

基礎程式設計 - Python

Week04

Po-Chieh Yu
pcyu@saturn.yzu.edu.tw

複習題

- 請撰寫一個程式，讓使用者輸入一個角度並計算其sin值且輸出。(提示: 利用數學函數sin()，sin()裡需為弧度，角度轉成弧度公式為 $\text{radians} = \text{degrees} * \text{PI} / 180$)

複習題

- 請撰寫一個程式，讓使用者輸入一個角度並計算其sin值且輸出。
(提示: 利用數學函數sin()，sin()裡需為弧度，角度轉成弧度公式為 $\text{radians} = \text{degrees} * \text{PI} / 180$)
- `import math`
- `input_deg = input("Please enter in degree:")`
- `input_deg = float(input_deg)` #先把輸入的內容轉成浮點數
- `input_radian = input_deg*(math.pi)/180` #把input_deg轉成弧度
- `output_sin = math.sin(input_radian)` #再使用sin函式
- `print(output_sin)`

字串函式

- my_string="The formation and evolution of supermassive black holes"
- **len()** : 取得字串的長度
- string_length = **len**(my_string)
- print(string_length)
- **split()**: 分割字串 (字串專用函式)
- 練習空白字元分割
- split_string = my_string.**split**(' ')
- print(split_string)

- **join(): 結合字串 (字串專用函式)**
- 練習將分割後的字串用 '-' 結合
- join_string = '-'.**join**(split_string)
- print(join_string)
- **replace(): 取代字串 (字串專用函式)**
- 練習用 '&' 取代 '-'
- replace_string = join_string.**replace**('-', '&')
- print(replace_string)

複習題

- 自行定義一個含有'&'符號的字串:

(1) 印出字串的長度

(2) 將字串以'&'分開

(3) 將(2)的結果字串以'-'結合

(4) 將(3)的結果用'@'取代'-'

大小寫轉換

- `my_string="the formation and evolution of supermassive black holes"`
- 把第一個字母改成大寫:
- `my_string = my_string.title()`
- `print(my_string)`
- 把所有字元改成大寫: `my_string.upper()`
- 把所有字元改成小寫: `my_string.lower()`
- 大小寫對調: `my_string.swapcase()`

Play with your string!

- 先得知字串的長度
- 傳回前6個字元
- 他的開頭是不是"The": `my_string.startswith()`
- 結尾是不是"The": `my_string.endswith()`
- 找出"and"第一次出現的位移值: `my_string.find()`
- 最後一次出現的位移值: `my_string.rfind()`
- "The"出現幾次? "the"出現幾次? `my_string.count()`

Play with your string!

- `my_string="The formation and evolution of supermassive black holes"`
- 先得知字串的長度: `len(my_string)`
- 傳回前6個字元: `my_string[0:6]`
- 他的開頭是不是"The": `my_string.startswith('The')`
- 結尾是不是"The": `my_string.endswith('The')`
- 找出"and"第一次出現的位移值: `my_string.find('and')`
- 最後一次出現的位移值: `my_string.rfind('and')`
- "The"出現幾次? "the"出現幾次? `my_string.count('the')`

練習題

- 將字串 `my_string='I like python!'` (1)都改成大寫並存到另一個變數輸出;(2)將新變數的字串以空白字元分割並輸出;(3)將分割的字串用'%'連接起來並輸出;(4)將連接後的字串，用'\$'取代 '%' 並輸出

練習題

- 將字串 `my_string='I like python!'` (1)都改成大寫並存到另一個變數輸出;(2)將新變數的字串以空白字元分割並輸出; (3)將分割的字串用'%'連接起來並輸出; (4)將連接後的字串，用'\$'取代 '%' 並輸出
- `u_string = my_string.upper()`
- `print(u_string)`
- `u_string = u_string.split(' ')`
- `print(u_string)`
- `u_string = '%'.join(u_string)`
- `print(u_string)`
- `u_string = u_string.replace('%','$')`
- `print(u_string)`

