Python
Numerical and String Processing

馬誠佑

2025/03/14

內建函數

函式	描述	範例
abs(x)	回傳 x 的絕對值	abs(-2) 是 2
max(x1, x2, ...)	回傳 x1, x2, ...的最大值	max(1, 5, 2) 是 5
min(x1, x2, ...)	回傳 x1, x2, ...的最小值	min(1, 5, 2) 是 1
pow(a, b)	回傳 a^b	pow(2, 3) 是 8
round(x)	回傳最接近 x 的整數。若與兩整數接近，則回傳偶數的整數。 四捨六入五成雙	round(5.4) 是 5 round(5.5) 是 6 round(4.5) 是 4
round(x, n)	回傳捨位到小數點後 n 位的浮點數。	round(5.466, 2) 是 5.47 round(5.463, 2) 是 5.46

import math

函式	描述	範例
fabs(x)	以浮點數回傳 x 的絕對值	fabs(-2) 是 2.0
ceil(x)	回傳大於 x 的最小整數。	ceil(2.1) 是 3 ceil(-2.1) 是 -2
floor(x)	回傳小於 x 的最大整數。	floor(2.1) 是 2 floor(-2.1) 是 -3
exp(x)	回傳 e^x	exp(1) 是 2.71828
log(x)	回傳 $\log_e(x)$	log(2.71828) 是 1.0
log(x,base)	回傳以指定基底的對數。	log(100, 10) 是 2.0
sqrt(x)	回傳 $\sqrt{x}$	sqrt(4.0) 是 2
sin(x)	回傳以弧度為單位的 sine 三角函式	sin(3.14159 / 2) 是 1 sin(3.14159) 是 0
asin(x)	回傳以弧度為單位的反 sine 三角函式	asin(1.0) 是 1.57 asin(0.5) 是 0.523599
cos(x)	回傳以弧度為單位的 cosine 三角函式	cos(3.14159 / 2) 是 0 cos(3.14159) 是 -1
acos(x)	回傳以弧度為單位的反 cosine 三角函式	acos(1.0) 是 0 acos(0.5) 是 1.0472
tan(x)	回傳以弧度為單位的 tangent 三角函式	tan(3.14159 / 4) 是 1 tan(0.0) 是 0
degrees(x)	將 x 角度從弧度(radian)轉換為度數(degree)	degrees(1.57) 是 90
radians(x)	將 x 角度從度數轉換為弧度	radians(90) 是 1.57

String (字串)

- 用引號建立 “ ” or ‘ ’ or ‘’’ ‘’’ or “”” “””
- print(‘ “I love python!” ’)
- print(“ ’I love python!!’ ”)
- s = ''' I love python
  and I don't like java.

C++ is the best programing language.
’’’
- “”, ‘’, ’’’’’’, “”””””” (空字串)

Escape sequence

- S = '\102\157\156'

print(S)

=>Bon

- S = '\x42\x6F\6E'

print(S)

=>Bon

No.	Code	Name	Escape Characters
1	\'	單引號 (Single Quote)	'
2	\\	反斜線 (Backslash)	\
3	\n	換行 (New Line)	換行
4	\t	空格 (Tab)	空格
5	\b	退格鍵 (Backspace)	去掉空格
6	\ooo	八進制值 (Octal value)	
7	\xhh	十六進制值 (Hex value)	
8	\f	換頁 (Form Feed)	

Raw string

- rawString = r‘Type a \n to get a new line in a normal string’
  print(rawString) => Type a \n to get a new line in a normal string

- c = '''123
  456
  789'’’

+ (concate)

- ‘Release the kraken! ’ + ‘No, wait!’
- ‘Release the kraken! ’ ‘No, wait!’
- Vowels =(‘a’”e”’’i”’’o”””u””)
- print(‘Duck’,’Duck’, ’Grey Duck!’)

* (duplicate)

```
start = ‘Na ’*4+’\n’
middle = ‘Hey ’*3 + ‘\n’
end = ‘Goodbye’
print(start+middle+end)
```

[] (get character)

letters = ‘abcdefghijklmnopqrstuvwxyz’

>>>letters[0]

‘a’

>>>letters[1]

‘b’

>>>letters[-1]

‘z’

>>>letters[-2]

‘y’

>>>letters[25]

‘z’

>>>letters[5]

‘f’

>>>letter[5] = ‘k’

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: 'str' object does not support item assignment

>>>name = ‘Henny’

>>>name.replace(‘H’,’P’)

‘Penny’

>>>‘P’+name[1:]

Slice

- [:] 取整個序列
- [start:] 從start一直到最後結束
- [:end] 從頭到end-1
- [start:end]從start到end-1
- [start:end:step]從start到end-1並每次跳step個

