Python 程式設計

林奇賦　daky1983@gmail.com

Outline

- 字串處理

跳脫序列

跳脫序列	意義
\\	反斜線符號 (\)
\'	單引號 (')
\"	雙引號 (")
\a	響鈴符號 (BEL)
\b	空格符號 (BS)
\f	換頁符號 (FF)
\n	換行符號 (LF)
\r	返回符號 (CR)
\t	水平縮排符號 (TAB)
\v	垂直跳格符號 (VT)
\ooo	ooo 是三個八進位的數字
\xhh	*hh* 是兩個十六進位的數字

字串處理

- 字串型態的切片(Slice)
  - Ex:

```
>>> toast="PYTHONSLICE“
>>> toast[-5]
'S‘
>>> toast[-5:]
'SLICE'
>>> toast[:6]
'PYTHON‘
>>> toast[:5]+toast[5:]
'PYTHONSLICE'
```

  - Ex:

```
>>> 'PYTHONSLICE'[:6]==toast[:6]
True
```

字串處理

- 元組型態的切片
  - Ex:

```
>>> toast="PYTHONSLICE“
>>> tuples=toast[0:3],toast[3:6],toast[6:9],toast[9:11]
>>> tuples
('PYT', 'HON', 'SLI', 'CE')
>>> tuples[0]
'PYT'
>>> tuples[2:4]
('SLI', 'CE')
>>> tuples[1][0]
'H'
```

  - Ex:

```
>>>
("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")[3]
'Wednesday'
```

字串處理

- 序列型態的切片
  - Ex:

```
>>>
["Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"][random.randint(0,6)]
'Saturday‘
```

  - Ex:

```
>>>
days=["Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"]
>>> days[2]='星期二'
>>> print(days[2])
星期二
```

字串處理

- 字典型態的切片
  - Ex:

```
>>>
days={1:"Sunday",2:"Monday",3:"Tuesday",4:"Wednesday",5:"T
hursday",6:"Friday",7:"Saturday"}
>>> days[2],days[3],days[4]
('Monday', 'Tuesday', 'Wednesday')
```

- 數字型態的切片
  - 必須先用str()函數轉換成字串

字串處理

- 字串函數處理與格式化
  - 旗標(Flag)指定格式化的字串變數
  - Ex:

```
>>> print("會員編號1:%d, 會員編號2:%d" % (10, 20))
會員編號1:10, 會員編號2:20
>>> print("會員編號2:%(#2)d, 會員編號1:%(#1)d"%{"#1":10, "#2":20})
會員編號2:20, 會員編號1:10
>>> print("會員編號1:%(num1)d, 會員編號2:%(num2)d"%{"num1":10, "num2":20})
會員編號1:10, 會員編號2:20
```

字串處理

- 字串函數處理與格式化
  - 補0
  - Ex:

```
>>> print("會員編號:%(#)08d" % {"#" : 123456})
會員編號:00123456
```

  - Ex:

```
>>> print("%8.2f" % (123.456))
  123.46
```

  - Ex:

```
>>> money=987.98
>>> print("$%*.2f" % (7, money))
$ 987.98
```

字串處理

- 字串函數處理與格式化
  - 輸出格式不只接受單純的數字和字串型態的變數，也可帶入整個字典型態變數
  - Ex:

```
>>> name={"game":"xbox", "apple":"iphone","camera":"nikon"}
>>> print("%(apple)s,%(camera)s,%(game)s"%name)
iphone, nikon, xbox
```

printf 風格的字串格式化

轉換	含義
%d	有符號整型十進制
%i	有符號整型十進制
%o	有符號八進製值
%u	過時類型 – 等同 %d
%x	有符號十六進制(小寫)
%X	有符號十六進制(大寫)
%e	浮點數指數格式 (小寫)
%E	浮點數指數格式 (大寫)
%f	浮點數十進制格式
%F	浮點數十進制格式
%g	浮點數格式。使用小寫指數格式，若指數小於 -4 或不低於精度的話；否則，會使用十進制格式
%G	浮點數格式。使用大寫指數格式，若指數小於 -4 或不低於精度的話；否則，會使用十進制格式
%c	單字符 (接受整數或單字符字符串)
%r	字符串 (以 repr() 內置函數轉換任何 Python 對象)
%s	字符串 (以 str() 內置函數轉換任何 Python 對象)
%a	字符串 (以 ascii() 內置函數轉換任何 Python 對象)
%%	不轉換自變量，會導致 ‘%’ 字符出現在結果中

