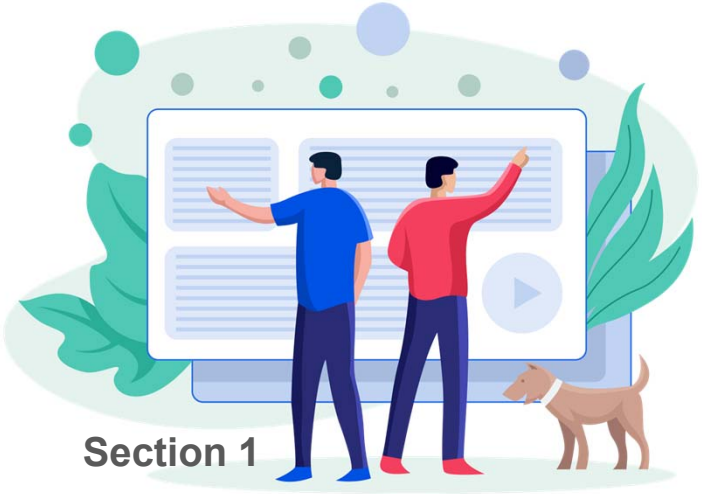
Python文字資料型態
與成員運算子

學習目標

- 使用 Python 文字資料型態
- 成員運算子(Membership Operators)
- 文字資料型態相關運算與方法

文字資料型態

- 介紹 Python 文字資料型態
- 了解文字資料型態的特性與用法

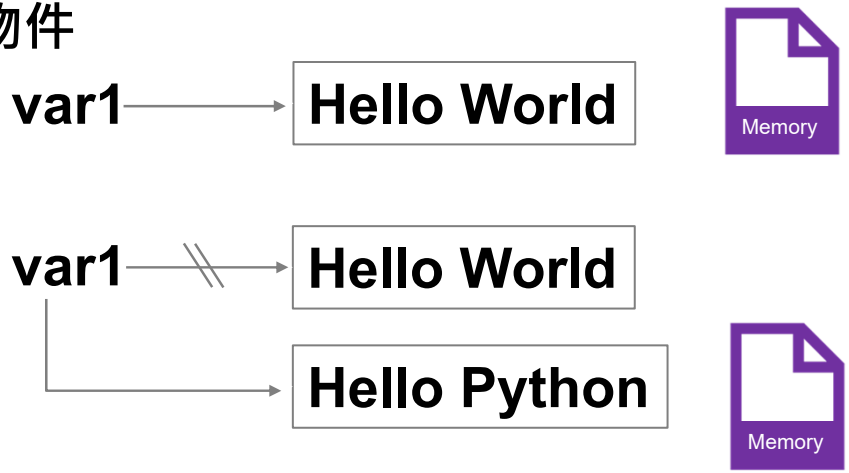


文字型態(Text)

- 常用於文字檔案的讀寫處理
- 用引號括起來
  - Python 單引號(')與雙引號(")視為相同
  - var1 = 'Hello World!'
  - var2 = "Python Programming"
- 可用中括號配合索引(index)或切片(slice)來取得字串內容值
  - var1[0]　　# 'H'
  - var2[1:5]　　# 'ytho'

更新文字

- 文字資料型態是有順序(sequence)，不可改變的(immutable)
- 可以重新參考到另一個全新的字串物件
  - var1 = 'Hello World'
  - var1 = var1[:6] + 'Python'
  - print(var1)
- 字串使用「+」來連結字串
- 字串使用「*」來重複連結字串
- 在 Python 3，所有字串都以 Unicode 格式存放



跳脫字元(Escape Characters)

Escape Sequence	Meaning
\newline	Backslash and newline ignored
\\	Backslash (\)
\'	Single quote (')
\"	Double quote (")
\a	ASCII Bell (BEL)
\b	ASCII Backspace (BS)
\f	ASCII Formfeed (FF)
\n	ASCII Linefeed (LF)
\r	ASCII Carriage Return (CR)
\t	ASCII Horizontal Tab (TAB)
\v	ASCII Vertical Tab (VT)
\ooo	Character with octal value ooo
\xhh	Character with hex value hh

三引號(Triple Quotes)

