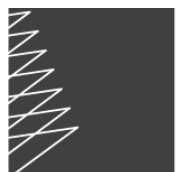
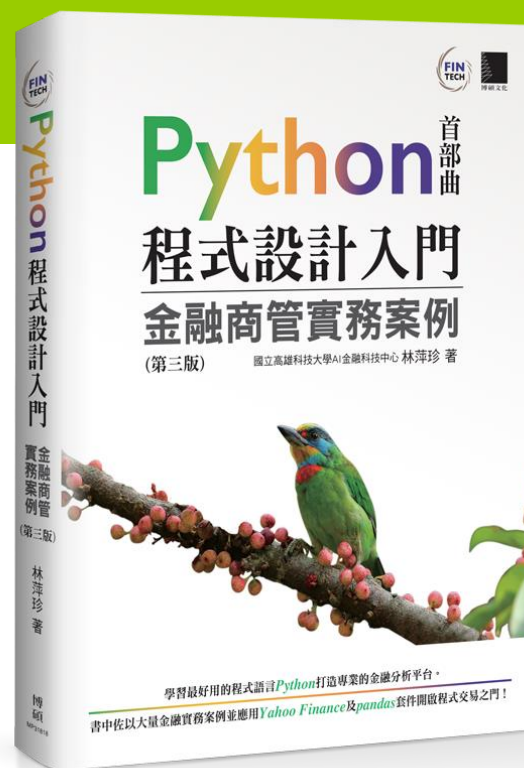


# 使用教材：Python 程式設計入門 金融商管實務 案例 第三版

作者：林萍珍



DrMaster

精選簡報・教師專用  
博碩文化・版權所有

[www.drmaster.com.tw](http://www.drmaster.com.tw)

## 第三章 資料型別

3-1 可變與不可變

3-2 變數命名規則

3-3 數值型別

3-4 字串型別

3-5 字串的方法

3-6 抽象資料型別

3-7 容器型別

# 可變與不可變

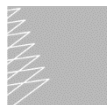
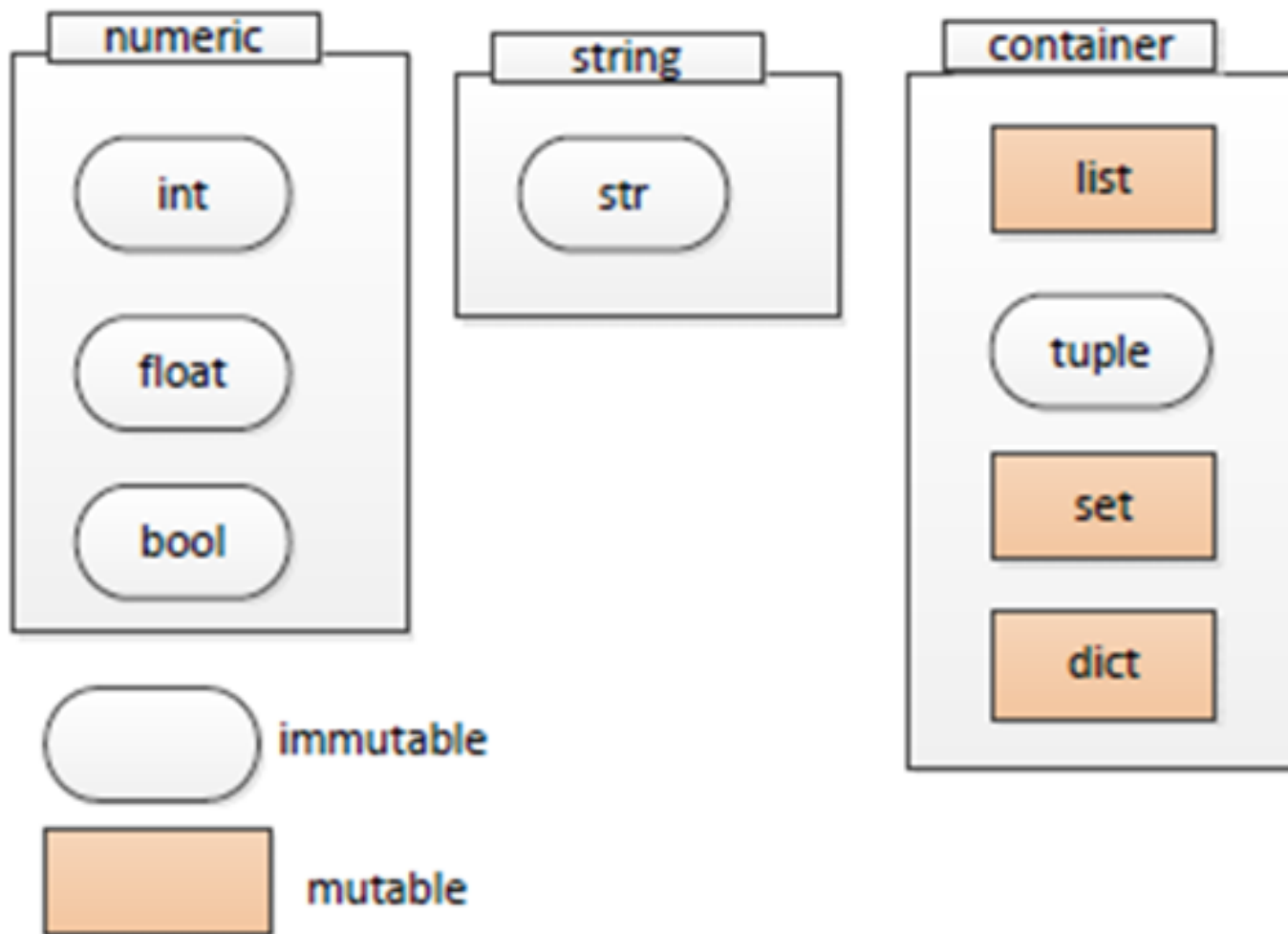
## ☑ 不可變的 (immutable)

