

# 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/>

The screenshot shows the Python.org website interface. At the top, there is a dark blue header with the Python logo and the word "python" in white. To the right of the logo is a search bar with a magnifying glass icon, a "GO" button, and a "Socialize" link. Below the header is a navigation menu with tabs for "About", "Downloads", "Documentation", "Community", "Success Stories", "News", and "Events". The "Downloads" tab is selected, and a dropdown menu is open, showing options: "All releases", "Source code", "Windows", "Mac OS X", "Other Platforms", "License", and "Alternative Implementations". The "Windows" option is highlighted, and a sub-menu is open, showing "Download for Windows" with buttons for "Python 3.6.4" and "Python 2.7.14". Below this, a note states: "Note that Python 3.5+ cannot be used on Windows XP or earlier. Not the OS you are looking for? Python can be used on many operating systems and environments. View the full list of downloads." The main content area features the "Python 3.6.4" logo, the release date "2017-12-19", and a list of major new features compared to 3.5. The list includes: PEP 468, PEP 487, PEP 495, PEP 498, PEP 506, PEP 509, PEP 515, PEP 519, PEP 520, PEP 523, PEP 524, PEP 525, and PEP 526.

**Python 3.6.4**

**Release Date:** 2017-12-19

Python 3.6.4 is the fourth maintenance release of Python 3.6. For more information, see the [Python 3.6.4 Release Notes](#).

**Download for Windows**

Python 3.6.4 Python 2.7.14

**Note that Python 3.5+ cannot be used on Windows XP or earlier.**

Not the OS you are looking for? Python can be used on many operating systems and environments. View the full list of downloads.

**Major new features of the 3.6 series, compared to 3.5**

Among the new major new features in Python 3.6 are:

- [PEP 468](#), Preserving Keyword Argument Order
- [PEP 487](#), Simpler customization of class creation
- [PEP 495](#), Local Time Disambiguation
- [PEP 498](#), Literal String Formatting
- [PEP 506](#), Adding A Secrets Module To The Standard Library
- [PEP 509](#), Add a private version to dict
- [PEP 515](#), Underscores in Numeric Literals
- [PEP 519](#), Adding a file system path protocol
- [PEP 520](#), Preserving Class Attribute Definition Order
- [PEP 523](#), Adding a frame evaluation API to CPython
- [PEP 524](#), Make `os.urandom()` blocking on Linux (during system startup)
- [PEP 525](#), Asynchronous Generators (provisional)
- [PEP 526](#), Syntax for Variable Annotations (provisional)

# 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/>

The screenshot shows the Python.org website interface. At the top, there is a dark blue header with the Python logo and the word "python" in white. To the right of the logo is a search bar with a magnifying glass icon, a "GO" button, and a "Socialize" button. Below the header is a navigation menu with links for "About", "Downloads", "Documentation", "Community", "Success Stories", "News", and "Events".

The main content area is titled "Python 3.6.4" with a "Release Date: 2017-12-19". Below this, there is a paragraph of text and a section titled "Major new features of the 3.6 series". A list of features follows, each preceded by a bullet point and a link to a PEP document.

A dropdown menu is open over the "Documentation" link in the navigation menu. The menu items are: "Docs", "Audio/Visual Talks", "Beginner's Guide", "Developer's Guide", "FAQ", "Non-English Docs", "PEP Index", "Python Books", and "Python Essays".

A tooltip is visible over the "Python 3.x Docs" button in the dropdown menu. The tooltip text reads: "Python's standard documentation: download, browse or watch a tutorial. Get started below, or visit the Documentation page to browse by version." Below this text are two buttons: "Python 3.x Docs" and "Python 2.x Docs". At the bottom of the tooltip, it says "See also Should I use Python 2 or 3?".

The list of new features for Python 3.6 includes:

- [PEP 468](#), Preserving Keyword Argument Order
- [PEP 487](#), Simpler customization of class creation
- [PEP 495](#), Local Time Disambiguation
- [PEP 498](#), Literal String Formatting
- [PEP 506](#), Adding A Secrets Module To The Standard Library
- [PEP 509](#), Add a private version to dict
- [PEP 515](#), Underscores in Numeric Literals
- [PEP 519](#), Adding a file system path protocol
- [PEP 520](#), Preserving Class Attribute Definition Order
- [PEP 523](#), Adding a frame evaluation API to CPython
- [PEP 524](#), Make `os.urandom()` blocking on Linux (during system startup)
- [PEP 525](#), Asynchronous Generators (provisional)
- [PEP 526](#), Syntax for Variable Annotations (provisional)

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

Source: <https://docs.python.org/3.6/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.

- 1. Whetting Your Appetite
- 2. Using the Python Interpreter
  - 2.1. Invoking the Interpreter
    - 2.1.1. Argument Passing
    - 2.1.2. Interactive Mode
  - 2.2. The Interpreter and Its Environment
    - 2.2.1. Source Code Encoding
- 3. An Informal Introduction to Python
  - 3.1. Using Python as a Calculator
    - 3.1.1. Numbers
    - 3.1.2. Strings
    - 3.1.3. Lists
  - 3.2. First Steps Towards Programming

### 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*

### Indices and tables:

#### Global Module Index

*quick access to all modules*

#### General Index

*all functions, classes, terms*

#### Glossary

*the most important terms explained*

#### 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!)*

#### 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/> )

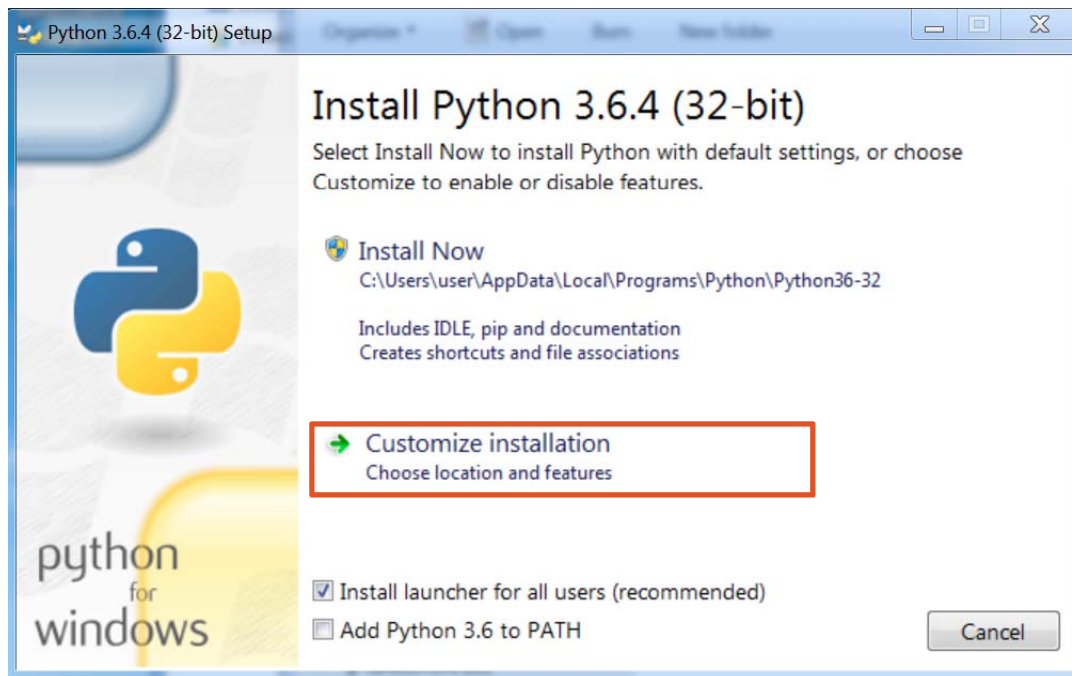
- If you have a question, it's a good idea to try the [FAQ](#), which answers the most commonly asked questions about Python.

# Looking to Help?

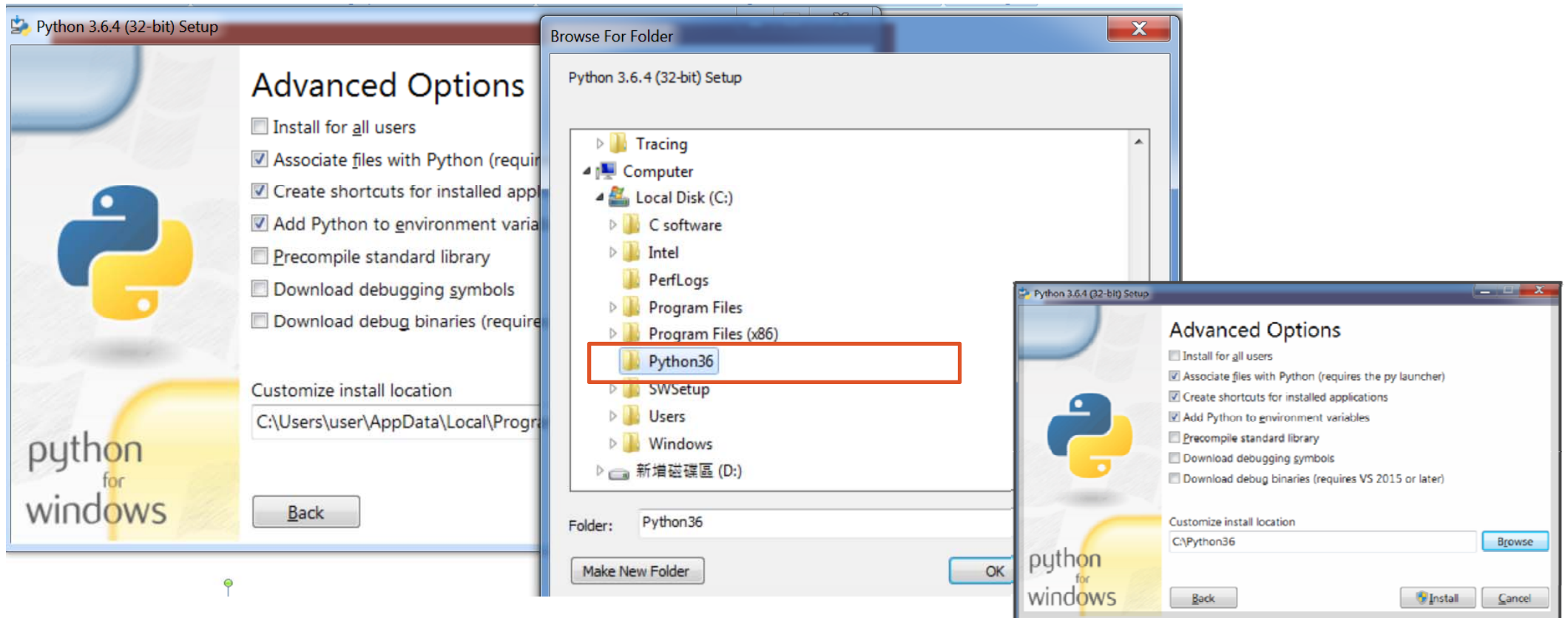
(Source URL: <https://www.python.org/about/gettingstarted/> )

- 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!

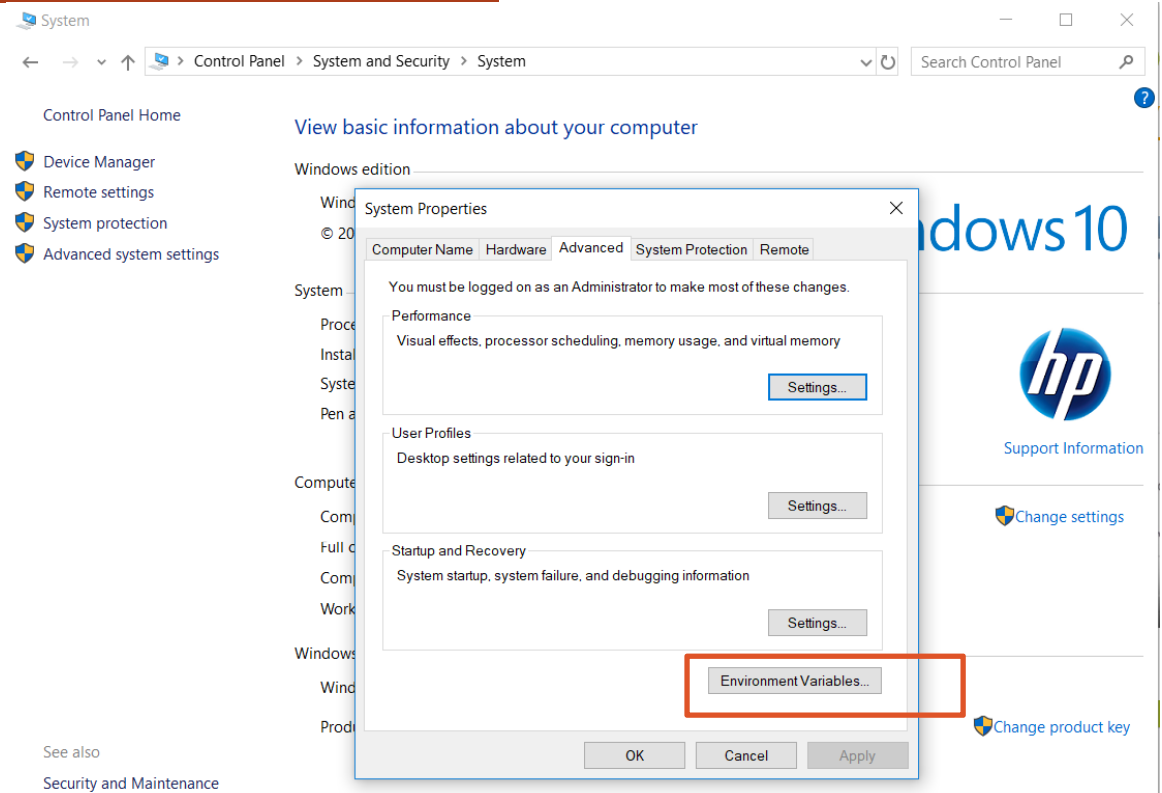
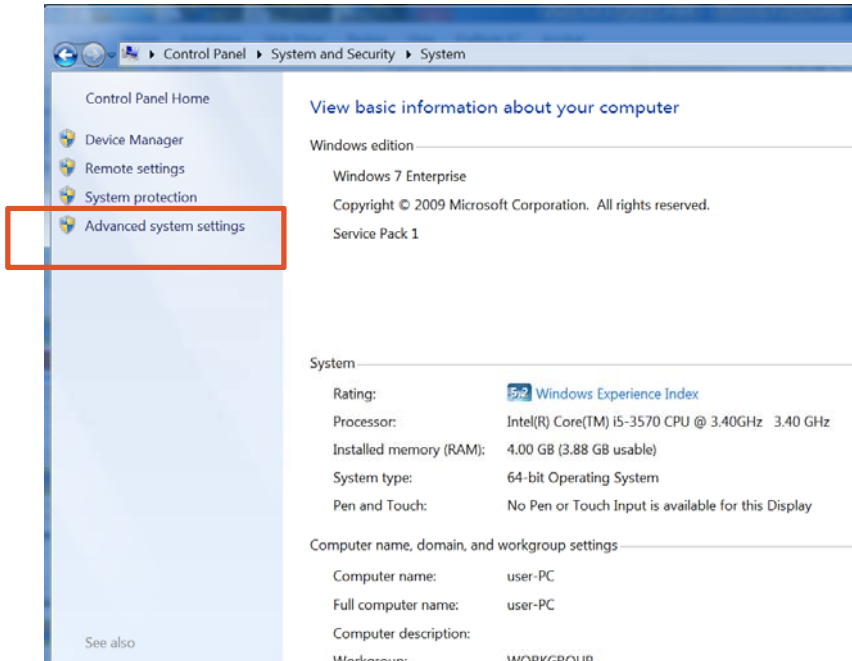
# 1. Installing Python 3.6.4 (1)



