

102

```
# 載入 xml.etree.ElementTree 模組並縮寫為 ET
import xml.etree.ElementTree as ET

# 載入 csv 模組
import csv

# 讀取 xml
tree = ET.parse("read.xml")
root = tree.getroot()
ubikefile = open("write.csv", "w", encoding='utf8')
csvwriter = csv.writer(ubikefile)
for row in root:
    ubike = []
    sno = row.find('sno').text
    ubike.append(sno)
    sna = row.find('sna').text
    ubike.append(sna)
    tot = row.find('tot').text
    ubike.append(tot)
    csvwriter.writerow(ubike)
ubikefile.close()
```

```
# 載入 xml.etree.ElementTree 模組並縮寫為 ET
import __ as __
# 載入 csv 模組
import __

# 讀取 xml
tree = __.__(__)
root = tree.getroot()

# 寫入 csv 檔案，編碼設定為 utf8
ubikefile = __(__, __, encoding=__)
csvwriter = csv.writer(ubikefile)

# 將其中 sno (站點代號)、sna (中文場站名稱)、tot (場站總停車格) 等三個欄位寫出
for row in root:
    ubike = []
    sno = row.find(__).text
    ubike.append(__)
    sna = row.find(__).text
    ubike.append(__)
    tot = row.find(__).text
    ubike.append(__)
    csvwriter.writerow(ubike)
ubikefile.close()
```

104

```
# 載入 json 模組
import json

people=[{'id': '1','name': 'Peter','country': 'Taiwan'},{'id': '2','name': 'Jack','country': 'USA'}, {"id": "3", "name": "Cindy", "country": "Japan"}]

peopleB={'people':people}

# 將資料寫入 json 檔案
with open('write.json', 'w') as outfile:
    json.dump(peopleB, outfile)
```

```
# 載入 json 模組
import __

# 建立資料
# 'id': '1'
# 'name': 'Peter'
# 'country': 'Taiwan'
#
# 'id': '2'
# 'name': 'Jack'
# 'country': 'USA'
#
# 'id': '3'
# 'name': 'Cindy'
# 'country': 'Japan'

# 將資料寫入json檔案
with __( '___', '___') as outfile:
    json.dump(__, __)
```

204

```
# 載入 requests 模組
```

```
import requests
```

```
# 載入 json 模組
```

```
import json
```

```
# 開放資料連結
```

```
url = 'http://tqc.codejudge.com:3000/target/5204.json'
```

```
# 發出 HTTP GET 請求
```

```
res = requests.get(url)
```

```
# 將回傳結果轉換成標準 JSON 格式
```

```
data = json.loads(res.text)
```

```
# 輸出新北市大專院校名單
```

```
print('新北市大專院校名單 : \n')
```

```
for record in data:
```

```
    if record['type'] == '大專院校':
```

```
        print('名稱 : {}'.format(record['name']))
```

```
        print('地址 : {}'.format(record['address']))
```

```
        print('聯絡電話 : {}'.format(record['tel']))
```

```
        print('網站 : {}'.format(record['website']))
```

```
        print('資料更新時間 : {}'.format(record['update_date']))
```

```
        print()
```

```
# 載入 requests 模組
import __
# 載入 json 模組
import __

# 開放資料連結
url = '____'
# 以 requests 模組發出 HTTP GET 請求
res = ____.(url)

# 將回傳結果轉換成標準JSON格式
data = json.loads(res.text)

# 輸出新北市大專院校名單
print('新北市大專院校名單 : \n')
for record in data:
    if record['type'] == '大專院校':
        print('名稱 : %s' % record['____'])
        print('地址 : %s' % record['____'])
        print('聯絡電話 : %s' % record['____'])
        print('網站 : %s' % record['____'])
        print('資料更新時間 : %s' % record['____'])
        print()
```

