

List 列表

- 列表List的建立與存取
- 列表List常用的方法

摘要：列表List

- Who(對象)：列表List
- Why(目的緣由)：為什麼需要列表List資料型態
 - 一個變數只能有一種資料型態
- When(應用時機)/Where(應用場域)：變數需要包含多種資料型態
 - 處理的資料內容需要同時儲存各種資料型態的資料
- What(內容定義)：列表List的類型結構
 1. 單層列表List
 2. 多層列表List
- How(操作機制)：
 1. 列表定義存取
 2. 相關函式
 3. 列表操作方法
 1. 定義存取運算：定義列表[]，並以索引值來存取列表資料
 2. 相關函式：len()、list()
 3. 列表運算方法：新增、刪除、修改等列表運算方法

摘要：列表建立與存取

列表List可以讓我們直接在此容器中放置其他型態的變數，
在我們需要用到大量變數時，列表不須重複宣告變數，
本文將介紹單層列表與多層列表結構，
以及列表的建立存取機制。

回想

在前幾章我們提到了變數的概念，都是一一對變數做指派。

如果小明家有五隻動物：

1.狗 2.貓 3.魚 4.鳥 5.烏龜

我們要用變數來表示這些動物的話，

依照前面章節的做法就需要利用到五個變數來分別表示。

回想

```
animal_1 = "dog"  
animal_2 = "cat"  
animal_3 = "bird"  
animal_4 = "fish"  
animal_5 = "turtle"
```

既然他們都是動物，有
沒有辦法用到一個變數
就包含4種動物？



List列表

概念：

List這個資料型態就像是一個置物櫃，置物櫃中有很多格子，每個格子都有自己的編號，並且格子裡面可以存放東西。

語法：

跟之前一樣利用等號做指派

記得列表要用中括號包起來

List變數
名稱

Alphabet = ["a", "b", "c", "d", "e", "f", "g"]

列表中存放的資料

List列表

- 列表必須用中括號框起來。
- 列表是一種資料型態，在建立時需要用一個變數去承接。
- 列表中的格子都有自己的編號，可利用對應的編號找到格子中的值。
- 切記：在列表中格子的編號從零開始。

程式 → Alphabet = ["a", "b", "c", "d", "e", "f", "g"]

圖示 → List >

0	1	2	3	4	5	6
a	b	c	d	e	f	g

列表中可存放的資料型態

列表可以存放多種不同資料型態的值，像是字串、整數、浮點數、變數...等，都可以一起被放進同一個列表中。

舉例：

```
a=5  
List = [a, 'a', 5, 5.0]
```

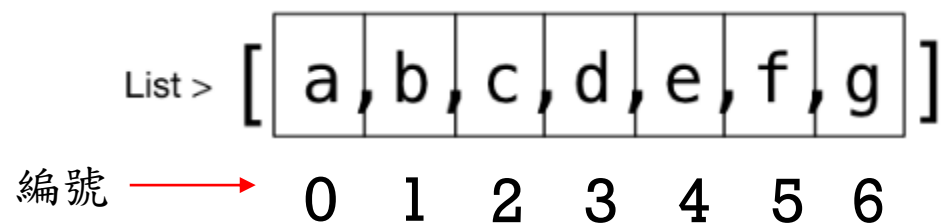
變數

字串

整數

浮點數

讀取列表中的值：索引值



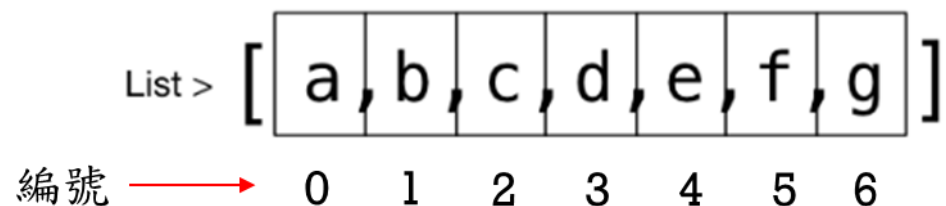
- 當我們要存取列表中的值，必須要使用列表編號來進行存取。
- 列表編號我們通常稱為索引(index)。
- 在程式語言中索引值是從0開始。

讀取列表中的值：索引值

語法：

List變數名稱[索引]

範例：




```
Alphabet = ["a", "b", "c", "d", "e", "f", "g"]  
print ("Alphabet[0] =", Alphabet[0])  
print ("Alphabet[3] =", Alphabet[3])
```

Alphabet[0] = a
Alphabet[3] = d

取出列表中的部分資料：切片(slicing)