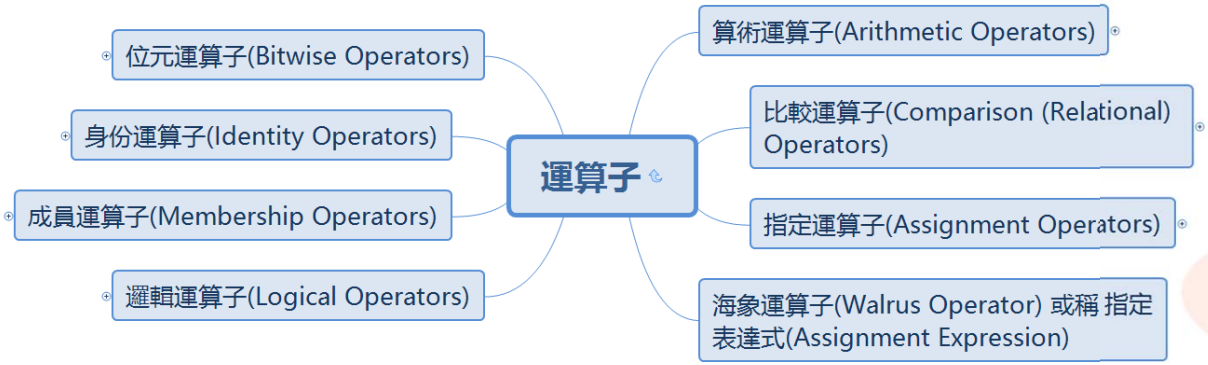
- 三引號為Python自有的語法
- Python 的三引號允許表達多列字串，會保留特殊字元，例如換行或[TAB]鍵
- 以三個單引號(''')或三個雙引號(""")括起來
  - text_para = """This is a long string that contains
    several lines and some non-printable characters,
    such as TAB(\t) or NEWLINE(\n). For triple-quoted strings,
    always use double quote characters to be consistent
    with the docstring convention in PEP 257."""
- 如果是 docstring，永遠使用三個雙引號表達

文字資料型態(Demo)

- 使用 Python 互動式環境使用文字型態的資料

成員運算子

使用成員運算子(Membership Operators)



Section 2

成員運算子

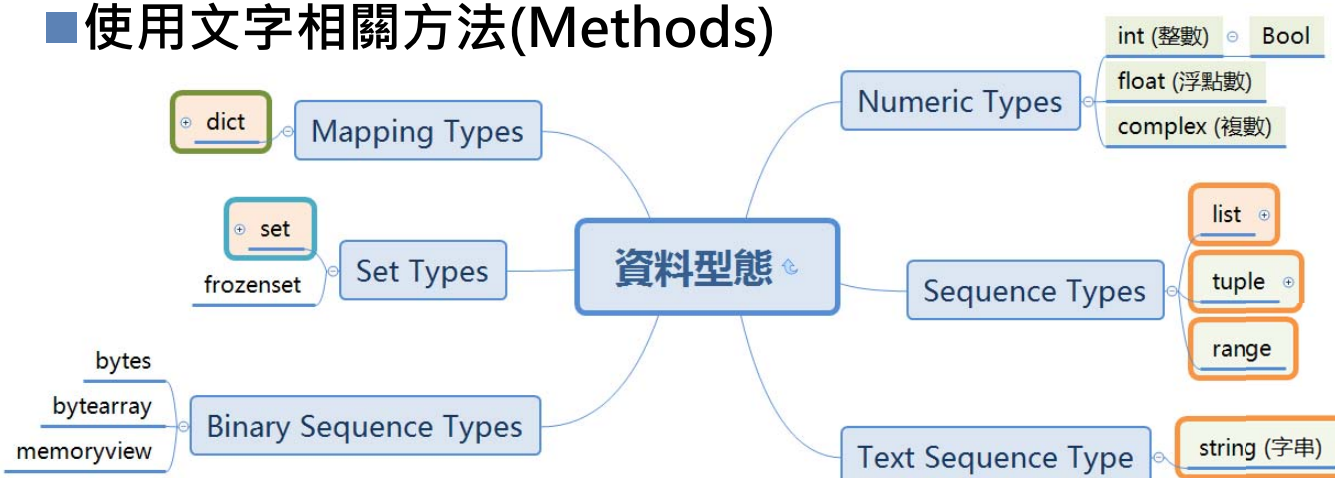
運算子	說明	舉例
in	如果某資料出現在一組資料內，結果為真，否則結果為假	x in y，結果為 True
not in	如果某資料沒有出現在一組資料內，結果為真，否則結果為假	x not in y，結果為 False