方法	描述
str.capitalize()	回傳將 str 改成首字母大寫，其餘字母小寫的字串
str.center(width[, fillchar])	回傳一個將 str 設置字串中央，長度 width 的新字串， fillchar 為填充字元，預設為空格
str.count(sub[, start[, end]])	計算 sub 出現的次數， start 為起始計算索引值， end 為結束索引值
str.encode(encoding="utf-8", errors="strict")	回傳 encoding 版本的 bytes 物件
str.endswith(suffix[, start[, end]])	判斷 str 是否以 suffix 結尾
str.expandtabs([tabsize])	將 tab 符號以 tabsize 的空格數替換
str.find(sub[, start[, end]])	回傳 sub 第一次出現的索引值
str.format(*args, **kwargs)	進行格式化字串運算
str.index(sub[, start[, end]])	回傳 sub 第一次出現的索引值
str.isalnum()	判斷字串中的字元是否至少一個是字母或數字
str.isalpha()	判斷字串中的字元是否至少一個是字母
str.isdecimal()	判斷字串中所有字元是否是十進位數字
str.isdigit()	判斷字串中所有字元是否是數字
str.isidentifier()	判斷字串是否可作為合法的識別字
str.islower()	判斷字串中所有字母字元是否都是小寫字母
str.isnumeric()	判斷字串中所有字元是否是數字
str.isprintable()	判斷字串中所有字元是否都屬於可見字元

方法	描述
str.isspace()	判斷字串是否為空格字元
str.istitle()	判斷字串是否適合當作標題
str.isupper()	判斷字串中所有字母字元是否都是大寫字母
str.join(iterable)	回傳將 str 連結 iterable 各元素的字串
str.ljust(width[, fillchar])	回傳將 str 在寬度 width 向左對齊的字串， fillchar 為填充字元，預設為空格
str.lower()	將 str 的英文字母都改成小寫
str.lstrip([chars])	回傳將 str 左邊具有 chars 字元去除的拷貝版本， chars 預設為空格符號
static str.maketrans(x[, y[, z]])	回傳 x 與 y 配對的 Unicode 編碼字典，若有提供 z ， z 中的字元會跟 None 配對
str.partition(sep)	以 sep 分割 str 為三個部份，結果回傳具有三個子字串的序對
str.replace(old, new[, count])	將 str 中的 old 子字串以 new 代換
str.rfind(sub[, start[, end]])	尋找最右邊的 sub ， 也就是索引值最大的 sub
str.rindex(sub[, start[, end]])	尋找最右邊的 sub ， 也就是索引值最大的 sub
str.rjust(width[, fillchar])	回傳將 str 在寬度 width 向右對齊的字串， fillchar 為填充字元，預設為空格

方法	描述
str.rpartition(sep)	以 sep 從最右端分割 str 為三個部份，結果回傳具有三個子字串的序對
str.rsplit([sep[, maxsplit]])	將 str 從最右端以 sep 分割成子字串，回傳儲存子字串的串列， maxsplit 為子字串最多的數量
str.rstrip([chars])	從 str 的最右端中移除 chars 字元，預設為空白字元
str.split([sep[, maxsplit]])	將 str 以 sep 分割成子字串，回傳儲存子字串的串列，maxsplit 為子字串最多的數量
str.splitlines([keepends])	將 str 以新行符號分割成子字串，回傳儲存子字串的串列
str.startswith(prefix[, start[, end]])	判斷 str 是否以 prefix 開頭
str.strip([chars])	從 str 中移除 chars 字元，預設為空白字元
str.swapcase()	將 str 中的英文字母進行大小寫轉換
str.title()	將 str 轉換成作為標題的字串
str.translate(map)	將 str 中的字元以 map 中配對的字元轉換
str.upper()	將 str 的英文字母都改成大寫
str.zfill(width)	回傳以 0 塞滿 width 的新字串
str.rpartition(sep)	以 sep 從最右端分割 str 為三個部份，結果回傳具有三個子字串的序對

