



網路程式設計

Python資料程式設計

Instructor: 馬豪尚

檔案和目錄管理

- › OS模組提供取得目錄、建立目錄、刪除目錄、刪除檔案、執行作業系統命令等函式
- › `os.getcwd`: 取得目前的工作目錄
- › `os.remove`: 刪除指定的檔案
- › `os.path`: 用以處理檔案路徑和名稱
 - `abspath()`: 傳回檔案完整的路徑名稱
 - `basename()`: 傳回檔案指定路徑名稱中, 最後的檔案或目錄名稱
 - `dirname()`: 傳回指定檔案的完整路徑
 - `exists()`: 檢查指定的檔案或路徑是否存在
 - `split()`: 將目標路徑分成目錄路徑和檔案名稱並回傳兩個值
 - `Join(path, fname)`: 組合目錄路徑和檔案名稱成為完整路徑

檔案和目錄管理

- › `os.walk()`: 搜尋指定目錄及其子目錄，會回傳包含三個元素的值，分別為目錄名稱、下一層的目錄名稱以及所有檔案名的串列(list)

檔案和目錄管理

- › glob模組: 可以取得指定條件的檔案串列
- › glob.glob("路徑名稱")
 - 路徑名稱可以明確指定檔案名稱也可以使用"*"萬用字元
 - 例如"os*.py": 代表列出開頭為"os"結尾為".py"的檔案
 - "*.py"代表列出所有結尾為".py"的檔案

檔案開啟

- › open函式有8個參數，最常用的是檔案名稱、模式以及編碼，只有檔案名稱不能省略其他都可以省略使用預設值
- › f=open(檔案名稱,[模式],[編碼])
 - open函式會建立一個物件，利用這物件就可以處理檔案
 - close()可以關閉這個物件

› 模式:

r: 讀取(預設)	r+: 可讀寫模式，會從第一行的開頭寫入(覆蓋)
w: 寫入模式，指定檔案不存在時會建立檔案再寫入，檔案若存在會覆蓋	w+: 先寫入再讀取，指定檔案不存在時會建立檔案再寫入，檔案若存在會覆蓋
a: 附加模式，指定檔案不存在時會建立檔案再寫入，檔案若存在會附加到檔案末端	a+: 先讀取再寫入，指定檔案不存在時會建立檔案再寫入，檔案若存在會附加到檔案末端

檔案開啟

- › with敘述: with結束後會自動關閉開啟的檔案，就不需要使用close來主動關閉檔案
- › with open("檔案名稱", 模式) as f

檔案讀寫方法

- › `read()`: 一次讀取全部的文字
- › `read([size])`: 讀取目前指標位置之後指定長度為`size`的字元
- › `readlines()`: 一次讀取所有列，回傳一個串列
- › `readline([size])`: 讀取目前指標位置所在列中指定長度為`size`的字元，忽略`[size]`參數即為讀取一整列
- › `write(str)`: 將指定字串寫入文件
- › `writelines(list)`: 將指定的串列寫入文件

檔案讀寫-編碼

- › 一般國際通用的編碼為UTF-8
 - `f=open(filename, 'r', encoding = 'utf-8')`
- › Windows系統的中文檔案編碼預設為ANSI
 - `f=open(filename, 'r', encoding = 'cp950')`

Python 字串(String)

› 分割與合併

- `s.split()`: 默認以空格、換行字元分割字串`s`，返回列表
- `s.join(seq)`: 以`s`為分隔符，將`seq`中的元素串起來成為一個新的字串

Example

```
S = "Horse-Horse-Tiger-Tiger"
print(S.split('-')) #['Horse', 'Horse', 'Tiger', 'Tiger']
print(S.split('Horse')) #['', '-', '-Tiger-Tiger']
```

```
SS = ["HOW", "ARE", "YOU"]
print(' '.join(SS)) # 印出 HOW ARE YOU
print('-'.join(SS)) # 印出 HOW-ARE-YOU
```

Python 字串(String)

› 查找

- `s.find(str)`: 返回`str`第一次在字串`s`中出現的index，若找不到則返回-1
- `s.count(str)`: 返回`str`在字串`s`中出現的次數

Example

```
s = "believe"
```

```
print(s.find("lie")) #印出2
```

```
print(s.find("le")) #印出-1
```

```
print(s.count("e")) #印出3
```

