



程式設計概論 Programming 101

一程式變數數值及字串的進階操作

授課老師：邱淑怡

Date: 3/18/2021

大綱

- 數值處理函式
- 數學模組
- 字串轉換及搜尋子字串
- 序列型別(sequence type)
 - list
- 亂數函式

數值處理函式_內建函式(function)

- `abs(x)`: 傳回數值參數`x`的絕對值
- `min(x1,x2 [, x3...])`: 傳回參數中的最小值
 - `min(5,1)`
 - `min(-1,3,-5,7,10)`
- `max(x1,x2 [, x3...])`: 傳回參數中的最大值
- `int(x)`: 傳回數值參數`x`的整數部分，小數部分直接捨去
- `round(x[,precision])`: 傳回與數值參數`x`最接近的整數(即四捨五入)，若要設定精確度為小數幾位，可加上選擇性參數`precision`
 - `round(3.6854,2)`
- `pow(x,y)`: 傳回數值參數`x`的參數`y`次方值

數值處理函式_數學模組

標準函式庫(standard library) 或是內建函式庫(build-in library)

- 模組是一個Python檔案，如: math模組，它有一些數學常數和數學函式，需要import
- **import math** (要執行以下函式需先下達該敘述)
- math.pi, math.e(自然對數的底數e)
- math.factorial(x): 傳回參數x的階乘(5!)
- math.gcd(x,y): 傳回參數x, y的最大公因數
- math.exp(x): 傳回自然對數之底數e的數值參數x次方值
- math.log(x[,base]): 傳回數值參數x的自然對數值，預設的底數是e，base可以設定底數，ex: math.log(2,2)
- math.sqrt(x): 傳回數值參數x的平方根

基本題

1. 從0, 98, 29, -56, 38, 27 等數字中找出最大值
2. 從0, 98, 29, -56, 38, 27 等數字中找出最小值
3. 用`math.pi`定義的圓周率求半徑為10的圓面積
4. $\sqrt{7}$ 的值
5. 616與1331的最大公因數

進階題

1. 已知 $x=\log 2$, $y=\log 3$ ，請撰寫Python程式令它計算 $10^{2x+3y+1}$ 並印出結果，將答案四捨五入取到整數

基本題_參考程式

1. 從0, 98 ,29, -56, 38, 27 等數字中找出最大值
2. 從0, 98 ,29, -56, 38, 27 等數字中找出最小值
3. 用math.pi定義的圓周率求半徑為10的圓面積
4. $\sqrt{7}$ 的值
5. 616與1331的最大公因數



```
print(min(0, 98 ,29, -56, 38, 27 ))  
print(max(0, 98 ,29, -56, 38, 27 ))  
import math  
print(math.pi*10*10)  
print(math.sqrt(7))  
print(math.gcd(616,1331))
```

指派運算子

運算子	語法	說明
=	a=b	將b指派給a, a的值設定為b的值
+=	a += b	a = a + b
-=	a -= b	a = a - b
*=	a *= b	a = a * b
/=	a /= b	a = a / b
//=	a //= b	a = a // b
%=	a %= b	a = a % b
**=	a **= b	a = a ** b

指派運算子_練習題

A=10

A += 2

print(A)

A -= 3

print(A)

A *= 2

print(A)

A /= 3

print(A)

