



# Introduction to Python for Data Science

## 资料科学入门 – 综合练习

June 2020

Microsoft Reactor | Ryan Chung

```
led by player to  
s.load_image("kg.png")  
  
[self]:  
    initialize Dog object and create Text o  
g, self).__init__(image = Dog.image  
x = games.mouse.x  
bottom = games.sc  
  
re = games.Text(value = 0, size = 24  
top = 5, right = gam  
  
reen.add(self.score)  
1 = games.Text(value = 0, size = 24  
top = 5, left = gam
```



# Ryan Chung

Instructor / DevelopIntelligence  
Founder / MobileDev.TW

@ryanchung403 on WeChat







# Reactor



[developer.microsoft.com/reactor/](https://developer.microsoft.com/reactor/)  
@MSFTReactor on Twitter

# Data Science Workshop agenda 资料科学在线研讨会议程

## Data Science Practices

## 数据科学练习

14:30	Welcome 开场
14:35	Data Science Practices - Family BMI Calculation 数据分析综合练习案例
15:10	Introduction to Matplotlib Matplotlib函数库介绍
15:30	10-minute break 中场休息
15:40	Import and Read files 档案读取与汇入
16:20	Q&A 问答与综整
16:30	Event end 研讨会结束

## 综合练习：家庭BMI值计算

## 案例：BMI值计算

1. 直接计算BMI值
2. 显示BMI值对应的结果
3. 用List建立全家人的BMI资料
4. 帮全家人检验，判断是否落于正常范围
5. 用BMI资料画图，观察一年的体重变化
6. 引用儿童的BMI指标来进行比对

# 使用Python计算BMI值

- 变数直接命名、直接给值
- `**n`：乘以自己n次

```
BMI > 01.HelloWorld.py > ...  
1 height = 1.78  
2 weight = 72.0  
3 bmi = weight / height **2
```

Python Interactive X



## Variables

Name	Type	Count	Value
bmi	float		22.724403484408533
height	float		1.78
weight	float		72.0

# BMI值标准

成人的体重分级与标准	
分 级	身体质量指数
体重过轻	$BMI < 18.5$
正常范围	$18.5 \leq BMI < 24$
过 重	$24 \leq BMI < 27$
轻度肥胖	$27 \leq BMI < 30$
中度肥胖	$30 \leq BMI < 35$
重度肥胖	$BMI \geq 35$
资料来源：食品资讯网 / 肥胖及体重控制	



# 显示判断结果

- if
- elif
- else

```
BMI > 01.HelloWorld.py > ...
1 height = 1.78
2 weight = 72.0
3 bmi = weight / height **2
4 print('你的BMI值为'+("%.2f" % bmi))
5 if bmi <18.5:
6     print('體重過輕')
7 elif 18.5 <= bmi < 24.0:
8     print('正常範圍')
9 elif 24.0 <= bmi < 27.0:
10    print('體重過重')
11 elif 27.0 <= bmi < 30.0:
12    print('輕度肥胖')
13 elif 30.0 <= bmi < 35.0:
14    print('中度肥胖')
15 else:
16    print('重度肥胖')
```

小数点后留两位数的浮点数

可以直接用一个范围来写

所有条件式不用加( )  
最后面打上冒号

```
Python Interactive X
[8] height = 1.78...
你的BMI值为22.72
正常範圍
```

# 用List建立全家的BMI资料

- 名称、身高、体重、BMI值

BMI > 02.FamilyBMI.py > ...

```
1 family_data = [  
2     ['Dad',178,72],  
3     ['Mom',155,44],  
4     ['Kid',117,19]  
5 ]  
6  
7 for each_one in family_data:  
8     this_bmi = each_one[2] / ((each_one[1]/100) **2)  
9     each_one.append(this_bmi)  
10  
11 family_data
```

计算完之后，加回原本的List

Python Interactive X

× ↶ ↷ □ ↻ 📊 📄 📁 📂

🔍

×

```
[['Dad', 178, 72, 22.724403484408533],  
 ['Mom', 155, 44, 18.314255983350673],  
 ['Kid', 117, 19, 13.879757469501062]]
```

10

HelloBMI.py

# 用List建立全家的BMI资料

BMI/02.FamilyBMI.py

- 名称、身高、体重、BMI值、BMI指标

```
10
11 for each_one in family_data:
12     if each_one[3] < 18.5:
13         bmi_index = '體重過輕'
14     elif 18.5 <= each_one[3] < 24.0:
15         bmi_index = '正常範圍'
16     elif 24.0 <= each_one[3] < 27.0:
17         bmi_index = '體重過重'
18     elif 27.0 <= each_one[3] < 30.0:
19         bmi_index = '輕度肥胖'
20     elif 30.0 <= each_one[3] < 35.0:
21         bmi_index = '中度肥胖'
22     else:
23         bmi_index = '重度肥胖'
24     each_one.append(bmi_index)
25
26 family_data
```

把判断完的结果也放回List

```
[['Dad', 178, 72, 22.724403484408533, '正常範圍'],
 ['Mom', 155, 44, 18.314255983350673, '體重過輕'],
 ['Kid', 117, 19, 13.879757469501062, '體重過輕']]
```