- 當我們要取出列表中部份的資料，可透過切片(slicing)來讀取。
- 格式：[起始索引值:終止索引值:間隔值]
- 包含起始值，但不包含終止值

```
Alphabet = ["a", [1,2,3], "b", 1, "c", "d", 2, 3, "e", "f", 4, "g"]  
print ("Alphabet[1:10:2] =", Alphabet[1:10:2])  
print ("Alphabet[::1] =", Alphabet[::1]) #由左到右的正向  
print ("Alphabet[::-1] =", Alphabet[::-1]) #由右到左的反向
```



```
Alphabet[1:10:2] = [[1, 2, 3], 1, 'd', 3, 'f']  
Alphabet[::1] = ['a', [1, 2, 3], 'b', 1, 'c', 'd', 2, 3, 'e', 'f', 4, 'g']  
Alphabet[::-1] = ['g', 4, 'f', 'e', 3, 2, 'd', 'c', 1, 'b', [1, 2, 3], 'a']
```

動動腦

我們在本章一開始提到的五種動物可以用一個list變數來解決了

```
Animal = ["dog", "cat", "bird", "fish", "turtle"]
```

➤ 那我們能不能將這些動物在這個列表中細分成更多種類呢？

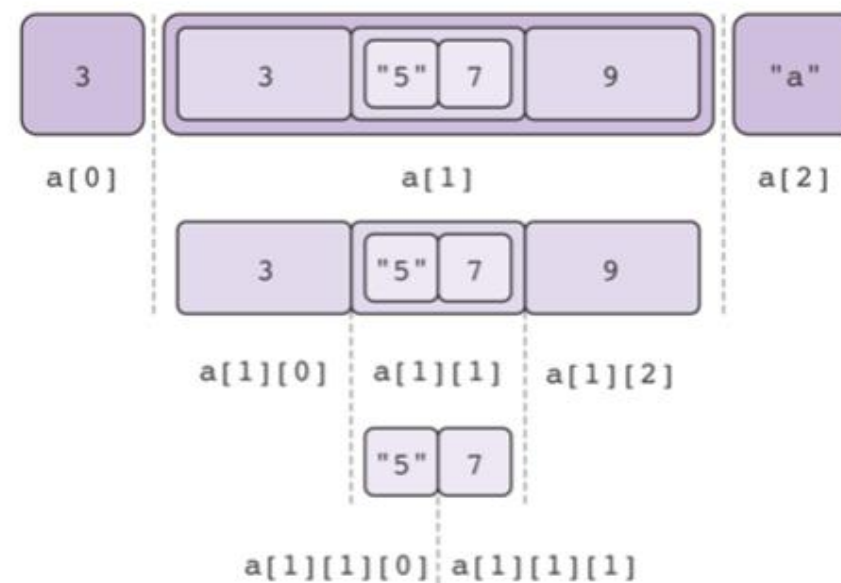
例如：在陸地上的分成一群，在空中的分成一群，水中的分成一群。

多層list

我們剛剛提到列表中可以塞各種資料型態的值，像是字串與整數，其實列表中還可以塞list，變成多層列表。

範例：

```
a = [3, [3, ["5", 7], 9], "a"]
```



多層List範例

寵物列表

```
Animal = ["dog", "cat", "bird", "fish", "turtle"]
```

- 分成三群：路上、空中、水中的動物

- land : dog, cat
- sky : bird
- sea : fish, turtle

- 程式範例1

```
land = ["dog", "cat"]
```

```
sky = ["bird"]
```

```
sea = ["fish", "turtle"]
```

```
animal = [land, sky, sea]
```

- 程式範例2

```
animal2 = [["dog", "cat"], ["bird"], ["fish", "turtle"]]
```

列表的常用方法

在講解了List列表的概念與使用方式，
在Python中提供了許多List的函式與方法供我們使用，
例如len()、append()、remove()等實用的函式與方法，
這些函式可以使我們更方便去操作列表，
本章也會透過範例來演示每一個函式與方法的運用。

強制資料型態轉換函式：list()

語法：

`list("字串")`

作用：

將括號內的字串拆成列表。

範例：

```
name = list("Anna")  
print("name :", name)  
print("type of name:", type(name))  
print("length of name:", len(name))
```

name : ["A", "n", "n", "a"]
type name : <class "list">
length name : 4

`name = list("Anna")`



name

"A"	"n"	"n"	"a"
0	1	2	3


```
score = input("Please input three scores : ")  
scoreL = list(score)
```

```
print("score type : ", type(score))  
print("score value : ", score)  
print("score len : ", len(score))
```

```
print("scoreL type : ", type(scoreL))  
print("score List content : ", scoreL)  
print("scoreL len : ", len(scoreL))
```