意指物件一旦被建立，它的內容值就是固定不變。如果內容值被改變，則原有的物件與變數的連結關係即重新建立，舊物件則由系統自動回收。Int, float, bool, str, tuple 是屬於不可變的資料型別。

## ☑ 可變的 (mutable)

意指元素內容可以改變，且不影響物件與變數的連結關係。list, set, dict 是屬於可變的資料型別。

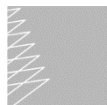
# 可變與不可變



# 變數命名規則

Python的世界是萬物皆物件，物件是需取**變數名稱**。  
變數的命名規則如下：

- ▶ 第一個字元必須是英文字母。
- ▶ 字元之間不能有空格。
- ▶ 變數名稱可以用底線，但不可用減號「-」。
- ▶ 不可使用特殊符號，例如：%、#、@等。
- ▶ 不建議使用中文命名。
- ▶ 大小寫有區別，例如：Agent 與 agent 視為兩個變數。
- ▶ 建議避免單獨使用的名稱：「I」（小寫是 i）、「L」（小寫是 l）、「O」（小寫是 o）  
乃因三字在字型視覺上，與數字的 1 和 0 容易混淆。

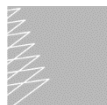
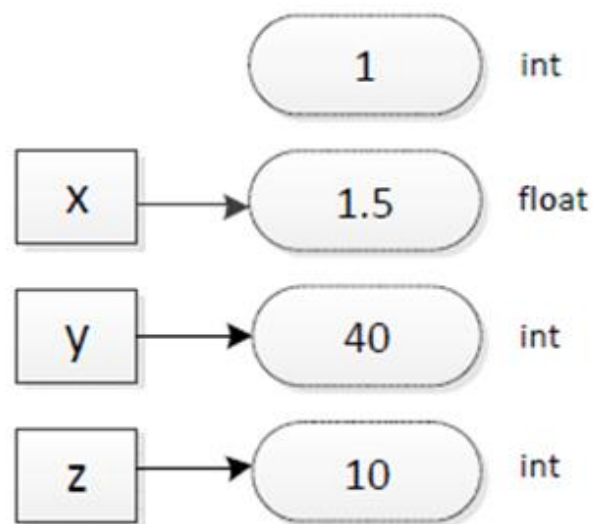


# 數值型別

## 3-3-1 整數與浮點數

Python 3.x 版後，整數不分固定精準度與無窮精準度，只有整數的資料型別一種。也就是說，Python3.x 版以後，整數的表示範圍沒有上限，只要記憶體可以處理，Python 就可以顯示。表示法（見第 1 到 2 列）。

