Python程式設計入門 - 執行環境

STEAM 教育學習網: https://steam.oxxostudio.tw/category/python/index.html
W3school: https://www.w3schools.com/python/default.asp

簡介

- Script Program Language
- Object-Oriented Program Language
- General-Purpose Program Language
- Easy to learn
- 誰在使用Python呢?
  - 大神Google
  - 美國太空總署(NASA)
  - ...
- [How to Become a Hacker] 一文中推薦使用
- Google 機器學習套件

Uses of Python

- shell tools
  - system admin tools, command line programs
- extension-language work
- rapid prototyping and development
- language-based modules
  - instead of special-purpose parsers
- database access
- Internet scripting
- 網頁設計、手機 App 撰寫、遊戲程式設計、硬體自動控制、生物醫學、大數據
- Machine learning (Deep learning)

Anaconda 安裝 (適合新手)

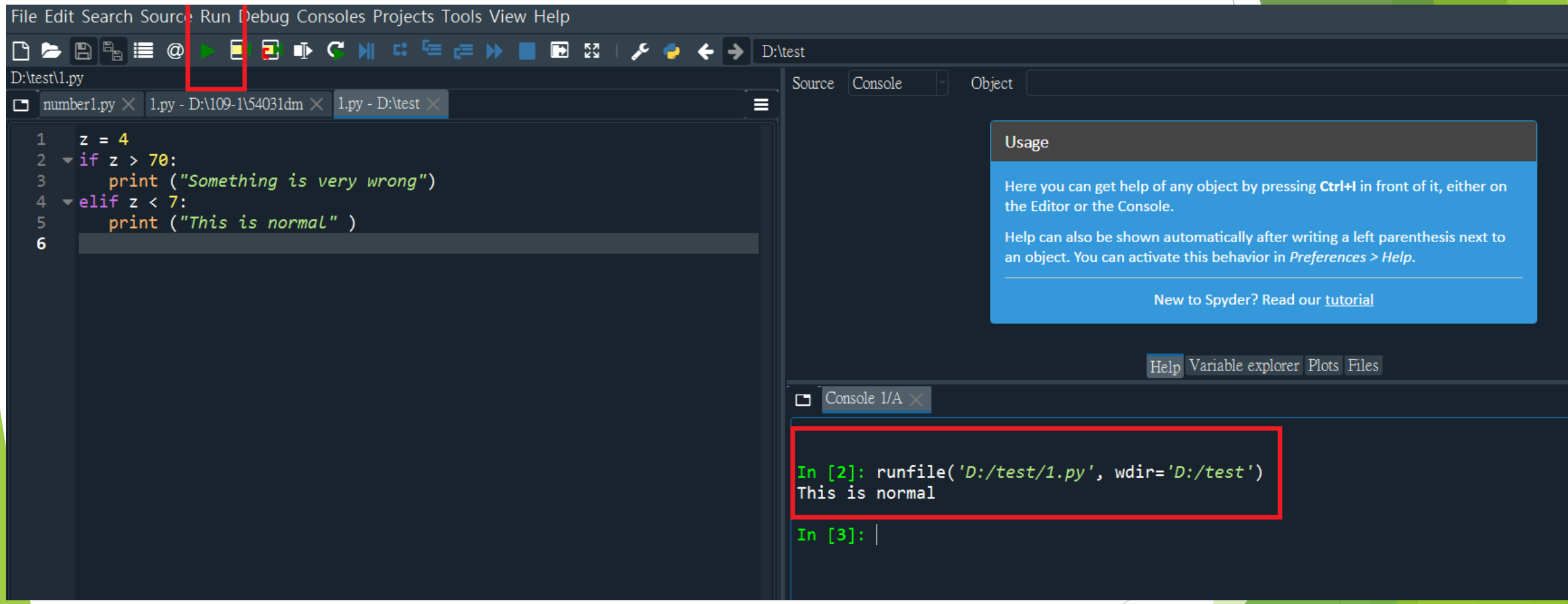
- Anaconda 是比 python 還大的蟒蛇
- 包含了眾多流行的科學、數學、工程、數據分析的 Python 套件
- 可切換python版本
- www.anaconda.com/download