# 用List建立全家的BMI资料

- 名称、身高、体重、BMI值、BMI指标、是否落于正常范围

```
25
26 for each_one in family_data:
27     if each_one[4] == '正常範圍':
28         is_normal = True
29     else:
30         is_normal = False
31     each_one.append(is_normal)
32
33 family_data
```

增加一个布尔值，来记录是否落在正常范围

```
[['Dad', 178, 72, 22.724403484408533, '正常範圍', True],
 ['Mom', 155, 44, 18.314255983350673, '體重過輕', False],
 ['Kid', 117, 19, 13.879757469501062, '體重過輕', False]]
```

# 前面用到的语法

- 资料型态
  - 字符串、整数、浮点数、布尔值、list
- 运算符
  - 等于、大于、大于等于、小于等于、小于
- 流程控制
  - 回圈
  - if..else if..else
- List 运算
  - 新增元素

# 检查资料型态

```
32
33 for each_one in family_data[0]:
34     print(type(each_one))
```

用type() 检查资料型态

0	1	2	3	4	5
Dad	178	72	22.7244034844	正常範圍	true

```
<class 'str'>
<class 'int'>
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>
```



# 观察list中的部分元素

- 起始 : 结束
- 取出元素不包含结束
- 不写开头，代表从最前面开始
- 不写结尾，代表从那一项开始走到全部结束

family\_data

全部的资料

```
[[ 'Dad', 178, 72, 22.724403484408533, '正常範圍', True],  
 [ 'Mom', 155, 44, 18.314255983350673, '體重過輕', False],  
 [ 'Kid', 117, 19, 13.879757469501062, '體重過輕', False]]
```

family\_data[0:2]

第0笔、第1笔

```
[[ 'Dad', 178, 72, 22.724403484408533, '正常範圍', True],  
 [ 'Mom', 155, 44, 18.314255983350673, '體重過輕', False]]
```

family\_data[1:]

第1笔、第2笔

```
[[ 'Mom', 155, 44, 18.314255983350673, '體重過輕', False],  
 [ 'Kid', 117, 19, 13.879757469501062, '體重過輕', False]]
```

family\_data[:2]

第0笔、第1笔

```
[[ 'Dad', 178, 72, 22.724403484408533, '正常範圍', True],  
 [ 'Mom', 155, 44, 18.314255983350673, '體重過輕', False]]
```

# 使用numpy array

- Element-Wise Calculation
- 可计算平均数、中位数、是否相关连、标准差
- 储存相同资料型态

# 将原本的数学资料搬至阵列，计算全家人的统计数字

把相同的资料型態抽出，放在一起

```
42 import numpy as np
43 family_name_array = np.array([family_data[0][0],family_data[1][0],family_data[2][0]])
44 family_height_array = np.array([family_data[0][1],family_data[1][1],family_data[2][1]])
45 family_weight_array = np.array([family_data[0][2],family_data[1][2],family_data[2][2]])
46 family_bmi_array = np.array([family_data[0][3],family_data[1][3],family_data[2][3]])
47 family_bmi_index_array = np.array([family_data[0][4],family_data[1][4],family_data[2][4]])
48 np.mean(family_height_array)
49 np.median(family_weight_array)
50 np.corrcoef(family_weight_array, family_bmi_array)[1,0]
```

计算平均数

```
np.mean(family_height_array)
```

**150.0**

计算中位数

```
np.median(family_weight_array)
```

**44.0**

计算关联性

```
np.corrcoef(family_weight_array, family_bmi_array)[1,0]
```

**0.9994132475732211**

加上[1,0]取出第1个物件中的第0个

```
array([[1.          , 0.99941325],
       [0.99941325, 1.          ]])
```

# Matplotlib 入门

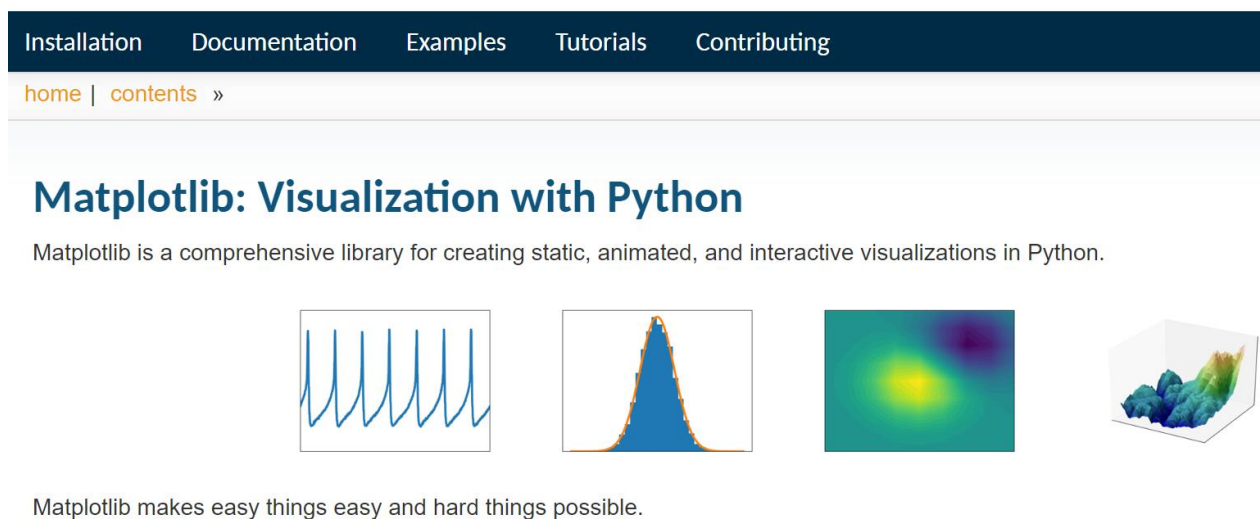
# Matplotlib Overview 功能综览

- Import & Setup 开始使用及设定
- 绘制
  - scatter() 点
  - xlabel(), ylabel(), title() 轴线标题、主标题
  - plot() 折线
  - fill\_between() 着色
  - xticks(), yticks() x轴刻度、y轴刻度