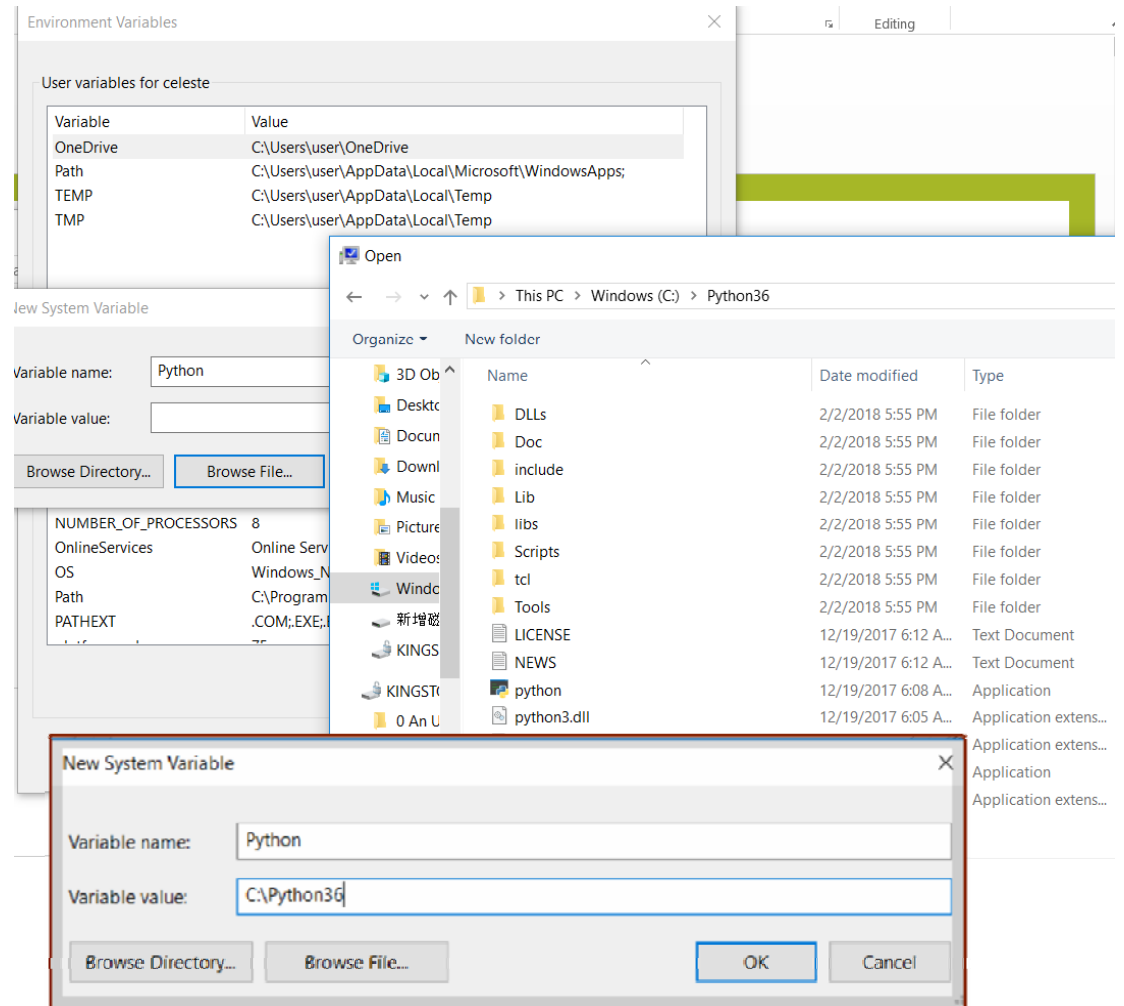
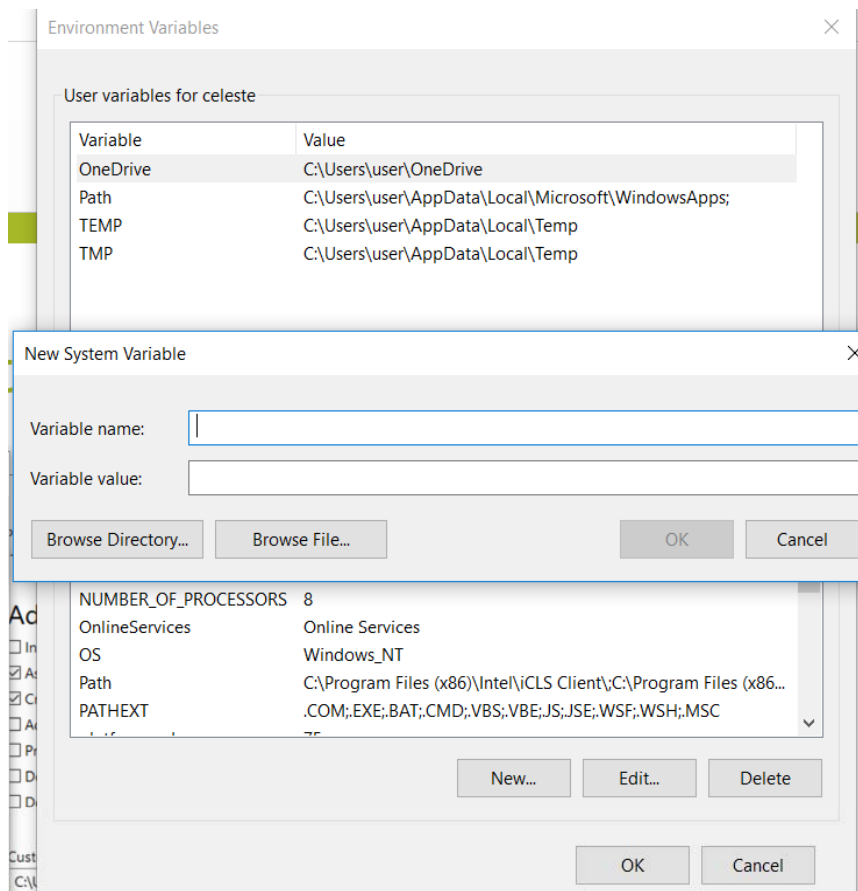
# 1. Installing Python 3.6.4 (2)



## 2. Setting environment (1)

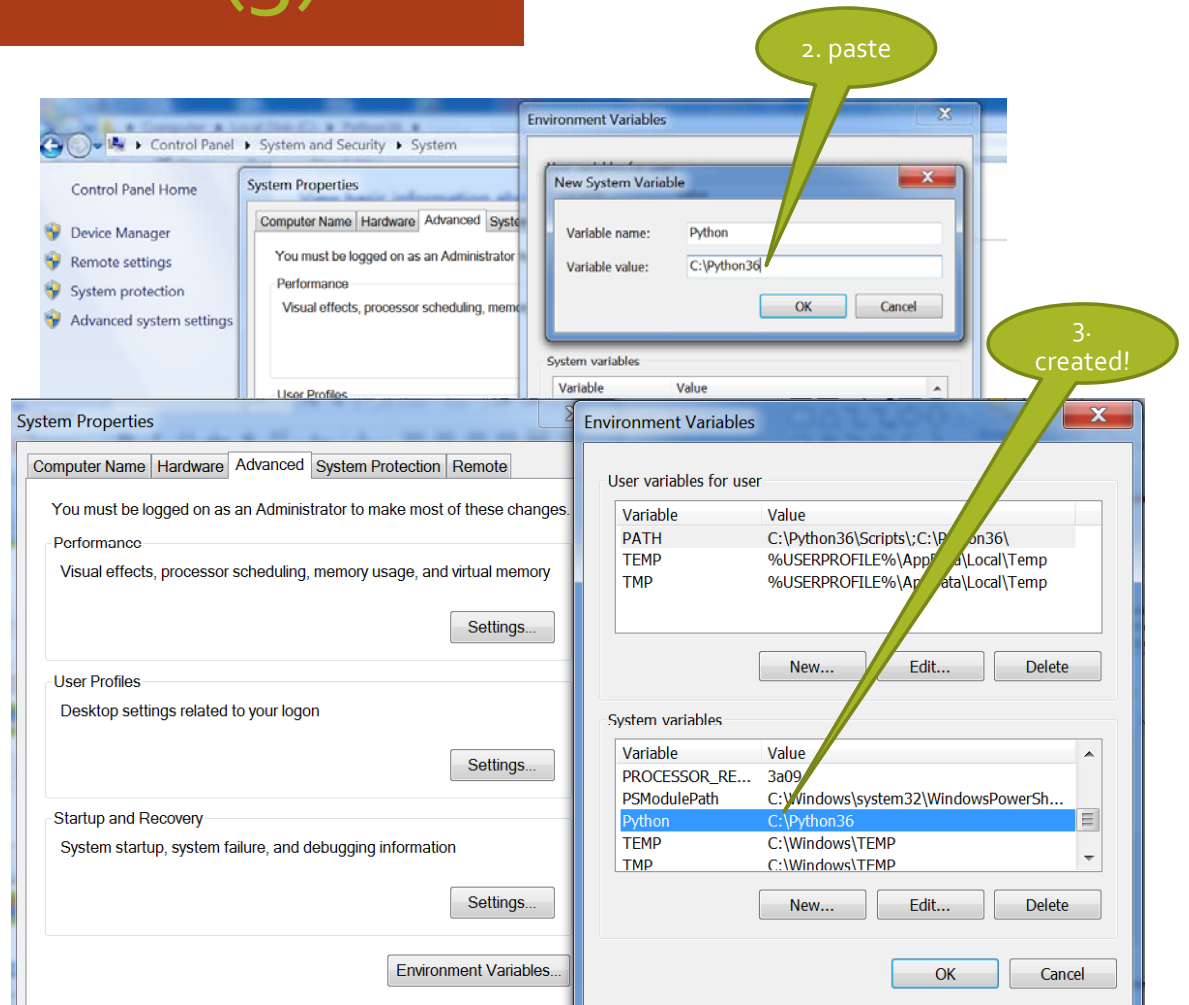
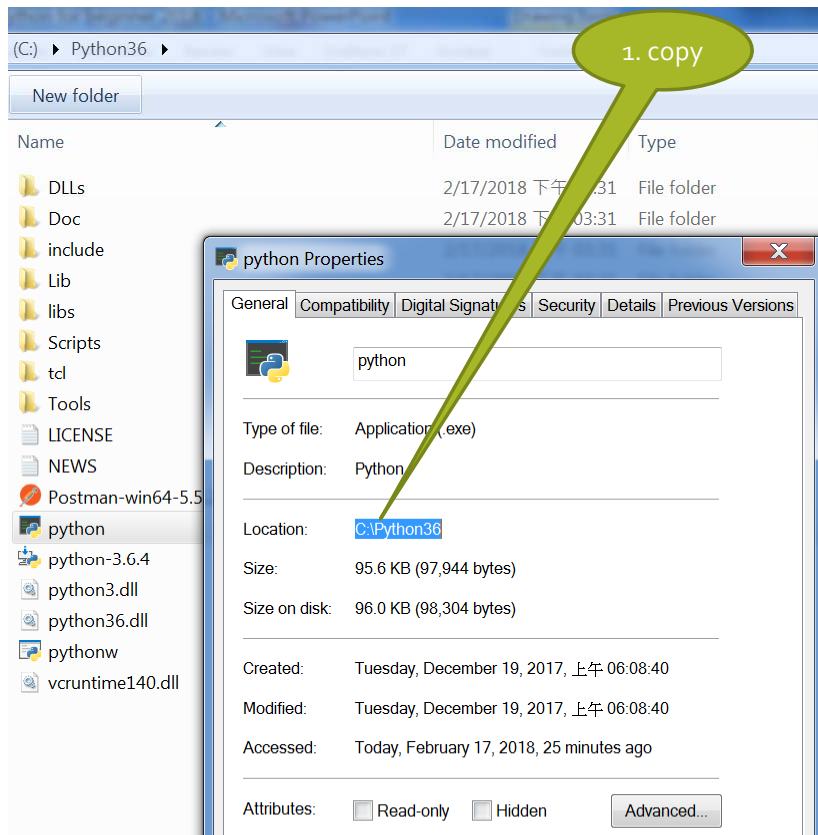