  (windows python 3.7 64bit version)

Python 整合開發環境 (IDE)

- Spyder : 較適合機器學習, 深度學習 等中小型規模開發, anaconda 已包含
- Jupyter notebook (Ipython) 可以使用瀏覽器登入server 操做python 相當多人使用的開發環境. anaconda 已包含
- Pycharm: 較適合網路應用 等中大規模開發(需安裝)

Spyder


File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\test
D:\test\1.py
number1.py
1.py - D:\109-1\54031dm
1.py - D:\test
z = 4
if z > 70:
print ("Something is very wrong")
elif z < 7:
print ("This is normal")
Source
Console
Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Help
Variable explorer
Plots
Files
Console 1/A
In [2]: runfile('D:/test/1.py', wdir='D:/test')
This is normal
In [3]:

使用Python

- Window 平台: Run IDE

- Linux、FreeBSD … 其他作業平台
- 有兩種主要使用python的方法
  - 使用互動式命令列
    - e.q. 直接鍵入python就會進入python的互動式命令列
  - 將程式寫成檔案，再由python執行
    - 直在將程式碼寫在檔案內，然後再執行python去讀取該檔案
      - Ex: python hello.py
    - 或是在檔案的第一個行寫著 #!/usr/bin/env python，然後在第二行之後輸入程式碼，如此可以直接執行該檔案
      - Ex: ./hello.py

Google colab

- 免費雲端運行Python深度學習框架

- https://colab.research.google.com

以google帳號登入

Google 提供了GPU，甚至是更專業化的TPU

TPU 為google專為AI/機器學習 等高速運算開發的架構, 比CPU, GPU 快 15-30 倍

- 每個程式限制12小時, 儲存空間有限制

Google Colaboratory (colab)

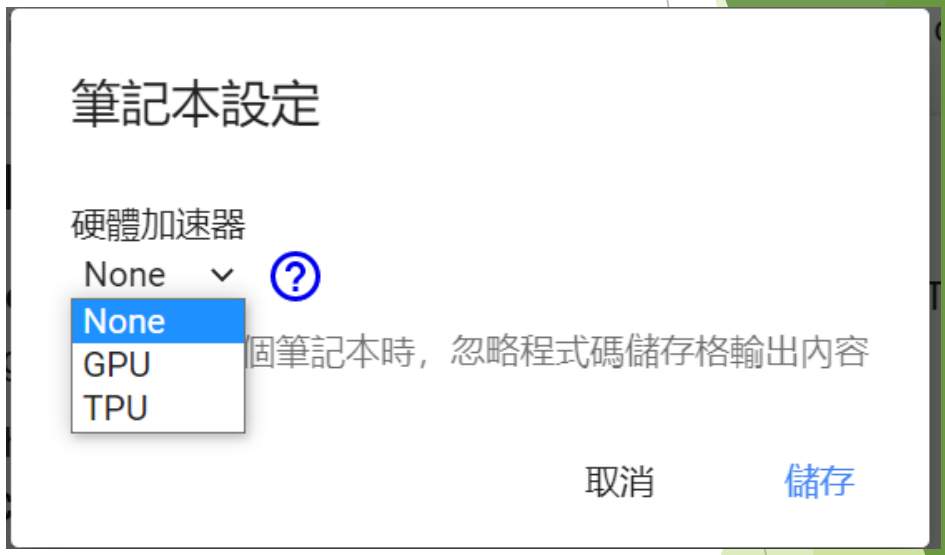
- 透過 Google Colaboratory 學習使用 Python 做機器學習等科學計算

- https://colab.research.google.com

- 你可以從官方提供的範例（EXAMPLES）、Google Drive、GitHub、自行上傳（UPLOAD）等方式，或是直接用最下面的按鈕建立全新的 notebook，