# Matplotlib Overview 功能综览

## 为什么要使用Matplotlib?

- 建立2维影像绘图
- 让Python有一套像是MATLAB的绘图界面
- 可以输出常见格式(PDF, SVG, JPG, PNG, BMP, GIF)



Matplotlib makes easy things easy and hard things possible.

<https://matplotlib.org/>



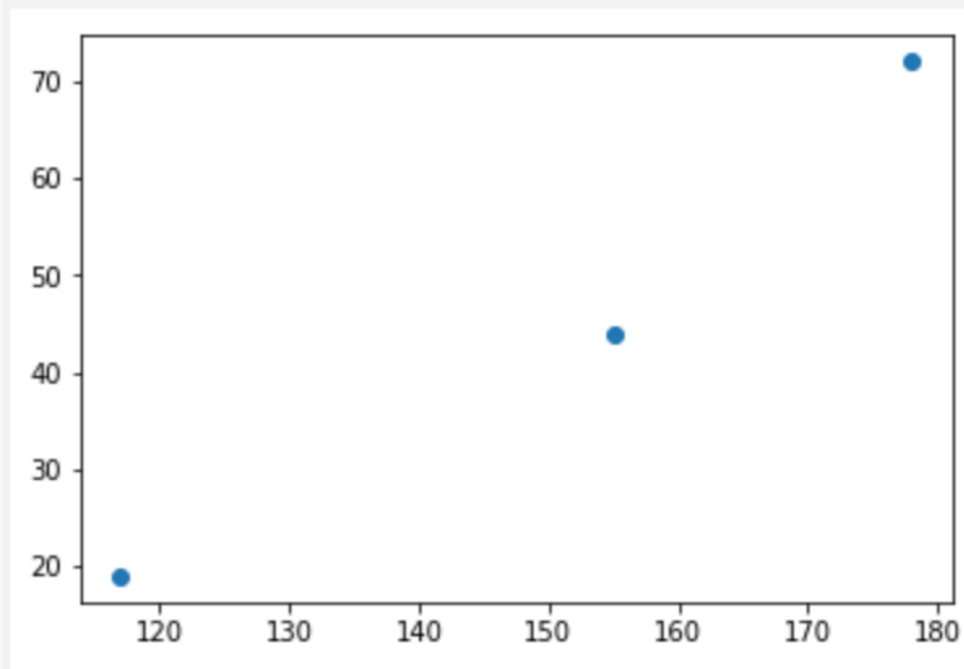
# 画图

```
49 import matplotlib.pyplot as plt
50 %matplotlib inline
51 plt.scatter(family_height_array, family_weight_array)
52 plt.show()
```