## 2. Setting environment (2) – my case



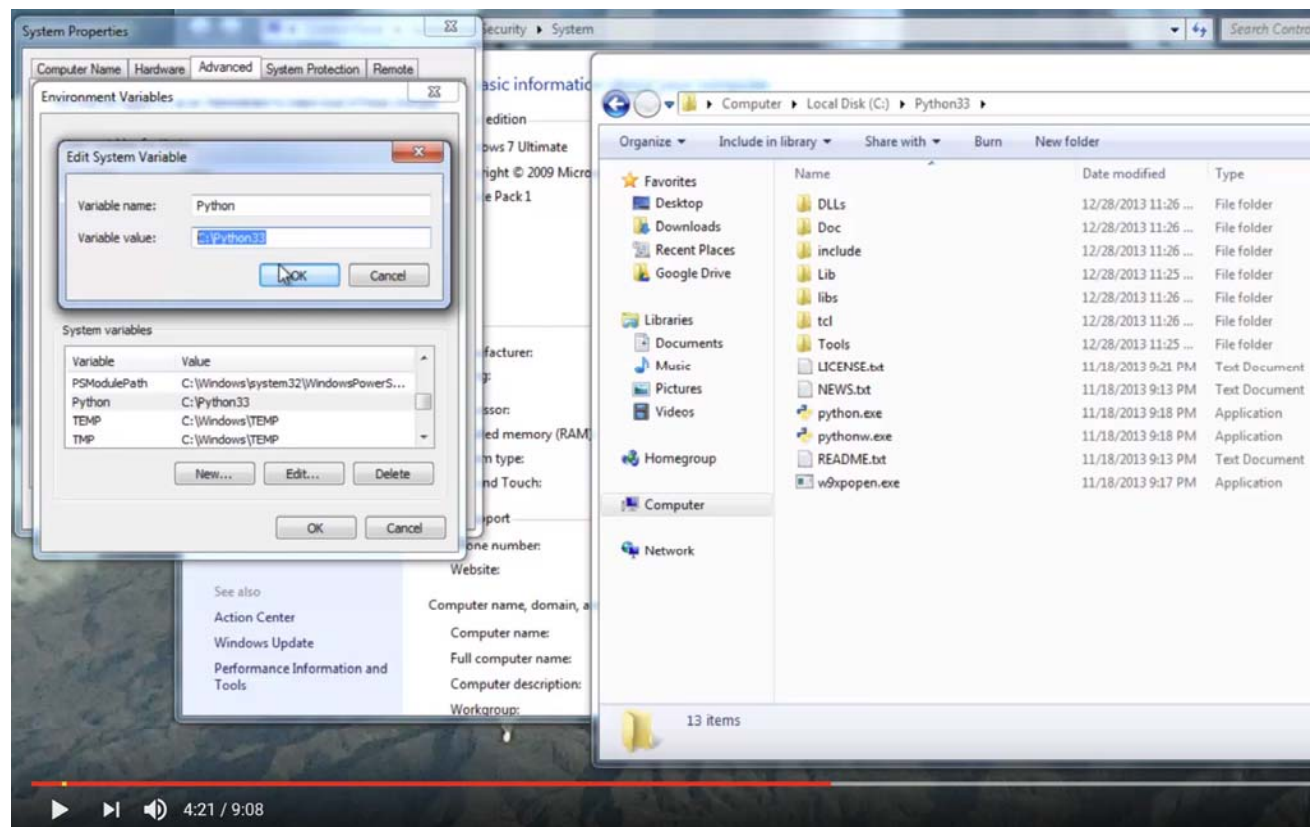
## 2. Setting environment (3)

Copy the path and paste on "Variable value"



## 2. Setting environment

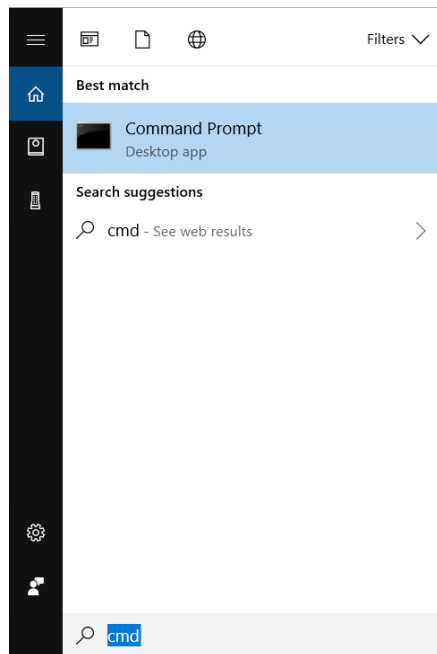
(Source URL: <https://www.youtube.com/watch?v=cpPGobKHYKc>)





# 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.

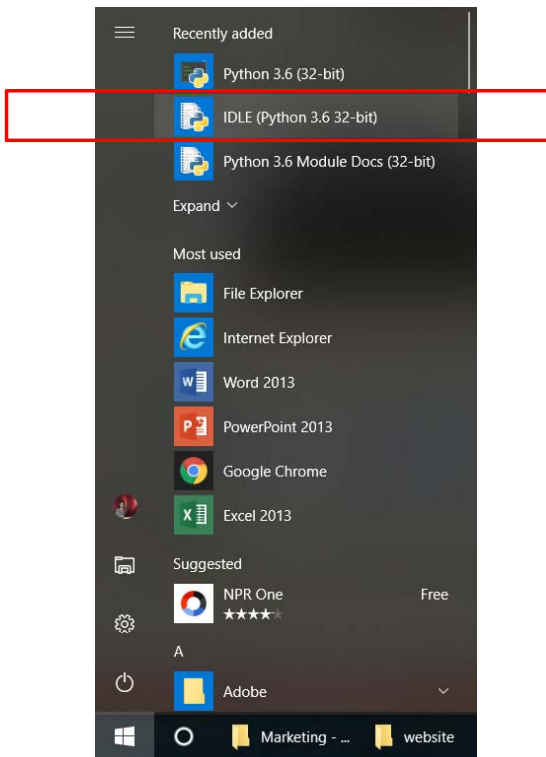
C:\Users\user>python
'python' is not recognized as an internal or external command,
operable program or batch file.

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
>>>
```

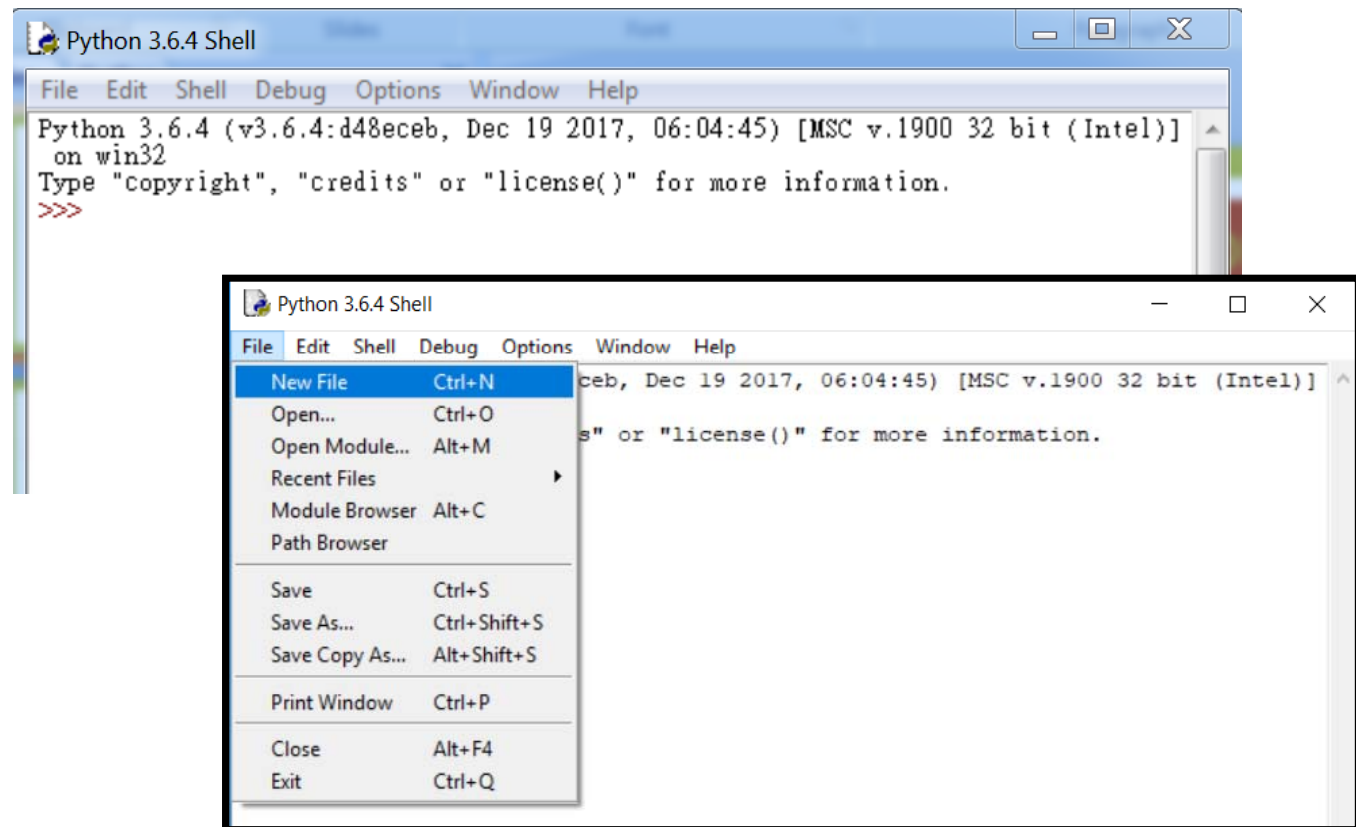
# 4. Hello world program

## Step 1 – Create a python program file (1)

(1) Open "IDLE" from the Start Menu



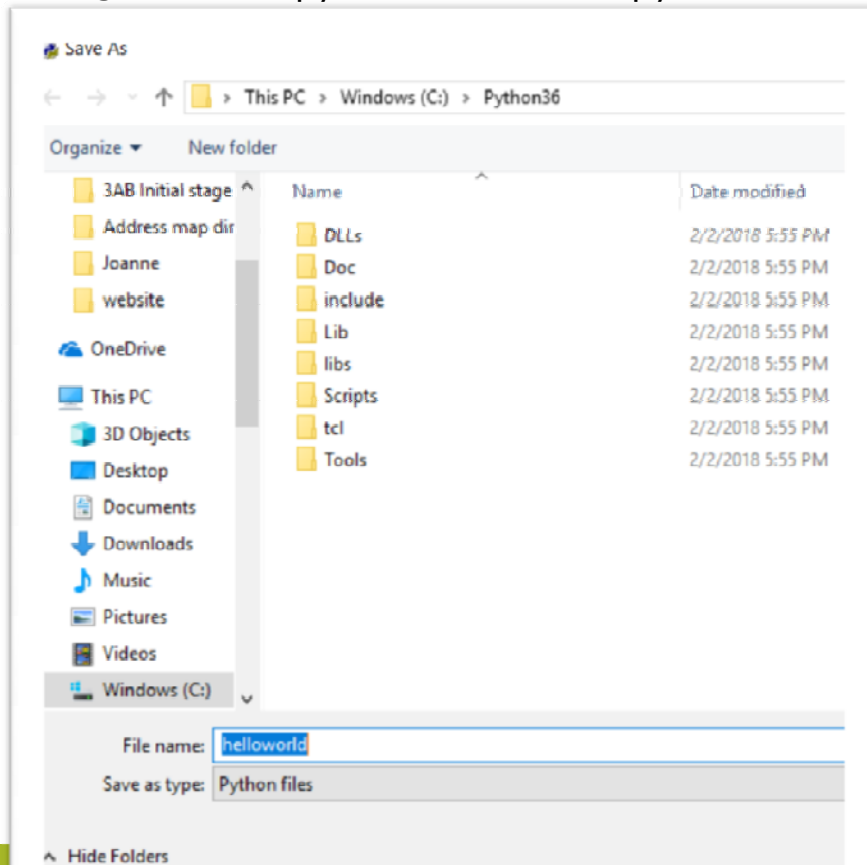
(2) From the "Python Shell, click File → "New File"



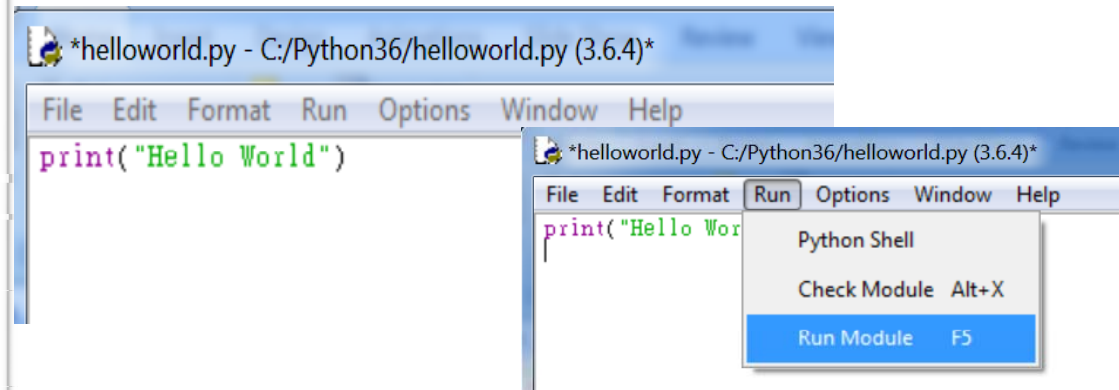
# 4. Hello world program

## Step 1 – Create a python program file (2)

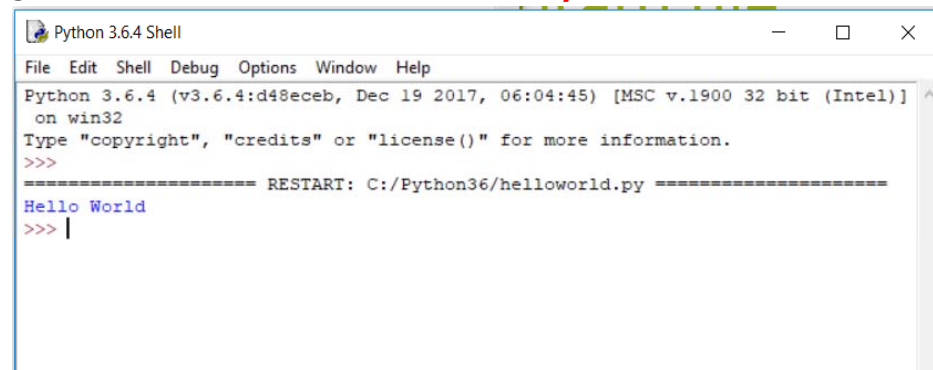
(3) Save the python file as “\*\*\*.py”



(4) Type the coding in the **NEW-File** → select “Run” → “Run Module”



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

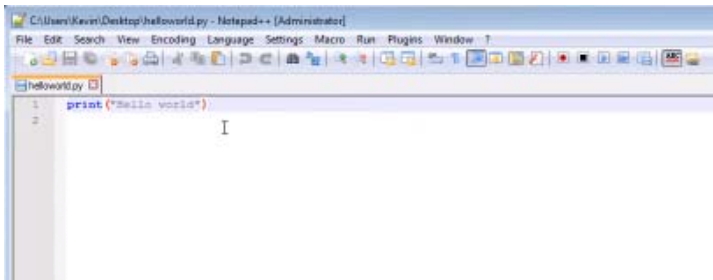


## 4. Hello world program

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

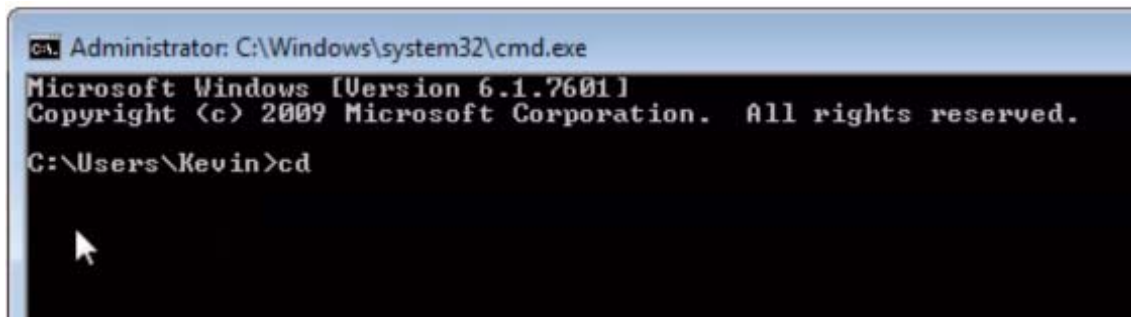
### Step 2 – Determine the path of a python program file

It is better type code & edit in “Notepad ++”



