

Software Implementation and computation experiments

Ex1 : imread and imshow

Ex2 : linspace, sin, scatter

EX3: input a number that is
multiple of 2 or 3

Pycharm是python程式語言的集成開發環境，本實驗將介紹如何使用pycharm來實作python程式。

我們要實作三個程式：

(30 points) 其中一個程式要讀入一張彩色照片並且顯示影像；

(30 points) 另一個程式是要畫出sin在 $[-\pi, \pi]$ 區間的函數圖；

(40 points) 第三個程式要求使用者輸入2或3的倍數

使用的packages包括，matplotlib、numpy，我們將介紹如何在pycharm中搜尋、安裝package。

實驗一(30 points)：

請自行準備一張照片，讀入
照片並將照片顯示在螢幕上

範例與說明

image3d.png



實驗一: Read and Show an image

- 使用PyCharm新增一個計畫
- 將照片image3d.png移到計畫檔案目錄中
- 安裝matplotlib
- 新增一個python檔案，撰寫以下程式：
 1. 匯入matplotlib.pyplot，matplotlib.image
 2. 使用matplotlib.image的imread方法，讀入影像
 3. 使用matplotlib.pyplot的imshow方法，顯示影像
 4. 使用matplotlib.pyplot的imshow()命令顯示圖形視窗

cv_show_image [~/Desktop/py_code_2020/cv_show_image] - .../main.py [cv_show_image]

cv_show_image > main.py

Project

- cv_show_image ~/Desktop/py_code_2020
 - venv
 - image3d.png
 - main.py
- External Libraries
- Scratches and Consoles

```
1 import matplotlib.pyplot as plt
2 import matplotlib.image as mpimg
3 plt.figure(1)
4 plt.imshow(mpimg.imread('image3d.png'))
5 plt.show()
6
7
```