让图案画在互动模式中，执行一次即可，执行后可批注

把身高体重当作 x, y 画上去

Python Interactive X

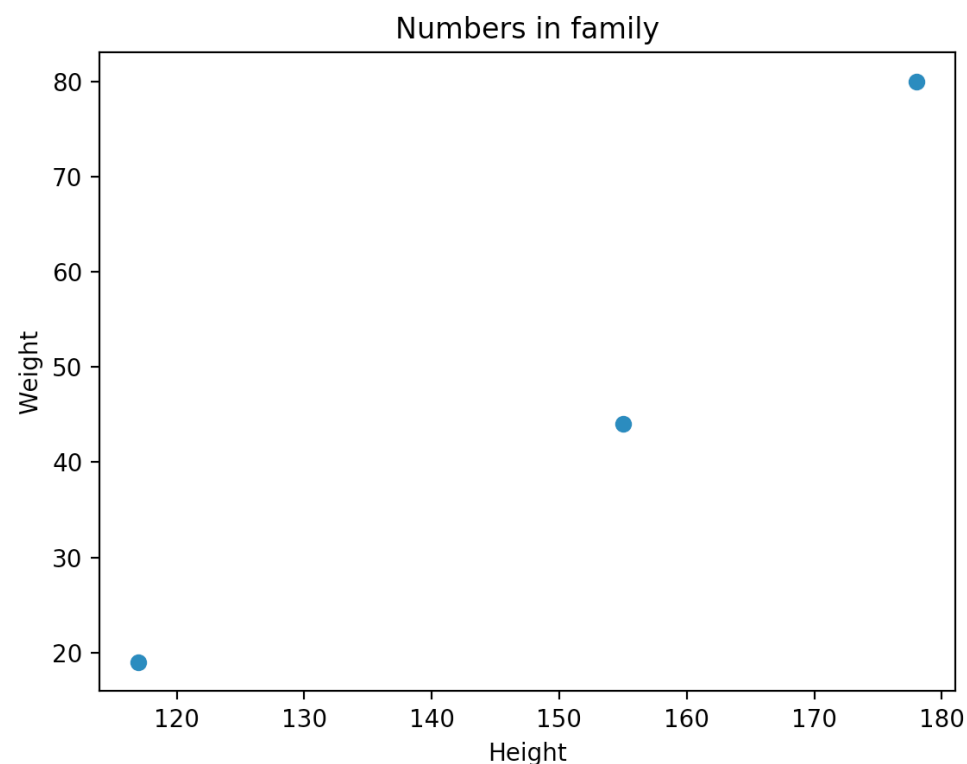


# 加上标签

BMI/02.FamilyBMI.py

```
49 import matplotlib.pyplot as plt
50 %matplotlib inline
51 plt.scatter(family_height_array, family_weight_array)
52 plt.xlabel('Height')
53 plt.ylabel('Weight')
54 plt.title('Numbers in family')
55 plt.show()
```

X轴 : xlabel()  
Y轴 : ylabel()  
标题 : title()



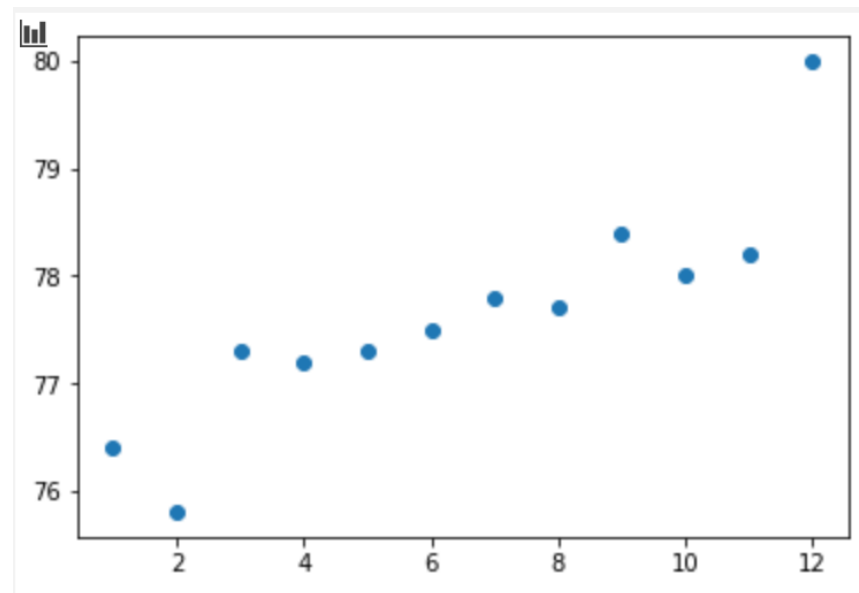
# 观察家人一年的体重变化

BMI/03.TrendGraph.py

BMI > 03.TrendGraph.py > ...

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 #產生月份數字 [1,2,3,4,5,6,7,8,9,10,11,12]
4 month_array = np.arange(1,13)
5 dad_weight_history_array = np.array([76.4, 75.8, 77.3, 77.2, 77.3, 77.5,
6 | | | | | | | | | | | | | 77.8, 77.7, 78.4, 78.0, 78.2, 80.0])
7 #畫出點
8 plt.scatter(month_array, dad_weight_history_array)
```

組合月份和每月体重数字，画在图表上

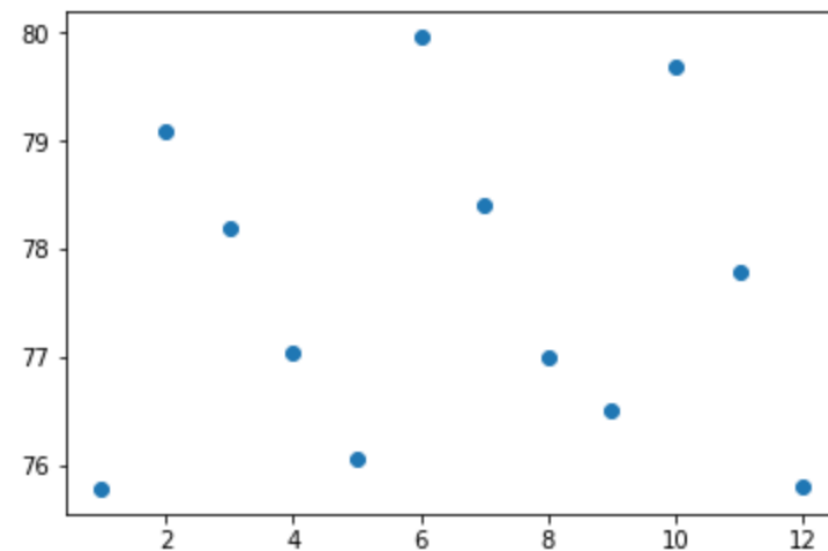


# 观察家人一年的体重变化 – 随机产生资料

BMI/03.TrendGraph.py

