7

Python
File I/O, os,
shutil, glob

open()

```
open(file_name[, mode][, encode])
```

- In Python, you can use this build-in method to open a file, and operate it.

- **mode**

r	Read mode. The cursor will be put at the beginning. This is the default mode.
r+	Read and write mode. The cursor will be put at the beginning.
w	Write mode. If the file isn’t created before, it will create a new file. If the file has already existed, it will put the cursor at the beginning which will overwrite the file.
w+	Read and write mode. If the file isn’t created before, it will create a new file. If the file has already existed, it will put the cursor at the beginning which will overwrite the file.
a	Write mode. If the file isn’t created before, it will create a new file. If the file has already existed, it will put the cursor at the end and continually adding new stuff.
a+	Read and write mode. If the file isn’t created before, it will create a new file. If the file has already existed, it will put the cursor at the end and continually adding new stuff.

open()

```
open(name[, mode][, encode])
```

- All the modes can be opened with binary format. You can add “b” at the end of each mode English name. Ex: rb, rb+, wb, wb+, ab, ab+

- Encoding: There’re some different encoding formats in Windows such as **cp950(Default in Chinese Windows), UTF-8…**
- **Since the most popular format is UTF-8, Linux use this format standard, too. so recommend using UTF-8 format.**

- **Ex**

```
f = open('test.txt', 'r' , encoding = 'UTF-8')
```

File operation method

read([size])	Read for specific(default is all) length of the character from the file.
readline()	Read specific (default is one) lines from the current cursor position; a newline character (\n) is appended to the end of the string.
readlines()	Read all lines from the file. It will create a list.
next()	Put the cursor to the next line.
write(str)	Write the string to the file. #must be string!!
seek(offset[, whence])	Put the cursor with offset position. Whence: 0: Count from the beginning (the default mode) 1: the current cursor position 2: from the end of the file
tell()	Return the current cursor position
close()	Close the file. After closing the file, you can’t operate the file anymore until you open it again.

open()

In test.txt:

```
Hello World!
Hello
World
!
```

-

```
f=open('test.txt','r',encoding='UTF-8’)
print(f.tell())
print(f.read(7))
f.seek(0)
print(f.readline())
print(f.tell())
print(f.readline())
print(f.readline())
print(f.readline())
f.seek(0)
x = f.readlines()
print(x)
f.close()
```

```
0
Hello W
Hello World!

14
Hello

World

!
['Hello World!\n', 'Hello\n', 'World\n', '!']
```

with open(...) as file_name:

- In Python, you can use file i/o with a function-like format. It will automatically close the file after running all the codes in the code section

```
with open(…) as file_name:
    #code
```

- Ex

```
with open('test.txt','r',encoding='UTF-8') as f:
    for line in f:
        print(line, end = '')
```

```
Hello World!
Hello
World
!
```

os

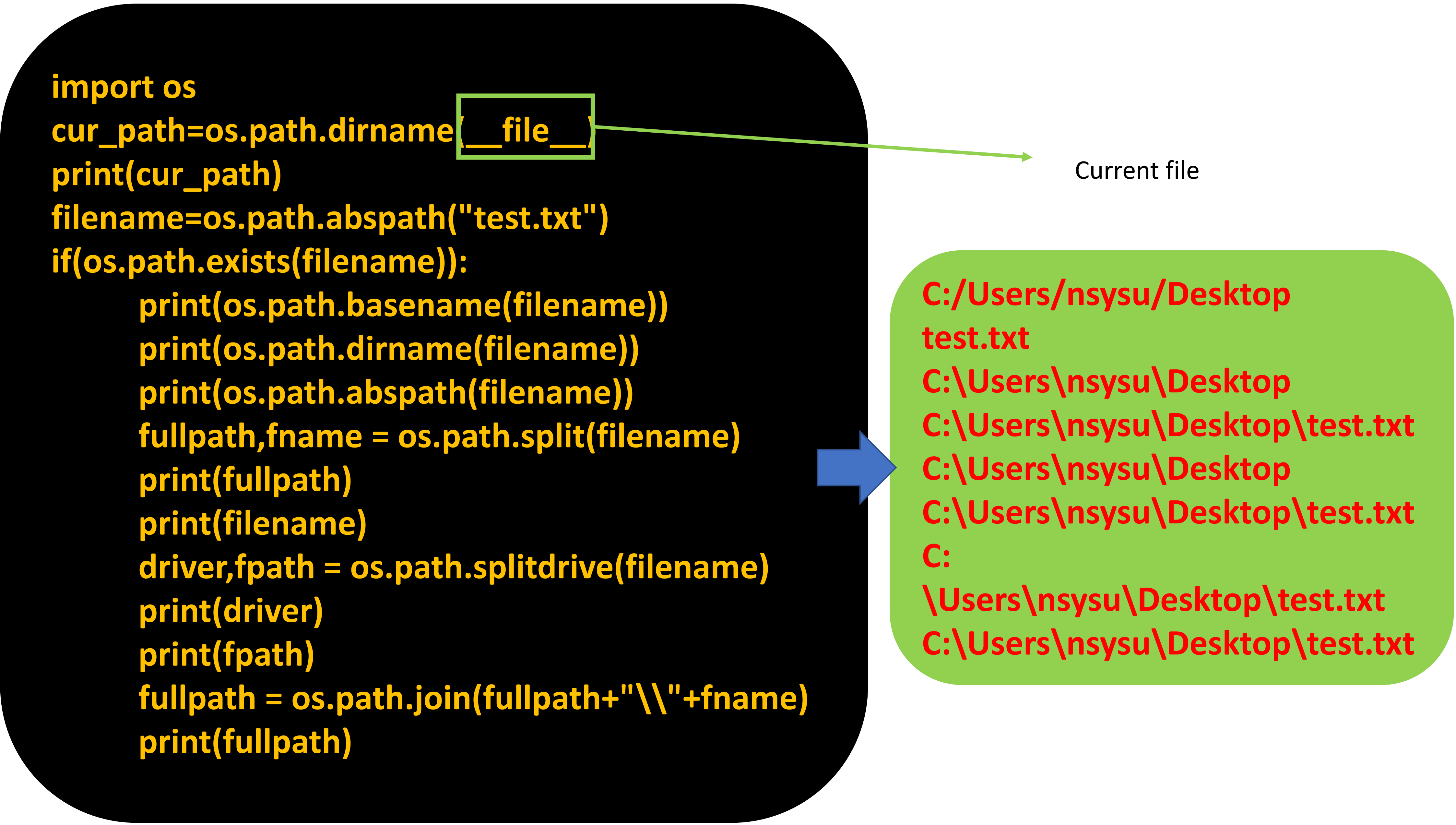
- **os** module allows you to easily create/delete path, and delete specific file, or even run the shell commands.

