

字典(Dict)

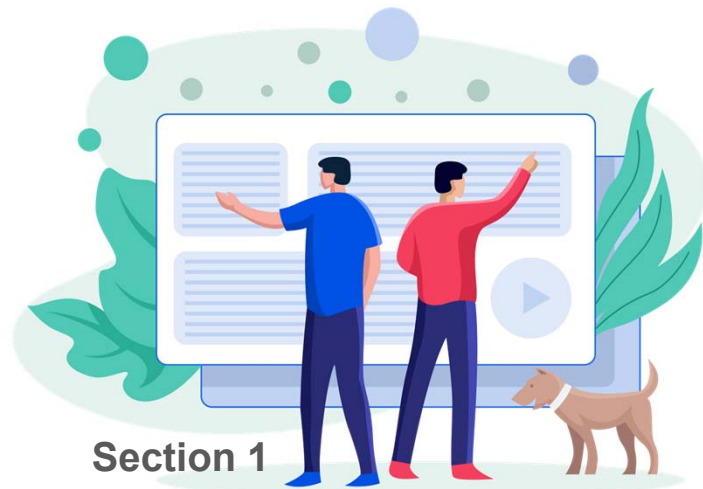
學習目標

- 字典(Dict)宣告與運算
- 字典(Dict)相關的函數與方法
- Dict Comprehensions



字典(Dict)的用法

- 介紹字典(Dict) 的特性
- 如何存取字典(Dict) 的內容



Section 1

字典(Dict)

- 字典(dict) 是沒有順序(non-sequence) , 可以改變(mutable)
- Key 與 value 用冒號(:) 隔開 , 每個項目以逗點隔開 , 最後整個字典用大括號括起來
- { } 代表空字典(dict)
- 在字典內 keys 必須唯一 , 但 values 可以相同
- 字典內的 values 可以是任意的資料型態 , 但 keys 必須是不可變的 (immutable) 資料型態 , 例如字串(strings) 、數值(numbers) 或元組(tuples)

建立、取得字典的資料

■ 也可以使用 `dict` 內建函數產生字典

- ▣ `dict1 = dict(one=1, two=2, three=3)`

- ▣ `dict1['two']` `# 2`

■ 可以使用中括號配合 `key` 值來存取字典的項目值

- ▣ `dict2 = {'Name': 'LabVIEW360', 'Year': 2000, 'Class': 'A'}`

- ▣ `dict2['Name']` `# 'LabVIEW360'`

- ▣ `dict2['Year']` `# 2000`

■ 如果 `key` 不存在字典內，則會觸發 `KeyError` 的例外(exception)

- ▣ `dict2['Stars']` `# trigger exception`

更新字典內容 -1

■可以透過 key-value 配對來修改或新增項目

□dict1 = {'Name': 'III', 'Year' : 1979, 'Class': 'A'}

□dict1['Class'] = 'A+' # update existing entry

□dict1['Stars'] = 5 # New entry

□dict1['Class'] # 'A+'

□dict1['Stars'] # 5

■可以透過成員運算子檢查字典是否有某個 key 值

□'Stars' in dict1 # True

□'Class' not in dict1 # False

■字典最強大的功能在於可以直接加入一個key : value，而這跟python資料分析的套件如NumPy、Pandas息息相關

更新字典內容 -2

■字典的 values 也可以是 List、Tuple 或 Set

- ▣ `dict1 = {'id':'A001', 'eng':[67,53,79]}`

- ▣ `dict1['id']` `# 'A001'`

- ▣ `dict1['eng']` `# [67, 53, 79]`

- ▣ `dict1['eng'][0]` `# 67`

- ▣ `dict1['math'] = [64,85,75]` `# new entry`

- ▣ `dict1['math'][1]` `# 85`

- ▣ `dict1['math'][1] = 88` `# update list value`

■串列(List) 的內容是可以變更的

刪除字典項目

- 可以刪除字典內某一項目，或是清除整個字典

- 也可以刪除整個字典

- `dict1 = {'Name': 'III', 'Year' : 1979, 'Class': 'A'}`

- `del dict1['Class']` # remove with key 'Class'

- `dict1.clear()` # remove all entries in dict

- `del dict1` # remove entire dict

- `dict1.clear()`：清除字典dict1內所有keys與values，但字典dict1本身還存在。

- `del dict1`：清除整個字典dict1

字典(Dict)的用法(Demo)

■如何使用字典(Dict)



字典(Dict)相關函數與方法

■介紹字典(Dict) 的函數與方法



內建字典相關函數 -1

■ len(dict)

- ▣ 取得 dict 項目個數

■ max(dict)

- ▣ 傳回 dict 內最大的 key 值
- ▣ 所有項目的 key 值必須是相同的資料型態

■ min(dict)