```
1 print("Hello world")
2
```

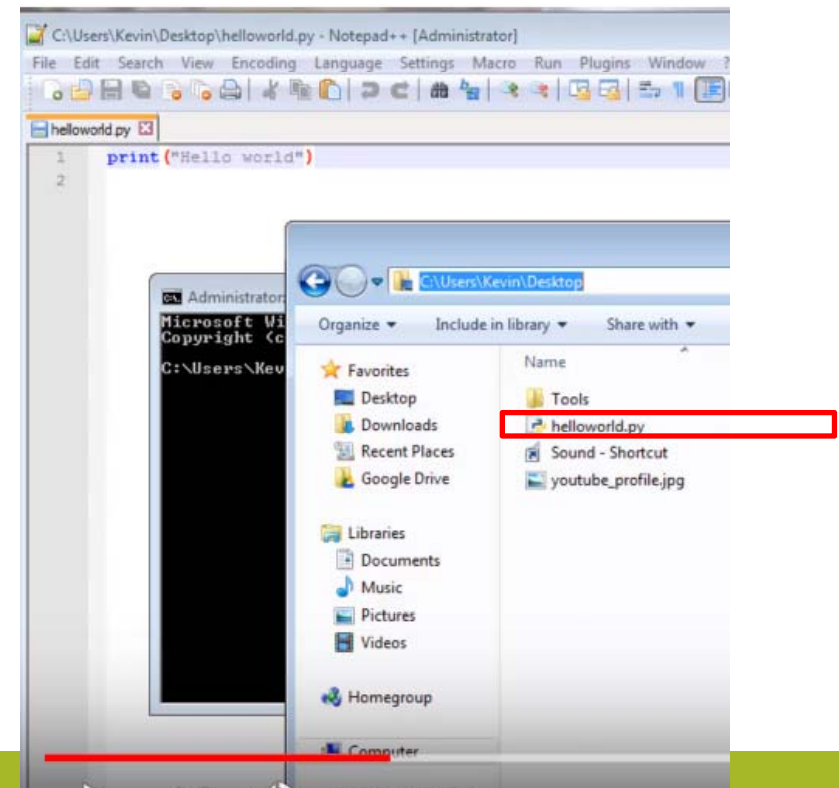
Call out the “command prompt” n type cmd



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Kevin>cd
```

Type “cmd” to change to the directory where you save python program file....  
Copy the file directory PATH

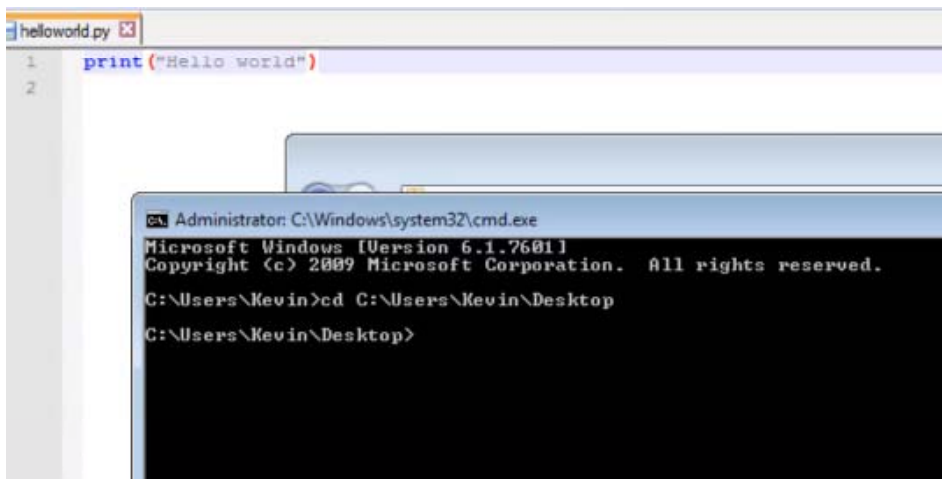


## 4. Hello world program

(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

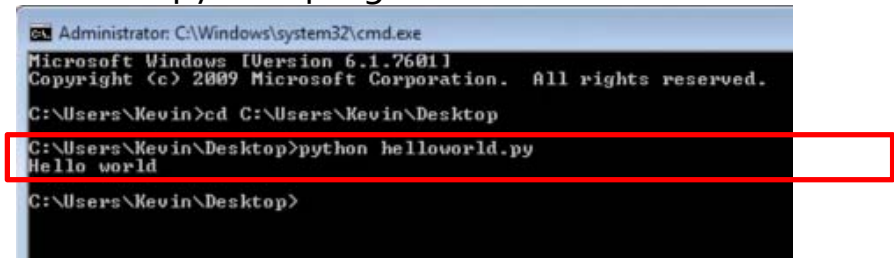


```
helloworld.py
1 print("Hello world")
2

Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Kevin>cd C:\Users\Kevin\Desktop
C:\Users\Kevin\Desktop>
```

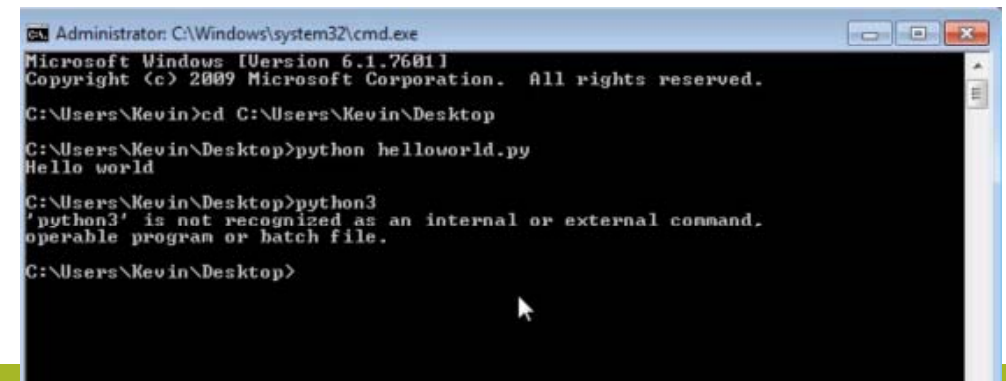
Run the python program



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Kevin>cd C:\Users\Kevin\Desktop
C:\Users\Kevin\Desktop>python helloworld.py
Hello world
C:\Users\Kevin\Desktop>
```

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**



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Kevin>cd C:\Users\Kevin\Desktop
C:\Users\Kevin\Desktop>python helloworld.py
Hello world

C:\Users\Kevin\Desktop>python3
'python3' is not recognized as an internal or external command,
operable program or batch file.

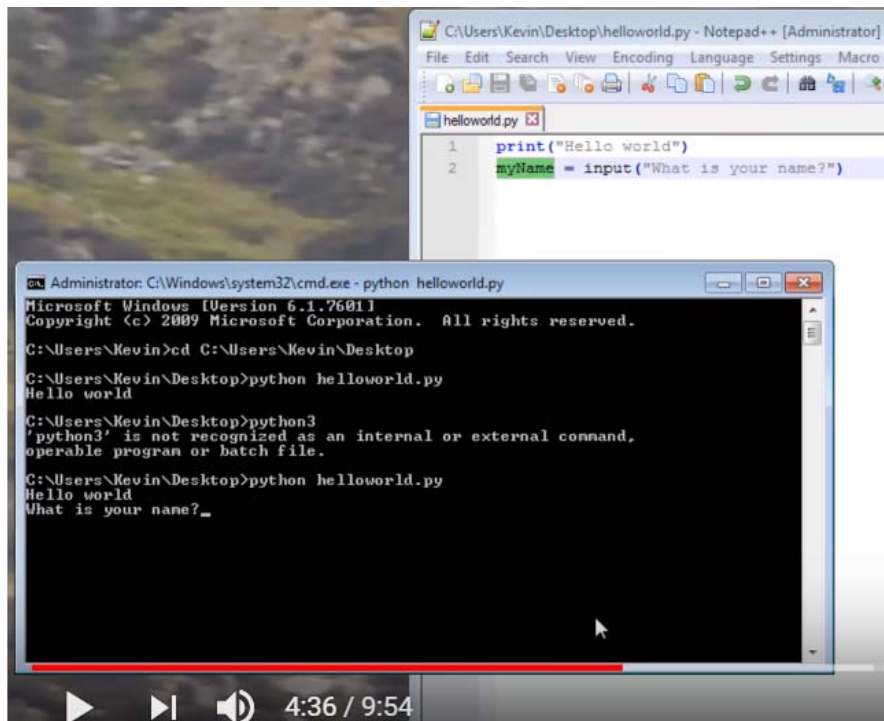
C:\Users\Kevin\Desktop>
```

## 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"



```
C:\Users\Kevin\Desktop\helloworld.py - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro
helloworld.py
1 print("Hello world")
2 myName = input("What is your name?")

Administrator: C:\Windows\system32\cmd.exe - python helloworld.py
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Kevin>cd C:\Users\Kevin\Desktop
C:\Users\Kevin\Desktop>python helloworld.py
Hello world

C:\Users\Kevin\Desktop>python3
'python3' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Kevin\Desktop>python helloworld.py
Hello world
What is your name?_
```

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

Home Remedies Index Feedback Contact

Top10 Home Remedies

HOME REMEDIES HOW TO SUPERFOODS KITCHEN INGREDIENTS NEWS & FACTS DIY HOMEMADE

Home » Kitchen Ingredients » 10 Health Benefits of Ginger

# 10 Health Benefits of Ginger

Home » Kitchen Ingredients » 10 Health Benefits of Ginger

10 Health Benefits of Ginger

Prev 1 of 3 Next

Ginger is a natural herb that is used across the globe as a spice. Due to this herb also is considered a virtual medicine chest. Various studies have proven that ginger is highly effective for treating a number of health problems.

Ginger has **anti-nausea, anti-spasmodic, antifungal, anti-inflammatory, antiseptic, antibacterial, antiviral and antitussive (cough suppressant) properties** that can do wonders for your health.

Also, ginger is a good source of vitamins A, C, E, and B-complex, magnesium, phosphorus, potassium, silicon, sodium, iron, zinc, calcium and beta-carotene.

Ginger is a natural herb that is used across the globe as a spice. Due to the various health benefits of ginger, this herb also is considered a virtual medicine chest. Various studies have proven that ginger is highly effective for treating a number of health problems.

Ginger has **anti-nausea, anti-spasmodic, antifungal, anti-inflammatory, antiseptic, antibacterial, antiviral and antitussive (cough suppressant) properties** that can do wonders for your health.

Also, ginger is a good source of vitamins A, C, E, and B-complex, magnesium, phosphorus, potassium, silicon, sodium, iron, zinc, calcium and beta-carotene.