- **You have to** `import os` **before using it.**

os.path

- os.path is for checking file/path is existed or not, seeing file/path information, and operating the file paths.

abspath()	Get the absolute pathname.
basename()	Get the base name of the pathname.
dirname()	Get the directory name of the pathname. dirname(__file__) can check the current directory name.
exists()	Check whether the file is existed or not.
getsize	Get the path size in Byte.
split()	Split the pathname path into a pair, (head, tail) where tail is the last pathname component and head is everything leading up to that.
splitdrive()	Split the pathname *path* into a pair (drive, tail) where head is the driver name, and tail is rest pathname.
join()	Merge one or more path components together.



```
import os
cur_path=os.path.dirname(__file__)
print(cur_path)
filename=os.path.abspath("test.txt")
if(os.path.exists(filename)):
        print(os.path.basename(filename))
        print(os.path.dirname(filename))
        print(os.path.abspath(filename))
        fullpath,fname = os.path.split(filename)
        print(fullpath)
        print(filename)
        driver,fpath = os.path.splitdrive(filename)
        print(driver)
        print(fpath)
        fullpath = os.path.join(fullpath+"\\"+fname)
        print(fullpath)
```

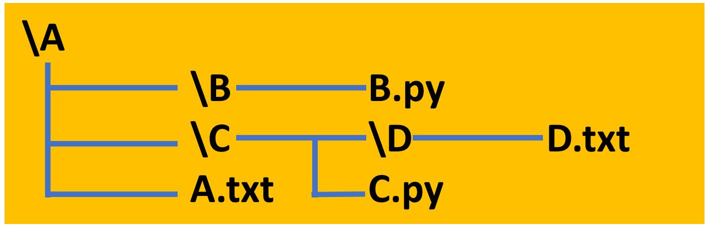
Current file

```
C:/Users/nsysu/Desktop
test.txt
C:\Users\nsysu\Desktop
C:\Users\nsysu\Desktop\test.txt
C:\Users\nsysu\Desktop
C:\Users\nsysu\Desktop\test.txt
C:
\Users\nsysu\Desktop\test.txt
C:\Users\nsysu\Desktop\test.txt
```

os.walk()

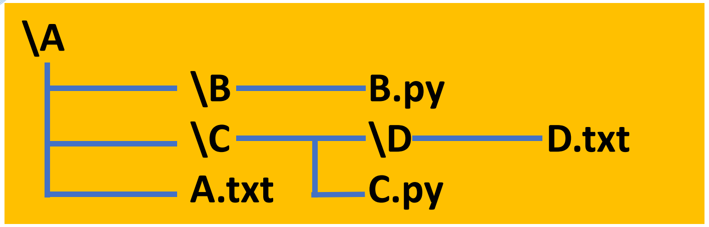
- os.walk() allow you to search specific directory, and its sub-directory. It will return a(or multiple) tuple(s) with 3 elements which are dirpath, dirnames, filenames. This function can be done recursively.

- **Example**
- **if we have a folder whose structure was:**



\A
\B
B.py
\C
\D
D.txt
A.txt
C.py

os.walk()



\A
\B
B.py
\C
\D
D.txt
C.py
A.txt

```
import os
cur_path=os.path.dirname(__file__)
filename = os.path.abspath("A")
s= os.walk(filename)
for dirpath,dirnames,filenames in s:
        print(dirpath)
        print(dirnames)
        print(filenames,'\n')
```

```
C:\Users\ihors\Desktop\A
['B', 'C']
['A.txt']

C:\Users\ihors\Desktop\A\B
[]
['B.py']

C:\Users\ihors\Desktop\A\C
['D']
['c.py']

C:\Users\ihors\Desktop\A\C\D
[]
['d.txt']
```

os.remove()

- os.remove() can remove specific file. You can use this function with os.path.exists() to check whether the file is existed or not.

- Ex

```
import os
file = "test.txt"
if os.path.exists(file):
    os.remove(file)
    print("Remove successfully.")
else:
    print("Failed to remove file.")
```

os.mkdir()

- os.mkdir() can create specific directory. You can use this function with os.path.exists() to check whether the directory has already existed or not.

- Ex

```
import os
dir = "NEW"
if not os.path.exists(dir):
        os.mkdir(dir)
        print("Create successfully.")
else:
        print(dir , "has already existed.")
```

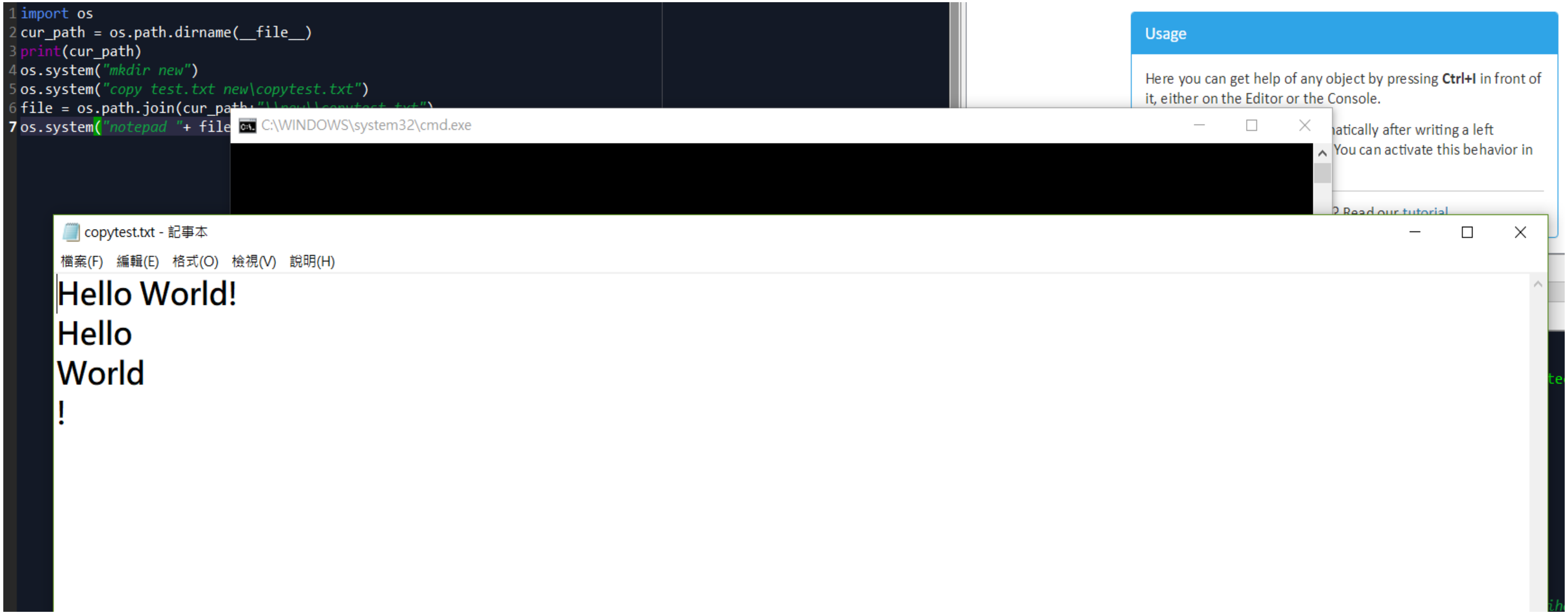
os.system()

- os.system() allow you to run shell commands.