以List來儲存使用者輸入的數個成績

字串中的每個字元都會被儲存成為List中的元素

```
Please input three scores : 12 34 56  
score type : <class "str">  
score value : 12 34 56  
score len : 8  
scoreL type : <class "list">  
score List content : ["1", "2", " ", "3", "4", " ", "5", "6"]  
scoreL len : 8
```

動動腦

能否將使用者輸入的數個成績，以逗號做區隔，將每個成績數值儲存成為List的元素？

例如：

Please input three scores :

輸入名字：67,86,95

輸出：

成績一是67，成績二是86，成績三是95

由使用者輸入
多個成績



以分隔符號對字串進行切片

語法：字串進行 split() 方法會得到一個 list 型態的資料

`str.split(str, num)`

- str參數：分隔符號 (預設值是空白字元)
- num參數：分割次數，分隔為 (num + 1) 個子字串 (預設值是對整個字串進行分割)

```
name1 = "My name is Anna."  
name1L = name1.split()  
print("name1 value : ", name1)  
print("name1 len : ", len(name1))  
print("name1L value : ", name1L)  
print("name1L len : ", len(name1L))  
  
name2 = "My&&name&&is&&Anna."  
name2L = name2.split("&&", 2)  
print("name2 len : ", len(name2))  
print("name2 value : ", name2)  
print("name2L len : ", len(name2L))  
print("name2L value : ", name2L)
```

```
name1 value : My name is Anna.  
name1 len : 16  
name1L value : ['My', 'name', 'is', 'Anna.']  
name1L len : 4  
name2 len : 19  
name2 value : My&&name&&is&&Anna.  
name2L len : 3  
name2L value : ['My', 'name', 'is&&Anna.']
```

以List來儲存使用者輸入的數個成績

字串中的每個字元都會被儲存成為List中的元素

```
score = input("Please input three scores : ")  
scoreL= list(score)  
scoreL2= score.split(",")
```

```
print("Score type : ", type(score))  
print("Score content : ", score)  
print("Score len : ", len(score))
```

```
print("ScoreL type : ", type(scoreL))  
print("Score List content : ", scoreL)  
print("ScoreL len : ", len(scoreL))
```

```
print("scoreL2 type : ",type(scoreL2))  
print("scoreL2 value : ",scoreL2)  
print("scoreL2 len : ",len(scoreL2))
```

Please input three scores : 12,34,56

Score type : <class "str">

Score content : 12,34,56

Score len : 8

ScoreL type : <class "list">

Score List content : ["1", "2", ",", "3", "4", ",", "5", "6"]

ScoreL len : 8

scoreL2 type : <class "list">

scoreL2 value : ["12", "34", "56"]

scoreL2 len : 3

C3_P3：水果攤的驚喜組合

小明是一位水果攤的老板，他最近推出了一個新的驚喜組合，顧客可以根據他們的喜好選擇多種水果。現在，你需要設計一個程式，讓小明能夠輸入這個組合的水果字串，然後將水果組合以空格來做分割，再儲存至列表中，方便他組合出獨特的水果組合，並輸出這個獨特的水果組合給顧客。

1. 輸入這個組合的水果字串
2. 將水果組合以空格來做分割，並儲存至列表中
3. 輸出這個獨特的水果組合給顧客。

List 串列的元素個術：len()

語法：

len(List)

作用：

計算List中有幾個元素。

範例：

```
name = ["A","n","n","a"]  
print("type name : ",type(name))  
print("length name : ",len(name))
```

type name : <class "list">
length name : 4

name	"A"	"n"	"n"	"a"
	0	1	2	3

len(name)



4 (name 中有4個元素)

C3_P1：長度計算

請計算一串數字，用空格分開，並計算數字的數量。


1. 輸入一串數字，用空格分開
2. 計算該串數字的數量
3. 輸出數字的數量

將串列中的中的元素以指定的連接字串連接生成一個新的字串

語法：

str變數名稱.join(list變數)

```
s = "***"  
name = ["f", "l", "o", "w"]  
  
print("before join name : ",name)  
print("name type==>",type(name))  
nameJ = s.join(name)  
print("after join nameJ : ",nameJ)  
print("nameJ type : ",type(nameJ))
```



```
before join name : ["f", "l", "o", "w"]  
name type==> <class "list">  
after join nameJ : f**l**o**w  
nameJ type : <class "str">
```