>>>letters = ‘abcdefghijklmnopqrstuvwxyz’
>>>letters[18:-3]
‘stuvw’
>>>letters[-6:-2]
‘uvwx’
>>>letters[4:20:3]
‘ehknqt’
>>>letters[::-1]
>>>letters[-50:]
‘abcdefghijklmnopqrstuvwxyz’
>>> letters[-51:-50]
‘’

len() (取長度)

```
>>>len(letters)
26
>>>empty = ‘’
>>>len(empyt)
0
```

split()

```
>>>tasks = ‘get gloves,get masks,give cat vitamins,call ambulance’
>>>tasks.split(‘,’)
[‘get gloves’, ‘get masks’, ‘give cat vitamins’, ‘call ambulance’]
>>>tasks.split()
[‘get’, ’gloves,get’, ’masks,give’, ’cat’, ‘vitamins,call’, ‘ambulance’]
```

join() <-> split()

```
>>>crypto_list = [‘Yeti’, ‘Bigfoot’, ‘Loch Ness Monster’]
>>>crypto_str = ‘, ’.join(crypto_list)
print(‘Found and signing book deals: ’ + crypto_str)
Found and signing book deals: Yeti, Bigfoot, Loch Ness Monster
```

replace()

>>>setup = ‘a duck goes into a bar…’

>>>setup.replace(‘duck’, ‘marmoset’)

‘a marmoset goes into a bar…’

>>>setup

‘a duck goes into a bar…’

>>> setup.replace(‘a ’, ‘a famous ’, 100)

‘a famous duck goes into a famous bar…’

>>>setup.replace(‘a’, ‘a famous’, 100)

’a famous duck goes into a famous ba famous…’

strip()

- 移除字串開頭或結尾的填補字元（’ ‘、’\t’、’\n’）

```
>>>world = ‘      earth          ’
>>>world.strip()
‘earth’
>>>world.strip(‘ ’)
‘earth’
>>>world.lstrip()
’earth          ’
>>>world.rstrip()
‘       earth’
>>>world.strip(‘!’)
‘      earth          ’
```

```
>>>blurt = ‘what the…!!?’
>>>blurt.strip(‘.?!’)
‘what the’
```

Search and selection

```
>>>poem = '''All that doth flow we cannot liquid name
Or else would fire and water be the same;
But that is liquid which is moist and wet
Fire that property can never get. Then 'tis not cold that doth the fire put out But 'tis the wet that makes it die, no doubt'''
>>>poem.startwith('All')
True
>>>poem.endwith('That\' s all, folks!')
False
>>>word = 'the'
>>>poem.find(word)
73
>>>poem.index(word)
73
>>>poem.rfind(word) #'the'最後一次出現時的offset
214
>>>poem.rindex(word)
214
```

Search and selection-2

#如果字串不存在

```
>>>poem.find(word)
-1
>>>poem.rfind(word)
-1
>>>poem.index(word) #poem.rindex(word)
ValueError:…
>>>poem.count(word) #word出現幾次
3
>>>poem.isalnum() #是否只有字母與數字字元？
False
```

大小寫

```
>>>setup = ‘a duck goes into a bar…’
>>>setup.capitalize() #將第一個單字的第一個字母改為大寫
‘A duck goes into a bar…’
>>>setup.title() #將每個單字的第一個字母改為大寫
‘A Duck Goes Into A Bar’
>>>setup.upper() #將所有字元改為大寫
>>>setup.lower() #將所有字元改為小寫
>>>setup.swapcase() #將大小寫對調
```

對齊方式

```
>>>setup.center(30) #在30個字元長度內置中字串
‘   a duck goes into a bar…   ’
>>>setup.ljust(30) #在30個字元長度靠左對齊字串
‘a duck goes into a bar…      ’
>>>setup.rjust(30) #在30個字元長度靠右對齊字串
‘     a duck goes into a bar…’
```

格式化

- 舊式 (python 2 & 3提供)
- 新式 (python 2.6以上)
- f-string (python 3.6以上)

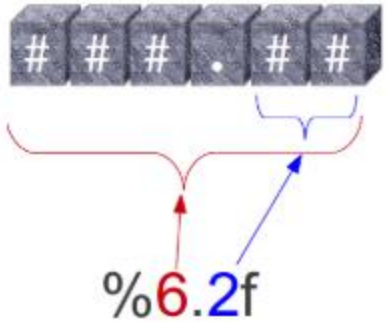
舊式(%)

```
print("Art: %5d,  Price per Unit: %8.2f" % (453, 59.058))
```