- **Ex**

```
import os
cur_path = os.path.dirname(__file__)
os.system("mkdir new") #create a folder “new”
os.system("copy test.txt new\copytest.txt")  #create test.txt to new\copytest.txt
file = os.path.join(cur_path, "new", "copytest.txt")
os.system("notepad "+ file) #use notepad to open copytest.txt
```

os.system()



```python
import os
cur_path = os.path.dirname(__file__)
print(cur_path)
os.system("mkdir new")
os.system("copy test.txt new\copytest.txt")
file = os.path.join(cur_path,"\\new\\copytest.txt")
os.system("notepad "+ file
```

C:\WINDOWS\system32\cmd.exe

copytest.txt - 記事本

```
Hello World!
Hello
World
!
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

atically after writing a left

You can activate this behavior in

? Read our tutorial

shutil

- shutil module can easily allow you to copy, delete, or move the file ordirectory.
- **You have to** `import shutil` **before using it.**

shutil

- shutil module can easily allow you to copy, delete, or move the file ordirectory.
- **You have to `import shutil` before using it.**

copy(src, dst)	Copy file src to file dst
copytree(src, dst)	Recursively copy an entire directory tree rooted at *src* to another location *dst*
rmtree(dir)	Delete an entire directory tree dir
move(src, dst)	Recursively move a directory *src* to another location *dst*

shutil

copy(src, dst)	Copy file src to file dst
copytree(src, dst)	Recursively copy an entire directory tree rooted at *src* to another location *dst*
rmtree(dir)	Delete an entire directory tree dir
move(src, dst)	Recursively move a directory *src* to another location *dst*

```
import os, shutil
destfile = os.path.join(os.path.dirname(__file__), "new", "copytest.txt")
shutil.copy("test.txt", destfile)
```

glob

- glob module can easily allow you to search for the specific directory or file.
- **You have to** `import glob` **before using it.**
- `glob.glob(pathname[,recursive = False])` will return a list which match pathname
- **Pathname support some special word like '?' can replace one word, '*' can replace multiples words, [0-9] can replace one number, [a-z] or [A-Z] can replace one alphabet(Not case sensitive).**

glob

```
import glob
finding = glob.glob("a.py") + glob.glob("?.txt") + glob.glob("?.py") + glob.glob("[0-9].*")
for file in finding:
        print(file)
```

```
a.py
a.txt
1.txt
1.py
12345.py
1.pdf
2.gif
```

8

Python Exception

Exception

- Just like C++. Python has exception, too.

```
try:
    #code
except exptiontype1:
    #code
except exptiontype2:
    # code
except:
   # code
else:
    #code
finally:
     #code
```

```
try:
    a=int(input("Input dividend: "))
    b=int(input("Input divisor: "))
    c=a/b
    print(a,"/",b,"= ",end="")
except ZeroDivisionError:
    print("Division by 0!")
except ValueError:
    print("Please input a valid integer numbers!")
else:
    print(c)
```

Exception

- The format of “except” is similar to if, elif, else. The one with no exception type name is for those exception not mentioned before.
- **Else** will be executed if there’s no exception occurred.

- And **finally** will be executed no matter what. It would be the end of the try…except block.

- You can put multiple exception types in one exception line.
- Ex

```
except ZeroDivisionError, ValueError:
    print("ZeroDivisionError or ValueError occured!")
```

Exception as

```
except exptiontype as name:
    #code
```

- You can add “as name”, then variable name will be bound to an exception instance with the arguments stored in instance.args which has some information about the occurred exception.
- Ex:

```
except ZeroDivisionError as errormessage:
    print(type(errormessage))
    print(errormessage)
```

```
<class 'ZeroDivisionError’>
division by zero
```

Pass

- If you want the program don’t do anything when the specific exception occurs, then you can add `pass` in the code section to make it continue.
- Ex

```
a, b = 5, 0
try:
    c=a/b
    print(a,"/",b,"= ",end="")
except ZeroDivisionError:
    pass
else:
    print(c)
```

No output in this example

Raise exception

```
raise exceptiontype [, args] [, traceback]
```

- You can raise exception manually by using raise function
- Ex:

```
try:
    raise RuntimeError('errorargs', 'errorargstraceback')
except Exception as errormessage:
    print(type(errormessage))
    print(errormessage.args)
    print(errormessage)
```

```
<class 'RuntimeError'>
('errorargs', 'errorargstraceback')
('errorargs', 'errorargstraceback')
```

9

Python Pandas

Overview

- **Pandas** is a powerful module which is good at data analysis and processing. For example, it supports read in SQL or spreadsheet data, etc. In addition, it also provides data filtration, reshape, merging , insertion, etc.

- It supports ***DataFrame*** object for data manipulation with integrated indexing.

- Recommend using `import pandas as pd` before using pandas module.

Data in ‘test.xlsx’ for example

	Math	English	Computer
A	90	90	85
B	75	77	65
C	80	97	90
D	80	55	33
E	77	20	53
F	22	80	66

Read in data function

read_table('file_name')	Read general delimited file into DataFrame
read_csv('file_name')	Read CSV (comma-separated) file into DataFrame
read_sql('file_name')	Read SQL query or database table into a DataFrame.
read_excel('file_name')	Read an Excel table into a pandas DataFrame
read_json('file_name')	Convert a JSON string to pandas object
read_html('file_name')	Read HTML tables into a list of DataFrame objects.

- Ex

```
import pandas as pd
tables = pd.read_excel('test.xlsx')
```

DataFrame.head()

- You can use `name.head(lines_number)` to get the **first lines_number row** of data

- **Ex**

```
import pandas as pd
tables = pd.read_csv('test.xlsx’)
tables.head(3)
```

```
      Math   English  Computer
A      90      90        85
B      75      77        65
C      80      97        90
```

Some useful variables/function to get the DataFrame layout

```
import pandas as pd
tables = pd.read_excel('test.xlsx')
```

		Answer
tables.**shape**	Show how many rows and columns in the dataframe	(6, 3) #6 rows, 3 columns
tables.**columns**	Show the column header names list	Index(['Math', 'English', 'Computer'], dtype='object')
tables.**index**	Show the rows header names list	Index(['A', 'B', 'C', 'D', 'E', 'F'], dtype='object')
tables.**info()**	Show the detailed dataframe information	-
tables.**describe()**	Show the whole table of the dataframe	-

Select specific columns of the DataFrame

- You can use `name[column_headername]` to get the specific columns of the DataFrame
- Ex

```
print(df["English"])

print(df[ ["Math","Computer"] ] )
```

```
     Math   Computer
A     90       85
B     75       65
C     80       90
D     80       33
E     77       53
F     22       66
```

```
     English
A      90
B      77
C      97
D      55
E      20
F      80
```

- If you want to select more than one column, then you have to add [] to make it become a list.

Select specific row range of the DataFrame

- You can use `name[row_index_range]` to get the specific row range of the DataFrame

- **Ex**

```
print(df[0:1])

print(df[3:6])
```

```
     Math  English  Computer
A     90      90        85
```

```
     Math  English  Computer
D     80      55        33
E     77      20        53
F     22      80        66
```

- Range in here couldn't only have **one** index number

- **Ex**

```
print(df[0])
```

```
Error!
```

Select specific row of the DataFrame

- You can use `name.loc[row_headername]` to get the specific row range of the DataFrame
- **Ex**