Run: main x

/Users/apple/Desktop/py_code_2020/cv_

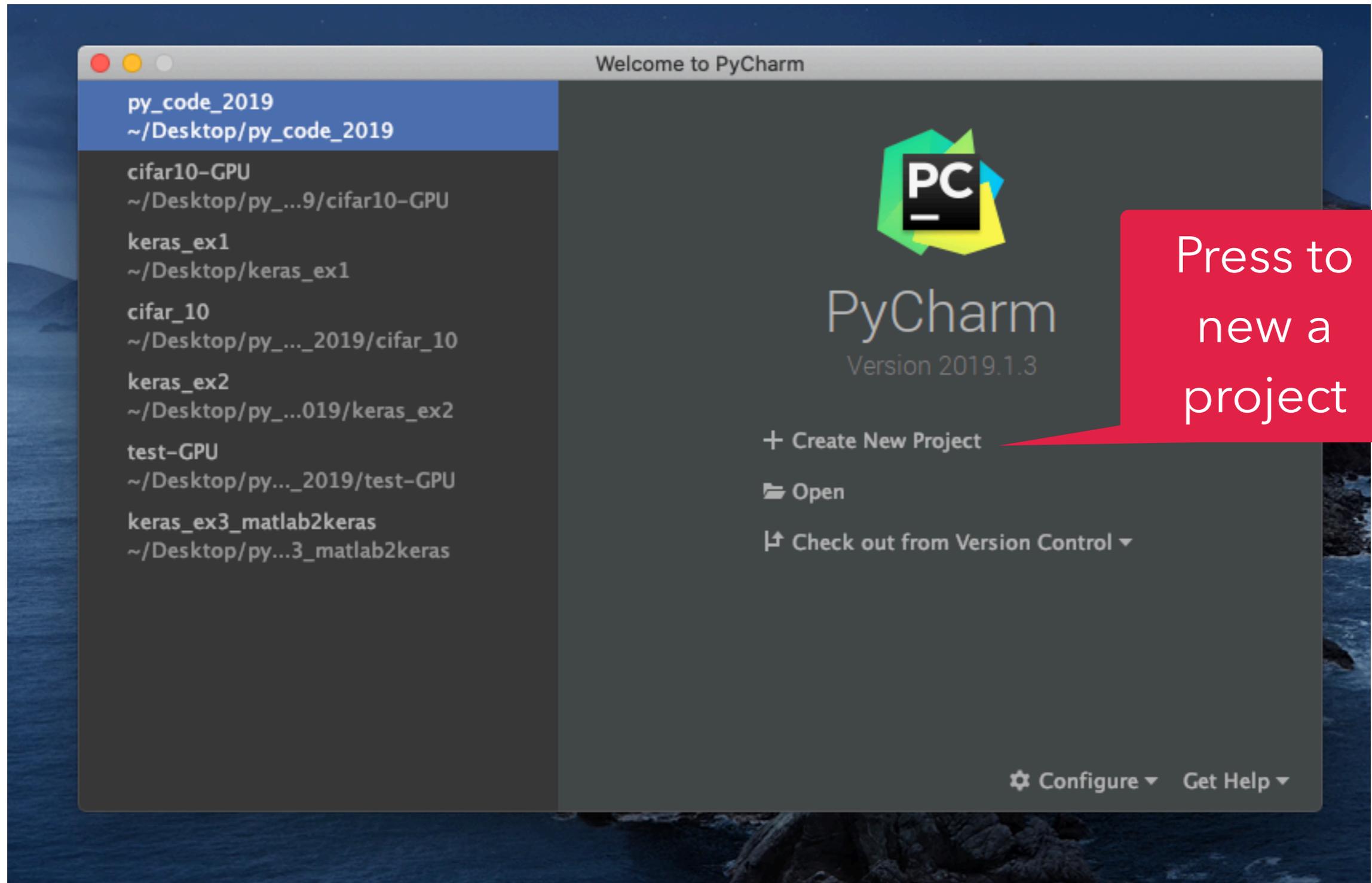
4: Run 6: TODO Terminal Python Console

PEP 8: blank line at end of file

Figure 1



使用PyCharm新增一個計畫



按下偏好preference，安裝軟體

Press to
Install
package

The screenshot shows the PyCharm IDE interface. The 'File' menu is open, and 'Preferences...' is selected. A red callout box points to this menu item with the text 'Press to Install package'. The main editor window displays a Python script named 'main.py' with the following code:

```
1 import matplotlib.pyplot as plt
2 import matplotlib.image as mpimg
3 plt.figure(1)
4 plt.imshow(mpimg.imread('image3d.png'))
5
6
7 import numpy as np
8 x = np.linspace(-np.pi, np.pi, 100)
9 y = np.sin(x)
10 plt.figure(2)
11 plt.scatter(x, y, marker='o')
12 plt.show()
```

At the bottom, the Run console shows the command executed and the resulting error message:

```
Run: /Users/apple/Desktop/py_code_2020/cv_show_image/venv/bin/python /Users/apple/Desktop/py_code_2020/cv_show_image/main.py
Process finished with exit code 137 (interrupted by signal 9: SIGKILL)
```

按下Project Interpreter搜尋套件
Press to search package

- ▶ Appearance & Behavior
- Keymap
- ▶ Editor
- Plugins
- ▶ Version Control
- ▼ Project: cv_show_image
 - Project Interpreter**
 - Project Structure
- ▶ Build, Execution, Deployment
- ▶ Languages & Frameworks
- ▶ Tools

Package	Version	Latest version
Pillow	8.1.0	8.1.0
cycler	0.10.0	0.10.0
kiwisolver	1.3.1	1.3.1
matplotlib	3.3.4	3.3.4
numpy	1.20.1	1.20.1
pip	19.0.3	▲ 21.0.1
pyparsing	2.4.7	2.4.7
python-dateutil	2.8.1	2.8.1
setuptools	40.8.0	▲ 53.0.0
six	1.15.0	1.15.0

+ - ▲ 🔍



Cancel

Apply

OK

matplotlib

1. Input package name

- hangar-matplotlib
- imatplotlib
- ing-theme-matplotlib
- japanize-matplotlib
- japanizematplotlib
- jirafs-matplotlib
- matplotlib**
- matplotlib-backend-qtquick
- matplotlib-chord-diagram
- matplotlib-colorbar
- matplotlib-doc-zh
- matplotlib-helper
- matplotlib-helpers
- matplotlib-hep
- matplotlib-inline
- matplotlib-label-lines
- matplotlib-pdf
- matplotlib-pgfutils
- matplotlib-scalebar
- matplotlib-stream
- matplotlib-subsets
- matplotlib-surface-plotting
- matplotlib-terminal
- matplotlib-tuda
- matplotlib-venn
- matplotlib-venn-wordcloud
- matplotlib2tikz

2. Select a package

Author
John D. Hunter, Michael Droettboom
<mailto:matplotlib-users@python.org>
<https://matplotlib.org>

3. Install

Specify version 3.4.0rc1

Options

Install Package

Manage Repositories

按右鍵，選擇run執行

The screenshot shows an IDE window with a Python file named `main.py` open. The code in the editor is:

```
1 import matplotlib.pyplot as plt
2 import matplotlib.image as mpimg
3 plt.figure(1)
4 plt.imshow(mpimg.imread('image3d.png'))
5 plt.show()
```