A //= 2

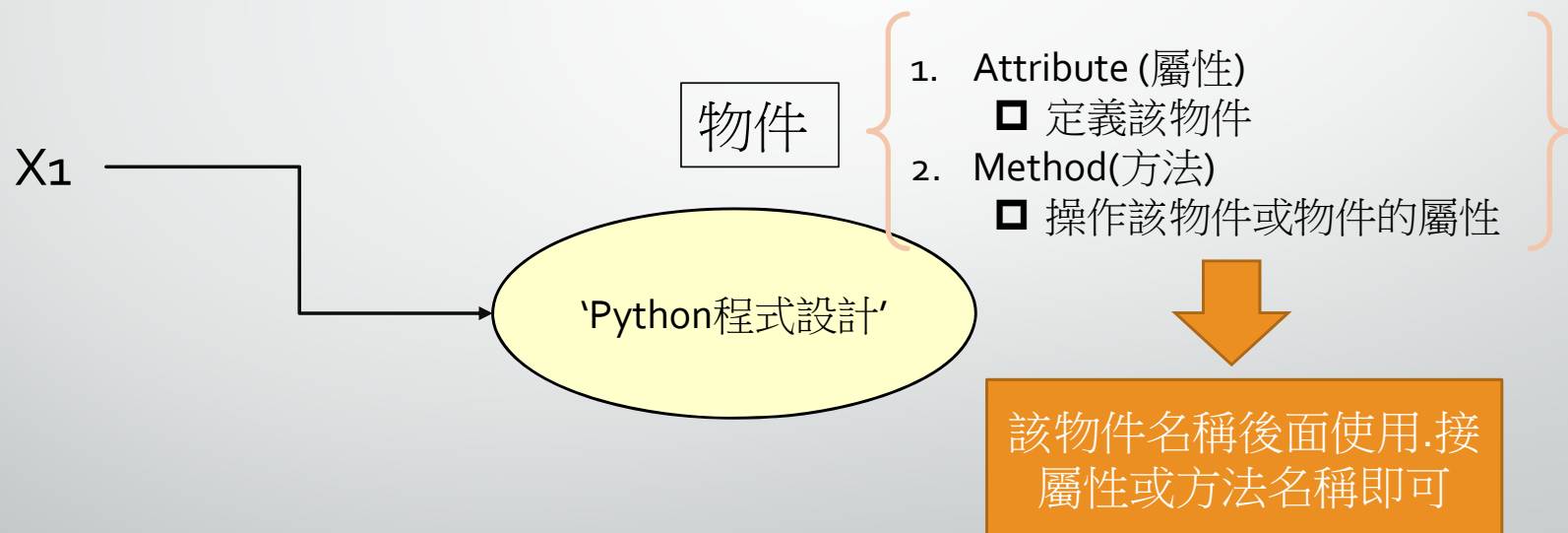
print(A)

A %= 2

print(A)

字串的其他操作 – 字串物件的操作(方法 method)

X1='Python程式設計'



字串轉換方法

- `x1='Python程式設計'`
- `print(x1.upper())`: 傳回字串`x1`的所有字元轉換成大寫的字串
- `print(x1.lower())`: 傳回字串`x1`的所有字元轉換成小寫的字串
- `print(x1.swapcase())`: 傳回字串`x1`大小寫互換的字串
- `print(x1.replace(old,new))`: 傳回字串`x1`的`old`字串取代為`new`的字串
- `print(x1.capitalize())`: 傳回字串`x1`的第一個字元轉換成大寫的字元
- `print(x1.title())`: 傳回字串`x1`的每個單字第一個字元轉換成大寫的字元

若 `old` 沒出現在 `x1` ?

針對特定子字串進行操作

- `find()` method: 找出特定子字串，找出字串裡面尋找子字串第一次出現的位置，傳回的結果是第一次出現的索引值，若沒找到則傳回-1
- `rfind()` method: r是reverse，從字串尾段開始搜尋，並傳回第一次出現該子字串出現的索引值
- `W1="me myself and I"`
- `W1.find('and')`
- `W1.find("you")`
- `W1.find("e")`
- `W1.rfind("e")`

字串中搜尋子字串的方法

```
var1='WowWowWowWowWow'  
print(var1.count('Wow')) #var1字串出現'Wow'的次數  
print(var1.startswith("Wow")) #var1字串是否以'Wow'開頭  
print(var1.endswith('Wow')) #var1字串是否以'Wow'結尾  
print(var1.find('Wow')) #var1字串中出現'Wow'的最小索引  
print(var1.rfind('Wow')) #var1字串中出現'Wow'的最大索引  
print(var1.startswith('Wow', 0, len(var1)-5 ))  
print(var1.endswith('Wow', 0, len(var1)-3 )) # 進階例子
```

字串內置方法(cont.)