```
print(df.loc["A"])

print(df.loc[["B", "D"]])
```

```
     Math  English  Computer
A     90      90        85
```

```
     Math  English  Computer
B     75      77        65
D     80      55        33
```

- Again, if you want to select more than one row, then you have to add **[]** to make it become a list.

Select specific row of the DataFrame

- `name.loc[row_headername]` supports range syntax to select from A to B row

- **Ex**

```
print(df.loc["B" : "E" : 2 ])
```

```
    Math  English  Computer
B    75     77        65
D    80     55        33
```

Select specific row of the DataFrame

- You can use `name.loc[row_headername, [row_headername]]` to get the specific rows with specific columns of the DataFrame he DataFrame

- **Ex**

```
print(df.loc["B" : "E" : 2], ["Math", "Computer" ])
```

```
      Math   Computer
B      75       65
D     80        33
```

Select specific row of the DataFrame

- You can use `name.iloc[row_headername, [row_index]]` to get the specific rows with specific columns of the DataFrame he DataFrame

- **Ex**

```
print(df.iloc["B" : "E" : 2], [0,2] )
```

→

```
      Math   Computer
B      75       65
D     80        33
```

Modifying DataFrame data

```
name[column][row] = …
```

- **Row** can be either **number or name**! But **column** can only be **name**
- **Ex**

```
df['Math']['B':'F'] = 60
```

	Math	English	Computer
A	90	90	85
B	60	77	65
C	60	97	90
D	60	55	33
E	60	20	53
F	60	80	66

Deleting DataFrame data

```
new_name = name.drop(row_or_column_name [, axis] )
```

- Axis in default is 0(row). It can be 0(row) or 1(column)

- Ex

```
df = df.drop('Math’, 1)
```

```
     English  Computer
A       90       85
B       77       65
C       97       90
D       55       33
E       20       53
F       80       66
```

DataFrame Filter

- You can use `name[condition]` to filter out the specific columns of the DataFrame

- **Ex**

```
print(df[ df["Math"] >=80 ] )
```

```
   Math  English  Computer
A   90     90       85
C   80     97       90
D   80     55       33
```

DataFrame Filter

- If you want to have more than one condition, then you can add | (or) or & (and) to connect them
- Don’t forget to use () to separate the condition
- Ex

```
print(df[ (df["Math"] >=80 ) & (df["English"] >=60) ])
```

```
   Math  English  Computer
A   90     90        85
C   80     97        90
```

Dealing with DataFrame with Empty data

- Sometimes, there’re some rows/columns with empty data, then you can use either `dropna()` **to delete those rows/columns, or** `fillna()`

to fill them with some values.

- `name.dropna(axis = ? )` **axis = 0/‘index’ in default, 1/ ‘columns’**
- `name.dropna(value = ?)`

Dealing with DataFrame with Empty data

- Ex pd=

```
     A     B     C    D
0  NaN   2.0   NaN   0
1   3.0   4.0  NaN   1
2  NaN  NaN   NaN  5
3   0      3.0   0    4
```

```
df = df.dropna()
```

```
        A    B    C     D
3       0   3.0   0     4
```

```
df = df.fillna(0)
```

```
       A     B     C     D
0   0.0   2.0   0.0   0
1   3.0   4.0   0.0   1
2   0.0   0.0   0.0  5
3   0.0   3.0   0.0  4
```

```
df = df.dropna(1)
```

```
     D
0   0
1   1
2   5
3   4
```

10

Python Pillow

Open Image

- **Pillow** is a powerful module for photo processing!

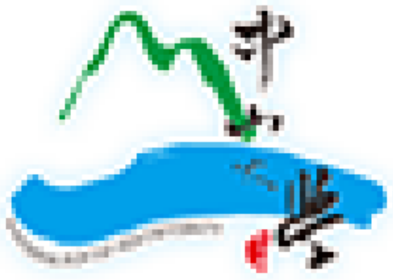
- Most of the code can be done with just “Image” package, so we can use `from PIL import Image` before using PIL module.

- When we want to demonstrate the images we’ve processed, we can use Pillow and Matplotlib simultaneously.

Import and plot

```
import matplotlib.pyplot as plt
%matplotlib inline
from PIL import Image

img = Image.open('logo.png')
plt.imshow(img)
plt.axis('off')
```



- In default, the axis is **on.**
  Or you can type “**on**”

```
plt.axis(on')
```

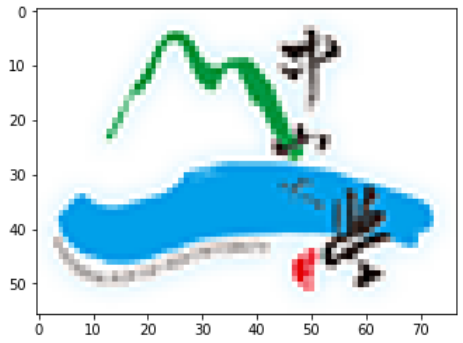


Image.convert()

- You can convert the color of the photo by using `img = img.convert('Type')`
- Popular Type: ‘**1**’, **dither** = **Image.NONE** : **black and white**
  ‘**L**’ : **greyscale**
  ‘**RGB**’ : **3x8-bit pixels, true color**
  ‘**RGBA**’ : **4x8-bit pixels, true color with transparency mask**
  ‘**CMYK**’ : **4x8-bit pixels, color separation**
- **Ex**

```
img = Image.open('logo.png')
img = img.convert('L')
plt.imshow(img)
plt.axis('off’)
```

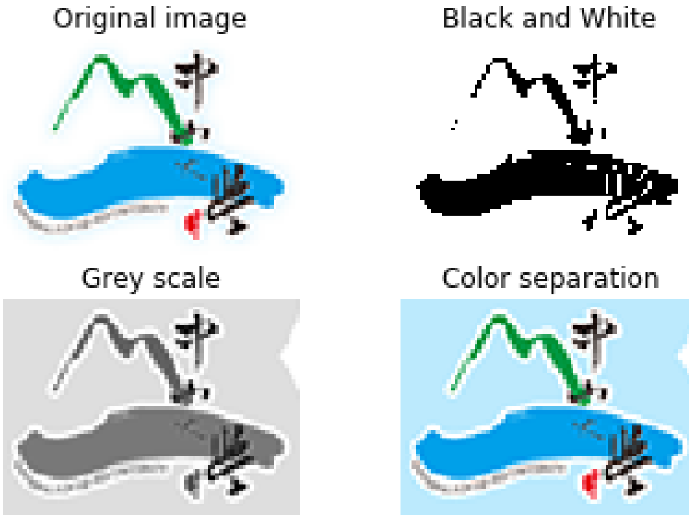
Image Subplot

```
img = Image.open('logo.png')
plt.subplot(2,2,1)
plt.imshow(img)
plt.axis('off')
plt.title('Original image')

plt.subplot(2,2,2)
img2=img.convert('1', dither = Image.NONE)
plt.imshow(img2)
plt.axis('off')
plt.title('Black and White')
```