- ▣ 傳回 dict 內最小的 key 值
- ▣ 所有項目的 key 值必須是相同的資料型態

內建字典相關函數 -2

■ sorted(dict)

- ▣ 預設傳回依字典 key 值升冪排序的串列(List)，可用 reverse=True 引數做降冪排序
- ▣ 所有項目的 key 值必須是相同的資料型態

■ dict(mapping|iterable|**kwarg)

- ▣ 將資料轉成 dict
- ▣ dict(one=1, two=2, three=3)
- ▣ dict([('two', 2), ('one', 1), ('three', 3)]) # list of tuples
- ▣ dict((('two', 2), ('one', 1), ('three', 3))) # tuple of tuples
- ▣ dict({'three': 3, 'one': 1, 'two': 2})

字典相關方法 -1

■ dict.clear()

- ▣ 移除 dict 所有項目

■ dict.copy()

- ▣ 從 dict 複製一份新的字典

■ dict.get(key[, default])

- ▣ 取得 “key” 對應的值，如果 “key” 不存在則傳回 default 值，此方法(Method) 不會觸發 KeyError 的例外(Exception)

■ dict.setdefault(key[, default])

- ▣ 類似 get()，但是如果 “key” 不存在則會產生一個新的 dict[key]=default 的項目

■ dict.get與dict.setdefault比較特別，可以迴避錯誤訊息產生。

字典相關方法 -2

■ dict.items()

- ▣ 傳回一個tuple格式 (key, value) 的 list

■ dict.keys()

- ▣ 傳回一個包含所有keys的list

■ dict.values()

- ▣ 傳回一個包含所有values的list

■ dict.update(other)

- ▣ 將 other 字典加到 dict 內
- ▣ 將兩個字典整合為同一個字典

字典相關方法 -3

■ dict.pop(key[, default])

- 如果 “key” 存在，則傳回該 “key” 對應的值，並將該項目從 dict 中移除
- 如果沒有那一個指定的key的話，則回傳預設(default)的訊息

■ dict.popitem()

- 從 dict 移除並傳回一個 (key, value) 配對的資料。配對是以 LIFO (後進先出)順序傳回
- 從 3.7 版以後，保證以 LIFO 順序傳回。之前的版本是隨意傳回一個配對

字典(Dict)相關函數與方法(Demo)

■ 如何使用字典(Dict) 相關函數與方法



Dict Comprehensions

- 介紹 Dict comprehensions 的語法
- 如何使用 Dict comprehensions



使用Dict Comprehensions

■ Python 的字典也有 Comprehensions

■ 語法

- `{ key_expression : value_expression for expression in iterable }`

■ 範例

- `word = 'letters'`

- `char_counts = {char: word.count(char) for char in set(word)}`

- `char_counts`

- `{'l': 1, 'e': 2, 't': 2, 'r': 1, 's': 1}`

Dict Comprehensions(Demo)

- 使用 Dict comprehensions 簡化程式



本章重點精華回顧

- 字典(Dict) 的特性與用途
- Dict comprehensions 的用法



Lab: 字典

- Lab01: 使用字典(Dict)
- Lab02: 使用 Dict comprehensions

Lab01: 使用字典(Dict) -1

■ 啟動Python互動式執行環境，做以下練習

■ >>> dict1 = {}

■ >>> dict1

■ >>> type(dict1)

■ >>> dict2 = {'eng':70,'math':82,'comp':63}

■ >>> dict2

■ >>> dict2['math']

■ >>> dict2['math'] = 90

■ >>> dict2

■ >>> dict2['phys'] = 50

■ >>> dict2

■ >>> len(dict2)

■ >>> dict1 = {'eng':70,'math':82}

■ >>> dict2 = {'comp':63,'phys':50,'chem':90}

■ >>> dict1

■ >>> dict2

■ >>> dict1.update(dict2)

■ >>> dict1

■ >>> 'eng' in dict1

■ >>> 'chem' not in dict1

Lab01: 使用字典(Dict) -2

```
■>>> dt = {'dept':'bdse', 'python':[82, 77, 59]}
■>>> dt['dept']
■>>> dt['python']
■>>> dt['python'][0]
■>>> dt['java'] = [64,85,75]
■>>> dt['java'][1]
■>>> dt['java'][1] = 88
■>>> dt
```

```
■>>> del dt['python']
■>>> dt
■>>> dt.clear()
■>>> dt
■>>> del dt
■>>> dt
```

Lab02: 使用Dict comprehensions

```
■>>> d1 = {x:x*x for x in range(10)}
```

```
■>>> d1
```

```
■>>> d2 = {x:x*x for x in range(10) if x % 2 == 1}
```

```
■>>> d2
```