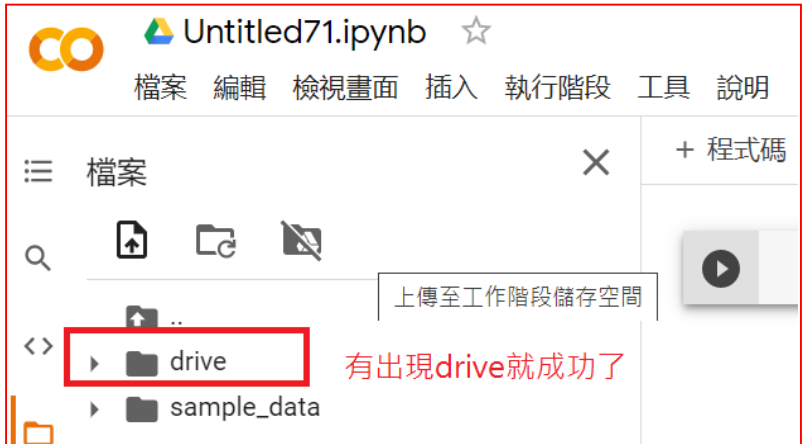
colab

- Edit (修改) -> notebook setting (筆記本設定)

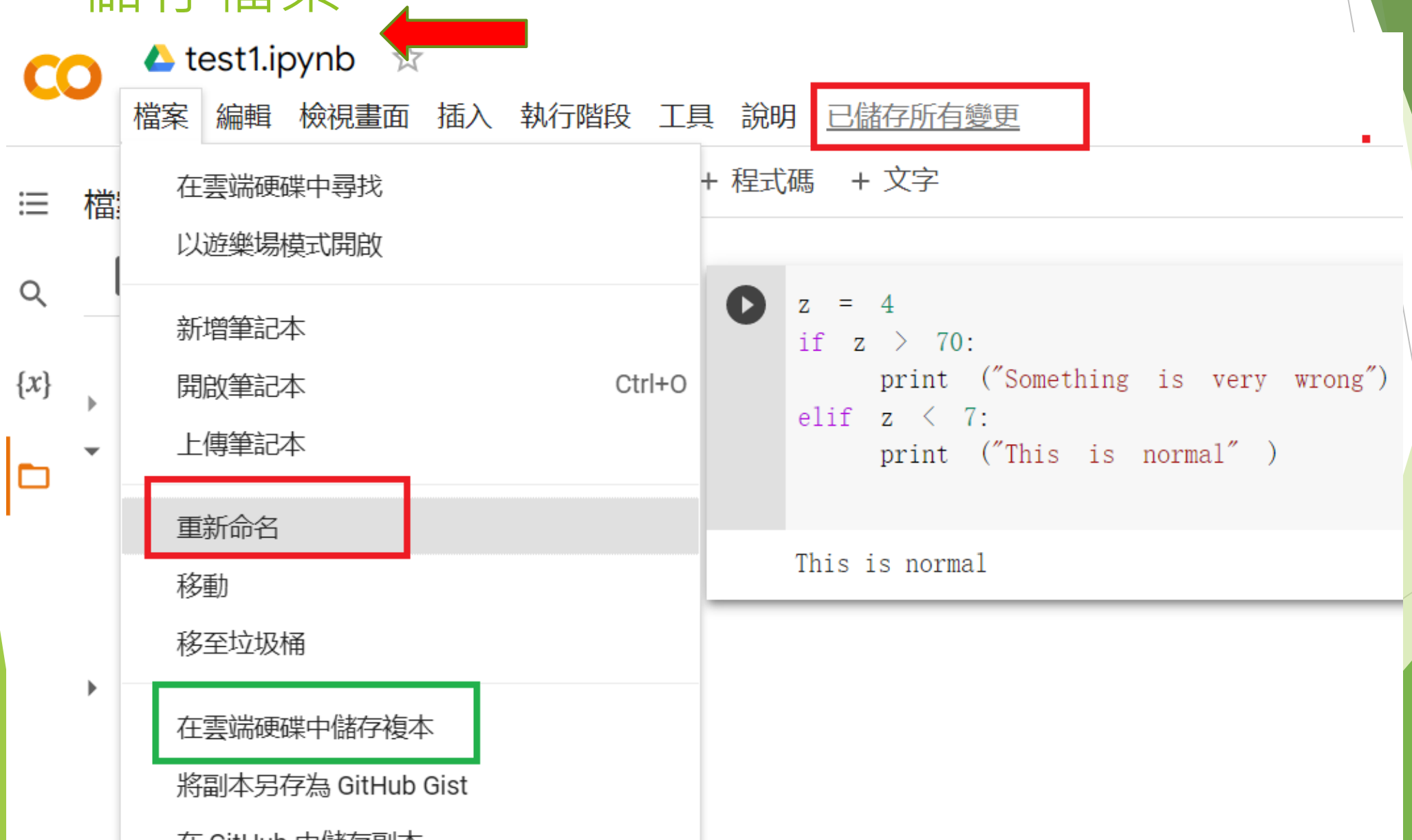


若不須用到GPU, TPU請勿選取, 節省系統資源

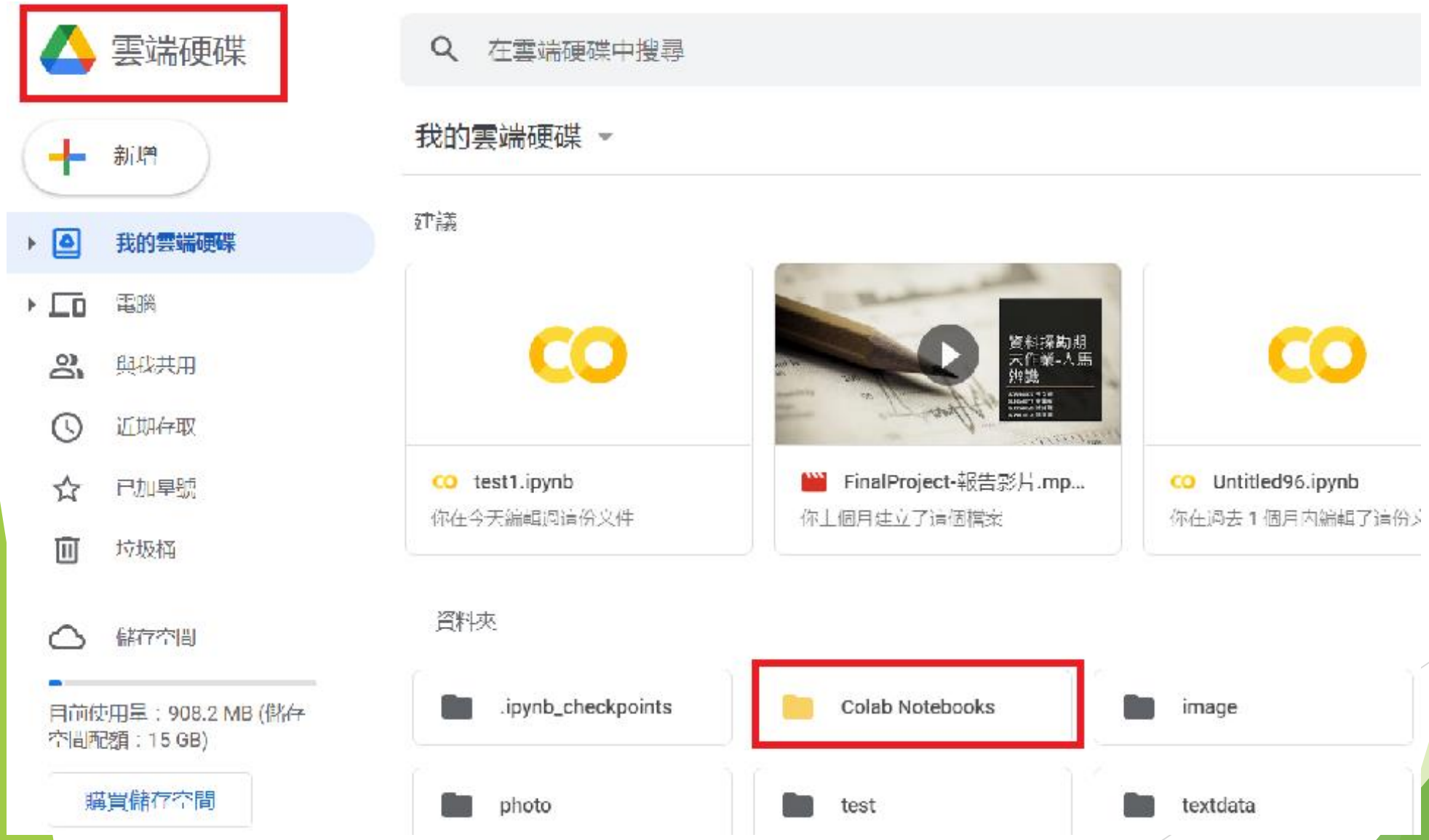
Colab 掛接雲端硬碟



儲存檔案


test1.ipynb
檔案 編輯 檢視畫面 插入 執行階段 工具 說明
已儲存所有變更
在雲端硬碟中尋找
以遊樂場模式開啟
新增筆記本
開啟筆記本 Ctrl+O
上傳筆記本
重新命名
移動
移至垃圾桶
在雲端硬碟中儲存複本
將副本另存為 GitHub Gist
+ 程式碼 + 文字
z = 4
if z > 70:
print ("Something is very wrong")
elif z < 7:
print ("This is normal")
This is normal

儲存檔案位置


雲端硬碟
在雲端硬碟中搜尋
新增
我的雲端硬碟
電腦
與我共用
近期存取
已加星號
垃圾桶
儲存空間
目前使用量：908.2 MB (儲存空間配額：15 GB)
購買儲存空間
我的雲端硬碟
建議
test1.ipynb
你在今天編輯過這份文件
FinalProject-報告影片.mp...
你上個月建立了這個檔案
Untitled96.ipynb
資料夾
.ipynb_checkpoints
Colab Notebooks
image