```
#月份:1~12
month_array = np.arange(1,13)
#产生12笔资料, 随机在75.5~80.0之间
dad_weight_history_array = []
for i in range(0,12):
    #指定范围、到小数点后几位数
    x = round(np.random.uniform(75.5, 80.0),2)
    dad_weight_history_array.append(x)

plt.scatter(month_array, dad_weight_history_array)
```



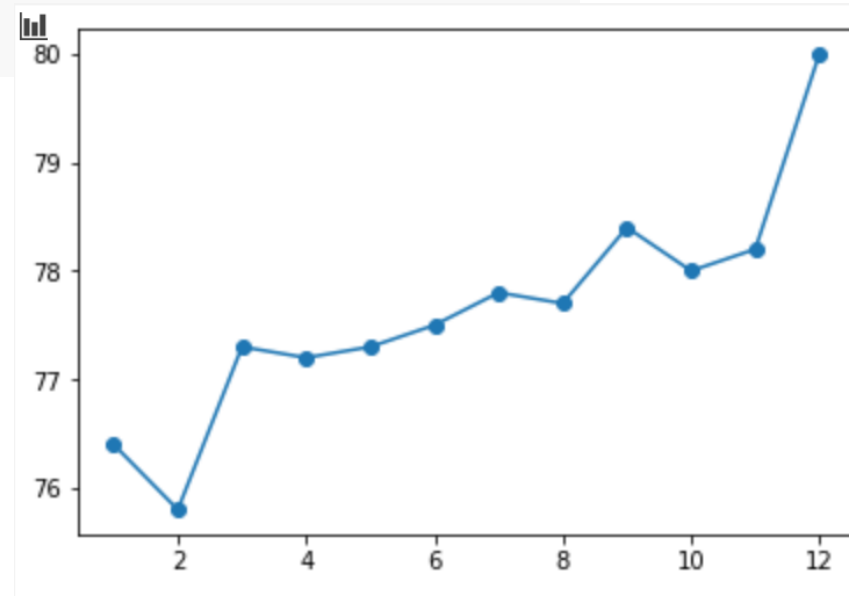
# 观察家人一年的体重变化

BMI/03.TrendGraph.py

BMI > 03.TrendGraph.py > ...

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 #產生月份數字 [1,2,3,4,5,6,7,8,9,10,11,12]
4 month_array = np.arange(1,13)
5 dad_weight_history_array = np.array([76.4, 75.8, 77.3, 77.2, 77.3, 77.5,
6 | | | | | | | | | | 77.8, 77.7, 78.4, 78.0, 78.2, 80.0])
7 #畫出點
8 plt.scatter(month_array, dad_weight_history_array)
9 #畫出折線
10 plt.plot(month_array, dad_weight_history_array)
```

加上折线 plot()



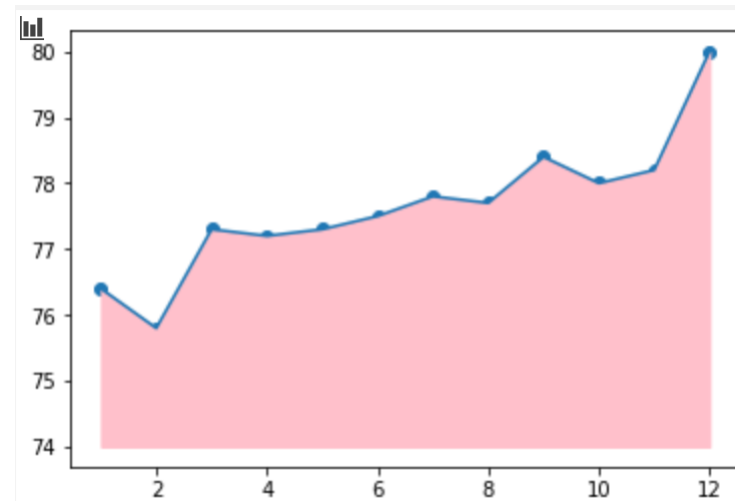
# 观察家人一年的体重变化

BMI/03.TrendGraph.py

BMI > 03.TrendGraph.py > ...

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 #產生月份數字 [1,2,3,4,5,6,7,8,9,10,11,12]
4 month_array = np.arange(1,13)
5 dad_weight_history_array = np.array([76.4, 75.8, 77.3, 77.2, 77.3, 77.5,
6                                     77.8, 77.7, 78.4, 78.0, 78.2, 80.0])
7 #畫出點
8 plt.scatter(month_array, dad_weight_history_array)
9 #畫出折線
10 plt.plot(month_array, dad_weight_history_array)
11 #塗上顏色
12 plt.fill_between(month_array, dad_weight_history_array, 74, color='pink')
```

再加上颜色-粉红色，同时设定了Y轴的起始值





# 观察家人一年的体重变化

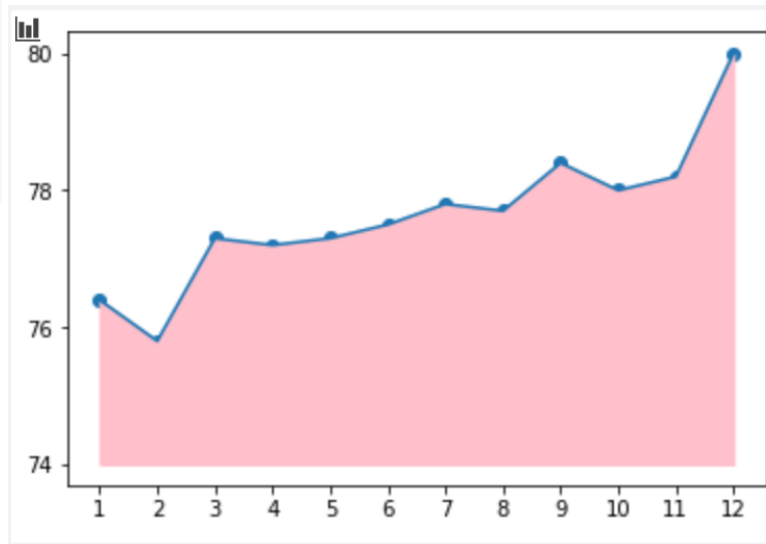
BMI/03.TrendGraph.py

BMI > 03.TrendGraph.py > ...

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 %matplotlib inline
4 #產生月份數字 [1,2,3,4,5,6,7,8,9,10,11,12]
5 month_array = np.arange(1,13)
6 dad_weight_history_array = np.array([76.4, 75.8, 77.3, 77.2, 77.3, 77.5,
7                                     77.8, 77.7, 78.4, 78.0, 78.2, 80.0])
8 #畫出點
9 plt.scatter(month_array, dad_weight_history_array)
10 #畫出折線
11 plt.plot(month_array, dad_weight_history_array)
12 #塗上顏色
13 plt.fill_between(month_array, dad_weight_history_array, 74, color='pink')
14 #決定X軸每個月份都顯示, y軸只顯示四個數字
15 plt.xticks(month_array)
16 plt.yticks([74, 76, 78, 80])
```

