

Python Tutorial

林逸軒

馮揚智

吳昶志

<http://lms.nthu.edu.tw/course/40724>

Department of Electrical Engineering National Tsing
Hua University

python

Print Function

```
>>> print ("hello world")
```

hello world

```
>>> print ("hello" + " NTHU")
```

hello NTHU

```
>>>print(33+36)
```

69

```
>>>print(123/456)
```

0.26973684210526316

```
>>>print(int(1.9)+3)
```

4

Define Variable and Basic Computational Operation

```
>>> a = 69
```

```
>>> 33*36
```

```
>>> type(a)
```

```
1188
```

```
int
```

```
>>> 2**10
```

```
>>> a = 'NTHU'
```

```
1024
```

```
>>> type(a)
```

```
>>> 8%3
```

```
Str
```

```
2
```

```
>>> a = 69.0
```

```
>>> type(a)
```

```
float
```

If-elif-else Loop

For python:

```
>>>a = 3
```

```
>>>b = 0
```

```
>>> if a > 2: 4 space or  
>>>     b = 123 1 Tab
```

```
>>> elif a < 0:
```

```
>>>     b = 456
```

```
>>>else :
```

```
>>>     b = 789
```

For C:

```
int a = 3;
```

```
int b = 0;
```

```
if(a>2){
```

```
    b = 123;
```

```
}
```

```
else if(a<0){
```

```
    b = 456;
```

```
}
```

```
else{
```

```
    b = 789;
```

```
}
```

While Loop

For python:

```
>>>condition = 10  
>>>while condition:  
>>>    print(condition)  
>>>    condition -= 1  
10  
9  
. . .  
1
```

For C:

```
int condition = 10;  
  
while(condition){  
    printf("%d\n",condition);  
    condition -= 1;  
}
```

For Loop(item sequence)

For python:

```
>>>example_list = [12,34,  
56,78,910]
```

```
>>>for i in example_list:  
>>>    print(i)
```

12

34

56

78

910

For C:

```
int example_list[] =  
{12,34,56,78,910};
```

```
int A = sizeof(example_list)  
/sizeof(int);
```

```
int i;
```

```
for(i = 0; i < A; i++){  
    printf("%d\n",  
    example_list[i]);  
}
```

For Loop(range function)

For python:

```
>>> for i in range(1, 10):  
>>>   print(i)
```

1

2

3

.

.

.

9

Not include 10

For C:

```
int i;  
for (i = 1; i < 10; i++){  
    printf("%d\n",i);  
}
```

For Loop(range function)

For python:

```
>>> for i in range(0,13,5):  
>>>   print(i)
```

```
0  
5  
10
```

For C:

```
int i;  
for (i = 0; i < 13; i=i+5){  
    printf("%d \n",i);  
}
```

Function

For python:

```
>>> def add(a, b):  
>>>     c = a + b  
>>>     print('c = ', c)  
>>>add(a =12,b=34)  
c = 46
```

For C:

```
#include <stdio.h>  
  
void add(int a,int b);  
  
int main(){  
    add(12, 34);  
    return 0;  
}  
  
void add(int a,int b){  
    int c = a + b;  
    printf("c = %d \n",c);  
}
```

Class(Account example)

```
class Account:  
  
    def __init__(self, number, name):  
        self.number = number  
        self.name = name  
        self.balance = 0  
  
    def deposit(self, amount):  
        if amount <= 0:  
            raise ValueError('must be positive')  
        self.balance += amount  
  
    def withdraw(self, amount):  
        if amount <= self.balance:  
            self.balance -= amount  
        else:  
            raise RuntimeError('balance not enough')
```

```
acct1 = Account ('123-456-789', 'Justin')  
  
# create an account  
acct1.deposit(100)  
acct1.withdraw(30)  
print (acct1.number)  
print (acct1.name)  
print (acct1.balance)
```

123-456-789

Justin

70

Input Function

```
>>> a_input = input('please input a number:')  
>>> print('this number is:',a_input)
```

please input a number:123

this number is: 123

List

```
>>>a = []  
>>> a = [1,1,2,3]  
>>> a.append(5678910)  
>>>print(a)
```