```
plt.subplot(2,2,3)
img3=img.convert('L')
plt.imshow(img3)
plt.axis('off')
plt.title('Grey scale')

plt.subplot(2,2,4)
img4=img.convert('CMYK')
plt.imshow(img4)
plt.axis('off')
plt.title('Color separation')
```



10

Python Web crawl

Overview

- We can use **requests** package to get the HTML code of the specific website. Then use **Beautifulsoup** package to get the certain part we want. Then you can start crawling the web!

- You can use **urlopen** from the **urllib** package to open a webpage and save the content. It's handy when you want to download some images.

- Recommend using

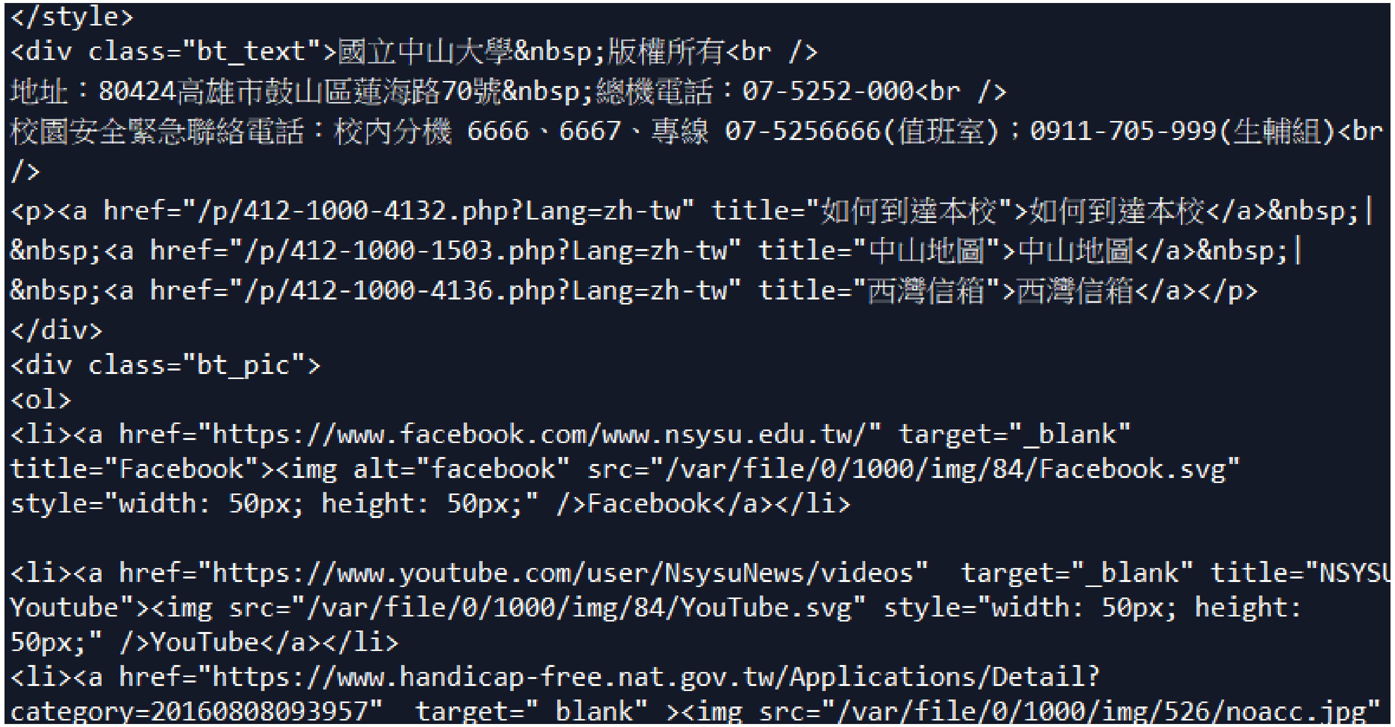
```
import requests
```

```
from bs4 import Beautifulsoup
```

```
from urllib.request import urlopen
```

requests

```python
import requests
html = requests.get('http://www.nsysu.edu.tw/')
print(html.text)
```

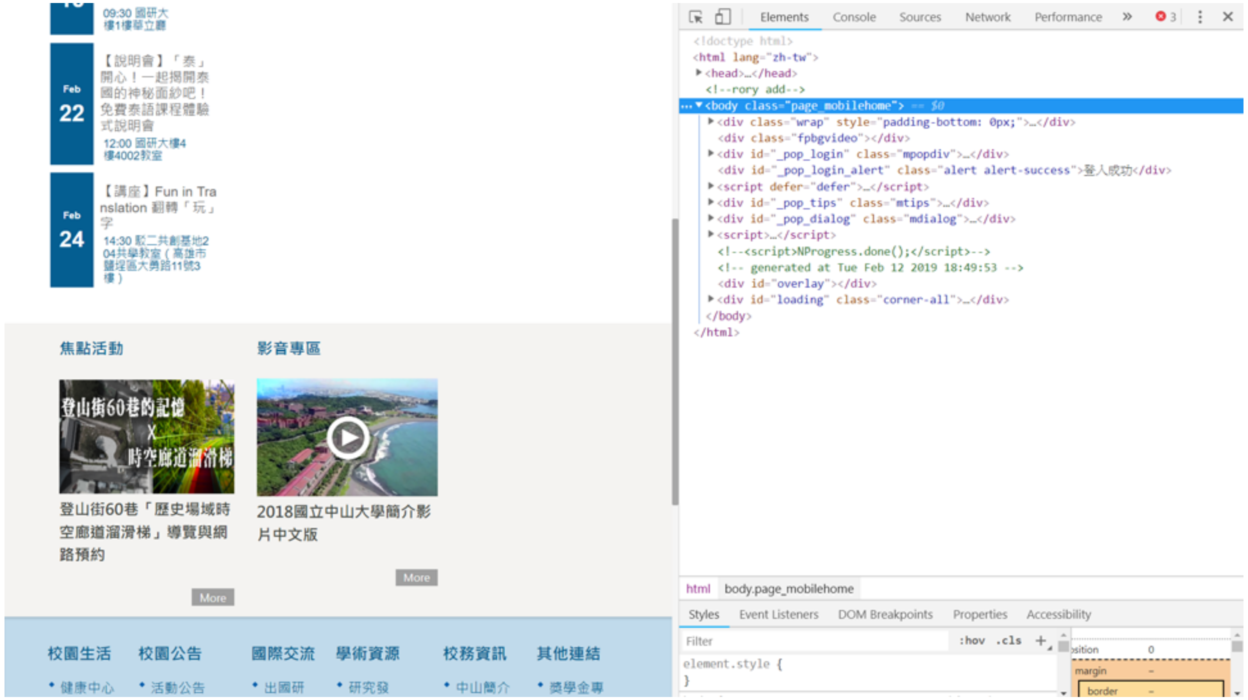


```
</style>
<div class="bt_text">國立中山大學&nbsp;版權所有<br />
地址：80424高雄市鼓山區蓮海路70號&nbsp;總機電話：07-5252-000<br />
校園安全緊急聯絡電話：校內分機 6666、6667、專線 07-5256666(值班室)；0911-705-999(生輔組)<br
/>
<p><a href="/p/412-1000-4132.php?Lang=zh-tw" title="如何到達本校">如何到達本校</a>&nbsp;|
&nbsp;<a href="/p/412-1000-1503.php?Lang=zh-tw" title="中山地圖">中山地圖</a>&nbsp;|
&nbsp;<a href="/p/412-1000-4136.php?Lang=zh-tw" title="西灣信箱">西灣信箱</a></p>
</div>
<div class="bt_pic">
<ol>
<li><a href="https://www.facebook.com/www.nsysu.edu.tw/" target="_blank"
title="Facebook"><img alt="facebook" src="/var/file/0/1000/img/84/Facebook.svg"
style="width: 50px; height: 50px;" />Facebook</a></li>