TiSPO

```
ginger-benefits tag-headache tag-health-benefits tag-immune-system tag-inflammation tag-nausea tag-pain" itemscope itemtype="http://schema.org/Article">
  <!-- start:article post header -->
  <!-- end:article post header -->
  <!-- start:article post content -->
  <!-- end:article post content -->
  <!-- start:article post addons - via & source -->
  <!-- end:article post addons -->
  <!-- start:article post footer -->
  <!-- WP QUADS v. 1.7.6 Shortcode Ad -->
```

div #page-content #main #post-610 div.article-post-content.clearfix

Filter: #hov .cls + user agent styles

```
display: block;
}
Inherited from body.post-template-default.single...
body {
  font-family: Arial,Helvetica,sans-serif;
  line-height: 24px;
  font-style: normal;
  color: #000;
  font-size: 16px;
  opacity: 1;
  visibility: visible;
  -webkit-transition: opacity 0.24s ease-in-out;
  -ms-transition: opacity 0.24s ease-in-out;
  transition: opacity 0.24s ease-in-out;
}
```

margin: - - - -

border: - - - -

padding: 666.667 x 3639.330 - - - -

40

Filter: Show all

- box-sizing border-
- color rgb(0, block
- display
- font-family Arial

# 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`

```
Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/user/Desktop/yahooidle.py =====
Traceback (most recent call last):
  File "C:/Users/user/Desktop/yahooidle.py", line 1, in <module>
    import requests
ModuleNotFoundError: No module named 'requests'
>>>
===== RESTART: C:/Users/user/Desktop/yahooidle.py =====
{'query': {'count': 1, 'created': '2018-02-03T06:26:31Z', 'lang': 'en-US', 'resu
lts': {'channel': {'atmosphere': {'humidity': '88', 'pressure': '33999.36', 'ris
ing': '0', 'visibility': '25.75'}, 'item': {'condition': {'code': '26', 'date':
'Sat, 03 Feb 2018 01:00 PM CST', 'temp': '11', 'text': 'Cloudy'}, 'forecast': {'
code': '11', 'date': '03 Feb 2018', 'day': 'Sat', 'high': '11', 'low': '9', 'tex
t': 'Showers'}}}}}}
>>>
```

```
python3.6.4 Shell
File Edit Shell Debug Options Window Help
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/user/Desktop/yahooidle.py =====
Traceback (most recent call last):
  File "C:/Users/user/Desktop/yahooidle.py", line 1, in <module>
    import requests
ModuleNotFoundError: No module named 'requests'
>>>
===== RESTART: C:/Users/user/Desktop/yahooidle.py =====
C:\Users\user>py -m pip install requests
Collecting requests
  Downloading requests-2.18.4-py2.py3-none-any.whl (88kB)
    100% |#####| 92kB 741kB/s
Collecting urllib3[secure]>=1.21.1 (from requests)
  Downloading urllib3-1.22-py2.py3-none-any.whl (132kB)
    100% |#####| 133kB 441kB/s
Collecting chardet>=3.1.0;=>3.0.2 (from requests)
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133kB)
    100% |#####| 143kB 1.5MB/s
Collecting certifi=2017.4.17 (from requests)
  Downloading certifi-2018.1.18-py2.py3-none-any.whl (151kB)
    100% |#####| 153kB 232kB/s
Collecting idna>=2.7;=>2.5 (from requests)
  Downloading idna-2.6-py2.py3-none-any.whl (56kB)
    100% |#####| 61kB 1.2MB/s
Installing collected packages: urllib3, chardet, certifi, idna, requests
Successfully installed certifi-2018.1.18 chardet-3.0.4 idna-2.6 requests-2.18.4 urllib3-1.22
C:\Users\user>
```



# Run the Yahoo Weather API program in Python



The image shows a screenshot of a Python IDE window titled "yahooidle.py - C:/Users/user/Desktop/yahooidle.py (3.6.4)". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code in the editor is as follows:

```
import requests  
  
result = requests.get("https://query.yahooapis.com/v1/public/yql?q=select%20item.forecast%2C%20item.condition%2C%20at:  
print(result.json())
```

# Run the Yahoo Weather API program in Postman

The screenshot displays the Postman interface with a GET request to the Yahoo Weather API. The request URL is `https://query.yahooapis.com/v1/public/yql?q=select%20item.forecast%2C%20item.condition%2C%20atmosphere%20%20%20from%20weather.forecast%20where%20woeid%20%3D%202306179%20and%20u%3D%22c%22%20limit%201&format=json&env=store%3A%2F%2Fdatatables.org%2Ffalltableswithkeys`. The response is a JSON object showing weather data for a specific location on February 3, 2018.

```
5  "lang": "en-US",
6  "results": {
7    "channel": {
8      "atmosphere": {
9        "humidity": "88",
10       "pressure": "33999.36",
11       "rising": "0",
12       "visibility": "25.75"
13     },
14     "item": {
15       "condition": {
16         "code": "26",
17         "date": "Sat, 03 Feb 2018 01:00 PM CST",
18         "temp": "11",
19         "text": "Cloudy"
20       },
21       "forecast": {
22         "code": "11",
23         "date": "03 Feb 2018",
24         "day": "Sat",
25         "high": "11",
26         "low": "9",
27         "text": "Showers"
28       }
29     }
30   }
31 }
```

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

```
yahooidle.py - C:/Users/user/Desktop/yahooidle.py (3.6.4)
File Edit Format Run Options Window Help
import requests
result = requests.get("https://query.yahooapis.com/v1/public/yql?q=select%20item.forecast%2C%20item.condition%2C%20at
jsonData = result.json()
print(jsonData["query"]["results"]["channel"]["item"]["forecast"])
```

# Result is shown in Python Shell

```
Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/user/Desktop/yahooidle.py =====
Traceback (most recent call last):
  File "C:/Users/user/Desktop/yahooidle.py", line 1, in <module>
    import requests
ModuleNotFoundError: No module named 'requests'
>>>
===== RESTART: C:/Users/user/Desktop/yahooidle.py =====
{'query': {'count': 1, 'created': '2018-02-03T06:26:31Z', 'lang': 'en-US', 'results': {'channel': {'atmosphere': {'humidity': '88', 'pressure': '33999.36', 'rising': '0', 'visibility': '25.75'}, 'item': {'condition': {'code': '26', 'date': 'Sat, 03 Feb 2018 01:00 PM CST', 'temp': '11', 'text': 'Cloudy'}, 'forecast': {'code': '11', 'date': '03 Feb 2018', 'day': 'Sat', 'high': '11', 'low': '9', 'text': 'Showers'}}}}}}
>>> dic = {"a": 1, "b": 2}
>>> dic
{'a': 1, 'b': 2}
>>> dic["b"]
2
>>>
===== RESTART: C:/Users/user/Desktop/yahooidle.py =====
Traceback (most recent call last):
  File "C:/Users/user/Desktop/yahooidle.py", line 7, in <module>
    print(jsonData["query"]["results"]["channel"]["item"]["forecast"])
KeyError: 'query'
>>>
===== RESTART: C:/Users/user/Desktop/yahooidle.py =====
{'code': '11', 'date': '03 Feb 2018', 'day': 'Sat', 'high': '11', 'low': '9', 'text': 'Showers'}
>>> |
```

# Simplify result in Shell

```
Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/user/Desktop/yahoodidle.py =====
Traceback (most recent call last):
  File "C:/Users/user/Desktop/yahoodidle.py", line 1, in <module>
    import requests
ModuleNotFoundError: No module named 'requests'
>>>
===== RESTART: C:/Users/user/Desktop/yahoodidle.py =====
({'query': {'count': 1, 'created': '2018-02-03T06:26:31Z', 'lang': 'en-US', 're
sults': {'channel': {'atmosphere': {'humidity': '88', 'pressure': '33999.36', 'r
ing': '0', 'visibility': '25.75'}, 'item': {'condition': {'code': '26', 'date
': 'Sat, 03 Feb 2018 01:00 PM CST', 'temp': '11', 'text': 'Cloudy', 'forecast':
'code': '11', 'date': '03 Feb 2018', 'day': 'Sat', 'high': '11', 'low': '9', 't
ext': 'Showers'}}}}}}})
>>> dic = {"a": 1, "b": 2}
>>> dic
{'a': 1, 'b': 2}
>>> dic["b"]
2
>>>
===== RESTART: C:/Users/user/Desktop/yahoodidle.py =====
Traceback (most recent call last):
  File "C:/Users/user/Desktop/yahoodidle.py", line 7, in <module>
    print(jsonData["query"]["results"]["channel"]["item"]["forecast"])
KeyError: 'query'
>>>
===== RESTART: C:/Users/user/Desktop/yahoodidle.py =====
({'code': '11', 'date': '03 Feb 2018', 'day': 'Sat', 'high': '11', 'low': '9',
ext': 'Showers'})
>>>
===== RESTART: C:/Users/user/Desktop/yahoodidle.py =====
11
9
>>>
```

```
yahoodidle.py - C:/Users/user/Desktop/yahoodidle.py (3.6.4)
File Edit Format Run Options Window Help
import requests
result = requests.get("https://query.yahooapis.com/v1/public/yql?q=select%20item.f
jsonData = result.json()
forecastData = jsonData["query"]["results"]["channel"]["item"]["forecast"]
print(forecastData["high"])
print(forecastData["low"])
```

# Install “BeautifulSoup” and “Openpyxl” in cmd

```
cmd  提取 命令提示字元
100% |████████████████████████████████████████████████████████████████████████████████| 61kB 1.2MB/s
Installing collected packages: urllib3, chardet, certifi, idna, requests
Successfully installed certifi-2018.1.18 chardet-3.0.4 idna-2.6 requests-2.18.4 urllib3-1.22

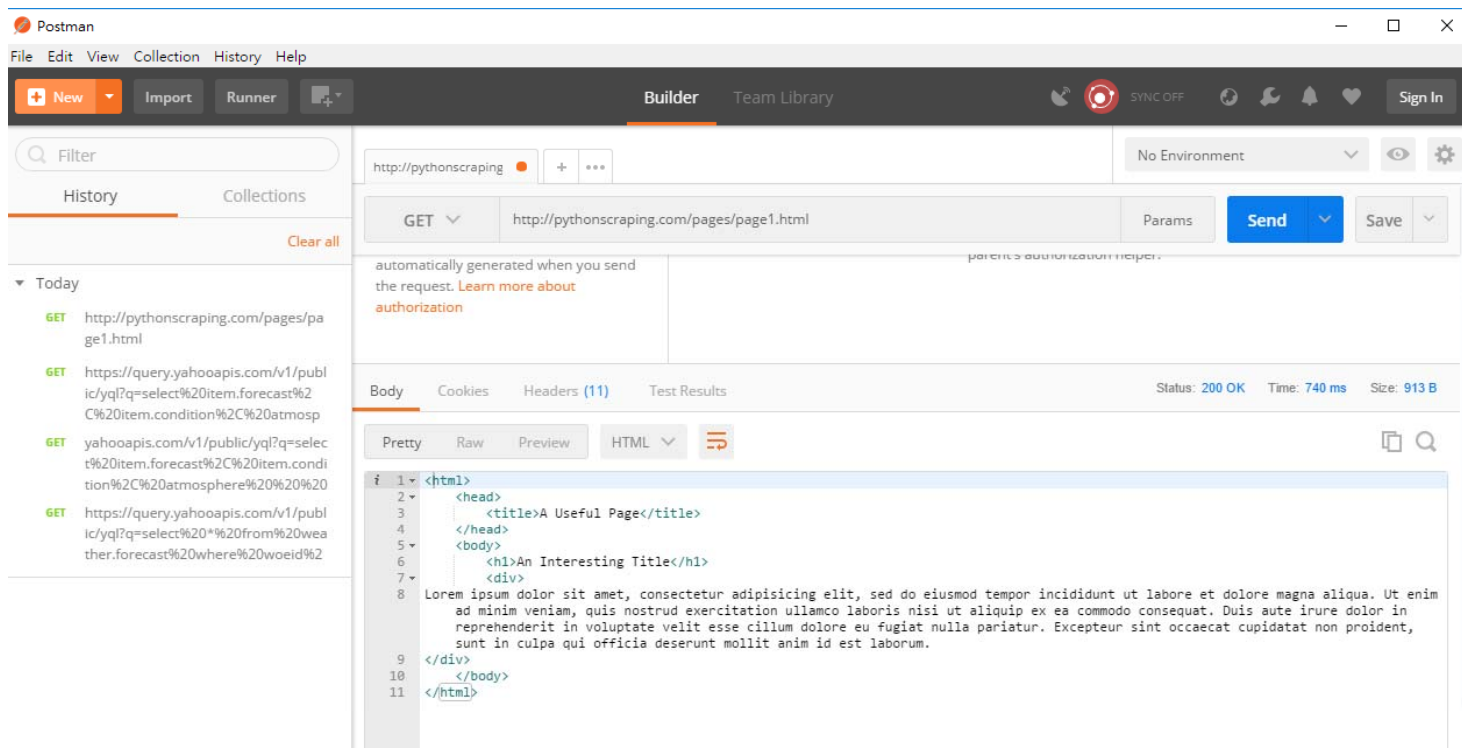
C:\Users\user>py -m pip install bs4
Collecting bs4
  Downloading bs4-0.0.1.tar.gz
Collecting beautifulsoup4 (from bs4)
  Downloading beautifulsoup4-4.6.0-py3-none-any.whl (86kB)
100% |████████████████████████████████████████████████████████████████████████████████| 92kB 153kB/s
Installing collected packages: beautifulsoup4, bs4
  Running setup.py install for bs4 ... done
Successfully installed beautifulsoup4-4.6.0 bs4-0.0.1

C:\Users\user>

C:\Users\user>

C:\Users\user>py -m pip install openpyxl
Collecting openpyxl
  Downloading openpyxl-2.5.0.tar.gz (169kB)
100% |████████████████████████████████████████████████████████████████████████████████| 174kB 725kB/s
Collecting jdcal (from openpyxl)
  Downloading jdcal-1.3.tar.gz
Collecting et_xmlfile (from openpyxl)
  Downloading et_xmlfile-1.0.1.tar.gz
Installing collected packages: jdcal, et-xmlfile, openpyxl
  Running setup.py install for jdcal ... done
  Running setup.py install for et-xmlfile ... done
  Running setup.py install for openpyxl ... done
Successfully installed et-xmlfile-1.0.1 jdcal-1.3 openpyxl-2.5.0
```