[1, 1, 2, 3, 5678910]

```
>>>a.insert(1,5566)  
>>> print(a)
```

[1, 5566, 1, 2, 3, 5678910]

```
>>>print(a[-1])
```

5678910

```
>>>print(a.count(1))
```

2

Numpy

```
>>>import numpy as np  
>>>array = np.array([[1,2,3],[2,3,4]])  
>>>print(array)
```

```
[[1 2 3]  
 [2 3 4]]
```

```
>>>print('number of dim:',array.ndim)
```

number of dim: 2

```
>>>print('shape :',array.shape)
```

shape : (2, 3)

```
>>>print('size:',array.size)
```

size: 6

Numpy

```
>>>a = np.array([2,23,4],dtype=np.float)  
>>>print(a.dtype)
```

float64

```
>>>a = np.array([2,23,4],dtype=np.float32)  
>>>print(a.dtype)
```

float32

```
>>>a = np.zeros((3,4))  
>>>print (a)
```

```
[[0. 0. 0. 0.]  
 [0. 0. 0. 0.]  
 [0. 0. 0. 0.]]
```

Numpy

```
>>>import numpy as np  
>>>a = np.arange(10,20,2)  
>>>print(a)
```

[10 12 14 16 18]

```
>>>a = np.arange(12).reshape((3,4))  
>>>print(a)
```

[[0 1 2 3]
 [4 5 6 7]
 [8 9 10 11]]

Numpy

```
>>>a = np.random.randint(1,10, size=(2, 4))  
>>>print (a)
```

```
[[1 5 6 8]  
 [3 3 4 9]]
```

```
>>> print(np.sum(a)) 39  
>>> print(np.min(a)) 1  
>>> print(np.max(a)) 9  
>>>print("sum =",np.sum(a, axis=1))
```

```
sum = [20 19]
```

```
>>>print("min =",np.min(a, axis=0))
```

```
min = [1 3 4 8]
```

```
>>>print("max =",np.max(a, axis=1))
```

```
max = [8 9]
```

Numpy

```
>>>import numpy as np  
>>>A = np.arange(14,2, -1).reshape((3,4))  
>>>print(A)
```

```
[[14 13 12 11]]
```

```
[10 9 8 7]
```

```
[ 6 5 4 3]]
```

```
>>>print(np.argmin(A)) 11  
>>> print(np.argmax(A)) 0  
>>> print(np.mean(A)) 8.5  
>>> print(np.sort(A))
```

```
[[11 12 13 14]]
```

```
[ 7 8 9 10]
```

```
[ 3 4 5 6]]
```

Numpy

```
>>> print(A.flatten())
```

```
[14 13 12 11 10  9  8  7  6  5  4  3]
```

```
>>>XX = np.array([1,1,1])
>>>YY = np.array([2,2,2])
>>ZZ = np.vstack((XX,YY))
>>>WW = np.hstack((XX,YY))
>>>print(ZZ)
```

```
[[1 1 1]]
```

```
[2 2 2]]
```

```
>>>print(WW)
```

```
[1 1 1 2 2 2]
```

```
>>>print(XX[:,np.newaxis])
```

```
[[1]]
```

```
[1]
```

```
[1]]
```

Numpy

```
>>>XX = np.array([1,1,1])  
>>>YY = np.array([2,2,2])  
>>ZZ = np.vstack((XX,YY))  
>>>WW = np.hstack((XX,YY))  
>>>print(ZZ)
```

[[1 1 1]]

[2 2 2]]

```
>>>print(WW)
```

[1 1 1 2 2 2]

