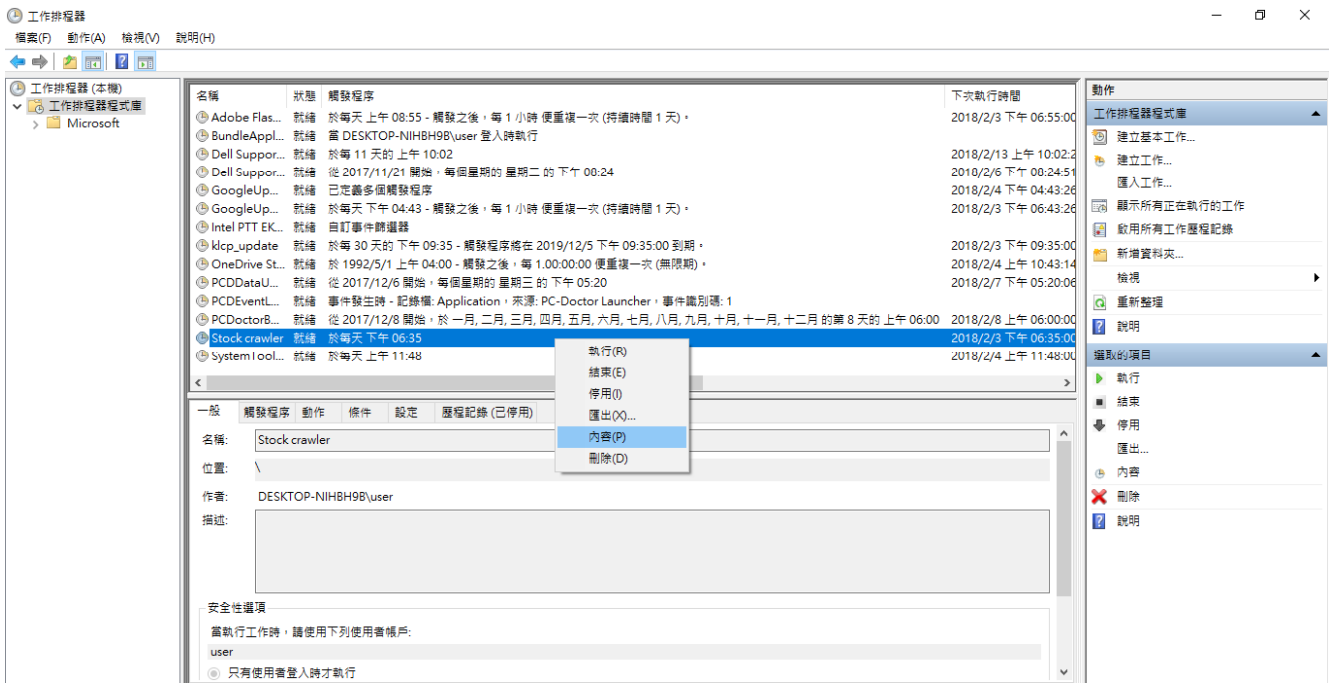


Scheduling (2)



Scheduling (3)

Scheduling (4)



Scheduling (5)