In	1	<code>2**100</code>
Out	2	1267650600228229401496703205376
In	3	<code>x=1</code>
	4	<code>x=1.5</code>
	5	<code>y=40</code>
	6	<code>z=10</code>

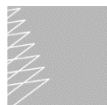


# 數值型別

## 3-3-2 布林

整數可以用來表示布林值，0 表示 False、1 表示 True。Python 提供 3 種適合布林型別運算的邏輯運算：且（and）、或（or）與非（not），相關討論見第 4 章。1 and 0 傳回 0（見第 3 至 4 列）；1 or 0 傳回 1（見第 5 至 6 列）；not 1 傳回 0（見第 7 至 8 列）。

In	1	x = 1
	2	y = 0
	3	x and y
Out	4	0
In	5	x or y
Out	6	1
In	7	not x

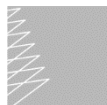


# 字串型別

## 3-4-2 文字資料儲存在變數中

文字資料可以儲存在變數中（見第 1 列）。以方便呼叫及儲存（見第 2 到 3 列）。若使用 變數名稱「str」儲存字串，字串內容要使用跳列指令「\n」（見第 5 列）。沒有顯示效果，即沒有跳列效果（見第 6 列）。若希望發揮跳列功能需使用 `print` 函數，以下介紹。

In	1	str=' Do not worry.' # 字串儲存在 str 變數中
	2	str # 呼叫 str 變數名稱
Out	3	' Do not worry.'
In	4	str='Do not worry. \n About tomorrow.'
	5	str
Out	6	'Do not worry. \n About tomorrow.'





# 字串型別

## 3-4-3 使用 print 函數顯示字串

使用 `print` 函數，除可以顯示字串外（見第 1 列），還可以發揮跳列功能（見第 2 到 3 列）。有時處理字串時，會需要印出反斜線，`Python` 會印出兩個反斜線（見第 4 到 6 列）。若改成 `print` 函數，即可正常顯示一個反斜線（見第 7 到 8 列）。

In	1	<code>print(str)</code>
Out	2	<code>Do not worry.</code>
	3	<code>About tomorrow.</code>
In	4	<code>path='c:\python'</code>
	5	<code>path</code>
Out	6	<code>'c:\\python'</code>
In	7	<code>print(path)</code>
Out	8	<code>c:\python</code>

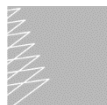
# 字串型別

## 3-4-4 r 前置字元

字串處理時不方便使用反斜線「\」，例如「\n」會跳列（見第 1 到 4 列）。「\name」是路徑名稱而非跳列，但 `print` 函數的字串中若有「\n」，會自動跳列。在字串變數等號的右邊，字串文字單引號之前加入「r」前置字元，則可擺脫跳列的功能，以原始字串（`raw string`）顯示。完整顯示路徑名稱（見第 5 到 7 列）。

In	1	<code>path='c:\python\name'</code>
	2	<code>print(path)</code>
Out	3	<code>c:\python</code>
	4	<code>ame</code>

In	5	<code>path='r'c:\python\name'</code>
	6	<code>print(path)</code>
Out	7	<code>c:\python\name</code>

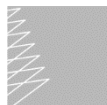


# 字串型別

## 3-4-5 字串相加

字串相加可以單純字串變數相加（見第 1 到 4 列）。或使用 `print` 函數，`print` 函數可以在字串單引號前後加入空白，以區別字句。若使用 `print` 函數顯示字串，不會出現單引號（見第 5 到 6 列）。

In	1	<code>x=' Hello! '</code>
	2	<code>y=' Python '</code>
	3	<code>x+y</code>
Out	4	<code>' Hello! Python '</code>
In	5	<code>print(x+y)</code>
Out	6	<code>Hello! Python</code>

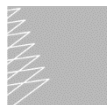


# 字串型別

## 3-4-6 複製字串

若要重覆複製字串，可以用字串變數相乘的方式處理，純字串變數亦可使用 `print` 函數處理，可以達到一樣的執行結果，二者差異只在顯示時純字串處理會多出單引號（見第 1 到 4 列）。另外，兩個字串要一起複製時，要將兩個變數前後加小括弧「`()`」（見第 5 與 8 列），否則僅最後一個字串變數的內容會重覆（見第 3 到 4 列）。