The file explorer on the left shows the project structure with a `venv` directory and an `image3d.png` file. A context menu is open over the code, with the `Run 'main'` option highlighted. The Run tool bar at the bottom left shows the Run button (a green play icon) is active. The terminal at the bottom displays the command:

```
/Users/apple/Desktop/py_code_2020/cv_show_image/venv/bin/python /Users/apple/Desktop/py_code_2020/cv_show_image/main.py
```

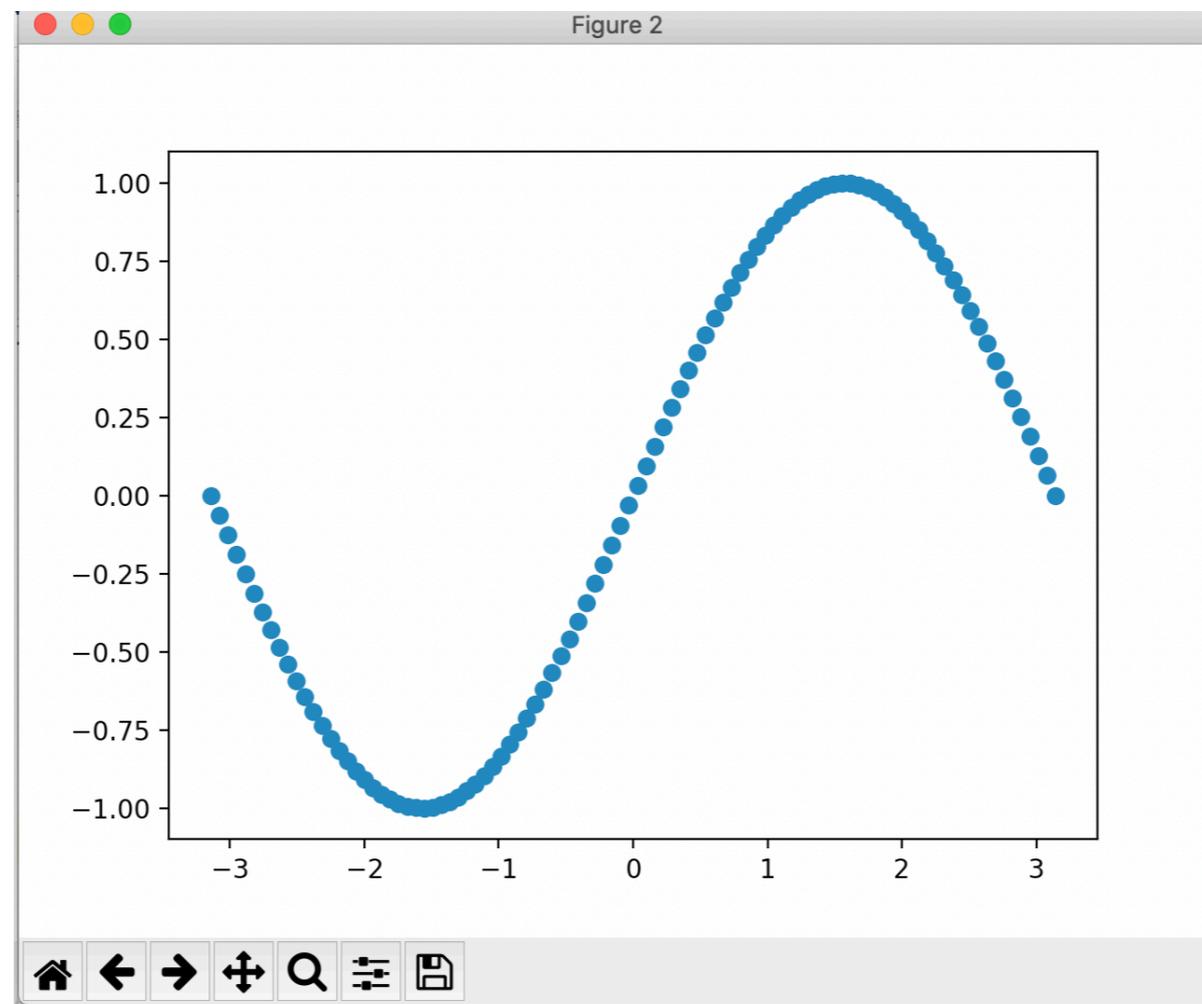
The status bar at the bottom right shows the file encoding as UTF-8, 4 spaces, and Python 3.7 (cv_show_image).