改变显示的刻度，X轴每个月份都显示、Y轴只显示74, 76, 78, 80  
练习：用np.arange来产生这4个数字

```
plt.yticks(np.arange(74,81,2))
```

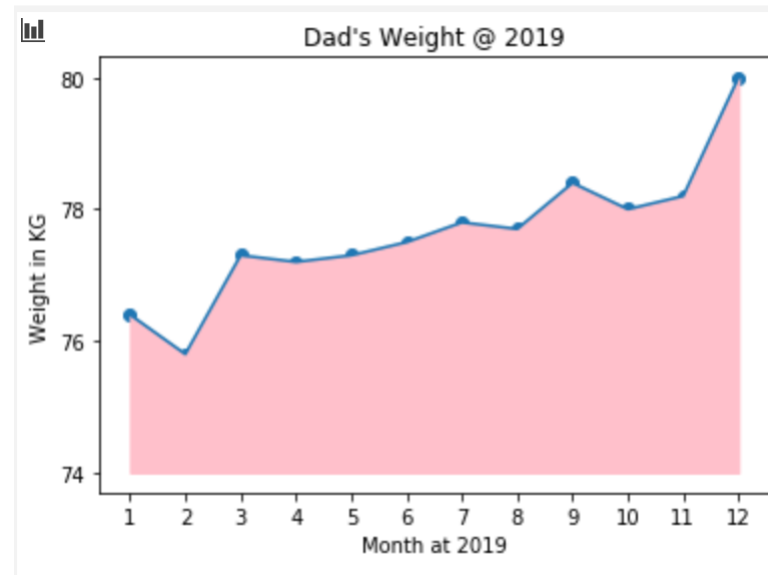


# 观察家人一年的体重变化

BMI/03.TrendGraph.py

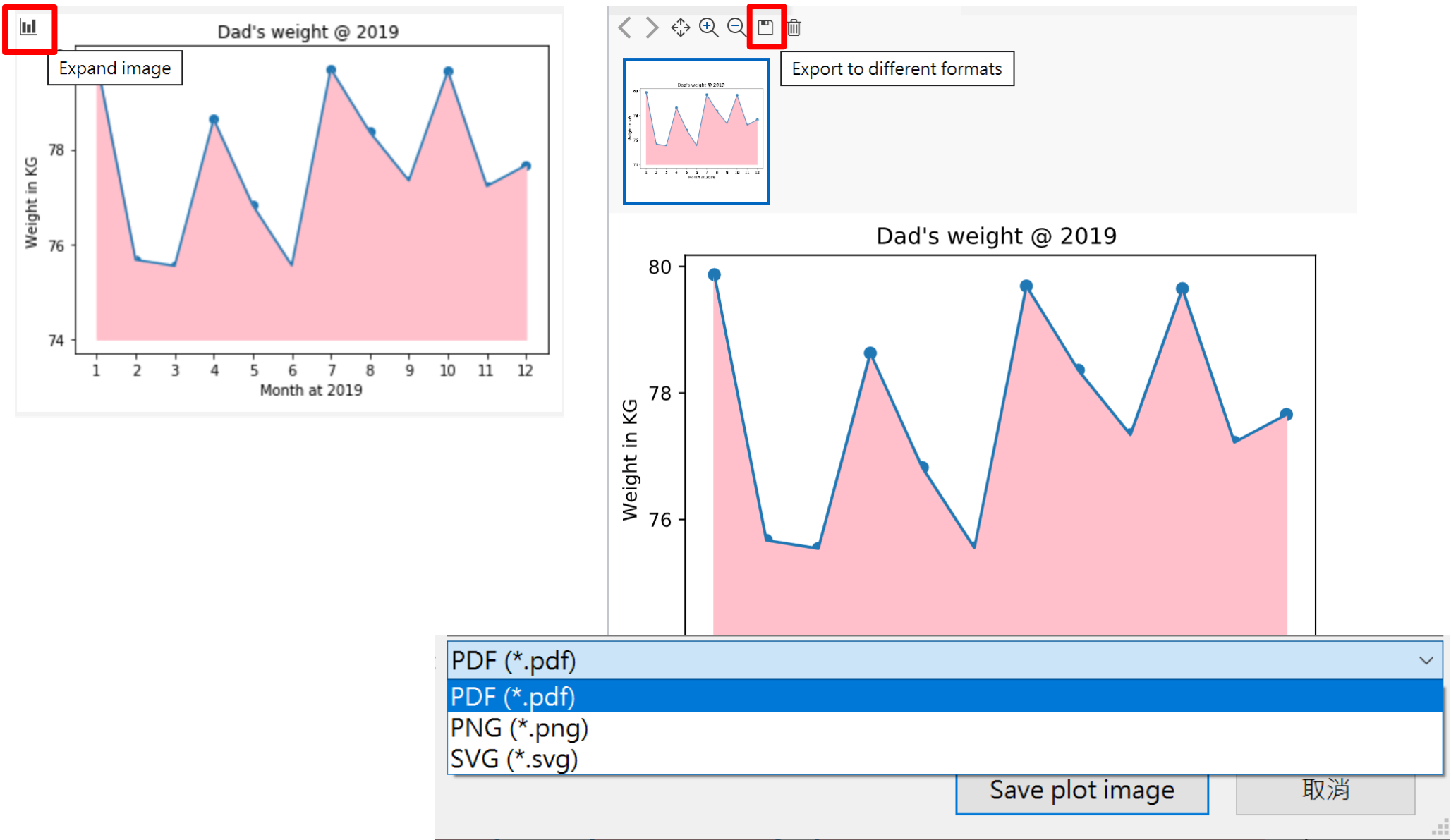
BMI > 03.TrendGraph.py > ...