302

```

# --開始--批改評分使用，請勿變動
set_seed = 123
# --結束--批改評分使用，請勿變動

import numpy as np

x = np.random.RandomState(set_seed).randint(low=5, high=16, size=15)
print('隨機正整數：', x)

x = x.reshape(3, 5)
print('X 矩陣內容：')
print(x)
print('最大：', x.max())
print('最小：', x.min())
print('總和：', x.sum())

print('四個角落元素：')
print(x[np.ix_([0, 2], [0, 4])])

```

```

# --開始--批改評分使用，請勿變動
set_seed = 123
# --結束--批改評分使用，請勿變動

import numpy as np

x = np.random.RandomState(set_seed).randint(low=5, high=16, size=15)
print('隨機正整數：', __)

x = x.reshape(__,__)
print('X矩陣內容：')
print(__)
print('最大：', __)
print('最小：', __)
print('總和：', __)
print('四個角落元素：')
print(x[np.ix_(__, __), __, __]))
```

304

```
# 載入 numpy 模組縮寫為 np
import numpy as np

# 載入 pandas 模組縮寫為 pd
import pandas as pd

# 使用 pd 的 read_csv 方法讀入 read.csv 檔案
df = pd.read_csv("read.csv")

# 將 df 中的 data 欄位取出為 ndarray
data = df["data"].values

# 使用 type 函數判斷資料型態
print('資料型態 : %s' % type(data))

# 使用 np.mean() 函數計算平均數
print('平均值 : %.2f' % np.mean(data))

# 使用 np.median() 函數計算中位數
print('中位數 : %.2f' % np.median(data))

# 使用 np.std() 函數計算標準差
print('標準差 : %.2f' % np.std(data))

# 使用 np.var() 函數計算變異數
print('變異數 : %.2f' % np.var(data))

# 使用 np.ptp() 函數計算極差值
print('極差值 : %.2f' % np.ptp(data))
```

```
# 載入 numpy 模組
# 載入 pandas 模組縮
# 讀入 read.csv 檔案

# 判斷資料型態
print('資料型態 : %s' % ____())
# 計算平均數
print('平均值 : %.2f' % ____())
# 計算中位數
print('中位數 : %.2f' % ____())
# 計算標準差
print('標準差 : %.2f' % ____())
# 計算變異數
print('變異數 : %.2f' % ____())
# 計算極差值
print('極差值 : %.2f' % ____())
```

402

--開始--批改評分使用，請勿變動

```
import matplotlib as mpl
mpl.use('Agg')
```

--結束--批改評分使用，請勿變動

載入 matplotlib.pyplot 並縮寫為 plt

```
import matplotlib.pyplot as plt
```

載入 csv 模組

```
import csv
```

```
x = []
```

```
y = []
```

讀入 read.csv

```
with open('read.csv', 'r', encoding='utf8') as csvfile:
```

```
    plots = csv.reader(csvfile, delimiter=',')
```

```
    for row in plots:
```

```
        x.append(row[0])
```

```
        y.append(float(row[1])))
```

```
x_ticks = range(1, len(x) + 1)
```

```
plt.plot(x_ticks, y, label='banana')
```

```
plt.xticks(x_ticks, x)
```

```
plt.xlabel('date')
```

```
plt.ylabel('NT$')
```

```
plt.ylim([15, 25])
```

添加圖表標題 title()

```
plt.title('Market Average Price')
```

```
plt.legend()
```

使用 savefig() 函數

```
plt.savefig('chart.png')
```

```
plt.close()
```

```
# --開始--批改評分使用，請勿變動
import matplotlib as mpl
mpl.use('Agg')
# --結束--批改評分使用，請勿變動

# 載入 matplotlib.pyplot 並縮寫為 plt
import __ as __
# 載入 csv 模組
import __

x = []
y = []

# 讀入 read.csv
with open(__, 'r', encoding='utf8') as csvfile:
    plots = csv.reader(csvfile, delimiter=',')
    for row in plots:
        x.append(row[0])
        y.append(float(row[1]))

x_ticks = range(1, len(x) + 1)
plt.__(x_ticks, y, label=__)
plt.xticks(x_ticks, x)
plt.xlabel(__)
plt.ylabel(__)
plt.ylim(__)
# 添加圖表標題 title()
plt.___('Market Average Price')
plt.legend()
# 使用 savefig() 函數
plt.___('chart.png')
plt.close()
```