程式碼參考

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
plt.imshow(mpimg.imread('image3d.png'))
plt.imshow()
```

實驗二 (30 points) :

畫出 \sin 在 $[-\pi, \pi]$ 區間的 函數圖



範例與說明

實驗二：畫出sin在 $[-\pi, \pi]$ 區間的函數圖

- 沿用實驗一的計畫
- 安裝numpy
- 新增一個python檔案，撰寫以下程式：
 1. 匯入numpy，匯入matplotlib.pyplot
 2. 使用numpy的linspace方法將 $[-\pi, \pi]$ 切割100點，並儲存在變數 x
 3. 使用numpy的sin方法求x的sin值，並儲存在變數 y中
 4. 使用matplotlib.pyplot中的figure, scatter方法繪製x,y的點中分佈圖
 5. 使用matplotlib.pyplot的imshow()顯示圖形

程式碼參考

```
import matplotlib.pyplot as plt  
import numpy as np
```

```
x = np.linspace(-np.pi, np.pi, 100)  
y = np.sin(x)  
plt.figure(2)  
plt.scatter(x, y, marker = 'o')  
plt.show()
```

ex2022_ex1 > ex2.py

Project | ex2022_ex1 ~/Desktop/py_co

- venv
- dice.py
- ex1.py
- ex2.py
- get_an_odd_number.py
- External Libraries
- Scratches and Consoles

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 x = np.linspace(-np.pi, np.pi, 100)
5 y = np.sin(x)
6 plt.figure()
7 plt.scatter(x, y, marker = 'o')
8 plt.show()
```