photo
test
textdata

Python程式設計入門 - 語法介紹

表示式(Expressions)

- 表示式

```
3 + 5
3 + (5 * 4)
3 ** 2
'Hello' + 'World'
```

運算子	說明	範例	結果
+	加法	1+2	3
-	減法	20-12	8
*	乘法	2*3	6
/	除法	9/2	4.5
//	除法取整數 (無條件捨去)	9//2	4
%	餘數	9%2	1
**	次方	2**3	8

變數的型別

- x= "Hello World" (str 型別)
- x=123456 (int 型別)
- x=123.456 (float 型別)
- x=[1, 2, 3, 4] (list 型別)

- type(object) (察看變數為何種型別)
- 變數指定

```
a = 4 << 3
b = a * 4.5
c = (a+b)/2.5
a = "Hello World"
```

  - 型別是動態的，會根據指定時的物件來決定型別
  - **變數單純只是物件的名稱，並不會和記憶體綁在一起。**

First example

```
text1 = 'good'
num = 100
print(text1 + str(num))
print( num +10)
print ("my number = ", num)
print ("my number = %d" %num)
print (" the data type of text1 =  " , type(text1))
```

Python 流程控制

- Python 程式碼注重格式
  - 撰寫程式時注重『縮排』
- If 敘述
  - 常使用於『邏輯判斷』
- For loop 迴圈控制
  - 常用來『重複做某件事』

基本概念

- 語法特色
  - 以冒號(:)做為敘述的開始
  - 不必使用分號(;)做為結尾
  - 井字號(#)做為註解符號，同行井字號後的任何字將被忽略
  - 使用tab鍵做為縮排區塊的依據 (空格)
  - 不必指定變數型態 (runtime時才會進行binding)

條件式敘述 和 比較運算子 (Conditional Statements) Part I

- if-else

```
if a < b:
    z = b
else:
    z = a
```

- pass 敘述 - 不做任何事時使用

```
if a < b:
    pass
else:
    z = a
```

運算子	說明	範例
>	大於 (a 是否大於 b)	a > b
<	小於 (a 是否小於 b)	a < b
>=	大於等於 (a 是否大於等於 b)	a >= b
<=	小於等於 (a 是否小於等於 b)	a <= b
==	等於 (a 是否等於 b)	a == b
!=	不等於 (a 是否不等於 b)	a != b

條件式敘述 (Conditional Statements) Part II

- elif敘述

```
if a == '+':
   op = PLUS
elif a == '-':
   op = MINUS
else:
   op = UNKNOWN
```