- Python提供一系列的isxxx的功能，可判斷字串是否有xxx屬性
 - `var='python programming 123'`
 - `print(var.isdigit())` # 是否全為數字
 - `print(var.isalpha())` # 是否全為字母
 - `print(var.isalnum())` # 是否全為數字或字母
 - `print(var.islower())` # 是否全為小寫
 - `print(var.isupper())` # 是否全為大寫
 - `print(var.isspace())` # 是否全是空白，是傳回True
 - `print(var.istitle())` # 是否每個單字第一個字元都是大寫字元，是傳回True

字串練習

- 基本題: `s1='Today is Friday'`
 1. `s1`是否包含'day'?
 2. 'day'出現在`s1`的次數
 3. 'day'出現在`s1`的最小索引
 4. 利用`s1`建立一個新字串變數`new1`，令`new1`的值為"Today is Saturday"
 5. 求`s1`的第2-8的字元
 6. `s1`的每個單字的第一個字元都是大寫嗎?

Python 內建函數: print() 進階語法_例子

- 放多個字串
- 字串格式 (string format)
 - 字串裡面「挖洞」放入你想要的變數
 - 使用 {:型別} 來完成
 - 放入字串變數 → {:s}
 - 放入整數 → {:d}
 - 放入浮點數 → {:f}
 -

範例一

```
print('girl', 'boy', sep='/') # result: girl/boy
```

範例二

```
print('Hi, {:s}. You have {:d} mails.'.format('Bill', 9))
```

||

```
a='Bill'  
b=9  
print("Hi " + a + " you have " + str(b) + " mails")
```

result

Hi, Bill. You have 9 mails.

序列型別(sequence type)

- list (串列)

- 串列(list)是由一群資料排在一起形成的，其定義與使用方式非常簡單
 - 定義時必須使用中括號([])，並在內容之間以逗點隔開
 - 使用時則是依照內容順序，由 0 開始數
- `a=[9, 1, 3, 5]`
- `print(a[0])`
- `print(a[1])`
- `print(a[-1])`
- `a[2]=10`
- `del a[1]`
- `print(a)`

list (cont.)

- `a = [12, 43, 56, 87, 9, 132, 456, 879, 1321, 4678, 13445]`
- `print(132 in a)`
- `print(678 not in a)`
- `print(a[4:7])` # `a[4]` 開始取到 `a[7-1]`，或者說 `a[4]` 開始取 7-4 個
- `print(a[3:8:2])` # `a[3]` 開始取到 `a[8-1]`，每 2 個取 1 個
- `print(a[-1])` # 倒數第一個
- `print(len(a))` # list 長度
- # 如果內容都是數字，可以使用以下的計算
- `print(min(a))`
- `print(max(a))`
- `print(sum(a))`
- `print(a[4:11])`

list練習題

- $A=[1,3,6,-13,5,67,-8]$
 - 求list A 最大值和最小值
 - 取出list A 中第五個數值
 - 在list A 刪除第三個數值
 - 在list A 第四個數值變成 100

亂數函式

- Python提供一些函式可以用來產生亂數
- `import random`
- `random.randint(x,y)`: 傳回數值x,y之間的數值(包含x,y)
- `random.random()`:傳回數值0.0, 1.0之間的浮點數
- `x = [1,2,6,4,8, 13]`
- `random.shuffle(x)` #將變數x中的元素隨機重排後再把結果給變數x
- `print(x)`
- `print(random.choice(x))` #將變數x中的元素隨機選出一個

最後，複習一下

- Python索引值從**0**開始
 - string slicing
 - list
- 資料型態:
 - **整數與浮點數**的處理
 - **Boolean**的判斷
 - **字串**的處理
 - **list** 基本處理

高

低

優先順序表

運算子	說明
(...), [...], {...}	list, set, tuple, dict
a[i], a[i:j]	索引
a**b	指數運算
+a, -a	正,負號
a*b, a/b, a//b, a%b	乘法, 除法, 餘數
a+b, a-b	加法, 減法
<, >, <=, >=, ==, !=	比較運算
Not a	邏輯not運算
a and b	邏輯 and運算
a or b	邏輯 or運算