Run: ex2 x

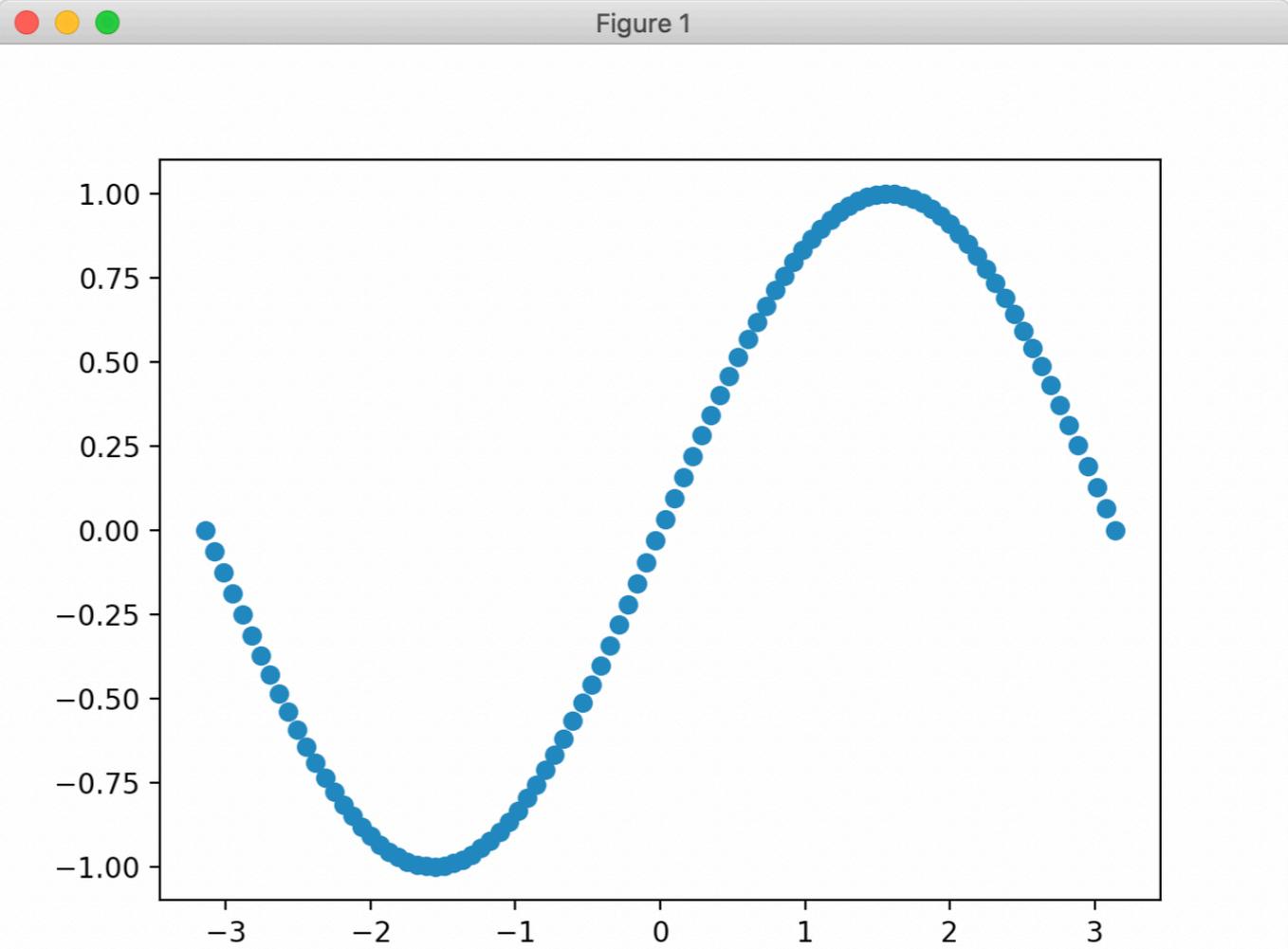
/Users/apple/Desktop/py

4: Run | 6: TODO | Terminal | Python Conso

PEP 8: no newline at end of file

plt.show()

Figure 1



skto

8

實驗三A(20 points) :
輸入一個奇數

實驗三B(20 points) :
輸入一個2或3的倍數

實驗三：輸入一個2或3的倍數

- 使用while True:迴圈，讓使用者輸入一個2或3的倍數
- 在迴圈中，撰寫以下程式：
 1. 使用input指令，讓使用者輸入一個字串
 2. 使用try except將輸入字串轉為int，轉換成功，儲存在變數num中，轉換失敗，輸出invalid，使用continue繼續執行迴圈
 3. 判斷num是否為2或3的倍數，如果是2或3的倍數，印出該數為2或3的倍數訊息，使用break中斷迴圈執行，否則印出該數不是2或3的倍數訊息

範例與說明：

本題為變化題，提供的範例為“輸入一個奇數”，請修改此範例

迴圈while True:說明

while True:

執行到continue
忽略continue後
面的指令，繼續
進行迴圈執行

...
continue

...
break

...

本例題的
while迴圈的進
入條件永遠為真

執行到break
忽略break後面
的指令，結束迴
圈執行

使用input要求使用者輸入字串

```
>>> ss = input("please input an odd number (between 0-100):")  
... print(ss)  
please input an odd number (between 0-100):>? abc  
abc
```

列印結果

輸入

```
>>> ss = input("please input an odd number (between 0-100):")  
... print(ss)  
please input an odd number (between 0-100):>? 1234  
1234
```

使用者輸入字串有兩種可能：
可轉換為整數或不可轉換為整數

使用try except進行例外處理

當int轉換失敗
執行
except :
指令

將ss的內容轉換為整數，如果轉換失敗，num的內容無定義

```
try:  
    num = int(ss)  
except:  
    print("Invalid number, please input again.")
```

```
>>> try:
...     print(int("abc"))
... except:
...     print("invalid")
...
invalid
```

轉換失敗

```
>>> try:
...     print(int("123"))
... except:
...     print("invalid")
...
123
```

轉換成功

判斷num是否為奇數，如果是奇數，印出該數為奇數訊息，使用break中斷迴圈執行，否則印出該數不是奇數訊息

num除2餘數
等於1

```
if num%2 == 1:  
    print(ss+" is odd")  
    break  
else:  
    print(ss+" is even. Input again")
```

中斷while回
圈執行

否則

參考程式碼：Get an odd number

```
while True:
    ss = input("please input an odd number (between 0-100):")
    try:
        num = int(ss)
    except:
        print("Invalid number, please input again.")
        continue
    if num%2 == 1:
        print(ss+" is odd")
        break
    else:
        print(ss+" is even. Input again")
```

執行直到迴圈進入條件不成立

將字串轉為整數

except代表轉換失敗

continue繼續到while迴圈的進入指令執行

如果餘數是1，代表奇數，break會中斷迴圈

程式輸出說明

輸入的字串
轉換為整
數，失敗

輸入的字串

```
please input an odd number (between 0-100):we2
Invalid number, please input again.
please input an odd number (between 0-100):22
22 is even. Input again
please input an odd number (between 0-100):21
21 is odd
```

輸入的字串轉
換為整數，判
斷為偶數

輸入的字串
轉換為整
數，是奇數