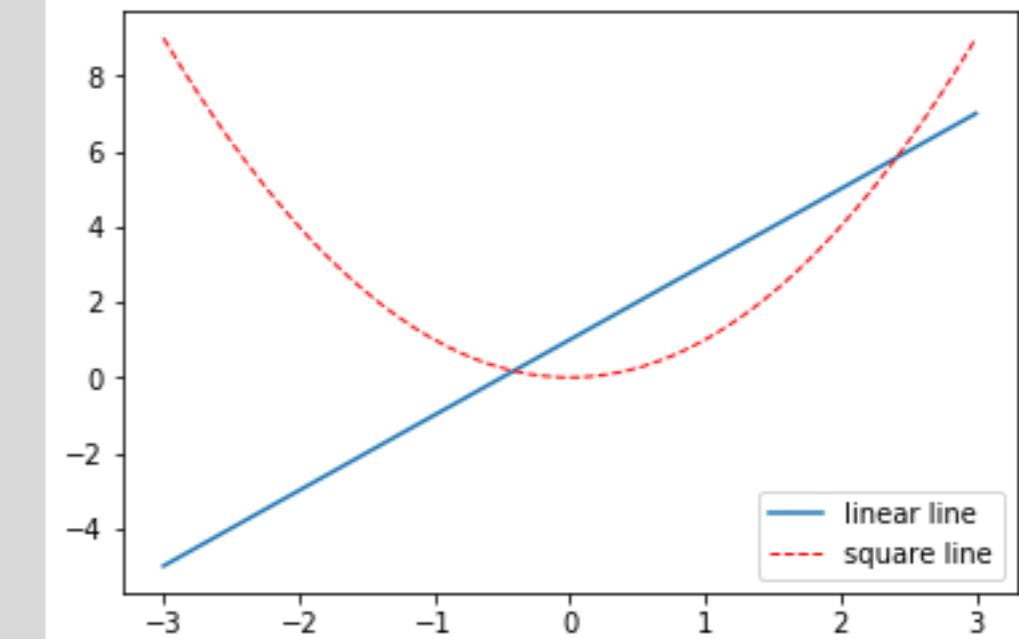
```
>>>[X,Y]=np.vsplit(ZZ, 2)  
>>>print(X)  
>>>print(Y)
```

[[1 1 1]]

[[2 2 2]]

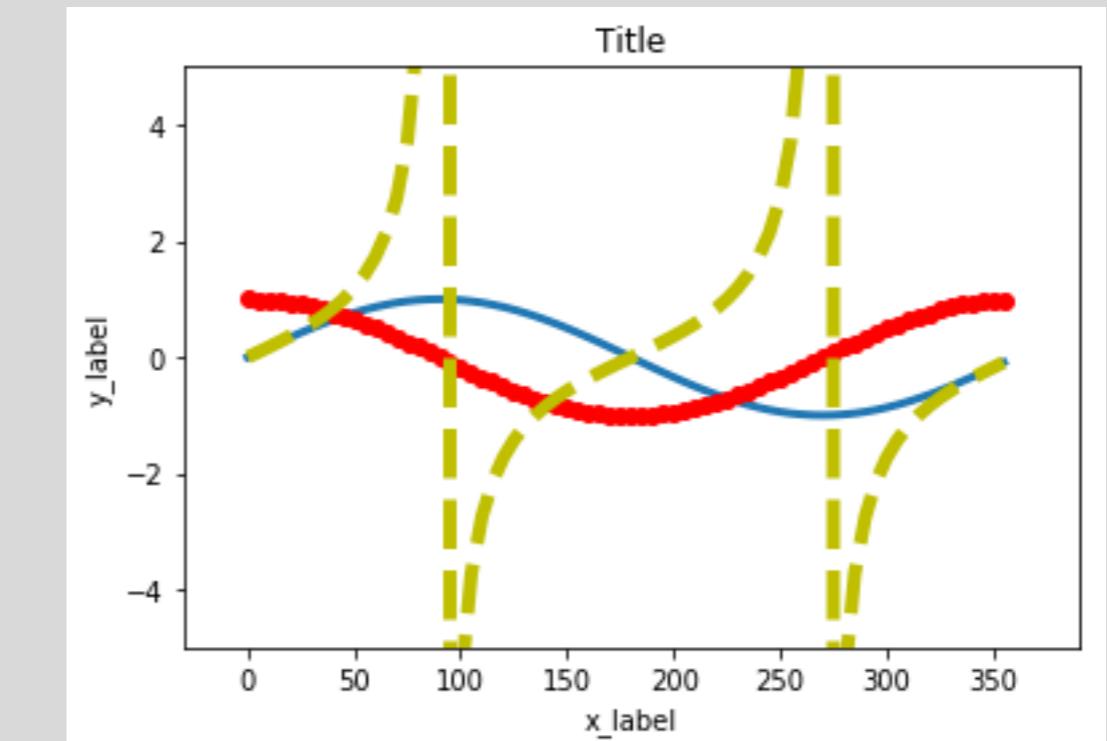
Matplotlib

```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(-3, 3, 50)
y1 = 2*x + 1
y2 = x**2
# set line styles
l1, = plt.plot(x, y1, label='linear line')
l2, = plt.plot(x, y2, color='red', linewidth=1.0, linestyle='--',
label='square line')
plt.legend(loc='best')
```