404

```
# --開始--批改評分使用，請勿變動
import matplotlib as mpl
mpl.use('Agg')

# --結束--批改評分使用，請勿變動

from matplotlib import pyplot as plt
import numpy as np
import pandas as pd

# 讀取學生分數資料

# 使用 pd.read_csv 讀取 read.csv
df = pd.read_csv('read.csv')
scores = df["scores"].values

# range_count[0]: range0~19
# range_count[1]: range20~39
# range_count[2]: range40~59
# range_count[3]: range60~79
# range_count[4]: range80~100

# 以 0 初始化計數串列
range_count = [0] * 5

# 計數過程
for score in scores:
    if score < 20:
        range_count[0] += 1
    elif score < 40:
        range_count[1] += 1
    elif score < 60:
        range_count[2] += 1
    elif score < 80:
        range_count[3] += 1
    else:
        range_count[4] += 1
```

```
# --開始--批改評分使用，請勿變動
import matplotlib as mpl
mpl.use('Agg')
# --結束--批改評分使用，請勿變動

from matplotlib import pyplot as plt
import numpy as np
import pandas as pd

# 讀取學生分數資料
# 讀取 read.csv
df = __(__)
scores = df["__"].values

# range_count[0]: range0~19
# range_count[1]: range20~39
# range_count[2]: range40~59
# range_count[3]: range60~79
# range_count[4]: range80~100
# 以0初始化計數串列
range_count = [0] * 5

# 計數過程
for score in scores:
    if score < 20:
        range_count[0] += 1
    elif score < 40:
        range_count[1] += 1
    elif score < 60:
        range_count[2] += 1
    elif score < 80:
        range_count[3] += 1
    else:
        range_count[4] += 1

# y軸標籤
index = np.arange(__, __, __)
# X軸刻度
labels = [__, __, '40~59', __, '80~100']
# 畫出長條圖
plt.bar(__, range_count, __)
# 設定X軸名稱
plt.xlabel(__, fontsize=__)
# 設定Y軸名稱
plt.ylabel(__, fontsize=__)
# 設定x軸標籤
plt.xticks(index, labels)
# 設定y軸標籤
plt.yticks(index)
# 設定圖名稱
plt.title(__, fontsize=__)
# 輸出圖片檔案
plt.__(__)
plt.close()
```

```

        range_count[3] += 1
else:
    range_count[4] += 1

# y 軸標籤
index = np.arange(0, 25, 5)

# x 軸標籤
labels = ['0~19', '20~39', '40~59', '60~79', '80~100']

# 畫出長條圖
plt.bar(index, range_count, width=2)

# 設定 x 軸名稱
plt.xlabel('Range', fontsize=14)

# 設定 y 軸名稱
plt.ylabel('Quantity', fontsize=14)

# 設定 x 軸標籤
plt.xticks(index, labels)

# 設定 y 軸標籤
plt.yticks(index)

# 設定圖名稱
plt.title('Score ranges count', fontsize=20)

# 輸出圖片檔案
plt.savefig('chart.png')
plt.close()

```

```

# y軸標籤
index = np.arange(__, __, __)
# X軸刻度
labels = [__, __, '40~59', __, '80~100']
# 畫出長條圖
plt.bar(__, range_count, __)
# 設定X軸名稱
plt.xlabel('__', fontsize=__)
# 設定Y軸名稱
plt.ylabel('__', fontsize=__)
# 設定x軸標籤
plt.xticks(index, labels)
# 設定y軸標籤
plt.yticks(index)
# 設定圖名稱
plt.title('__', fontsize=__)
# 輸出圖片檔案
plt.__('__')
plt.close()

```