List經常使用的方法

- 在上一個小節我們提到List的概念，
- 列表能夠使用一個容器來包含很多元素，
- 列表中的內容可以隨時做變動，這是一個很大的優點。
- 在Python中，提供很多方法給list使用，這些方法提供我們可以更方便的操作列表及管理列表內的元素。
- 本章將會介紹幾個經常用到的list方法。

List方法：append()

語法：

List變數名稱.append(obj)

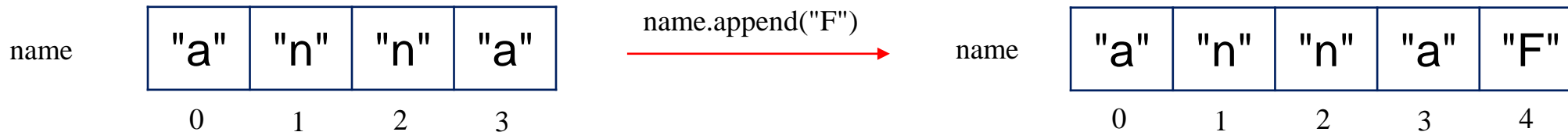
作用：

在列表的尾端加上一個元素。

範例：

```
name = ["A", "n", "n", "a"]  
name.append("F")  
print(name)
```

["A", "n", "n", "a", "F"]



List方法：append()

語法：

List變數名稱.append(obj)

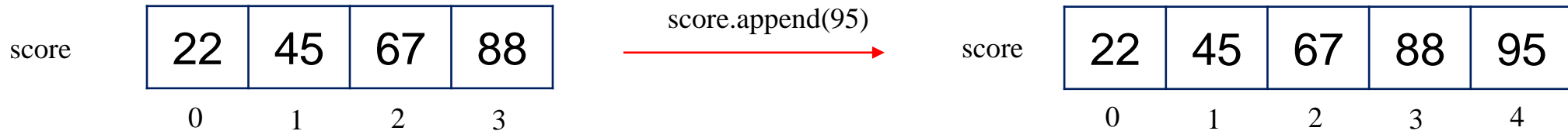
作用：

在列表的尾端加上一個元素。

範例：

```
score = [22,45,67,88]  
print("Before append, score :",score)  
score.append(95)  
print("after append, score :",score)
```

Before append, score : [22, 45, 67, 88]
after append, score : [22, 45, 67, 88, 95]



List方法：insert()

語法：

List變數名稱.insert(index, obj)

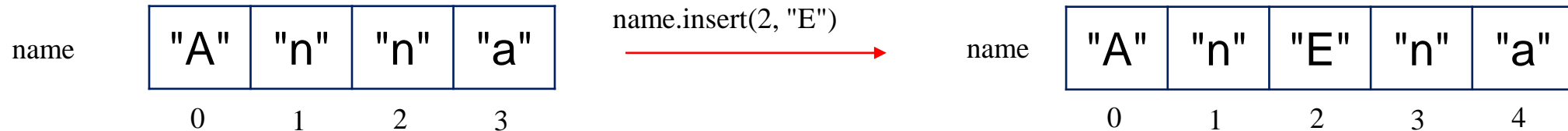
作用：

在列表的指定位置加上一個元素。

範例：

```
name = ["A", "n", "n", "a"]  
name.insert(2, "E")  
print(name)
```

["A", "n", "E", "n", "a"]



List方法：insert()

語法：

List變數名稱.insert(index, obj)

作用：

在列表的指定位置加上一個元素。

範例：

```
score = [22,45,67,88]
print("Before insert, score :", score)
score.insert(2,60)
print("after insert, score :", score)
```

Before insert, score : [22, 45, 67, 88]
after insert, score : [22, 45, 60, 67, 88]

score

22	45	67	88
0	1	2	3

score.insert(2,60)

score

22	45	60	67	88
0	1	2	3	4

List方法：remove()

語法：

List變數名稱.remove(obj)

範例：

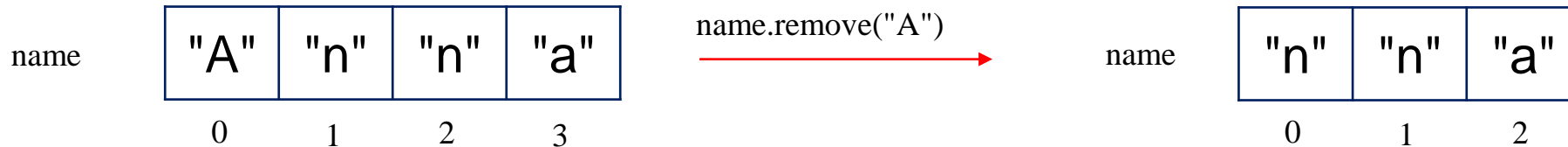
```
name = ["A", "n", "n", "a"]  
print("Before remove A :",name)  
name.remove("A")  
print("After remove A :",name)
```

