

# Python

串列(list)的使用

# 班級成員會變動

- 有新加入的轉學生
- 有轉學離開的學生
- 整班學生畢業，換下一屆學生
- ....

串列呢？



寫一個程式處理以下生活細節：

**小列**在便利商店購買水果，  
共買了三樣水果: 蘋果, 香蕉, 櫻桃

**小列**結帳時改變心意，  
不買香蕉了，改買草莓，並再加買個橘子，

**小列**要確認是否真的有移除香蕉改拿草莓，  
最後結帳時，當店員輸入蘋果價格後，

**小列**臨時插入多買一個芒果，  
請問**小列**最終買了幾樣水果？

問題



# 串列記錄生活 – 購物篇

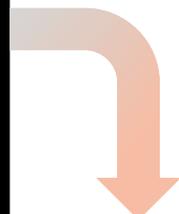
```
# 小列買了三樣水果：蘋果，香蕉，櫻桃
buylist = ['蘋果', '香蕉', '櫻桃']

# 不買香蕉，改買草莓
buylist[1] = '草莓'

# 並再加買個橘子，
buylist.append('橘子')

# 小列要確認是否真的有移除香蕉改拿草莓，
print('香蕉' in buylist)
print('草莓' in buylist)
# 當店員輸入蘋果價格後，小列臨時插入多買一個芒果，
buylist.insert(1, '芒果')

# 請問小列最終買了幾樣水果？
print(len(buylist))
print(buylist)
```



執行結果

```
False
True
5
['蘋果', '芒果', '草莓', '櫻桃', '橘子']
```

# 串列記錄生活 – 購物篇

## 串列名稱[索引值]=新值

將串列此索引位置的項目  
質改為新值

## 找尋的值 in 串列名稱

確認在串列裡面是否存  
在要找尋的值

## len(串列名稱)

列出串列裡面有幾  
個項目

```
# 小列買了三樣水果: 蘋果, 香蕉, 櫻桃
```

```
buylist = ['蘋果', '香蕉', '櫻桃']
```

```
# 不買香蕉, 改買草莓
```

```
buylist[1] = '草莓'
```

```
# 並再加買個橘子,
```

```
buylist.append('橘子')
```

```
# 小列要確認是否真的有移除香蕉改拿草莓,
```

```
print('香蕉' in buylist)
```

```
print('草莓' in buylist)
```

```
# 當店員輸入蘋果價格後, 小列臨時插入多買一個芒果,
```

```
buylist.insert(1, '芒果')
```

```
# 請問小列最終買了幾樣水果?
```

```
print(len(buylist))
```

```
print(buylist)
```

## 串列名稱.append(新項目)

在串列最後端新增一個項目

## 串列名稱.insert(索引值,新項目)

在串列指定的索引位置新增項目  
(原本在此索引值位置的資料會往  
後延一個位置, 索引值位置後面的  
資料也照此處理)

延續上面的生活記錄，  
繼續用程式處理以下生活細節：

**小列**將購買的水果依購買順序放入購物袋中  
回家後，先拿了最上面的水果出來吃，  
並發現袋子最下方的水果已經壞掉，  
需要拿出來丟掉，隔了幾天後，  
**小列**發現那袋水果已經全壞了，  
因此就整袋水果都丟掉了！

問題



# 串列記錄生活 – 食物整理篇

```
# 小列將購買的水果依購買順序放入購物袋中
buylist = ['蘋果', '芒果', '草莓', '櫻桃', '橘子']

# 先拿最上面的水果出來吃
print(buylist.pop())

# 將袋子最下方的水果丟掉
del buylist[0]
print(buylist[0])

# 整袋水果都丟掉
buylist.clear()
print(buylist)

del buylist
print(buylist)
```

## del 串列名稱

會刪除整個串列，包含  
整個變數都會移除，因  
此會出現錯誤訊息

執行結果

```
橘子
芒果
[]
Traceback (most recent call last):
  File "12-2.py", line 13, in <module>
    print(buylist)
NameError: name 'buylist' is not defined
```

# 串列記錄生活 – 食物整理篇

## 串列名稱.pop()

取出串列最尾端的項目，  
並將該項目從串列移除

## 串列名稱.clear()

將串列內所有項目都移  
除，串列會變成空串列

```
# 小列將購買的水果依購買順序放入購物袋中
buylist = ['蘋果', '芒果', '草莓', '櫻桃', '橘子']

# 先拿最上面的水果出來吃
print(buylist.pop())

# 將袋子最下方的水果丟掉
del buylist[0]
print(buylist[0])

# 就整袋水果都丟掉
buylist.clear()
print(buylist)

del buylist
print(buylist)
```

## del 串列名稱[索引值]

將串列內指定索引值位置  
的項目移除，並將該索引  
值後面的資料往前遞補

## del 串列名稱

刪除整個串列，包含整  
個變數都會移除

# 整理一下

工具使用方式	使用說明
串列名稱[索引值]=新值	將串列此索引位置的項目值改為新值
串列名稱.append(新項目)	在串列最後端新增一個項目
找尋的值 in 串列名稱	確認在串列裡面是否存在要找尋的值
串列名稱.insert(索引值,新項目)	在串列指定的索引位置新增項目 (原本在此索引值位置的資料會往後延一個位置,索引值位置後面的資料也照此處理)
len(串列名稱)	列出串列裡面有幾個項目
串列名稱.pop()	取出串列最尾端的項目,並將該項目從串列移除
del 串列名稱[索引值]	將串列內指定索引值位置的項目移除,並將該索引值後面的資料往前遞補
串列名稱.clear()	將串列內所有項目都移除,串列會變成空串列
del 串列名稱	刪除整個串列,包含整個變數都會移除

PYTHON

延伸的概念

# 概念 1: 建立串列有他法

中括號內放入要記錄在串列中的項目，各個項目使用逗點(,)分開

# 方法1: 使用中括號建立新串列

```
data1 = ['apple', 'banana']
```

# 方法2: 使用 list() 建立新串列

```
data2 = list(data1)
```

```
data3 = list(['apple', 'banana'])
```

```
data4 = list(('apple', 'banana'))
```

```
data5 = list('apple', 'banana')
```

使用 list() 建立新串列時，只能傳入一個變數值，因此需要將傳入的變數值設為串列才可

傳入一個串列型態的變數值

傳入一個元組型態的變數值  
(元組型態會在後面章節介紹)

傳入兩個字串型態的變數值，會造成錯誤  
TypeError: list() takes at most 1 argument (2 given)

# 概念 2: 刪除要小心

## del thislist

- 會將 thislist 包含串列內容與這個名稱都刪除，所以刪掉後若要再使用 thislist，就要先給 thislist 一個新內容

```
thislist = ['apple', 'banana']  
del thislist  
print(thislist)
```

## thislist.clear()

- 會將 thislist 串列內容刪除，但 thislist 仍會是一個串列，只是是一個空串列

```
thislist = ['apple', 'banana']  
thislist.clear()  
print(thislist)
```

串列跟字串一樣都自帶很多工具

更多的工具箱，可參考[這裡](https://www.w3schools.com/python/python_ref_list.asp)  
([https://www.w3schools.com/python/python\\_ref\\_list.asp](https://www.w3schools.com/python/python_ref_list.asp))