# Crawling a webpage – using Postman



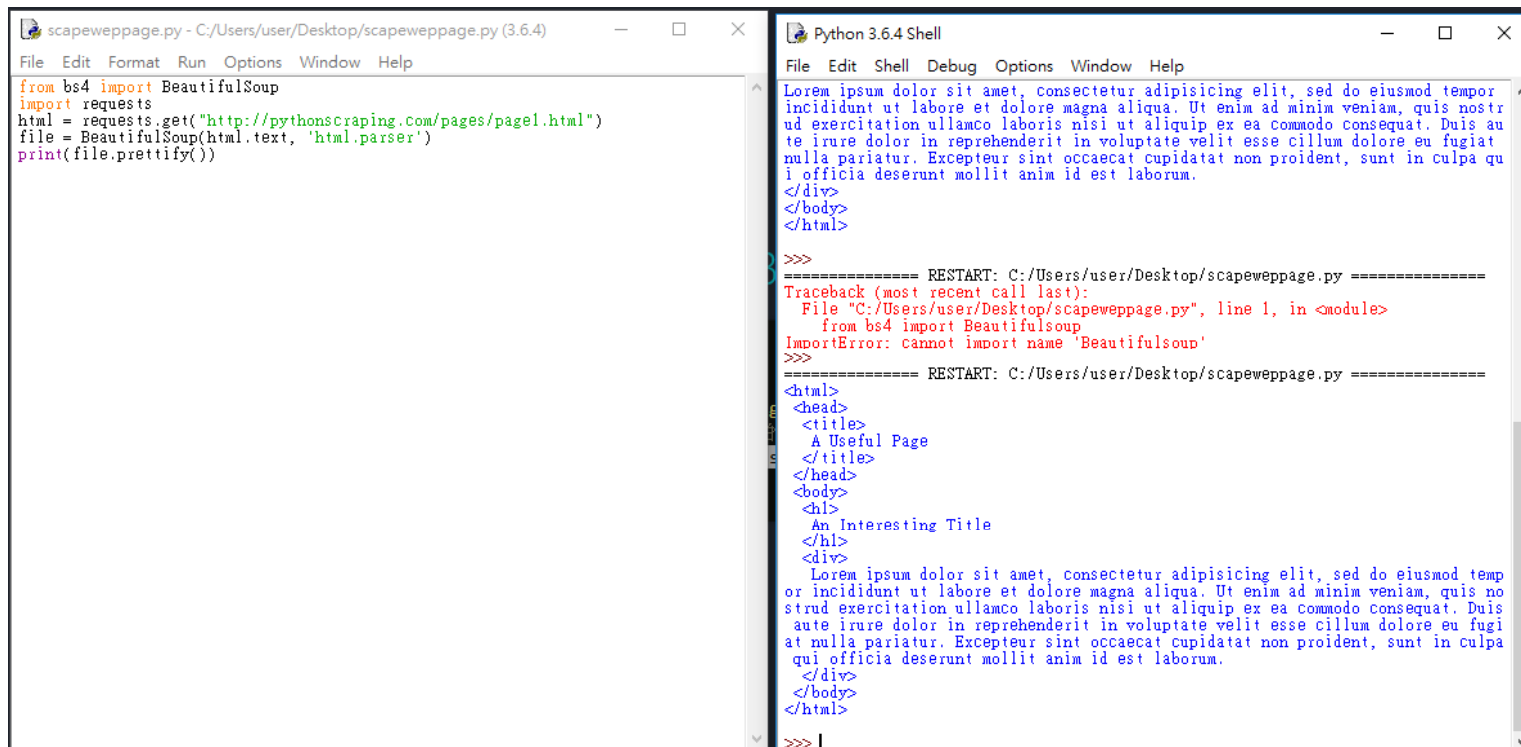
# Crawling a webpage – using Python

```
scapeweppage.py - C:/Users/user/Desktop/scapeweppage.py (3.6.4)
File Edit Format Run Options Window Help
import requests
html = requests.get("http://pythonscraping.com/pages/page1.html")
print(html.text)

Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
t': 'Showers'}}}}}}
>>> dic = {"a": 1, "b": 2}
>>> dic
{'a': 1, 'b': 2}
>>> dic["b"]
2
>>>
===== RESTART: C:/Users/user/Desktop/yahoodle.py =====
Traceback (most recent call last):
  File "C:/Users/user/Desktop/yahoodle.py", line 7, in <module>
    print(jsonData["query"]["results"]["channel"]["item"]["forecast"])
KeyError: 'query'
>>>
===== RESTART: C:/Users/user/Desktop/yahoodle.py =====
{'code': '11', 'date': '03 Feb 2018', 'day': 'Sat', 'high': '11', 'low': '9', 't
ext': 'Showers'}
>>>
===== RESTART: C:/Users/user/Desktop/yahoodle.py =====
11
9
>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
<html>
<head>
<title>A Useful Page</title>
</head>
<body>
<h1>An Interesting Title</h1>
<div>
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostr
ud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis au
te irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat
nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qu
i officia deserunt mollit anim id est laborum.
</div>
</body>
</html>
>>> |
```



# Beautifying output using "BeautifulSoup – in Python Program



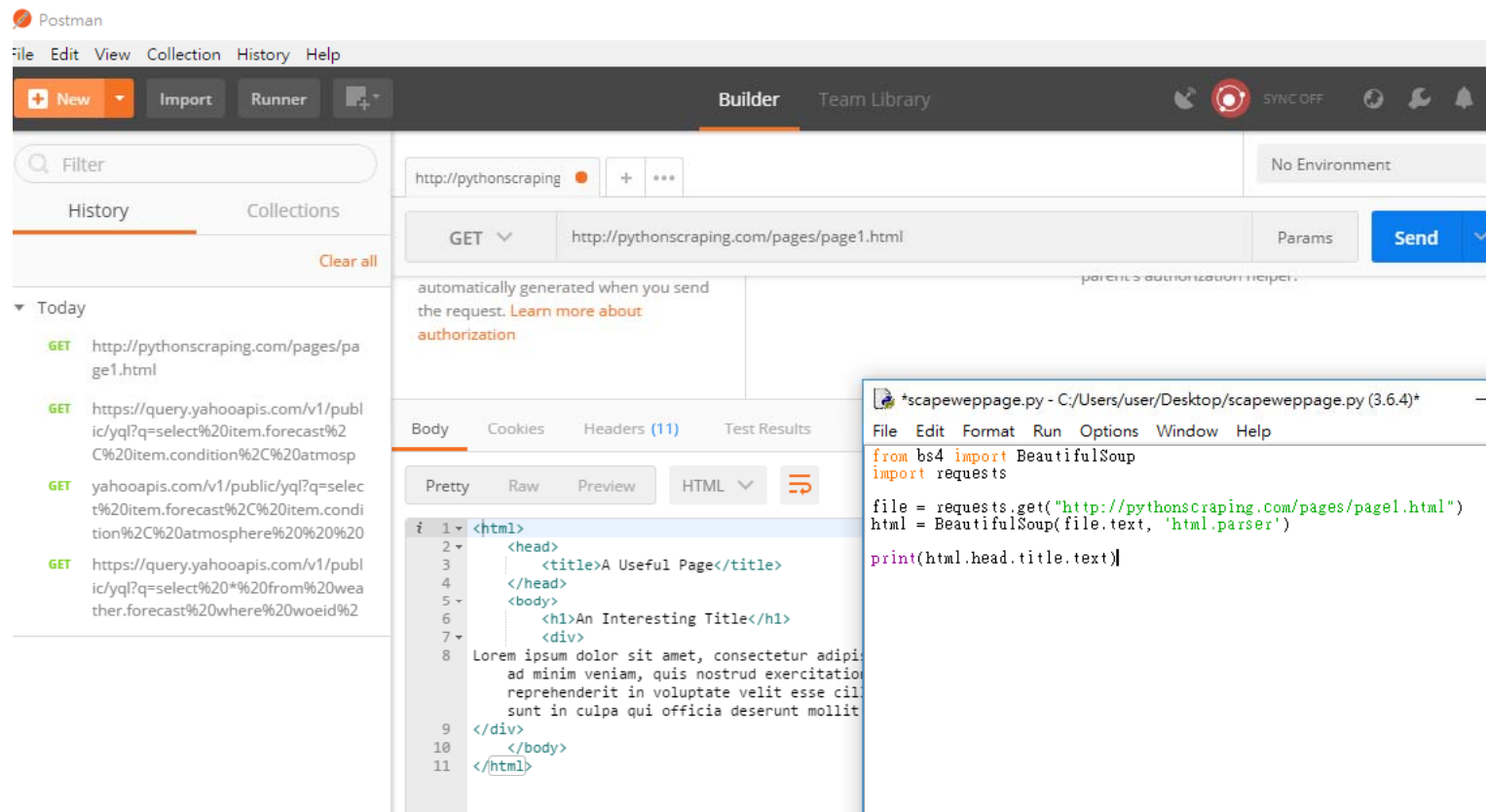
```
scapeweppage.py - C:/Users/user/Desktop/scapeweppage.py (3.6.4)
File Edit Format Run Options Window Help
from bs4 import BeautifulSoup
import requests
html = requests.get("http://pythonscraping.com/pages/page1.html")
file = BeautifulSoup(html.text, 'html.parser')
print(file.prettify())

Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostr
ud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis au
te irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat
nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qu
i officia deserunt mollit anim id est laborum.
</div>
</body>
</html>

>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
Traceback (most recent call last):
  File "C:/Users/user/Desktop/scapeweppage.py", line 1, in <module>
    from bs4 import Beautifulsoup
  ImportError: cannot import name 'Beautifulsoup'
>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
<html>
<head>
  <title>
    A Useful Page
  </title>
</head>
<body>
  <h1>
    An Interesting Title
  </h1>
  <div>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod temp
or incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis no
strud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis
aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugi
at nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa
qui officia deserunt mollit anim id est laborum.
  </div>
</body>
</html>

>>> |
```

# Extract text – using Python



The image shows a screenshot of the Postman web client interface and a Python script. The Postman interface displays a GET request to `http://pythonscraping.com/pages/page1.html`. The response body is shown in HTML format, containing a page title and a paragraph of Lorem Ipsum text. The Python script, named `*scapeweppage.py`, uses the `requests` and `BeautifulSoup` libraries to fetch the page and extract the title.

Postman Interface:

- Request: `GET http://pythonscraping.com/pages/page1.html`
- Response Body (HTML):

```
1 <html>
2   <head>
3     <title>A Useful Page</title>
4   </head>
5   <body>
6     <h1>An Interesting Title</h1>
7     <div>
8       Lorem ipsum dolor sit amet, consectetur adipi
9       ad minim veniam, quis nostrud exercitatio
10      reprehenderit in voluptate velit esse cil
11      sunt in culpa qui officia deserunt mollit
```

Python Script:

```
from bs4 import BeautifulSoup
import requests

file = requests.get("http://pythonscraping.com/pages/page1.html")
html = BeautifulSoup(file.text, 'html.parser')

print(html.head.title.text)
```

# Css

10 Surprising Benefits x 一個下午教你用 Python x A Useful Page x pythonscraping.com/p...

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,

...a penetrating intuition, natural to a man of importance who had grown old in

```
<html>
<head>
  <style>
    .green{
      color:#55ff55;
    }
    .red{
      color:#ff5555;
    }
    #text{
      width:50%;
    }
  </style>
</head>
<body>
  <h1>War and Peace</h1>
  <h2>Chapter 1</h2>
  <div id="text">
    "
    "
    "
    ><span class="red">...</span> == $0
    "
    "
    ><p>...</p>
</body>
</html>
```

html body div#text span.red

Styles Event Listeners DOM Breakpoints Properties

Filter :hov .cls +

```
element.style {
}
.red {
  color: #ff5555;
}
```

margin -  
border -  
padding-  
auto x auto

# Scrape green text only & array

10 Surprising Benefits of Python | 一個下午教你用Python | A Useful Page | pythonscraping.com/

## 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,

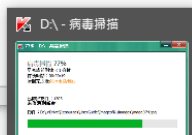
```
scapeweppage.py - C:/Users/user/Desktop/scapeweppage.py (3.6.4)
File Edit Format Run Options Window Help
from bs4 import BeautifulSoup
import requests

file = requests.get("http://pythonscraping.com/pages/warandpeace.html")
html = BeautifulSoup(file.text, 'html.parser')

nameList = html.findAll("span", {"class": "green"})
print(nameList)
```

```
Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
<div>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod temp
  or incididunt ut labore et dolore magna aliqua. Ut enim ad minima veniam, quis no
  strud exercitacion ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis
  aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugi
  at nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa
  qui officia deserunt mollit anim id est laborum.
</div>
</body>
</html>

>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
A Useful Page
>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
[<span class="green">Anna
Pavlovna Scherer</span>, <span class="green">Empress Marya
Fedorovna</span>, <span class="green">Prince Vasili Kuragin</span>, <span class="
green">Anna Pavlovna</span>, <span class="green">St. Petersburg</span>, <span c
lass="green">the prince</span>, <span class="green">Anna Pavlovna</span>, <span
class="green">Anna Pavlovna</span>, <span class="green">the prince</span>, <span
class="green">the prince</span>, <span class="green">the prince</span>, <span c
lass="green">Prince Vasili</span>, <span class="green">Anna Pavlovna</span>, <sp
an class="green">Anna Pavlovna</span>, <span class="green">the prince</span>, <S
pan class="green">Wintzingerode</span>, <span class="green">King of Prussia</spa
n>, <span class="green">le Vicomte de Mortemart</span>, <span class="green">Mont
morencys</span>, <span class="green">Rohans</span>, <span class="green">Abbe Mor
io</span>, <span class="green">the Emperor</span>, <span class="green">the princ
e</span>, <span class="green">Prince Vasili</span>, <span class="green">Dowager
Empress Marya Fedorovna</span>, <span class="green">the baron</span>, <span clas
s="green">Anna Pavlovna</span>, <span class="green">the Empress</span>, <span cl
ass="green">the Empress</span>, <span class="green">Anna Pavlovna's</span>, <spa
n class="green">Her Majesty</span>, <span class="green">Baron
Funke</span>, <span class="green">the prince</span>, <span class="green">Anna
Pavlovna</span>, <span class="green">the Empress</span>, <span class="green">The
prince</span>, <span class="green">Anatole</span>, <span class="green">the prin
ce</span>, <span class="green">The prince</span>, <span class="green">Anna
Pavlovna</span>, <span class="green">Anna Pavlovna</span>]
```



```
Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
strud exercitacion ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis
aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugi
at nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa
qui officia deserunt mollit anim id est laborum.
</div>
</body>
</html>

>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
A Useful Page
>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
[<span class="green">Anna
Pavlovna Scherer</span>, <span class="green">Empress Marya
Fedorovna</span>, <span class="green">Prince Vasili Kuragin</span>, <span class="
green">Anna Pavlovna</span>, <span class="green">St. Petersburg</span>, <span c
lass="green">the prince</span>, <span class="green">Anna Pavlovna</span>, <span
class="green">Anna Pavlovna</span>, <span class="green">the prince</span>, <span
class="green">the prince</span>, <span class="green">the prince</span>, <span c
lass="green">Prince Vasili</span>, <span class="green">Anna Pavlovna</span>, <sp
an class="green">Anna Pavlovna</span>, <span class="green">the prince</span>, <S
pan class="green">Wintzingerode</span>, <span class="green">King of Prussia</spa
n>, <span class="green">le Vicomte de Mortemart</span>, <span class="green">Mont
morencys</span>, <span class="green">Rohans</span>, <span class="green">Abbe Mor
io</span>, <span class="green">the Emperor</span>, <span class="green">the princ
e</span>, <span class="green">Prince Vasili</span>, <span class="green">Dowager
Empress Marya Fedorovna</span>, <span class="green">the baron</span>, <span clas
s="green">Anna Pavlovna</span>, <span class="green">the Empress</span>, <span cl
ass="green">the Empress</span>, <span class="green">Anna Pavlovna's</span>, <spa
n class="green">Her Majesty</span>, <span class="green">Baron
Funke</span>, <span class="green">the prince</span>, <span class="green">Anna
Pavlovna</span>, <span class="green">the Empress</span>, <span class="green">The
prince</span>, <span class="green">Anatole</span>, <span class="green">the prin
ce</span>, <span class="green">The prince</span>, <span class="green">Anna
Pavlovna</span>, <span class="green">Anna Pavlovna</span>]

>>> li = ["A", "B", "C"]
>>> li[1]
'B'
>>>
```