In	1	<code>x=' Hello! '</code>
	2	<code>y=' Python '</code>
	3	<code>x+y*3</code>
Out	4	<code>' Hello! Python Python Python '</code>
In	5	<code>(x+y)*3</code>
Out	6	<code>' Hello! Python Hello! Python Hello! Python '</code>
In	7	<code>print((x+y)*3)</code>
Out	8	<code>Hello! Python Hello! Python Hello! Python</code>



# 字串型別

## 3-4-8 取出部分字串

x		H	e	l	l	o	!	
索引值	0	1	2	3	4	5	6	7

圖 3-3 x 字串索引值對照圖

y		P	y	t	h	o	n	
索引值	0	1	2	3	4	5	6	7

圖 3-4 y 字串索引值對照圖

In	1	x[-2]
Out	2	'!'
In	3	y[0:2]
Out	4	' P'
In	5	y[0:3]
Out	6	' Py'

# 字串型別

## 3-4-9 字串比較

Python 提供簡單的字串比較方法，即使用關係運算，此見後續章節。其中，相等「==」（見第 1 到 2 列）；或不相等「!=」（見第 3 到 4 列），有關 **x** 與 **y** 的字串比較，結果回傳值是 **True** 或 **False**。

In	1	<code>x==y</code>
Out	2	<code>False</code>
In	3	<code>x!=y</code>
Out	4	<code>True</code>

# 字串的方法

字串是不可變的序列（sequence），所有可應用在不可變序列資料型別的功能，都可以在字串型別上使用。

物件名稱 . 方法名稱 ( 相關參數 )

表 3-2 常用字串方法 (1)

分類	方法名稱	功能說明
大小寫轉換	capitalize()	將字串的第一個字母轉大寫，其餘小寫。
	upper()	將字串全部轉大寫。
	lower()	將字串全部轉小寫。
	title()	將每個單字第一個字轉大寫，其餘轉小寫。
	swapcase()	小寫轉大寫，大寫轉小寫。

# 字串型別--置換子字串

表 3-3 常用字串方法 (2)

分類	方法名稱	功能說明
置換合併	<code>replace(old, new, count)</code>	將 <code>old</code> 子字串置換成 <code>new</code> 子字串， <code>count</code> 是指定置換幾個。
子字串	<code>find()</code>	搜尋子字串出現的位置。
	<code>count()</code>	計算目標字串出現的次數。
	<code>startswith()</code>	檢查子字串是否出現在原字串中。
	<code>endswith()</code>	同 <code>startswith</code> ，但是倒著數。
	<code>split([sep[, maxsplit]])</code>	字串分割。
	<code>join(stt1, str2)</code>	連接字串。



# 字串型別--置換子字串

s	h	e	l	l	o		p	y	t	h	o	n
索引值	0	1	2	3	4	5	6	7	8	9	10	11

圖 3-5 s 串索引值

t	h	i	!		p	y	t	h	o	n
索引值	0	1	2	3	4	5	6	7	8	9

圖 3-6 t 字串索引值

# 字串型別--置換子字串

(1)replace()方法可將新字串hi! 取代舊字串hello，見1到2列。

In	1	s.replace('hello', 'hi!')
Out	2	'hi! python'

(2)常用子字串搜尋的find()方法，可以從s字串中，找出子字串py，回傳值是s，字串的索引值為6，見3到4列。

In	3	s.find('py')
Out	4	6

## 字串型別--置換子字串

- (3)若將 `replace()` 方法的回傳值儲存在 `t` 變數中（見第 5 到 7 列），再用 `t.find()` 方法，則是以新字串 `t` 為搜尋目標，找到符合目標索引值（4,5）的第一個索引值 4（見圖 3-6），以及指令（見第 8 到 9 列）。

In	5	<code>t= s.replace('hello', 'hi!')</code>
	6	<code>t</code>
Out	7	<code>'hi! python'</code>
In	8	<code>t.find('py')</code>
Out	9	<code>4</code>