Python的資料儲存

- 串列 (list)
- 字組 (tuple)
- 字典 (dict)
- 集合 (set)

串列

- 串列 (list) 是一串 python的資料
- 串列裡的每個項目稱為元素
- 可以有不同的資料形態
- `my_num=[1,2,3,4]`
- `my_week=['Mon','Wed','Sun']`
- `all_list=[20,'Sat',2.1134]`
- 空串列:`empty_list=[]` 或 `empty_list=list()`

串列

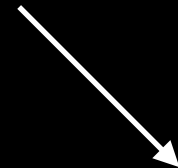
- `a=list('astronomy')`
- `print(a)`
- Review: 分割字串形成一個串列
- `today='2018/Oct/1'`
- `s_today=today.split('/')`
- `print(s_today)`

串列

- `my_num=[1,2,3,4]`



`my_num[0]`



`my_num[3]`

- 串列長度: `len(my_num)`

巢狀串列

- 串列裡的元素，也可以是另一個串列
- `my_nest=[23,'test',3.4,[1,2,3]]`
- `my_nest[3]=?`
- 更改內容 `my_nest[0]=11`
- 串列長度是最上層的元素數量
- `len(my_nest)=?`

串列的存取

- 利用索引值: `my_nest[0]`
- 巢狀串列也相同: `my_nest=[23,'test',3.4,[1,2,3]]`



`my_nest[3][0]`

如何得知某個項目的索引值?

`index()`

- 巢狀串列也相同: `my_nest=[23,'test',3.4,[1,2,3]]`
- `my_nest.index('test')`

如何得知某個項目的索引值？

index()

- 將以下成績60,59,100,12,30,存到串列score, 利用index()找出59分的索引值，並且將該成績改成60。
- `score = [60,59,100,12,30]` #定義串列score內容
- `tmp = score.index(59)` #找出59的索引值並且存到tmp
- `score[tmp]=60` #利用tmp的索引值改掉該元素的內容
- `print(score)`

判斷是否是元素成員

- 利用布林運算子 in
- `a=[0,1,2,3,4]`
- `print(1 in a) =?`
- `print(5 in a) =?`

串聯與重複

- 利用運算子+或是extend()來結合串列
- `a=['Mon','Tue','Wed']`
- `b=[1,2,3,4]`
- `c=a+b`
- `print(c) = ?`
- `a.extend(b)`
- `print(a) = ?`

串聯與重複

- 利用運算子*來複製list
- `a=['Mon','Tue','Wed']`
- `a=a*4`
- `print[a] = ?`
- `my_nest=[23,'test',3.4,[1,2,3]]`
- `my_nest*2`

修改串列

- 利用索引值
- `my_week=['Tue','Thu','Fri']`
- `my_week[0]='Mon'` #修改第一個元素
- `my_week[-1]='Wed'` #修改最後一個元素

修改串列

- `my_list=['a','b','c','d','e','f','g','h','i']`
- `my_list[1:4]=['x','y','z']` #修改`my_list[1]`到`[3]`的值
- `print(my_list) = ?`

- 複習:
- 請自行定義一個串列，含有文字與數字:
- (1) 請求出此串列的長度; (2) 判斷字元 'b' 是否在串列中; (3) 將第三個元素改成 'OK!'; (4) 將串列 [2,3,4,5,6] 加到你的串列中; (5) 讓使用者可以輸入一個元素，你幫他加入串列中。

- `mylist=[1,3,5,7,'a','b']`
- `print('b' in mylist)`
- `mylist[2]='OK!'`
- `tmp=[2,3,4,5,6]`
- `mylist.extend(tmp)`
- `new=input("Please enter your element:")`
- `mylist.extend(new)`
- `new=[input("Please enter your element:")] #有什麼不一樣?`
- `mylist.extend(new)`