# Get 1<sup>st</sup> set of data & last (-1)

```
scapeweppage.py - C:/Users/user/Desktop/scapeweppage.py (3.6.4)
File Edit Format Run Options Window Help
from bs4 import BeautifulSoup
import requests

file = requests.get("http://pythonscraping.com/pages/warandpeace.html")
html = BeautifulSoup(file.text, 'html.parser')

nameList = html.findk"span", {"class": "green"})
print(nameList)

Ln: 7 Col: 20

Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
</div>
</body>
</html>
>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
A Useful Page
>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
[<span class="green">Anna Pavlovna Scherer</span>, <span class="green">Empress Marya Fedorovna</span>, <span class="green">Prince Vasili Kuragin</span>, <span class="green">Anna Pavlovna</span>, <span class="green">St. Petersburg</span>, <span class="green">the prince</span>, <span class="green">Anna Pavlovna</span>, <span class="green">the prince</span>, <span class="green">Anna Pavlovna</span>, <span class="green">the prince</span>, <span class="green">the prince</span>, <span class="green">Prince Vasili</span>, <span class="green">Anna Pavlovna</span>, <span class="green">Anna Pavlovna</span>, <span class="green">the prince</span>, <span class="green">Wintzingerode</span>, <span class="green">King of Prussia</span>, <span class="green">le Vicomte de Mortemart</span>, <span class="green">Montmorencys</span>, <span class="green">Rohans</span>, <span class="green">Abbe Morello</span>, <span class="green">the Emperor</span>, <span class="green">the prince</span>, <span class="green">Prince Vasili</span>, <span class="green">Dowager Empress Marya Fedorovna</span>, <span class="green">the baron</span>, <span class="green">Anna Pavlovna</span>, <span class="green">the Empress</span>, <span class="green">Anna Pavlovna's</span>, <span class="green">Her Majesty</span>, <span class="green">Baron Funke</span>, <span class="green">The prince</span>, <span class="green">Anna Pavlovna</span>, <span class="green">the Empress</span>, <span class="green">The prince</span>, <span class="green">Anatole</span>, <span class="green">the prince</span>, <span class="green">Anna Pavlovna</span>, <span class="green">Anna Pavlovna</span>]
>>> li = ["A", "B", "C"]
>>> li[1]
'B'
>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
<span class="green">Anna Pavlovna Scherer</span>
>>>

Ln: 75 Col: 1200
```

# Len of a phrase (number of occurrence)

The screenshot shows a web browser window displaying a page with text. Below the browser, a Python script is open in a text editor, and a Python 3.6.4 Shell window shows the execution of the script with several errors.

**Browser Content:**

10 Surprising Benefits of... 一个下午教你用Python x A Useful Page x pythonscraping.com/p...  
pythonscrapping.com/pages/warandpeace.html  
...ump or patermity.  
"Don't joke; I mean to have a serious talk with you. Do you know I am dissatisfied with your younger son? Between ourselves" (and her face assumed its melancholy expression), "he was mentioned at Her Majesty's and you were pitied..."  
The prince answered nothing, but she looked at him significantly, awaiting a reply. He frowned.  
"What would you have me do?" he said at last. "You know I did all a father could for their education, and they have both turned out fools. Hippolyte is at least a quiet fool, but Anatole is an active one. That is the only difference between them." He said this smiling in a way more natural and animated than usual, so that the wrinkles round his mouth very clearly revealed something unexpectedly coarse and unpleasant.  
"And why are children born to such men as you? If you were not a father there would be nothing I could reproach you with," said Anna Pavlovna, looking up pensively.  
"I am your faithful slave and to you alone I can confess that my children

**Python Script (scapeweppage.py):**

```
from bs4 import BeautifulSoup
import requests

file = requests.get("http://pythonscraping.com/pages/warandpeace.html")
html = BeautifulSoup(file.text, 'html.parser')
nameList = html.find(text='Anna Pavlovna')
print(len(nameList))
```

**Python Shell Output:**

```
File Edit Shell Debug Options Window Help
span class="green">Wintzingerode</span>, <span class="green">King of Prussia</span>, <span class="green">le Viconte de Mortemart</span>, <span class="green">Montmorencys</span>, <span class="green">Rohans</span>, <span class="green">Abbe Morio</span>, <span class="green">the Emperor</span>, <span class="green">the prince</span>, <span class="green">Prince Vasili</span>, <span class="green">Dowager Empress Marya Fedorovna</span>, <span class="green">the barons</span>, <span class="green">Anna Pavlovna</span>, <span class="green">the Empress</span>, <span class="green">Anna Pavlovna's</span>, <span class="green">Her Majesty</span>, <span class="green">Baron Funke</span>, <span class="green">The prince</span>, <span class="green">Anna Pavlovna</span>, <span class="green">the Empress</span>, <span class="green">The prince</span>, <span class="green">Anatole</span>, <span class="green">the prince</span>, <span class="green">The prince</span>, <span class="green">Anna Pavlovna</span>]
>>> li = ["A", "B", "C"]
>>> li[1]
'B'
>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
<span class="green">Anna Pavlovna Scherer</span>
>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
le Viconte de Mortemart
>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
Traceback (most recent call last):
  File "C:/Users/user/Desktop/scapeweppage.py", line 8, in <module>
    print(len(nameList))
NameError: name 'len' is not defined
>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
Traceback (most recent call last):
  File "C:/Users/user/Desktop/scapeweppage.py", line 8, in <module>
    print(len(nameList))
AttributeError: 'builtin_function_or_method' object has no attribute 'nameList'
>>>
===== RESTART: C:/Users/user/Desktop/scapeweppage.py =====
13
>>> |
Ln: 104 Col: 4
```

# Stock closing price (1)

The screenshot shows a web browser window displaying the Yahoo! stock page for TSMC (2330). The browser's developer tool is open, showing the HTML structure of the page. The table structure for the closing price is as follows:

```
<table border="2" width="750">
  <tbody>
    <tr>
      <td align="center" width="105">
      <td align="center" bgcolor="#FFFFFF" nowrap:14:30</td>
      <td align="center" bgcolor="#FFFFFF" nowrap:</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:</td>
      <td align="center" bgcolor="#FFFFFF" nowrap:25,682</td>
      <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" width="137" class="tt">
    </tr>
  </tbody>
</table>
```

The table in the browser shows the following data for TSMC (2330):

股票代號	時間	成交	買進	賣出	漲跌	張數	昨收	開盤
2330台積電	14:30	259.5	259.0	259.5	0.00	25,682	259.5	259.5

The developer tool also shows the CSS styles for the table cell:

```
white-space: -webkit-nowrap;
display: table-cell;
vertical-align: inherit;
```

# 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)

Choose the specific item-to-be-crawled by Python

The screenshot shows a web browser displaying a stock price table for '2330 台積電'. The table has columns for '股票代號', '時間', '成交', '買進', '賣出', '漲跌', '張數', and 'td'. The closing price is 259.5. The developer tools show the HTML structure of the table, with a context menu open over a specific cell containing the closing price '259.5'. The context menu includes options like 'Copy', 'Copy selector', and 'Copy XPath'. A green arrow points from the text 'Choose the specific item-to-be-crawled by Python' to the selected cell in the table.

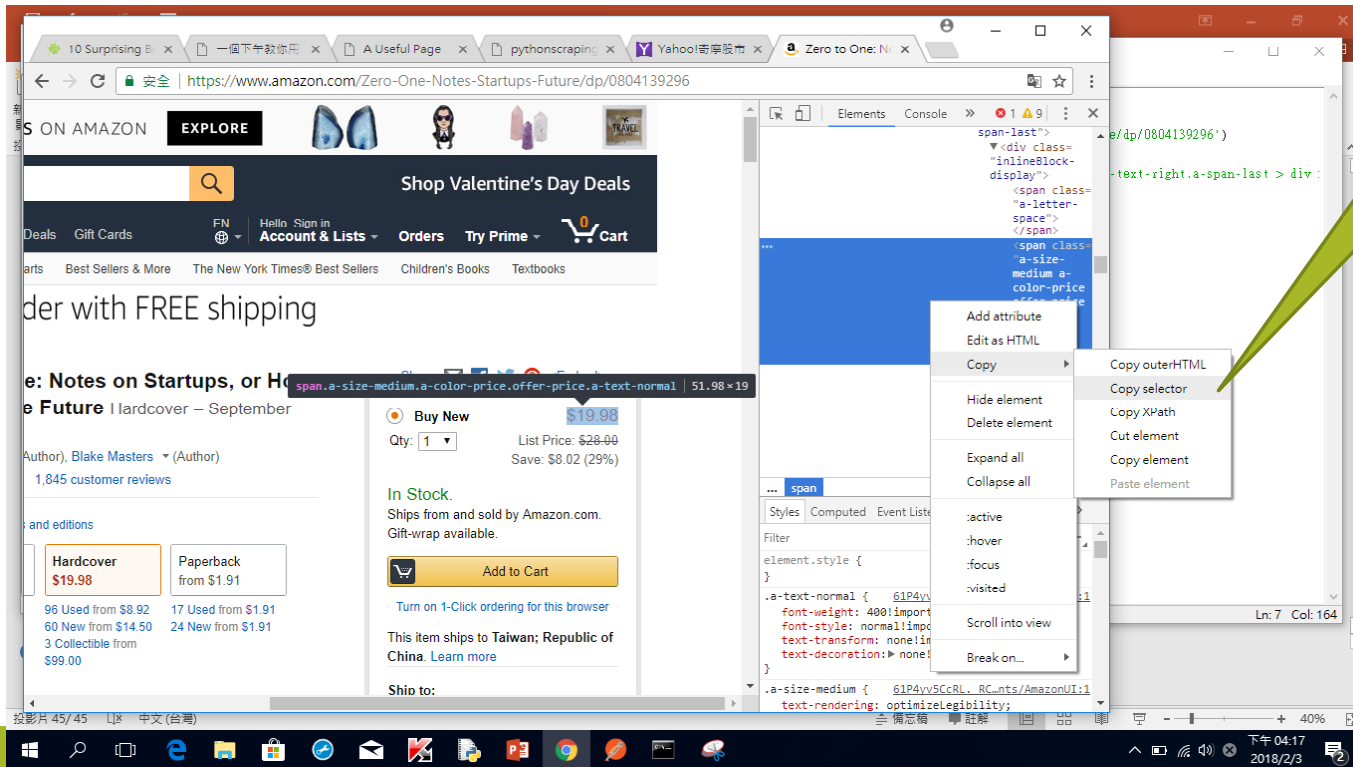
股票代號	時間	成交	買進	賣出	漲跌	張數	td
2330 台積電	14:30	259.5	259.0	259.5	0.00	25,682	259.5



# 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

Choose the specific item-to-be-crawled by Python



# Book price (2)

```
scapewepage.py - C:/Users/user/Desktop/scapewepage.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="#ffff01">
<th align="center">股票</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">漲數</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"><a href="/q/bc?s=2330">2330台積電</a><br/><a href
="/pf/pfisel?sstocklist=2330;"><font size="-1">加到投資組合</font><br/></a></td>
<td align="center" bgcolor="#FFFFFF" nowrap="">14:30</td>
<td align="center" bgcolor="#FFFFFF" nowrap=""><b>259.5</b></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=""><font color="#000000">0.00
<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">
<a href="/q/ts?s=2330">成交明細</a><br/><a href="/q/ta?s=2330">技術</a> <a href
="/q/h?s=2330">新聞</a><a href="/d/s/company_2330.html"><br/>基本</a> <a href="
/d/s/credit_2330.html">籌碼</a><br/><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</a></td></font></td>
</tr>
</table>
>>>
===== RESTART: C:/Users/user/Desktop/scapewepage.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='價股資料')[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 Window
<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><br>
"/q/h?s=2330">新聞</a><a href="/d/s/
/d/s/credit_2330.html">籌碼</a><br><
ock/other/STG=125v47s73/**https://tw.
tal?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/scapeweb.py", line 10, in <module>
    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/scapeweb.py", line 10, in <module>
    sheet = workbook.get_sheet_by_name('TW2330')
DeprecationWarning: Call to deprecated method get_sheet_by_name()
>>>
```

日期	收盤價
2018/1/11	235
2018/1/12	237
2018/01/13	235
2018/01/14	235
2018/01/20	248.5
2018/01/21	248.5
2018/01/22	255.5
2018/01/23	266
2018/01/24	258
2018/01/25	258
2018/01/26	255
2018/01/27	258
2018/01/28	258
2018/01/29	258.5
2018/01/30	253
2018/01/31	255
2018/2/1	259.5
2018/02/02	259.5
2018/2/1	259.5