<li><a href="https://www.youtube.com/user/NsysuNews/videos"  target="_blank" title="NSYSU
Youtube"><img src="/var/file/0/1000/img/84/YouTube.svg" style="width: 50px; height:
50px;" />YouTube</a></li>
<li><a href="https://www.handicap-free.nat.gov.tw/Applications/Detail?
category=20160808093957"  target="_blank" ><img src="/var/file/0/1000/img/526/noacc.jpg"
```

Chrome F12 Web Developer Tool

- For better understanding which HTML code lines correspond to which part of the web page, we can use chrome and go to the web page you want to analyze, then press F12.



09:30 國研大樓1樓華立廳

Feb 22 【說明會】「泰」開心！一起揭開泰國的神秘面紗吧！免費泰語課程體驗式說明會 12:00 國研大樓4樓4002教室

Feb 24 【講座】Fun in Translation 翻轉「玩」字 14:30 駁二共創基地204共學教室（高雄市鹽埕區大勇路11號3樓）

焦點活動

登山街60巷「歷史場域時空廊道溜滑梯」導覽與網路預約

影音專區

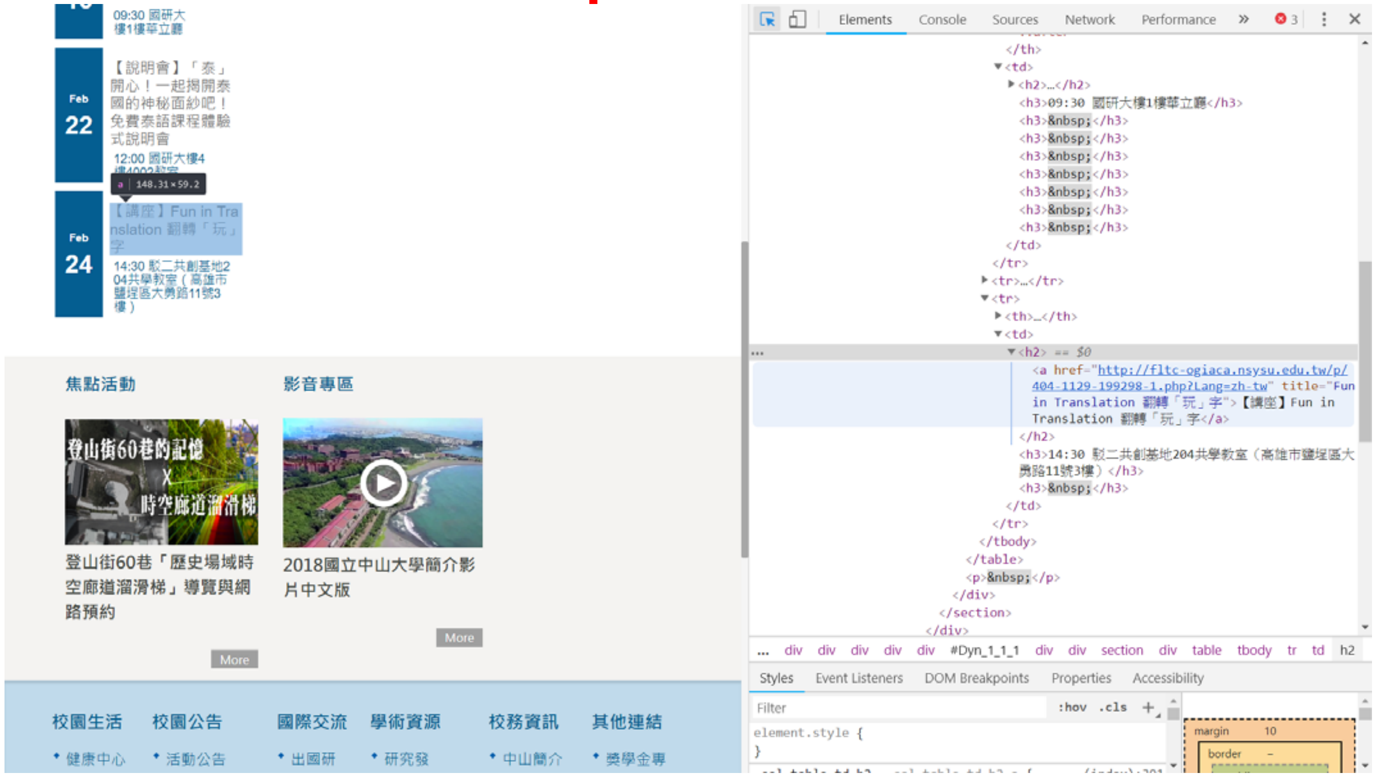
2018國立中山大學簡介影片中文版

```
<!doctype html>
<html lang="zh-tw">
 ▶<head>…</head>
  <!--rory add-->
…▼<body class="page_mobilehome"> == $0
   ▶<div class="wrap" style="padding-bottom: 0px;">…</div>
    <div class="fpbgvideo"></div>
   ▶<div id="_pop_login" class="mpopdiv">…</div>
    <div id="_pop_login_alert" class="alert alert-success">登入成功</div>
   ▶<script defer="defer">…</script>
   ▶<div id="_pop_tips" class="mtips">…</div>
   ▶<div id="_pop_dialog" class="mdialog">…</div>
   ▶<script>…</script>
    <!--<script>NProgress.done();</script>-->
    <!-- generated at Tue Feb 12 2019 18:49:53 -->
    <div id="overlay"></div>
   ▶<div id="loading" class="corner-all">…</div>
  </body>
</html>
```

```
element.style {
}
```

Chrome F12 Web Developer Tool

- We can press the first button on the left top corner (or Ctrl+Shift+C) to choose an element to inspect it.



```
          </th>
        ▼<td>
          ▶<h2>…</h2>
           <h3>09:30 國研大樓1樓華立廳</h3>
           <h3>&nbsp;</h3>
           <h3>&nbsp;</h3>
           <h3>&nbsp;</h3>
           <h3>&nbsp;</h3>
           <h3>&nbsp;</h3>
           <h3>&nbsp;</h3>
           <h3>&nbsp;</h3>
          </td>
        </tr>
      ▶<tr>…</tr>
      ▼<tr>
        ▶<th>…</th>
        ▼<td>
...       ▼<h2> == $0
             <a href="http://fltc-ogiaca.nsysu.edu.tw/p/
             404-1129-199298-1.php?Lang=zh-tw" title="Fun
             in Translation 翻轉「玩」字">【講座】Fun in
             Translation 翻轉「玩」字</a>
           </h2>
           <h3>14:30 駁二共創基地204共學教室（高雄市鹽埕區大
           勇路11號3樓）</h3>
           <h3>&nbsp;</h3>
          </td>
        </tr>
      </tbody>
    </table>
    <p>&nbsp;</p>
  </div>
</section>
</div>
```

... div div div div div #Dyn_1_1_1 div div section div table tbody tr td h2

```
element.style {
}
```

BeautifulSoup

- After using requests package to download the HTML code, we can further use Beautifulsoup to get the part we want.