# 字串型別--置換子字串

(4) count()方法可以計算子字串出現的次數，如o在s字串出現2次，見10到11列。

In	10	s.count('o')
Out	11	2

## 字串型別--置換子字串

(8)split()可以切割字串，預設切割判定為單一空白，使用方法(見18到19列)，成為切割判定的依據字符會消失，使用方法(見20到21列)。

In	18	s.split()
Out	19	['hello', 'python']
In	20	s.split('p')
Out	21	['hello ', 'python']

# 字串型別--判斷字串

表 3-4 常用字串方法 (3)

分類	方法名稱	功能說明
判斷字串	isidentifier()	檢查是否為識別字或保留字。
	isalnum()	檢查字串是否為字母或數字。
	isalpha()	檢查字串是否為字母。
	isdigit()	檢查字串是否為數字。
	isupper()	檢查字串是否為大寫。
	islower()	檢查字串是否為小寫。
	istitle()	檢查字串字首是否為大寫。
	isspace()	檢查字串字首是否為空白字元。

# 字串型別--判斷字串

- (2) `isalnum()` 方法可以檢查字串是否含有字母或數字，只要符合其中一項，即傳回 `True`（見第 9 到 10 列）；字串中含有字母或數字以外的字元（含特殊字元或空字元），即傳回 `False`（見第 11 到 12 列）。

In	9	'show123'.isalnum(), 'show'.isalnum(), '123'.isalnum()
Out	10	(True, True, True)
In	11	'&abc123'.isalnum(), ''.isalnum(), '%---%'.isalnum()
Out	12	(False, False, False)

## 字串型別--判斷字串

(3) `isalpha()`方法是檢查字串是否全部為字母，見13到14列。

In	13	'show'.isalpha(), '123'.isalpha(), 'show123'.isalpha()
Out	14	(True, False, False)

(4) `isdigit()`方法是檢查字串是否全部為數字，見15到16列。

In	15	'show'.isdigit(), '123'.isdigit(), 'show123'.isdigit()
Out	16	(False, True, False)



# 字串型別--刪除/切割/填滿

表 3-5 常用字串方法(4)

分類	方法名稱	功能說明
刪除	strip()	刪除字串開頭(左)與尾端(右)的空白字元。
	lstrip()	刪除字串開頭的空白字元。
	rstrip()	刪除字串尾端的空白字元。
切割	partition(sep)	字串切割。
	split(sep)	字串連續分割。
填滿	center(width,[fillchar])	字串置中對齊。
	ljust(width,[fillchar])	字串置左對齊。
	rjust(width,[fillchar])	字串置右對齊。
	zfill(width)	字串填滿。

# 字串型別--刪除/切割/填滿

(1)strip()方法的功能是刪除字串左右，即開頭與尾端的空白字元。

(2)lstrip()方法只刪除字串左邊的空白字元。

(3)rstrip()方法則是刪除右邊空白字元，見1到2列。

In	1	' show123 '.strip(), ' show123'.lstrip(), 'show123 '.rstrip()
Out	2	('show123', 'show123', 'show123')

(4) partition()方法可以使sep參數做為分割符號準則，將字串分割成3個部份，見第3到4列。

In	3	'Good, Job'.partition(',')
Out	4	('Good', ',', 'Job')

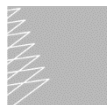
## 字串型別--刪除/切割/填滿

(5) `split()` 方法可以將字串連續分割多個子字串（見第 5 到 6 列），回傳值以串列「`[]`」儲存，其他方法的回傳值多以元組「`()`」儲存，串列與元組的使用方法（見第本章後續章節）。

In	5	'弟子規 聖人訓 首孝弟 次謹信'.split(' ')
Out	6	['弟子規', '聖人訓', '首孝弟', '次謹信']

# 容器型別

- ◆ Python 最常用的容器型別（container type）是串列（list）、元組（tuple）、集合（set）以及字典（dict），四者都是屬於抽象型別。
- ◆ 容器型別可以儲存不同資料特徵，例如：整數、浮點數與字串以及串列自己本身等。
- ◆ list、tuple 等資料型別以儲存格（cell）為基本單位，各自獨立，所以可以個別儲存不同的資料型別。
- ◆ Python 內建資料型別並不包含陣列，必須引用外部模組函數庫。