假設 x=3, y=[1,2,3,4,5]

- 測試資料是否為成員的一份子，適用於 text(strings)、lists、tuples 或 sets

文字相關運算與方法

- 介紹文字相關運算
- 使用文字相關方法(Methods)



Section 3

文字相關運算

- 假設字串 a 為 'Hello'，字串 b 為 'Python'

運算	說明	舉例
+	文字相加	a + b 等於 'HelloPython'
*	文字重複	a*2 等於 'HelloHello'
[ ]	索引(Index)	a[1] 得到 'e'
[:]	切片(Slice)	a[1:4] 得到 'ell'
in	成員運算	'H' in a 得到 1(True)
not in	成員運算	'm' not in a 得到 1(True)
r/R	Raw String，保留特殊字元，“r” 不分大小寫，必須緊連著左引號	print(r'\n') : 印出 '\n' print(R'\n') : 印出 '\n'
len()	內建函數，求文字長度	len(a) 結果為 5
str()	內建函數，將資料轉成文字	str(45) 結果為 '45'

內建文字型態方法(methods) -1

- Python 內建字串處理的方法，可以有效率的處理文字
- 這些 method 是傳回修改後的字串，因為文字型態是不可改變的 (immutable)
- s.upper()
  - 小寫字串換成大寫字串
- s.lower()
  - 大寫字串換成小寫字串

內建文字型態方法(methods) -2

- s.isalpha(), s.isdigit(), s.isalnum(), s.isspace()
  - 判斷是否為文字、數字、文數字或空白字元(whitespace)
- s.islower(), s.isupper()
  - 判斷小寫字串或大寫字串
- s.split([sep[, maxsplit]])
  - 把字串切割為子字串
  - “sep” 參數可以指定切割字元，預設為空白字元(whitespace)
  - “maxsplit” 參數可以設定最大分割數，預設全部切割(-1)

內建文字型態方法(methods) -3

- s.rsplit([sep[, maxsplit]])
  - 從右邊做字串分割
- s.strip([chars])
  - 傳回一個左右兩邊都移除特定字元的字串，“char” 字串指定要移除的字元，預設為whitespace
- s.rstrip([chars])
  - 傳回一個只有移除右邊特定字元的字串

文字相關運算與方法(Demo)

- 使用索引(Index)、切片(Slice)
- 使用文字相關方法(Methods)

本章重點精華回顧

- Python 文字型態的特性
- 熟悉 Python 成員運算子
- 文字型態的索引(Index) 與切片(Slicing)
- 文字型態相關方法(Methods)

Lab: Python的文字資料型態

- Lab01: 透過互動式環境使用文字資料型態
  - 使用索引(Index)、切片(Slice)
  - 使用文字型態相關函數(function) 與方法(method)

Lab01: 透過互動式環境使用文字資料型態

- 啟動Python互動式執行環境，做以下練習

```
>>> s1 = 'hello'
>>> s1
>>> type(s1)
>>> id(s1)
>>> s1[0], s1[3]
>>> s1 = 'python'
>>> s1
>>> id(s1)
>>> s1[-1], s1[-3]
```

```
>>> s2 = 'Python Basics'
>>> s2[:]
>>> s2[4:]
>>> s2[:-2]
>>> s2[:11]
>>> s2[2:-1]
>>> s2[3:8:2]
>>> 'y' in s2
>>> len(s2)
```

```
>>> s3='''This string
contains [\t] and [\n] chars'''
>>> s3
>>> print(s3)

>>> s4 = '###PYTHON###'
>>> s4.lower()
>>> s4.lstrip('#')
>>> s4.rstrip('#')
>>> s4.strip('#')
```