```
Beautifulsoup(html_code_with_text_fortmat, parser_tool)
```

- Parser tool recommend using "html.parser" or "lxml"(need to download lxml package in advance.
- Ex

```
import requests
from bs4 import BeautifulSoup
html = requests.get('http://www.nsysu.edu.tw/’)
sp = BeautifulSoup(html.text, 'html.parser')
```

Some BeautifulSoup Common Function

title	Get the title tag of the HTML code.	sp.title #Result: <title>國立中山大學 National Sun Yat-sen University </title>
find()	Return the first result of the specific tag	sp.find("a") #Result: <a class="focusable" href="#start-C" title="跳到主要內容區塊">跳到主要內容區塊</a>
find_all()	Return all results of the specific tag	sp.find_all("img")
select()	Use CSS selector to return a list with all results of the specific tag, id or class in HTML code Must add “#” before id name “.” before class name	sp.select("img") #find tag with “img” sp.select("#Dyn_head") #find id “Dyn_head” sp.select(".mbox") #find class “mbox”
text	Get the specific HTML code part with just the content text.	sp.find("a").text #Result: 跳到主要內容區塊

Find and find all

- Find and find all can have second parameter

```
find(tagName,{attributeName:attributeContent})
```

- Ex

```
sp.find_all("img",{"class":"img-responsive"})
sp.find("img",{"class":"img-responsive"})
```

- If you want to search multiple tag in one time, you can include them with “[]”
- Ex

```
x=sp.find_all(['h','a'])
```

get()

- If you want to get the content in particular attribute, you can use get()

```
get(attributeName)
```

- Ex: If sp=

```
<a href="http://science.nsysu.edu.tw/p/406-1020-196791,r16.php?Lang=zh-tw" title="大數據分析在電力系統應用論壇">【論壇】大數據分析在電力系統應用論壇</a>
```

```
print(sp.find("a").get("href"))
```

→

```
http://science.nsysu.edu.tw/p/406-1020-196791,r16.php?Lang=zh-tw
```

urlopen()

- You can use urlopen with requests and beautifulsoup to download photos/document in the website.

```
urlopen(webpage)
```

Download all the images in Appledaily homepage

```
import requests
from bs4 import BeautifulSoup
from urllib.request import urlopen
html=requests.get('https://tw.appledaily.com/new/realtime')
sp = BeautifulSoup(html.text,'html.parser')
for i in sp.find_all("img"):
    print(i)
```

```
<img
src="http://b.scorecardresearch.com/p?c1=2&amp;c2=8028476&amp;cv=2.0&amp;cj=1
"/>
<img src="//img.appledaily.com.tw/pay/img/applelogo_realtime.png"/>
<img src="//img.appledaily.com.tw/appledaily/images/header/img/apple.png"/>
<img alt="只有5天！阿嬤顧的孫變了樣　養成雙下巴爆可愛" class="owl-lazy" data-
src="//img.appledaily.com.tw/images/thumbnail/other/c8317dda08bcdcbca5d6bbd7b6
7796cd.jpg"
src="data:image/gif;base64,R0lGODlhAQABAIAAAAAAAP///yH5BAEAAAAALAAAAAABA
AEAAAIBRAA7" tcode="只有5天！阿嬤顧的孫變了樣　養成雙下巴爆可愛"/>
<img alt="電信戰再起？中華電才嗆「別惹我」　台灣大、遠傳就推低價吃到飽"
class="owl-lazy" data-
src="//img.appledaily.com.tw/images/thumbnail/other/3a896eed3a394b99218d0df5d2
c24f4d.jpg"
src="data:image/gif;base64,R0lGODlhAQABAIAAAAAAAP///yH5BAEAAAAALAAAAAABA
AEAAAIBRAA7" tcode="電信戰再起？中華電才嗆「別惹我」　台灣大、遠傳就推低
價吃到飽"/>
<img alt="4歲童滑雪、衝浪都行　陽光帥爸教出「最快樂的孩子」" class="owl-lazy"
data-
src="//img.appledaily.com.tw/images/thumbnail/other/a8bb82e235c8d2733542fdd737
3aad8d.jpg"
src="data:image/gif;base64,R0lGODlhAQABAIAAAAAAAP///yH5BAEAAAAALAAAAAABA
AEAAAIBRAA7" tcode="4歲童滑雪、衝浪都行　陽光帥爸教出「最快樂的孩子」"/>
…
```

In here:

Picture name in “alt”
Image site in “data-src”

Download all the images in Appledaily homepage

```
import …
…
sp = BeautifulSoup(html.text,'html.parser')
for i in sp.find_all("img"):
    if(i.get("data-src")):
        print(i.get("data-src"))
```

```
//img.appledaily.com.tw/images/thumbnail/other/2fd1ba2213c4e5993b47127c2b5a5d69.jpg
//img.appledaily.com.tw/images/thumbnail/other/c8317dda08bcdcbca5d6bbd7b67796cd.jpg
//img.appledaily.com.tw/images/thumbnail/other/3a896eed3a394b99218d0df5d2c24f4d.jpg
//img.appledaily.com.tw/images/thumbnail/other/a8bb82e235c8d2733542fdd7373aad8d.jpg
//img.appledaily.com.tw/images/thumbnail/other/536812980fe2beb79a08f138375f3d12.jpg
//img.appledaily.com.tw/images/thumbnail/other/345a06b949414a6aa8e7314d4fa53496.jpg
//img.appledaily.com.tw/images/thumbnail/other/7b3602e2702dfcf774164ea4cdcbab7e.jpg
//img.appledaily.com.tw/images/thumbnail/other/7d59fcf2ec3663a277696e365abf4dac.jpg
//img.appledaily.com.tw/images/thumbnail/other/1d8dae72d1b3880473ccd0a33c546960.jpg
//img.appledaily.com.tw/images/thumbnail/other/7528ae080fd12345128d998f94bd3dda.jpg
//img.appledaily.com.tw/images/thumbnail/other/72645aaef226a879f52000132c974fa2.jpg
//img.appledaily.com.tw/images/thumbnail/other/abbc2936ece1b6085c9c50be9b2ecf3c.jpg
//img.appledaily.com.tw/images/thumbnail/other/c878c84d4fdc88bbeb3234551dc140cc.jpg
//img.appledaily.com.tw/images/thumbnail/other/d71df2feb0aac3ff1a937e9fa1f6c92c.jpg
```

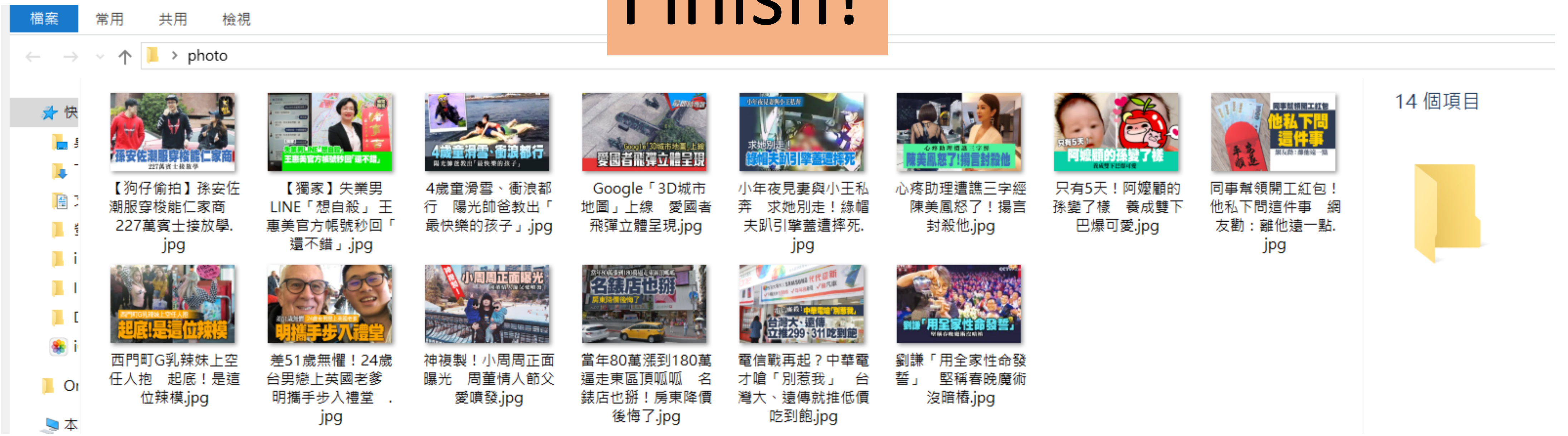
Download all the images in Appledaily homepage