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 %matplotlib inline
4 #產生月份數字 [1,2,3,4,5,6,7,8,9,10,11,12]
5 month_array = np.arange(1,13)
6 dad_weight_history_array = np.array([76.4, 75.8, 77.3, 77.2, 77.3, 77.5,
7 | | | | | | | | | | 77.8, 77.7, 78.4, 78.0, 78.2, 80.0])
8 #畫出點
9 plt.scatter(month_array, dad_weight_history_array)
10 #畫出折線
11 plt.plot(month_array, dad_weight_history_array)
12 #塗上顏色
13 plt.fill_between(month_array, dad_weight_history_array, 74, color='pink')
14 #決定x軸每個月份都顯示, y軸只顯示四個數字
15 plt.xticks(month_array)
16 plt.yticks([74, 76, 78, 80])
17 #加上x軸標題、y軸標題、大標題
18 plt.xlabel('Month at 2019')
19 plt.ylabel('Weight in KG')
20 plt.title("Dad's Weight @ 2019")
21 #顯示圖表-非互動模式才需要
22 plt.show()
```



# 图表导出 – VS Code

BMI/03.TrendGraph.py



# 幼童的BMI值标准不同

· 儿童与青少年生长身体质量指数(BMI)建议值

	男性			女性		
年齡	正常範圍	過重	肥胖	正常範圍	過重	肥胖
	(BMI介於)	(BMI≥)	(BMI≥)	(BMI介於)	(BMI≥)	(BMI≥)
0	11.5-14.8	14.8	15.8	11.5-14.7	14.7	15.5
0.5	15.2-18.9	18.9	19.9	14.6-18.6	18.6	19.6
1	14.8-18.3	18.3	19.2	14.2-17.9	17.9	19
1.5	14.2-17.5	17.5	18.5	13.7-17.2	17.2	18.2
2	14.2-17.4	17.4	18.3	13.7-17.2	17.2	18.1
2.5	13.9-17.2	17.2	18	13.6-17.0	17	17.9
3	13.7-17.0	17	17.8	13.5-16.9	16.9	17.8

# 更新儿童的BMI指标与是否标准

- 在资料中增加性别、年龄、称谓
- 判断年龄是否小于18岁
- 依年龄、性别比对BMI标准值



31

稍后请记得填写课程回馈问卷  
<https://aka.ms/ReactorFeedback>

# 汇入比对的标准资料

59

import pandas as pd

透过pandas的read\_csv() 方法，将资料读入转换成DataFrame格式

60

kid\_bmi\_standard\_english = pd.read\_csv('bmirecommendation-English.csv')

61

kid\_bmi\_standard\_english.head()

利用 head() 方法，取出前面5笔

62

63

family\_gender\_array = np.array(['Male', 'Female', 'Female'])

64

family\_age\_array = np.array([38, 37, 6])

	Age	Normal Male Should Between	Male Over Weight if Higher	Male Fat if Higher	Normal Female Should Between	Female Over Weight if Higher	Female Fat if Higher
0	0.0	11.5-14.8	14.8	15.8	11.5-14.7	14.7	15.5
1	0.5	15.2-18.9	18.9	19.9	14.6-18.6	18.6	19.6
2	1.0	14.8-18.3	18.3	19.2	14.2-17.9	17.9	19.0
3	1.5	14.2-17.5	17.5	18.5	13.7-17.2	17.2	18.2
4	2.0	14.2-17.4	17.4	18.3	13.7-17.2	17.2	18.1



# 汇入比对的标准资料