Matplotlib

```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(0,360,5)
y1 = np.sin(x * np.pi / 180.0)
y2 = np.cos(x * np.pi / 180.0)
y3 = np.tan(x * np.pi / 180.0)
plt.show()
plt.plot(x,y1,lw=3)
plt.plot(x,y2,"ro")
plt.plot(x,y3,"y--",lw=5)
plt.ylabel("y_label")
plt.xlabel("x_label")
plt.title("Title")
plt.xlim(-30,390)
plt.ylim(-5,5)
```



Matplotlib

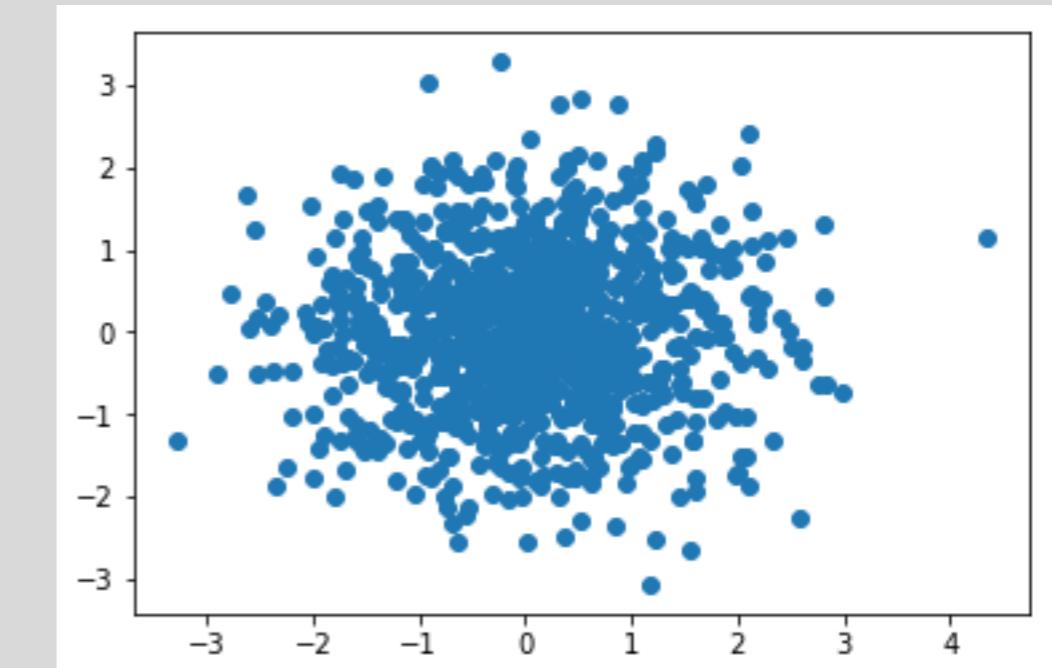
character	description
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
::	dotted line style
.'	point marker
;	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'+'	plus marker

character	color
'b'	blue
'g'	green
'r'	red
::	cyan
'.'	magenta
;;	yellow
'o'	black
'v'	white

[Matplotlib.pyplot API](#)

Matplotlib

```
import matplotlib.pyplot as plt  
  
n = 1024  
  
X = np.random.normal(0,1,n)  
  
Y = np.random.normal(0,1,n)  
  
plt.scatter(X,Y)
```



Read file

```
import csv  
  
f = open('data.csv', 'r')  
file_read=csv.reader(f)  
for row in file_read :  
    print(row)  
f.close()
```

data.csv			
	A	B	C
1	1	3	3
2	4	5	6
3	7	8	9

['1', '2', '3']

['4', '5', '6']

['7', '8', '9']

Installation

Installation-Anaconda

 Windows |  macOS |  Linux

Anaconda 2019.07 for Windows Installer

Python 3.7 version

[Download](#)