```
import …
…
sp = BeautifulSoup(html.text,'html.parser')
for i in sp.find_all("img"):
    if(i.get("data-src")):
        temp=i.get("data-src")
        filename=i.get("alt")
        with open("photo/"+filename+".jpg","wb") as f:
            f.write(urlopen("http:"+temp).read())
```

Must use “wb” which means “write in binary mode”

Download all the images in Appledaily homepage

Finish!



【狗仔偷拍】孫安佐潮服穿梭能仁家商 227萬賓士接放學.jpg

【獨家】失業男LINE「想自殺」 王惠美官方帳號秒回「還不錯」.jpg

4歲童滑雪、衝浪都行　陽光帥爸教出「最快樂的孩子」.jpg

Google「3D城市地圖」上線　愛國者飛彈立體呈現.jpg

小年夜見妻與小王私奔　求她別走！綠帽夫趴引擎蓋遭摔死.jpg

心疼助理遭譙三字經　陳美鳳怒了！揚言封殺他.jpg

只有5天！阿嬤顧的孫變了樣　養成雙下巴爆可愛.jpg

同事幫領開工紅包！他私下問這件事　網友勸：離他遠一點.jpg

西門町G乳辣妹上空任人抱　起底！是這位辣模.jpg

差51歲無懼！24歲台男戀上英國老爹　明攜手步入禮堂　.jpg

神複製！小周周正面曝光　周董情人節父愛噴發.jpg

當年80萬漲到180萬逼走東區頂呱呱　名錶店也掰！房東降價後悔了.jpg

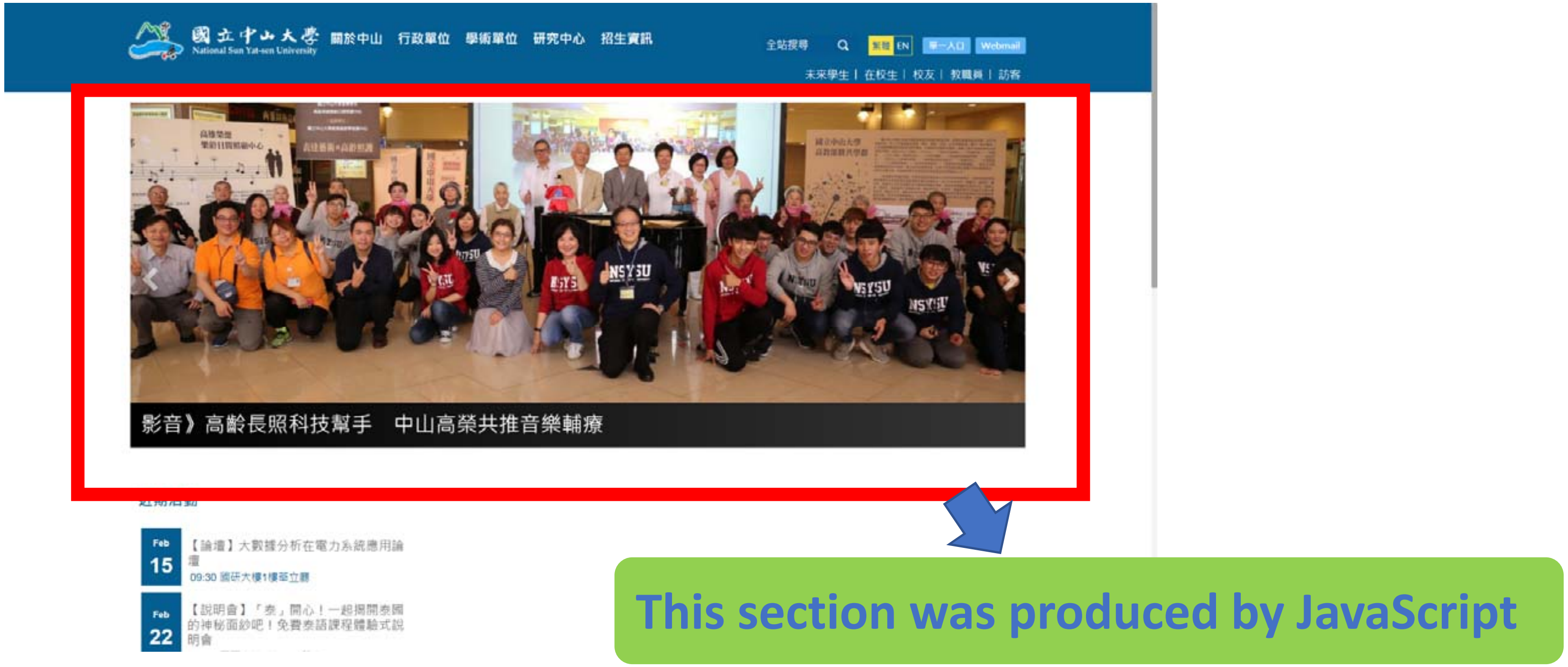
電信戰再起？中華電才嗆「別惹我」　台灣大、遠傳就推低價吃到飽.jpg

劉謙「用全家性命發誓」　堅稱春晚魔術沒暗樁.jpg

14 個項目

Cons of using requests

- If the HTML code is produced by JavaScript, it would be hard to fetch them, since requests didn't support dynamic content.



影音》高齡長照科技幫手　中山高榮共推音樂輔療

Feb 15 【論壇】大數據分析在電力系統應用論壇
09:30 國研大樓1樓華立廳

Feb 22 【說明會】「泰」開心！一起揭開泰國的神秘面紗吧！免費泰語課程體驗式說明會

This section was produced by JavaScript