```
for this_index, each_age in enumerate(family_age_array):
    if each_age < 18:
        #看看是第几个未满18岁
        print('第'+str(this_index)+'个的年龄是'+str(each_age)+'小于18岁')
        #找出比对数据的该年龄那一列
        only_this_age = kid_bmi_standard_english['Age']==each_age
        this_bmi_standard = kid_bmi_standard_english[only_this_age]
        #取得这一个未满18岁孩子的性别
        this_gender = family_gender_array[this_index]
        #只取出该性别的那三项指标(会从Normal Male/Female Should Between开始)
        this_bmi_standard_gender = this_bmi_standard.loc[:, 'Normal '+this_gender+' Should Between':this_gender+" Fat if Higher"]
        #开始跟筛选过的DataFrame进行比对

        if family_bmi_array[this_index]>float(this_bmi_standard_gender.iloc[0,2]):
            family_bmi_index_array[this_index] = '肥胖' #Fat if Higher
        elif family_bmi_array[this_index]>float(this_bmi_standard_gender.iloc[0,1]):
            family_bmi_index_array[this_index] = '体重过重' #Over Weight if Higher
        elif family_bmi_array[this_index]>float(this_bmi_standard_gender.iloc[0,0][:4]):
            family_bmi_index_array[this_index] = '正常范围' #Normal Should Between
        else:
            family_bmi_index_array[this_index] = '体重过轻'

family_bmi_index_array
```

Age		Normal Male Should Between	Male Over Weight if Higher	Male Fat if Higher	Normal Female Should Between	Female Over Weight if Higher	Female Fat if Higher
12	6.0	13.5-16.9	16.9	18.5	13.1-17.2	17.2	18.8

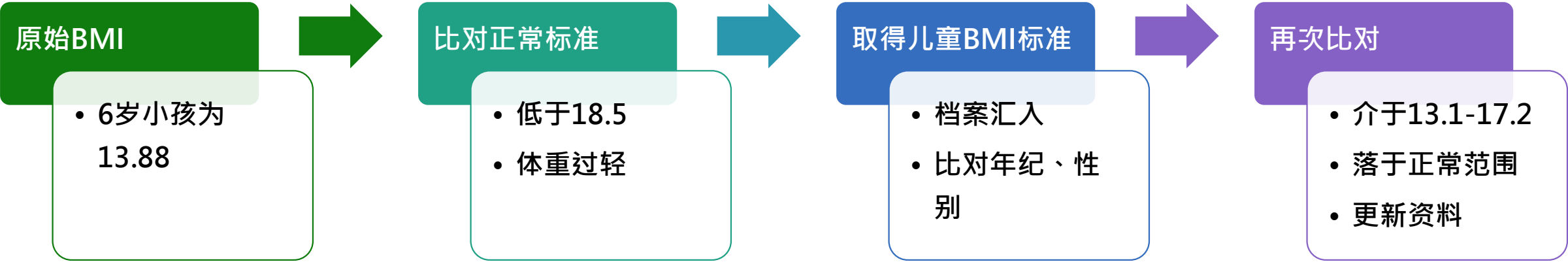
this\_bmi\_standard

		Normal Female Should Between	Female Over Weight if Higher	Female Fat if Higher
12		13.1-17.2	17.2	18.8

this\_bmi\_standard\_gender

['正常範圍' '體重過輕' '正常範圍']  
family\_bmi\_index\_array 33

# 资料流程



成人的体重分级与标准	
分 级	身体质量指数
体重过轻	BMI < 18.5
正常范围	$18.5 \leq \text{BMI} < 24$
过 重	$24 \leq \text{BMI} < 27$
轻度肥胖	$27 \leq \text{BMI} < 30$
中度肥胖	$30 \leq \text{BMI} < 35$
重度肥胖	$\text{BMI} \geq 35$
资料来源：食品资讯网 / 肥胖及体重控制	

青少年与儿童 女性BMI 标准			
年龄	正常范围	体重过重	肥胖
6	13.1 – 17.2	17.2	18.8

34

小孩的BMI指标已更新！

# 转换成DataFrame来显示

```
90
91 family_data_frame = pd.DataFrame(data=family_bmi_index_array,
92 | | | | | | | | | | index= family_name_array,
93 | | | | | | | | | | columns = ['BMI指標'])
94 family_data_frame['年齡'] = family_age_array
95 family_data_frame
```

决定资料、索引值、栏位名称

新增资料至DataFrame

	BMI指標	年齡
Dad	正常範圍	38
Mom	體重過輕	37
Kid	正常範圍	6

# 综合练习重点提示

- Python Syntax
  - 资料型态、运算符、流程控制、List 运算
- Numpy Array
  - 平均数、中位数、关联性、同质储存
- DataFrame
  - 汇整、新增、筛选、比对
- Matplotlib
  - 绘图、点、折线、涂色、XY轴标签、范围
- 档案运用
  - 汇入、查询



# Reactor



[developer.microsoft.com/reactor/](https://developer.microsoft.com/reactor/)  
@MSFTReactor on Twitter

# 议程结束 感谢聆听



请记得填写课程回馈问卷  
<https://aka.ms/ReactorFeedback>

© 2019 Microsoft Corporation. All rights reserved. The text in this document is available under the Creative Commons Attribution 3.0 License, additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are not included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.