[64-Bit Graphical Installer \(486 MB\)](#)

[32-Bit Graphical Installer \(418 MB\)](#)

Python 2.7 version

[Download](#)

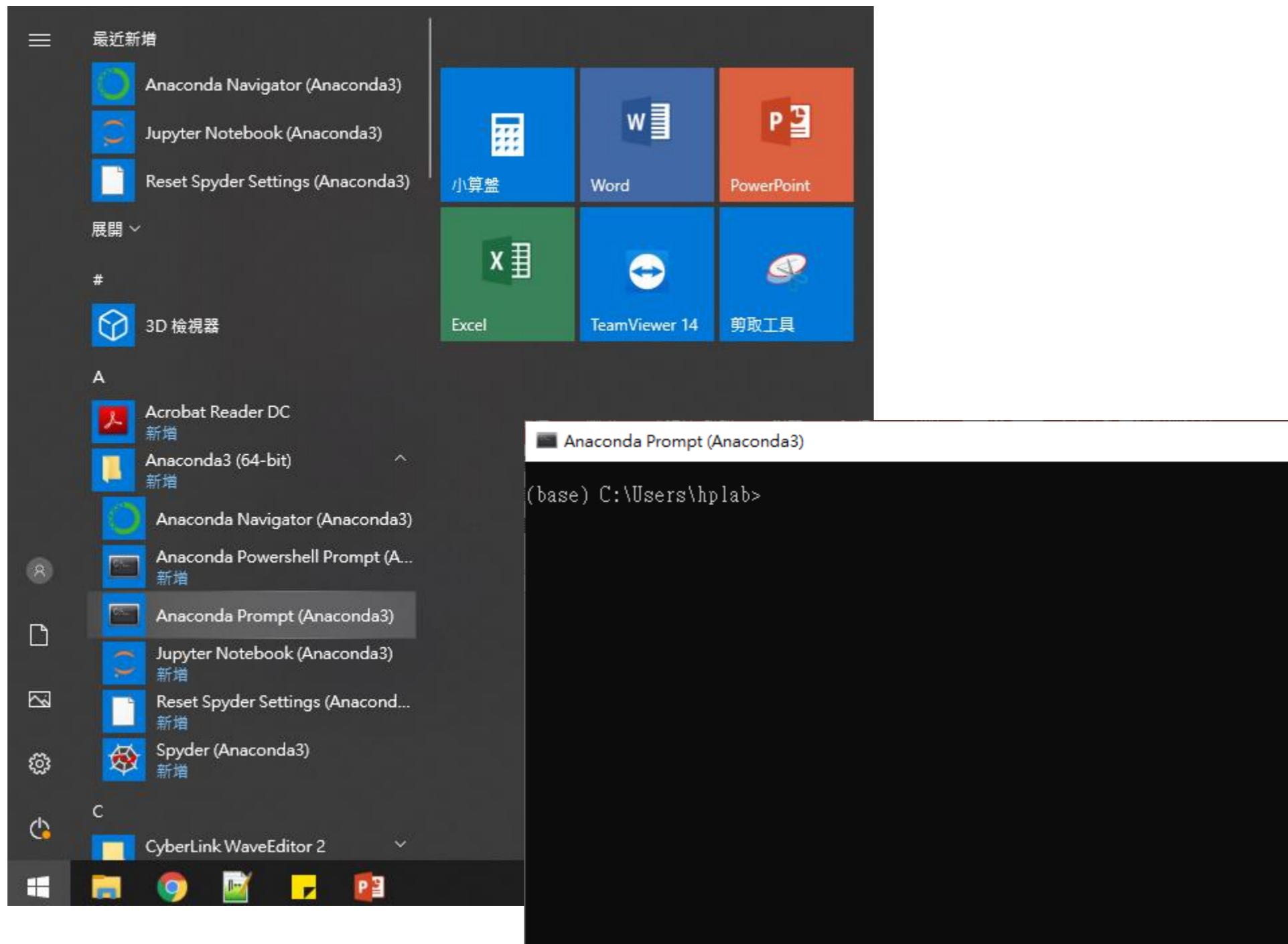
[64-Bit Graphical Installer \(427 MB\)](#)