  - 沒有像C語言一樣，有switch的語法

- 布林表示式 - and, or, not

```
if b >= a and b <= c:
    print ("b is between a and c")
if not (b < a or c > c):
    print ("b is still between a and c")
```

If else

```
z = 4
if z > 70:
    print ("Something is very wrong")
elif z < 7:
    print ("This is normal" )
```

If else

```
a = "h"
b = "k"

if a == "a":
    print("Yes!")
else:
    if a == b:
        print("No!")
    else:
        print("What?")
```

讓使用者輸入資料

- input(prompt)

```
x=input("please enter your name : ")
print (x+ "how are you ! " )
```

```
xString = input("Enter a number: ")
x = int(xString)
yString = input("Enter a second number: ")
y = int(yString)
print('The sum of ', x, ' and ', y, ' is ', x+y, '.' )
```

迴圈 (Loops)

- while敘述

```
while a < b:
    # Do something
    a = a + 1
```

- for敘述 (走訪序列的元素)

```
for i in [3, 4, 10, 25]:
    print (i)

# Print characters one at a time
for c in "Hello World":
    print (c)

# Loop over a range of numbers
for i in range(0,100):
    print (i)
```

range

- range(start , pastEnd , step)

```
for i in range(10, 0, -1):  # countdown...
    print(i)
print('Blastoff!')
```

Example

```
for i in 1,2,3,4,5,6,7,8,9:
    for j in 1,2,3,4,5,6,7,8,9:
        print ("%4d" %(i*j), end=" ")
    print ()
```

Example

```
temperature = 115
while temperature > 112:  # first while loop code
    print(temperature)
    temperature = temperature - 1

print('The tea is cool enough.')
```

函式 (Functions)

- def敘述

```
# Return the remainder of a/b
def remainder(a,b):
    q = a/b  # q = int (a/b)
    r = a - q*b
    return r
# Now use it
a = remainder(42,5)          # a = 2
print (a)
```

- 回傳一個以上的值

```
def divide(a,b):
    q = int (a/b)
    r = a - q*b
    return q,r

x,y = divide(42,5)       # x = 8, y = 2
print (x)
print (y)
```

Example 1

```
def a(name):
        print ("My name is",name)

a("Denny")
```

- My name is Denny

Example 2

```
def varLenArgFunc(*varvallist ):
    print ("The Output is: ")
    for varval in varvallist:
        print (varval)
    return;
print("Calling with single value")
varLenArgFunc(55)
print("Calling with multiple values")
varLenArgFunc(50,60,70,80)
```

Example 3

```
# Define our 3 functions
def my_function():
  print("Hello From My Function!")
def my_function_with_args(username, greeting):
  print("Hello, %s , From My Function!, I wish you %s"%(username,
greeting))
def sum_two_numbers(a, b):
  return a + b
# print(a simple greeting)
my_function()
#prints - "Hello, John Doe, From My Function!, I wish you a great year!“
my_function_with_args("John Doe", "a great year!")
# after this line x will hold the value 3!
x = sum_two_numbers(1,2)


print (x)
```

模組 (Modules)

- 程式可分成好幾個模組

```
# number1.py
def add(a, b):
    print ("ADDING %d + %d" % (a, b))
    return a + b

def multiply(a, b):
    print ("MULTIPLYING %d * %d" % (a, b))
    return a * b
```

- import敘述

```
import number1
x = number.add(42,5)
y = number.multiply(5, 8)
print ("the value of x = ", x)
print ("the value of y = ", y)
```

字串格式化

- 在 Python3 以後，開始引進新串格式化
- Python 3.6 又新增了格式字串字面值

```
def my_function_with_args(username, greeting):
    print("Hello, {} , From My Function!, I wish you {}
".format(username, greeting))
def sum1(a,b):
  return f"in function a + b  = {a+b}"
my_function_with_args("John Doe", "a great year!")
x=sum1(1,2)
print(x)
a=1
b=2
print(f"a+b = {a+b}")
#格式字串字面值可以,把 Python 運算式嵌入在字串常數中
```