Python 字串(String)

› 替換

– `s.replace(str1, str2)` 將s中的str1替換成str2

Example

```
s1 = "ABBABBAAB"
```

```
print(s1.replace('A','C')) #印出 CBBCBBCCCB
```

Python 字串(String)

- › `s.lower()`: 將字串s裡的字母全部改成小寫
- › `s.upper()`: 將字串s裡的字母全部改成大寫
- › `s.swapcase()`: 將字串s的字母大小寫翻轉
- › `s.lstrip()`: 去除字串s左邊的空格
- › `s.rstrip()`: 去除字串s右邊的空格
- › `s.strip()`: 去除字串s左、右兩邊的空格
- › `s.center(width)` 返回一個居中的字串，將左右兩邊填充至長度width

Numpy – 建立ndarray

- › Import numpy as np
- › array()可以使用list或tuple來建立一維陣列
 - np.array([1, 2, 3, 4]) -List
 - np.array((5, 6, 7, 8)) -Tuple
- › dtype 可以設定資料的型態
 - np.array([1, 2, 3, 4], dtype=int)
 - np.array([1, 2, 3, 4], dtype=float)

Numpy – 建立ndarray

› 全部值一樣的陣列

- zeros() : 產生全部為零的陣列
- ones() : 全部為1的陣列
- empty() : 無初始值的陣列

› 等差陣列

- arange(start, stop, step, dtype=None) : step為等差值
- linspace(start, stop, num, endpoint=True) : num代表陣列的大小

Numpy – 建立ndarray

- › 建立多維陣列

- listdata = [[1,2,3,4,5],
 [6,7,8,9,10],
 [11,12,13,14,15]]

- `np.array(listdata)`

Numpy – 建立ndarray

› 隨機數列

- `random.random(size)`
- `random.ranf(size)`
- `random.sample(size)`
- `random.randint(low, high, size)`
- `random.normal(loc='loc', scale='scale', size=None)` : `loc`為平均值、`scale`為標準差

Numpy 陣列屬性

- › `ndim()` : 取得陣列的維度數量
- › `shape()` : 陣列的形狀
- › `size()` : 陣列的數量
- › `dtype()` : 資料型態
- › `itemsize()` : 陣列中元素的大小(位元組為單位)
- › `nbytes()` : 陣列的大小(位元組為單位) 一般來說 $nbytes = itemsize * size$

Numpy 陣列操作

- › `reshape()`: 改變陣列的形狀
 - `np.array([1, 2, 3, 4])` → `shape=1*4`
 - `reshape(2, 2)` → `shape =2*2`
- › 串接 `x=[1, 2, 3]` `y=[4, 5, 6]`
 - `concatenate([x,y])`: 串接
 - › `[1, 2, 3, 4, 5, 6]`
 - `vstack([x,y])`: 垂直串接
 - › `[[1, 2, 3], [4, 5, 6]]`
 - `hstack([x,y])`: 水平串接
 - › `[1, 2, 3, 4, 5, 6]`



Numpy 陣列操作

- › `z = np.array([1,2,3,4,5,6,7,8,9])`
 - `np.split(z,[3,5])` → `([1,2,3], [4,5], [6,7,8,9])`
- › `z = np.array([[1,2,3],[4,5,6],[7,8,9]])`

1	2	3
4	5	6
7	8	9

– `np.vsplit(z,[2])` →

1	2	3
4	5	6
7	8	9

– `np.hsplit(z,[2])` →

1	2	3
4	5	6
7	8	9

Numpy 陣列取值

› 一維陣列

0	1	2	3	4	5
---	---	---	---	---	---

–a[index]

› a[0]

0	1	2	3	4	5
---	---	---	---	---	---

–a[start:end]

› a[1:5]

0	1	2	3	4	5
---	---	---	---	---	---

› a[:3]

0	1	2	3	4	5
---	---	---	---	---	---

–a[start:end:gap]

› a[1:5:2]

0	1	2	3	4	5
---	---	---	---	---	---

Numpy 陣列取值

› 多維陣列 `a = np.arange(1, 17).reshape(4, 4)`

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

`a[2, 3]`

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

`a[1:3, 1:3]`

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

`a[1, 1:3]`

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

`a[:, 2]`

Numpy 陣列運算

- › `a = np.arange(1,10).reshape(3,3)`
- › `b = np.arange(10,19).reshape(3,3)`
- › 對所有元素都加一個值