[32-Bit Graphical Installer \(361 MB\)](#)

[Anaconda download site](#)

Installation-Tensorflow

• Anaconda Prompt



Installation-Tensorflow

- Create a conda environment

```
C:> conda create -n tensorflow pip python=3.5
```

- Activate the conda environment

```
C:> activate tensorflow
```

- Issue the appropriate command

```
C:> pip install --ignore-installed --upgrade tensorflow
```

- Install the required matplotlib

```
C:> pip install matplotlib
```

Installation-Jupyter

- Installation-Jupyter

```
pip install jupyter
```

```
jupyter notebook
```

```
C:\Windows\System32
```

```
C:\users> activate tensorflow
```

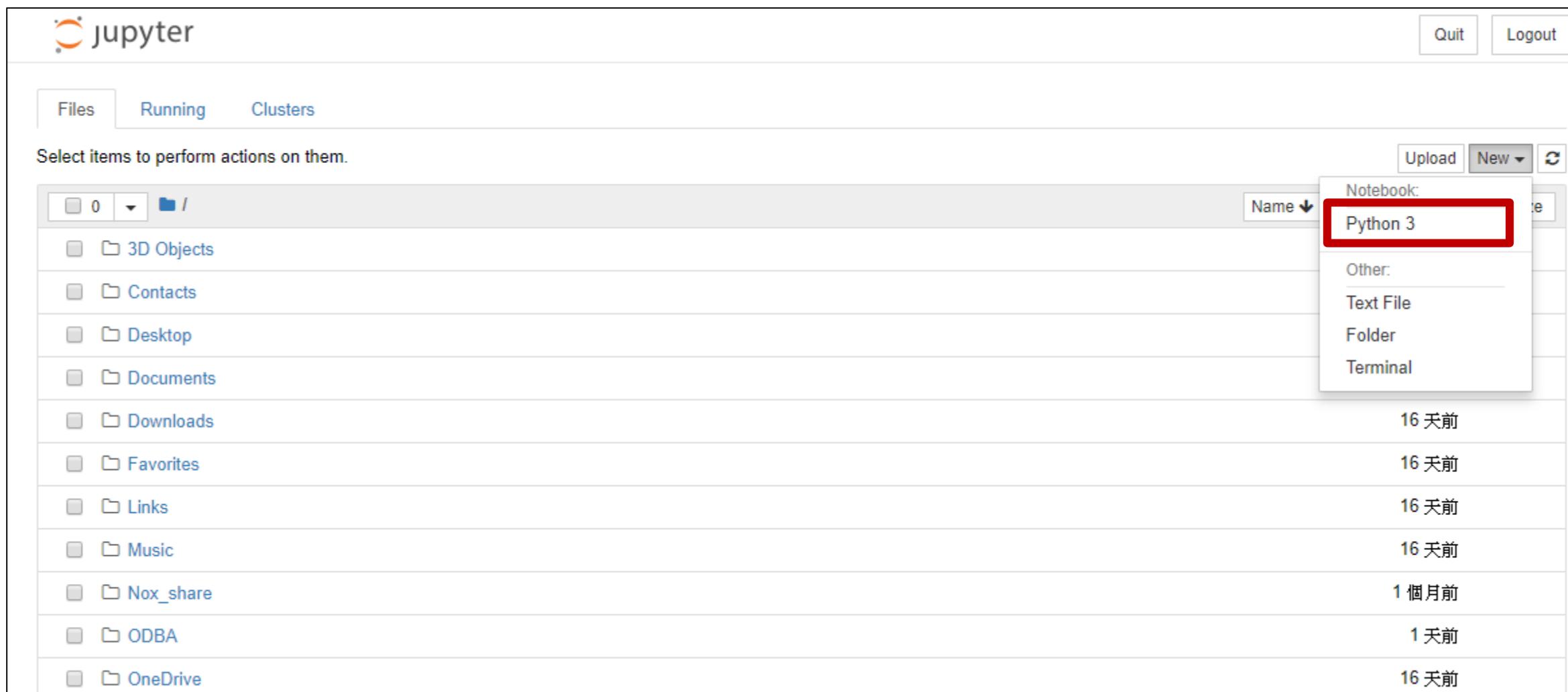
```
C:\users> jupyter notebook
```

How to hand in the report

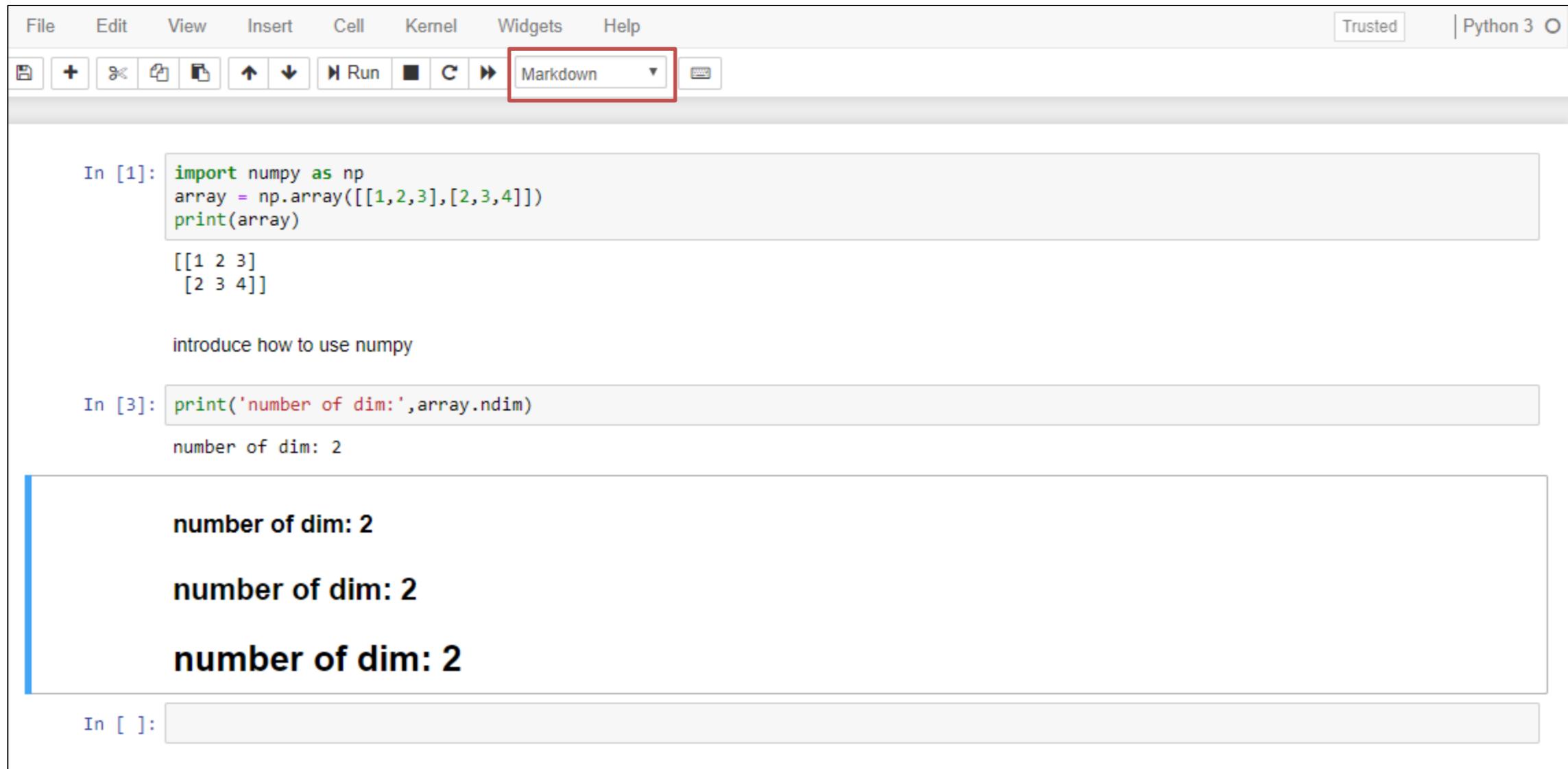
Start on jupyter

```
C:\Windows\System32
C:\users> jupyter notebook
```

Start on jupyter



Start on jupyter



The screenshot shows a Jupyter Notebook interface with the following details:

- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, Python 3.
- Cell 1 (Markdown):** Contains the code:

```
import numpy as np
array = np.array([[1,2,3],[2,3,4]])
print(array)
```

 and its output:

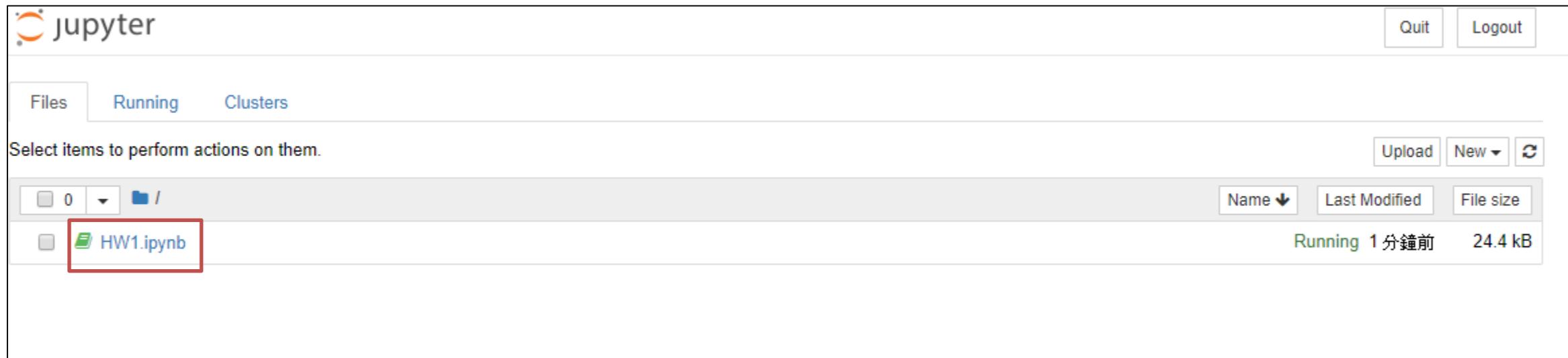
```
[[1 2 3]
 [2 3 4]]
```
- Text:** "introduce how to use numpy"
- Cell 3:** Contains the code:

```
print('number of dim:',array.ndim)
```

 and its output:

```
number of dim: 2
```
- Output Area:** Displays three bolded outputs: **number of dim: 2**, **number of dim: 2**, and **number of dim: 2**.
- In []:** An empty input cell for the next command.

Start on jupyter



Latex in jupyter notebook

```
from IPython.display import Latex
```

```
Latex(r"$f(x) = 3x + 7$")
```

$$f(x) = 3x + 7$$

```
Latex(r"\sum_{i=1}^n a_i = 0")
```

$$\sum_{i=1}^n a_i = 0$$

```
Latex(r"\frac{7x+5}{1+y^2} \sqrt{x^2+y^2} \sqrt[n]{x^n+y^n}")
```

$$\frac{7x + 5}{1 + y^2} \sqrt{x^2 + y^2} \sqrt[n]{x^n + y^n}$$

[List of LaTeX mathematical symbols](#)

Practice Example

Practice Example1

Guessing number game

Generate a random number between 1 to 200 (including 1 and 200). Ask the user to guess the number, then tell them whether the answer is

1. lower than previous answer,
2. higher than previous answer,
3. or exactly the correct number

Calculate how many times does the user guess until the number is hit in the end.

(Hint: use the input func and random generate func)

Practice Example1

Guessing number game

What's your guess?

What's your guess? 55

Too high!

What's your guess? 35

Too low!

What's your guess? 43

Too low!

What's your guess? 47

Too low!

What's your guess? 52

You got it!

And it only took you 5 tries!

Practice Example2

matplotlib

Read the file (data.csv). Convert data to array type. Calculate how many positive elements are in every raw.

Then separate the array (2x20) into two array(1x20). Treat the two arrays as x-axis data and y-axis data and use **matplotlib** to draw a scatter plot.

Practice Example2

matplotlib

```
number of positive in first row = 13
number of positive in second row = 6
```

```
Out[108]: <matplotlib.collections.PathCollection at 0x2562e8d8dd8>
```