作用：

在列表中刪除指定的元素。

(若指定的元素有重複則刪第一個)

Before remove A : ["A", "n", "n", "a"]
After remove A : ["n", "n", "a"]



List方法：remove()

語法：

List變數名稱.remove(obj)

作用：

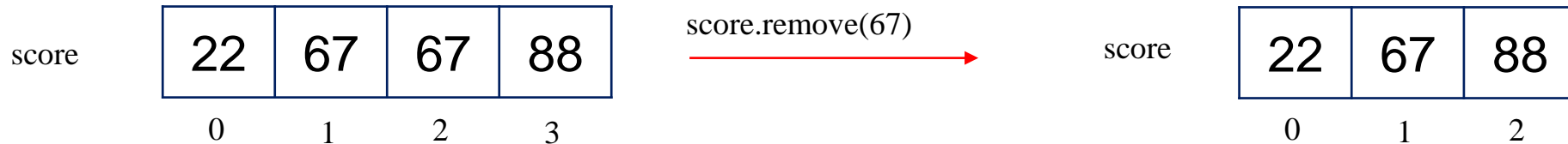
在列表中刪除指定的元素。

(若指定的元素有重複則刪第一個)

範例：

```
score = [22,67,67,88]
print("Before remove, score :",score)
score.remove(67)
print("after remove, score :",score)
```

Before remove, score : [22, 67, 67, 88]
after remove, score : [22, 67, 88]



List方法：count()

語法：

List變數名稱.count(obj)

作用：

統計某個元素在列表出現的次數。

範例：

```
name = ["A", "n", "n", "a"]  
num_n = name.count("n")  
num_a = name.count("a")  
print("number of n :", num_n)  
print("number of a :", num_a)
```

number of n : 2
number of a : 1

name

"A"	"n"	"n"	"a"
0	1	2	3

name.count("n")

2 (n出現兩次)

List方法：index()

語法：

List變數名稱.index(obj)

作用：

找出某個元素第一次出現的索引值。
(若找不到就傳回錯誤訊息)

範例：

```
name = ["A", "n", "n", "a"]  
print("index of A :", name.index("A"))  
print("index of a :", name.index("a"))  
print("index of n :", name.index("n"))
```

index of A : 0
index of a : 3
index of n : 1

name	"A"	"n"	"n"	"a"
	0	1	2	3

name.index("a")

3 (a第一次出現在位置3)

List方法：reverse()

語法：

List變數名稱.reverse()

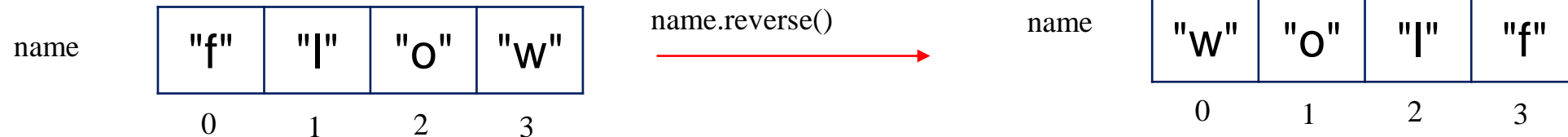
作用：

將整個List裡的元素順序做反轉。

範例：

```
name = ["f", "l", "o", "w"]  
print("beforereverse :", name)  
name.reverse()  
print("after reverse :", name)
```

beforereverse : ["f", "l", "o", "w"]
after reverse : ["w", "o", "l", "f"]



List方法：sort()

語法：


List變數名稱.sort()

作用：

將整個List裡的元素進行排序

範例：

```
myList = ["Owen","Jeff","Owen","Louisa"]  
print("before sort :",myList)  
myList.sort()  
print("aftter sort :",myList)
```



```
before sort : ["Owen", "Jeff", "Owen", "Louisa"]  
after sort  : ["Jeff", "Louisa", "Owen", "Owen"]
```

C3_P2：Find keyword

輸入一串單字、關鍵字，計算共有多少個單字，
及關鍵字共出現幾次

1. 輸入一串以空格分隔的字串，並以
空格為分隔儲存為列表
2. 輸入一個要計算出現次數的字串
3. 計算這個字串在列表中出現的次數
4. 輸出計數結果

章節總結活動



本章關鍵字
本章重點概念簡述



請將你的章節總結寫在Memo上
，Memo的title請設定為 **總結**