# 容器型別--list

list是有順序、可變的物件集合體，其特性有：

1. 內容是可以索引的，若長度為 $n$ 的串列，有效的索引值是0到 $n-1$ ，即從0開始編號。
2. 元素不限統一資料型別，可以放不同型別的資料。
3. 元素的長度可以變動。
4. 元素的內容是可變的。

# 容器型別--list

```
In      1      lis1=[30, 50, 99]
          2      lis1[0]
Out     3      30
In      4      lis1[10]
Out     5      Traceback (most recent call last):
          6      File "<stdin>" , line 1, in <module>
          7      IndexError: list index out of range
In      8      lis2=[10422152, 'John', 85.51]
          9      score=lis2[2]
         10      score
Out     11      85.51
```

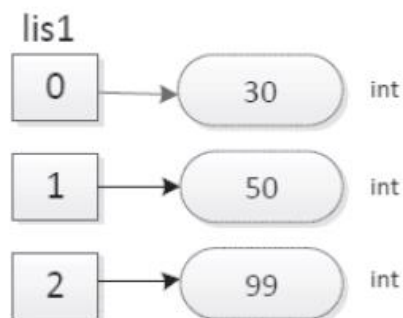


圖 3-8 串列儲存一維整數

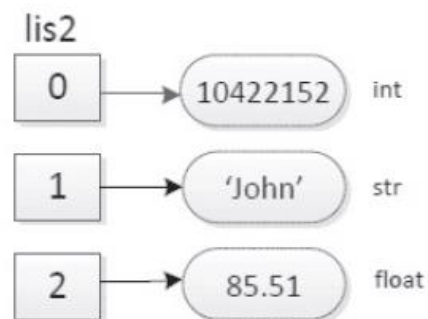


圖 3-9 串列儲存不同資料型別

# 以串列模擬二維陣列

In	12	lis3=[[6, 3, 1], [5, 2, 4]]
	13	total=lis3[0][0]+lis3[1][0]+lis3[0][1]+lis3[1][1]+lis3[0][2]+lis3[1][2]
	14	total
Out	15	21

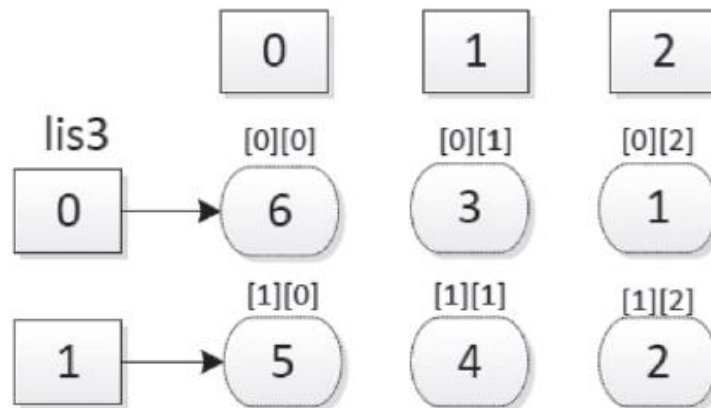


圖 3-10 串列包含串列模擬二維陣列

# 定位 *position*

## ◆ 定位position

使用range產生一維之5的倍數的資料，再藉list函數將數值轉成串列，並指派給lis1，見第1列。

In	1	lis1=list(range(5, 25, 5))
----	---	----------------------------

● (1)全部顯示：串列中括弧內輸入冒號「:」。如  
lis1[:] 代表從將元素從頭至尾顯示（見第2到3列）。

	2	lis1[:]
Out	3	[5, 10, 15, 20]



## 定位 *position*

- (2)指定顯示某個特定位置：串列中括弧內輸入索引編號，如lis1[1]，即可顯示該位置的元素。  
lis1[1]表示索引值為1的第2個元素10，（見第4到5列）。