字串函數使用

- 字串函數使用
  - string.capitalize()函數
    - 變數第一個字轉變為大寫
    - Ex:

```
>>> print("how are you?".capitalize())
How are you?
```

字串函數使用

- 字串函數使用
  - string.center(width)函數
    - width引數決定對齊的總長度
    - Ex:

```
>>> text1="first line....."
>>> text1.center(50)
'                  first line.....                '
```

字串函數使用

- 字串函數使用
  - string.count(sub[, start[, end]])
    - 回傳此字串裡有多少個sub引數字元
    - Ex:

```
>>> text='abbggccdeefgggijklgglmo'
>>> text.count('g')
7
>>> text.count('g',4,-4)
5
```

字串函數使用

- 字串函數使用
  - str.endswith(suffix[, start[, end]])
    - 判斷字串內是否有符合suffix引數的值
    - Ex:

```
>>> images="xbox.gif, iphone.jpg"
>>> images.endswith(".jpg")
True
>>> images.endswith(".gif",0, 8)
True
>>> images.endswith(".gif")
False
```

字串函數使用

- 字串函數使用
  - string.find(sub[, start[, end]])
    - 搜尋字串變數裡符合sub引數的字元位置
    - Ex:

```
>>> text='abcdefgabcdefg'
>>> text.find('a')
0
>>> text.find('a',1)
7
```

字串函數使用

- 字串函數使用
  - str.format(format_string, *args, **kwargs)
    - 將輸入的format_string引數變數進行格式化
    - 數字可以省略
    - Ex:

      >>> "{0} makes a full man, and {1} an exact man.".format("Reading", "writing")
      'Reading makes a full man, and writing an exact man.'
    - 沒有強調限制 "{ }" 符號內的名稱一定要數字
    - Ex:

      {a}...{b}....format(a="Reading“, b="writing")
    - 允許對參照關係符號定義寬度，長度不夠會自動填滿

字串函數使用

- 字串函數使用
  - string.index(sub[, start[, end]])
    - 與string.find()類似，差異在當s字串變數內搜尋不到sub字串會回傳ValueError錯誤訊息
    - Ex:

```
>>> text="abcdeabcde"
>>> text.index('d', 4)
8
```

字串函數使用

- 字串函數使用
  - str.isalnum()
    - 判斷該變數裡的內容是否為[a-z]、[A-Z]與[0-9]的字元
    - 不可以判別多行宣告
  - str.isalpha()
    - 與str.isalnum()的差異在於這個函數只接受字串內有英文字母

字串函數使用

- 字串函數使用
  - str.isdigit()
    - 判斷字串內的數字
  - str.islower()
    - 判斷字串變數內的字元是否全部都是小寫
  - str.isspace()
    - 判斷字串變數是否為空白字元

課堂練習

使用者可以輸入任意整數 n

當輸入的n不為整數，提示使用者輸入型態錯誤，並 且重新讓使用者繼續輸入

若輸入的值為整數，將其print至螢幕上

- ex.

  n=100

字串函數使用

- 字串函數使用
  - str.istitle()
    - 判斷字串變數裡的第一個字是否為大寫
    - 如果宣告一句英文句子，句子裡的每一個單字都會判斷
  - str.isupper()
    - 判斷字串變數內的所有字母都必須要大寫
    - 不會理會特殊字元

字串函數使用

- 字串函數使用
  - string.ljust(width)
    - 將傳入的s字串進行向左對齊，width引數是指定對齊的總寬度
    - Ex:

```
>>> text="abcdefghijkl"
>>> text.ljust(20)
'abcdefghijkl        '
```