# Input data to excel --- for all stock

```
import time
import os
import openpyxl

def getStockPriceByld(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:
    getStockPriceByld(stockID, workbook)
workbook.save("stock_price_data.xlsx")
```

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

def getStockPriceByld(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:
    getStockPriceByld(stockID, workbook)
```

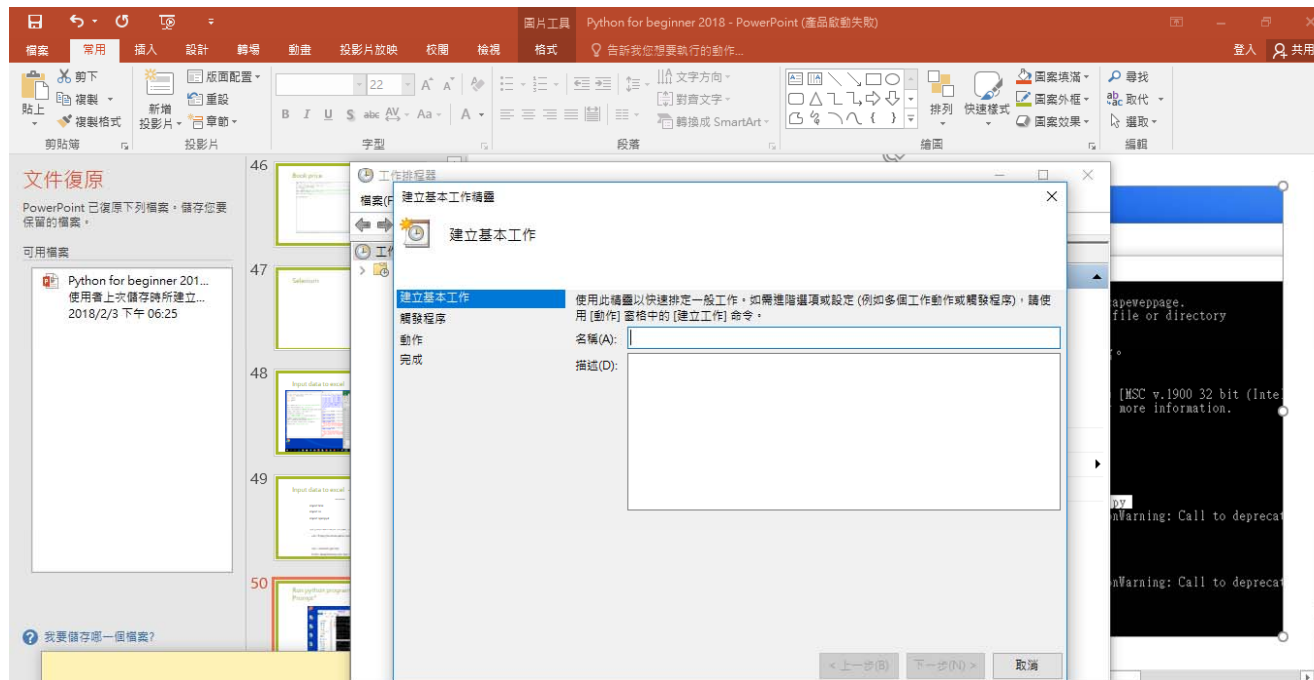
Source, URL:

<https://gist.github.com/yuyueugene84/80bb61ebb3762c97d642ecb935e3ca84>

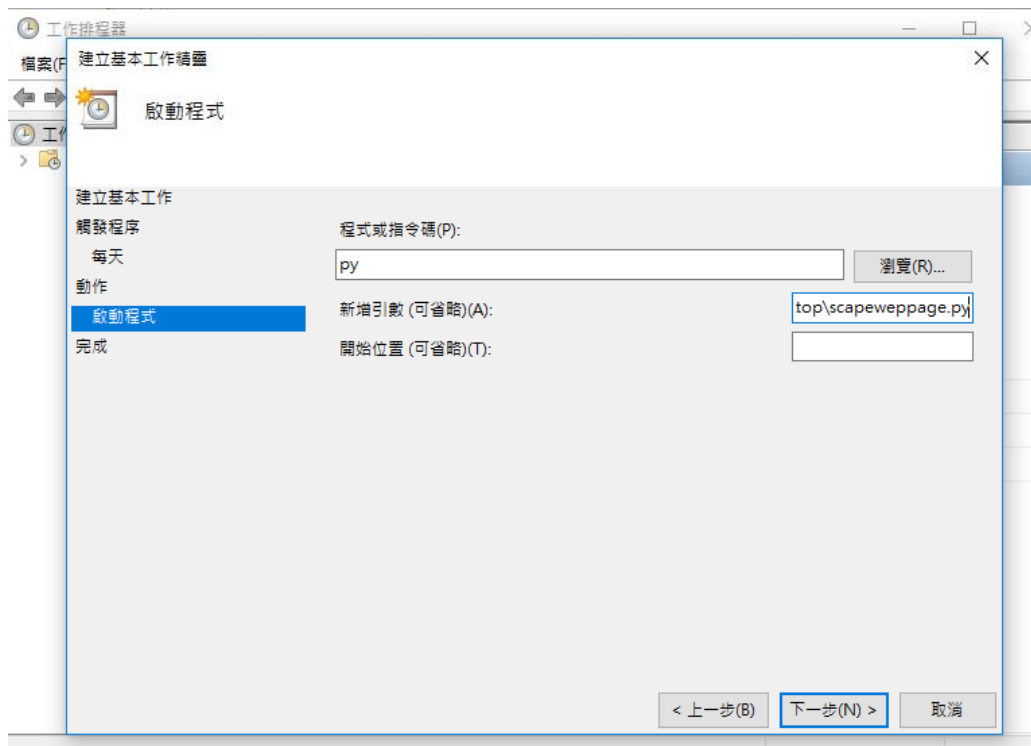
# Run Python program from “Command Prompt”

```
CA 選取 命令提示字元
C:\Users\user>python python C:\Users\user\Desktop\scapewepage.
python: can't open file 'python': [Errno 2] No such file or directory
C:\Users\user>C:\Python 2018\python-3.6.4.exe
'C:\Python' 不是內部或外部命令、可執行的程式或批次檔。
C:\Users\user>py
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)]
Type "help", "copyright", "credits" or "license()" for more information.
>>>
KeyboardInterrupt
>>>
KeyboardInterrupt
>>> exit()
C:\Users\user>py C:\Users\user\Desktop\scapewepage.py
C:\Users\user\Desktop\scapewepage.py:84: DeprecationWarning: Call to deprecated method get_sheet_by_name(sheetName).
  sheet = workbook.get_sheet_by_name('Portfolio')
10
2018/02/03 1301收盤價：103.0
C:\Users\user\Desktop\scapewepage.py:46: DeprecationWarning: Call to deprecated method get_sheet_by_name(sheetName).
  sheet = workbook.get_sheet_by_name(sheetName)
3
2018/02/03 1302收盤價：80.1
```

# Scheduling (1)



# Scheduling (2)



# Scheduling (3)

The screenshot displays the Windows Task Scheduler interface. The main window shows a list of scheduled tasks with columns for Name, Status, Trigger, and Next Run Time. The task 'Adobe Flash Player Updater' is selected, and its details are shown in the lower pane. The details pane includes fields for Name, Location, Author, and Description. The Description states: '這個工作會讓您的 Adobe Flash Player 安裝與最新增強功能和安全性修正保持同步。如果停用或移除這個工作，Adobe Flash Player 將無法自動透過最新安全性修正保護您的電腦。' The Security Options section is also visible, showing 'SYSTEM' as the user and the option 'Only run when user is logged on' is selected.

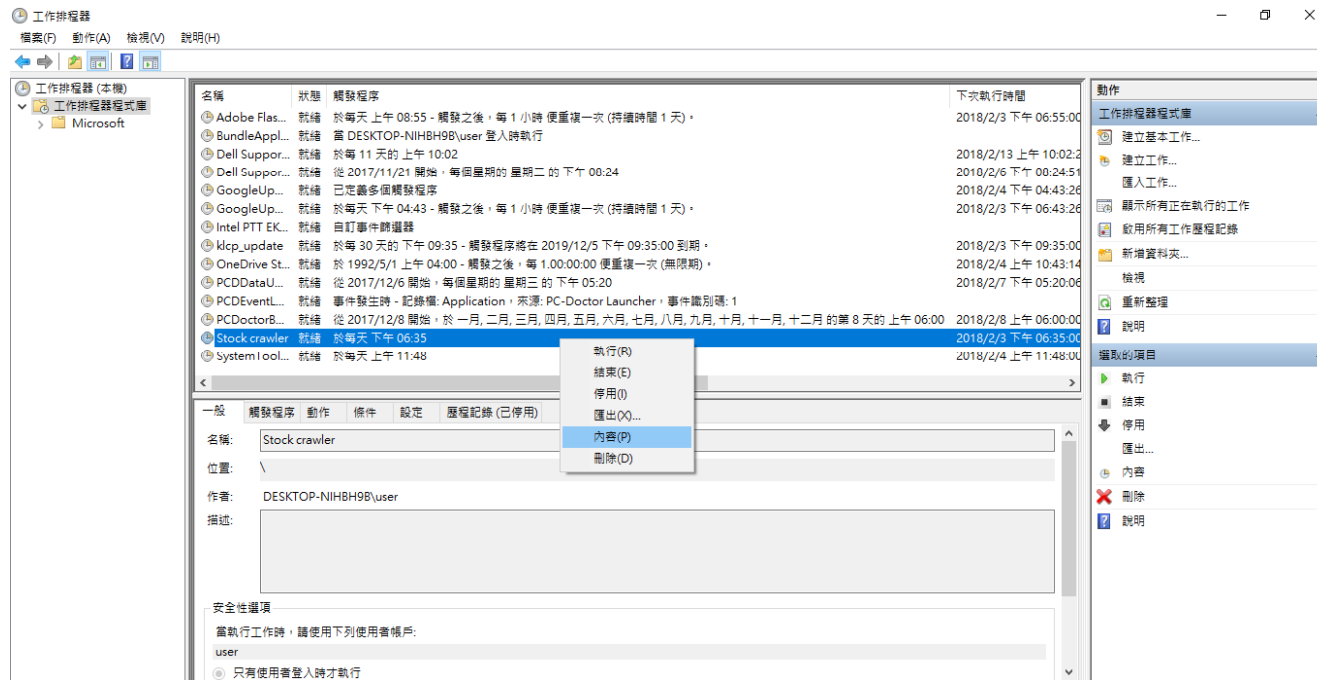
名稱	狀態	觸發程序	下次執行時間
Adobe Flas...	就緒	於每天 上午 08:55 - 觸發之後，每 1 小時 便重複一次 (持續時間 1 天)。	2018/2/3 下午 06:55:00
BundleAppl...	就緒	當 DESKTOP-NIHBH98\user 登入時執行	
Dell Suppor...	就緒	於每 11 天的 上午 10:02	2018/2/13 上午 10:02:2
Dell Suppor...	就緒	從 2017/11/21 開始，每個星期的 星期二的 下午 08:24	2018/2/6 下午 08:24:5
GoogleUp...	就緒	已定義多個觸發程序	2018/2/4 下午 04:43:26
GoogleUp...	就緒	於每天 下午 04:43 - 觸發之後，每 1 小時 便重複一次 (持續時間 1 天)。	2018/2/3 下午 06:43:26
Intel PTT EK...	就緒	自訂事件觸發器	
kicp_update	就緒	於每 30 天的 下午 09:35 - 觸發程序將在 2019/12/5 下午 09:35:00 到期。	2018/2/3 下午 09:35:00
OneDrive St...	就緒	於 1992/5/1 上午 04:00 - 觸發之後，每 1.00:00:00 便重複一次 (無限期)。	2018/2/4 上午 10:43:14
PCDDataU...	就緒	從 2017/12/6 開始，每個星期的 星期三的 下午 05:20	2018/2/7 下午 05:20:06
PCDEventL...	就緒	事件發生時。記錄檔: Application，來源: PC-Doctor Launcher，事件識別碼: 1	
PCDoctorB...	就緒	從 2017/12/8 開始，於 一月, 二月, 三月, 四月, 五月, 六月, 七月, 八月, 九月, 十月, 十一月, 十二月 的 第 8 天的 上午 06:00	2018/2/8 上午 06:00:00
Stock crawler	就緒	於每天 下午 06:32	2018/2/3 下午 06:32:00
SystemTool...	就緒	於每天 上午 11:48	2018/2/4 上午 11:48:00

**Adobe Flash Player Updater 詳細資訊:**

- 名稱: Adobe Flash Player Updater
- 位置: \
- 作者: Adobe Systems Incorporated
- 描述: 這個工作會讓您的 Adobe Flash Player 安裝與最新增強功能和安全性修正保持同步。如果停用或移除這個工作，Adobe Flash Player 將無法自動透過最新安全性修正保護您的電腦。
- 安全性選項: 當執行工作時，請使用下列使用者帳戶: SYSTEM
- 只有使用者登入時才執行



# Scheduling (4)



# Scheduling (5)

The screenshot displays the Windows Task Scheduler interface. The main window shows a list of tasks, including Adobe Flash Player updates, Dell support updates, and a 'Stock crawler' task. The 'Stock crawler' task is selected, and its properties are shown in the lower pane. The 'Conditions' tab of the 'Stock crawler' task is open, showing the following settings:

- 指定與續發程式共同決定是否應該執行工作的條件:**  只有電腦開置下列時間後才啟動工作(C): 10 分鐘
- 等待開置時間(A):** 1 小時
- 如果電腦不再開置即停止(E)
- 如果再度開置就重新啟動(U)
- 開啟/關閉**
- 只有在電腦是使用 AC 電源時才啟動這個工作(P)
- 如果電腦切換到使用電池時即停止(B)
- 喚醒電腦以執行此工作(W)
- 網路**
- 只有在下列網路連線可以使用時才啟動(V): 任何連線

The 'Security options' section at the bottom indicates that the task should only be executed when the user is logged on.