In	4	lis1[1]
Out	5	10

- (3)負1代表最後位置：串列中括弧內輸入-1。如lis1[-1]，代表顯示最後一個元素，（見第6到7列）。

In	6	lis1[-1]
Out	7	20

## 定位 *position*

(4)數字加冒號：串列中括弧內輸入數字加冒號。如 `lis1[1:]`，代表顯示從數字索引值之後到最後一個元素，(見8到9列)。

In	8	<code>lis1[1:]</code>
Out	9	<code>[10, 15, 20]</code>

(5)冒號加負1：串列中括弧內輸入冒號加數字。如 `lis1[:-1]`，代表從0索引值的元素開始顯示到索引值 `n-1` 個元素，(見10到11列)。

In	10	<code>lis1[:-1]</code>
Out	11	<code>[5, 10, 15]</code>

## 定位 *position*

(6)數字冒號負號：串列中括弧內輸入數字接冒號接負1。如lis1[1:-1]，代表從1索引值的元素開始顯示到索引值n-1個元素。(見12到13列)。

In	12	lis1[1:-1]
Out	13	[10, 15]

(7)反向顯示：串列中括弧內輸入2個冒號加負1。如lis1[::-1]，代表最後一個元素開始往前顯示，(見14到15列)。

In	14	lis1[::-1]
Out	15	[20, 15, 10, 5]

# 定位 *position*

● (8)遞增顯示：串列中括弧內輸入數字加 2 個冒號加數字。如 `lis1[1::2]`，代表從第 1 個索引值開始到最後一個元素，每次遞增 2，即從 1 開始每次跳過一個顯示元素（見第 16 到 17 列）。語法如下：`[start:end:step]`

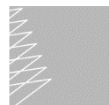
`step`（增減量）為正，表示有小到大遞增；為負，表示由大到小遞減。

In	16	<code>lis1[1::2]</code>
Out	17	<code>[10, 20]</code>

# 串列型別專屬的方法

表 3-6 串列專屬的方法

方法名稱	功能說明
append()	把一個元素加入到串列尾端。
extend()	把一組元素加入到串列的尾端。
insert()	把一個元素加入到串列中。
remove()	從串列尾端刪除一個元素。
pop()	可以取出並回傳指定索引值的元素。
sort()	元素由小到大排序，加入參數 <code>reverse=True</code> 做由大到小排序。
reverse()	串列的元素由大到小排序。
copy()	串列的元素做複製。
clear()	清除串列內所有元素。



# 串列型別專屬的方法

(1)建立一個lis1串列後，再使用append()方法在尾端加入一個指定元素，(見1到4列)。

In	1	lis1=[5, 10, 15, 20]
	2	lis1.append(25)
	3	lis1
Out	4	[5, 10, 15, 20, 25]

(2)若要一次加入多個元素，可以使用extend()方法，預設在尾端加入，(見5到7列)。

In	5	lis1.extend([30, 35])
	6	lis1
Out	7	[5, 10, 15, 20, 25, 30, 35]

## 串列型別專屬的方法

(3) insert()方法可以指定索引位置、插入指定元素，例如：可以在第0個索引值加入數值0的元素，(見8到10列)。

In	8	lis1.insert(0,0)
	9	lis1
Out	10	[0, 5, 10, 15, 20, 25, 30, 35]

(4) remove()方法可以從尾端移除一個元素，(見11到13列)。

In	11	lis1.remove(35)
	12	lis1
Out	13	[0, 5, 10, 15, 20, 25, 30]

# 串列型別專屬的方法

(5) pop()方法可以移除指定索引位置的元素，並可以回傳此值，若無帶入參數（索引值），預設是移除並回傳尾端的元素，見(14到17列)。

In	14	a=lis1.pop(3)
	15	b=lis1.pop()
	16	a,b
Out	17	(15, 30)



# 串列型別專屬的方法

- (6) `sort()` 方法可以將串列的元素由小到大排序（見第 18 到 20 列；若參數帶入 `reverse=True` 則可以做大到小排序（見第 21 到 23 列）。

In	18	<code>lis1.sort()</code>
	19	<code>lis1</code>
Out	20	<code>[0, 5, 10, 20, 25]</code>
In	21	<code>lis1.sort(reverse=True)</code>
	22	<code>lis1</code>
Out	23	<code>[25, 20, 10, 5, 0]</code>

## 串列型別專屬的方法

(7) `reverse()` 方法可以將串列的值倒著排列（見第 24 到 26 列）。功能 `sort (reverse=True)` 相同。

In	24	<code>lis1.reverse()</code>
	25	<code>lis1</code>
Out	26	<code>[0, 5, 10, 20, 25]</code>