字串函數使用

- 字串函數使用
  - string.lower()
    - 將string內的字元從大寫字母轉換為小寫字母
  - str.replace(old, new, count)
    - 將字串內所有符合old引數以new引數的字元來替代，而count引數是指定只要代替的數目
  - sring.rfind(sub[,start[, end]])
    - 從右到左尋找，sub引數是預計要搜尋的字元

字串函數使用

- string.lstrip([chars])
  - 將s字串變數內左邊的多於空白字元去掉，chars引數必須傳入字串型態
  - chars決定string.lstrip()函數要去掉字串變數內的那些字元，預設只會刪去空白字元
  - Ex:
    - >>> text = " aaaaa bbbbbbaaaccccc"
    - >>>

字串函數使用

- 字串函數使用
  - str.partition(sep)
    - 將字串做分割，但只會分割第一個符合sep引數的字元，形成3-tuple
    - Ex:

```
>>> names = 'Tom,John,Mary,Bob,Sunny'
>>> names.partition(',')
('Tom', ',', 'John,Mary,Bob,Sunny')
>>> names.partition(',')[-1]
'John,Mary,Bob,Sunny'
>>> names.rpartition(',')
('Tom,John,Mary,Bob', ',', 'Sunny')
```

字串函數使用

- 字串函數使用
  - string.split(sep, maxsplit)
    - 由左至右，將string字串變數內的字元以sep引數字元為分隔字元進行分割
    - 找不到符合sep的值，就會回傳整個字串

字串函數使用

- 字串函數使用
  - str.splitlines(keepends)
    - 將字串進行分割
    - 以“\n”和“\r”作為分割的區隔字元
    - 以序列型態回傳
    - Keepends引數預設False，設為True會連同脫逸字元一併回傳
  - str.startswith(prefix[, start[, end]])
    - 判斷傳入的prefix字串字元是否為開始字元

字串函數使用

- 字串函數使用
  - string.strip([chars])
    - 將string字串變數裡的左右兩邊的空白字元刪除掉
    - chars引數不為None時會決定string.strip()函數要刪除的字元
  - string.swapcase()
    - 將string字串裡的字母大小寫互轉

字串函數使用

- 字串函數使用
  - string.rjust(width)
    - 與string.ljust()有相反的意思
  - str.rpartition(sep)
    - 與string.partition()類似
  - string.rsplit(sep[, maxsplit]])
    - 將字串變數s裡面的值進行分割，分割的參考字元是sep引數裡的字元，其結果以序列型態儲存

字串函數使用

- 字串函數使用
  - string.rindex(sub[, start[, end]])
    - 由右至左搜尋，s字串變數搜尋不到sub字串將會回傳ValueError錯誤訊息
  - string.rstrip([chars])
    - 將x字串變數內右邊的多於空白字元去掉

字串函數使用

- 字串函數使用
  - string.title()
    - 將字串內所有為[a-z]的單字第一個字元轉換成大寫
  - string.translate(map)
    - 將 string 中的字元以 map 中配對的字元轉換
    - 搭配str.maketrans(from, to)

字串函數使用

- 字串函數使用
  - string.upper()
    - 將string字串變數內的字母從小寫轉換為大寫
  - string.zfill(width)
    - 將string變數內的字串前面補0，直到string變數的長度等於width引數設定的長度

Homework 3

https://gist.github.com/chifu/f044779487741c829734#file-ex03_hw-py

- 題目: 在特定的文章字串中，搜尋輸入的字串
  - 條件1: 若有符合的字串，將其索引值印出 (全部印出， 並非印出第一個符合的索引值)
  - 條件2: 最後印出 總共有9個“的”, (若輸入的字為“的”)

- 格式: py254_中文姓名_hw3.py
- 上傳至:
  - https://goo.gl/Uv6LVo

範例圖

```
>>>
請輸入要找的字:銷售
39
89
547
572
641
786
925
966
總共有8個"銷售"
>>>
```