– `a+1` →

2	3	4
5	6	7
8	9	10

- › 對所有元素取平方

– `a**2` →

1	4	9
16	25	36
49	64	81

1	2	3
4	5	6
7	8	9

a

10	11	12
13	14	15
16	17	18

b

Numpy 陣列運算

- › 取出指定陣列的元素進行運算

– $a[0, :] + 1 \rightarrow$

2	3	4
---	---	---

- › 矩陣相加

– $a+b \rightarrow$

11	13	15
17	19	21
23	25	27

- › 矩陣元素相乘

– $a*b \rightarrow$

10	22	36
52	70	90
112	136	162

- › 矩陣內積

– $\text{np.dot}(a, b) \rightarrow$

84	90	96
201	216	231
318	342	366

Numpy 常用的計算與統計函數

- › `sum()`: 加總
- › `prod()`: 乘積
- › `mean()`: 平均值
- › `min()`: 最小值
- › `max()`: 最大值
- › `std()`: 標準差
- › `var()`: 變異數
- › `median()`: 中位數
- › `argmin()`: 最小元素值索引
- › `argmax()`: 最大元素值索引
- › `cumsum()`: 陣列元素累加
- › `cumprod()`: 陣列元素累積
- › `percentile()`: 以百分比顯示陣列中的指定值
- › `ptp()`: 最大值與最小值的差

Numpy 排序函數

› 一維陣列排序

- `np.sort()`: 對陣列中的值進行排序並返回結果
- `np.argsort()`: 對陣列中的值進行排序並返回索引值

45	28	21	47	11	26	30	22	15	16
----	----	----	----	----	----	----	----	----	----

sort()	11	15	16	21	22	26	28	30	45	47
--------	----	----	----	----	----	----	----	----	----	----

argsort()	4	8	9	2	7	5	1	6	0	3
-----------	---	---	---	---	---	---	---	---	---	---

Numpy 排序函數

- › 多維陣列排序，可以用axis軸來設定排序方式

a

2	4	1	6	2
2	2	4	6	9
5	1	6	0	9

- › `np.sort(a, axis=0)` → 對直行進行排序

2	1	1	0	2
2	2	4	6	9
5	4	6	6	9

- › `np.sort(a, axis=1)` → 對直行進行排序

1	2	2	4	6
2	2	4	6	9
0	1	5	6	9

練習1

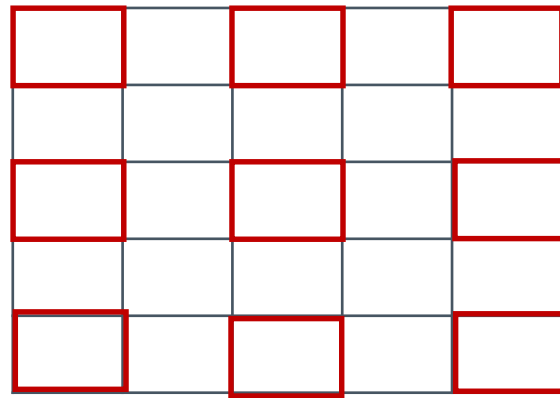
- › 請將輸入字串中英文字中的母音(aeiou)都去除
 - 假設輸入apple → 印出ppl
 - 假設輸入animal → 印出nml

練習2

- › 請產生一個隨機1-4的整數, 6個數字的數列
- › 請將這個數列用one-hot encoding的方式來表示
 - $1=[1,0,0,0]$, $2=[0,1,0,0]$, $3=[0,0,1,0]$, $4=[0,0,0,1]$

練習3

- › 請產生一個隨機0-10整數的5*5二維陣列
- › 請把每隔一行(row)且隔一欄(column)的值取出, 不能用指定位置的方式



- › 請將取出的值, 依照大小從左上角開始, 依序擺放在陣列中
 - 每一列的最左邊為該列最小值, 每一欄最上方也是該欄最小

練習4

- › 請讀取一個文件檔, 該文件檔是一篇論文的内容, 請將除了章節標題以外的文字做反序
 - 例如: 我的興趣是打籃球and play piano → onaip yalp dna球籃打是趣興的我