(8) `copy()` 方法可以複製一份串列內容，並回傳給新串列，變數名稱 `lis2`，（見 27 到 29 列）。

In	27	<code>lis2=lis1.copy()</code>
	28	<code>lis2</code>
Out	29	<code>[0, 5, 10, 20, 25]</code>

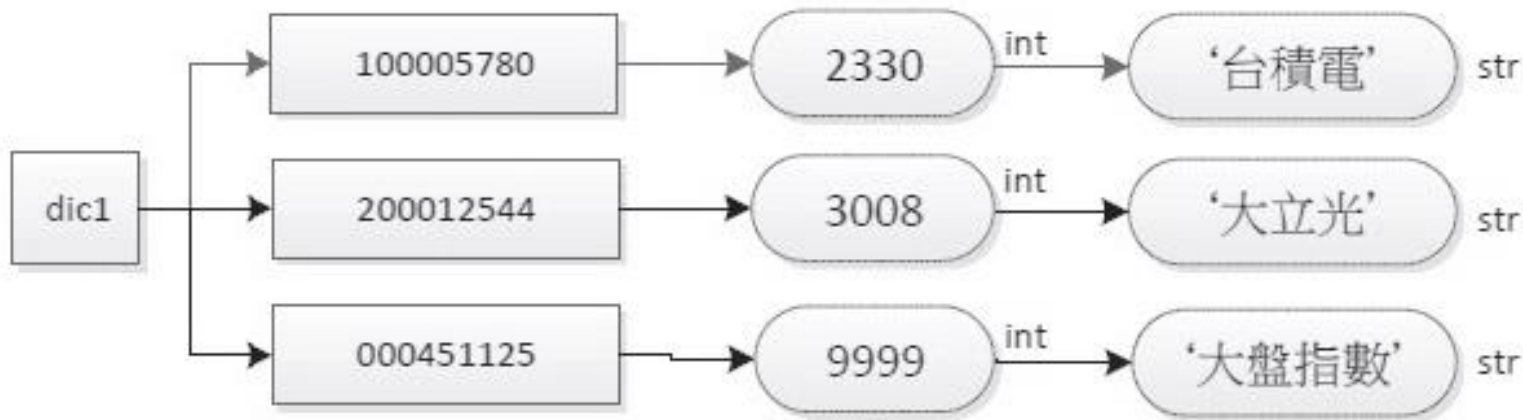
# 串列型別專屬的方法

(9) `clear()`方法可以清除串列內的所有元素，成為空串列，(見30到32列)。

In	30	<code>lis2.clear()</code>
	31	<code>lis2</code>
Out	32	<code>[]</code>

## 容器型別--dict

字典的資料型別比較特別，其元素是配對好的鍵值 (key: value)。同set，dict也是用大括弧「{ }」將元素前後括起來，dict內的鍵 (key) 必須要是唯一識別，此與set一樣必須要經過雜湊運算成為雜湊值，雜湊值先指向鍵，鍵再指向值(value)。



## 容器型別--dict

(1) dict用大括弧「{ }」將元素前後括起來，「鍵」在前、「值」在後，中間用冒號「:」隔開，每一組鍵值需以逗號隔開，(見第1到3列)。

In	1	dic1={2330: '台積電', 3008: '大立光', 9999: '大盤指數'}
	2	dic1
Out	3	{2330: '台積電', 3008: '大立光', 9999: '大盤指數'}

(2) dict的物件實體dic1取值時用中括弧「[]」，並輸入鍵值，(見第4到5列)。

In	4	dic1[2330]
Out	5	'台積電'

## 容器型別--dict

(3)更新某鍵的值時，新值放在等號右邊，指定鍵的名稱放在左邊物件實體（如dic1）的中括弧內，（見第6到8列）。

In	6	dic1[9999]='台股指數'
	7	dic1
Out	8	{2330: '台積電', 3008: '大立光', 9999: '台股指數'}

(4)若存取不存在的鍵，會發生錯誤，見9到11列。

In	9	dic1[1101]
Out	10	...
	11	KeyError: 1101

# 本章講解完畢

現場同學們如有不懂的地方，  
請提出問題。

# 練習

1. 建立一個字串變數內容為 '1101 台泥' 指派給 s1 並印出。
2. 計算 (1) 建立的變數中，輸出 '1' 出現的次數和字串長度。