output

String Modulo Operator

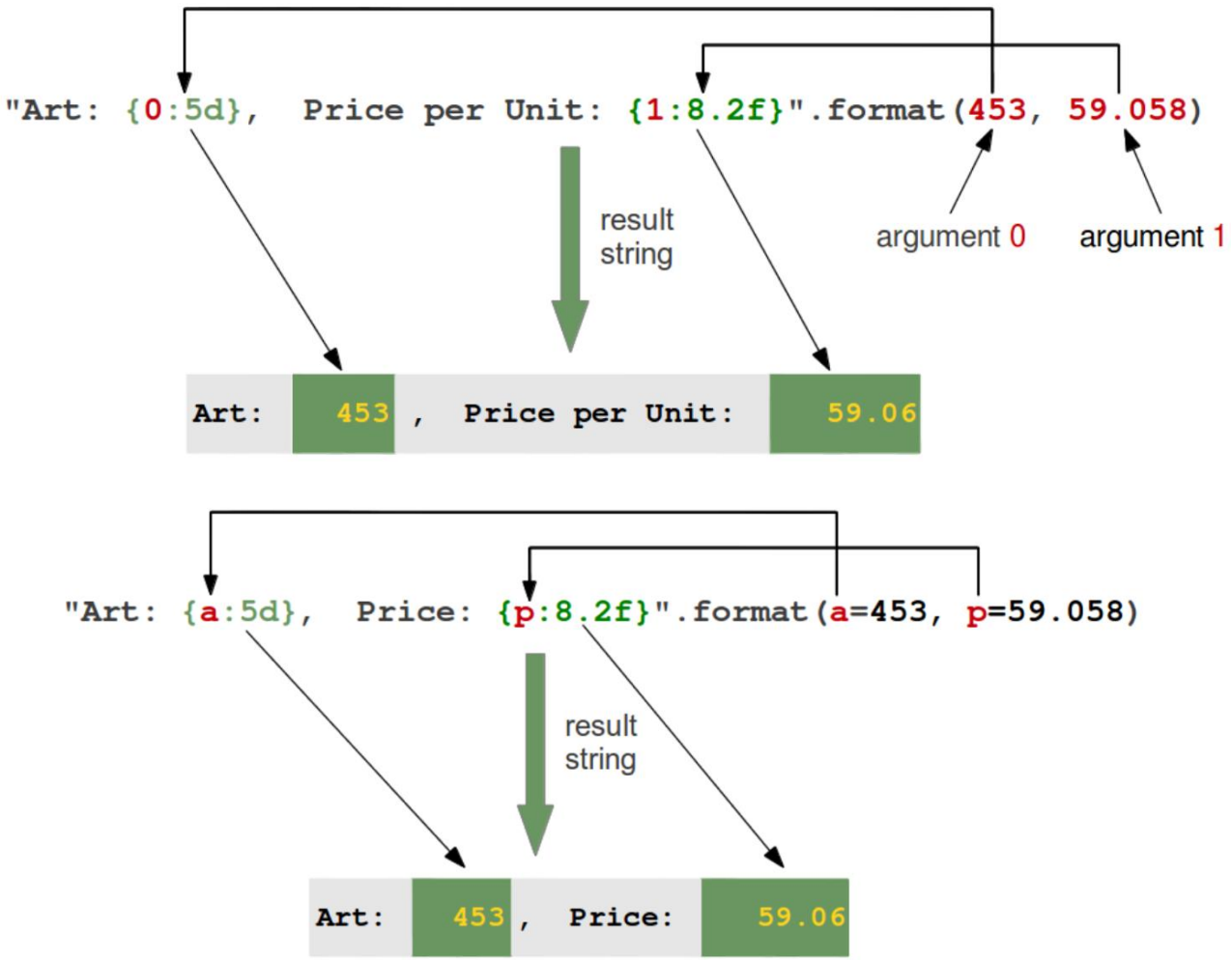
```
Art:   453,  Price per Unit:    59.06
```



Conversion	Meaning
d	Signed integer decimal.
i	Signed integer decimal.
o	Unsigned octal.
u	Obsolete and equivalent to 'd', i.e. signed integer decimal.
x	Unsigned hexadecimal (lowercase).
X	Unsigned hexadecimal (uppercase).
e	Floating point exponential format (lowercase).
E	Floating point exponential format (uppercase).
f	Floating point decimal format.
F	Floating point decimal format.
g	Same as "e" if exponent is greater than -4 or less than precision, "f" otherwise.
G	Same as "E" if exponent is greater than -4 or less than precision, "F" otherwise.
c	Single character (accepts integer or single character string).
r	String (converts any python object using repr()).
s	String (converts any python object using str()).
%	No argument is converted, results in a "%" character in the result.

- 初始的 '%' 字元。
- 選用的對齊字元：不加入任何東西，或 '+' 代表靠右對齊，'-' 代表靠左對齊。
- 選用的最小寬度（*minwidth*）欄位。
- 選用的 '.' 字元，用來隔開最小寬度（*minwidth*）與最大字元（*maxchars*）。
- 選用的最大字元（如果轉換型態是 s），代表從資料值印出多少字元。如果轉換型態是 f，它代表精確度（印出小數點後多少位）。
- 上一個表格的轉換型態字元。

新式與format()



- 有個開始的冒號（':'）。
- 有個選用的填補（*fill*）字元（預設為' '），在字串比最小寬度（*minwidth*）短時填補它的值。
- 有個選用的對齊字元。這次預設的選項是左對齊。'<' 也代表左，'>' 代表右，'^' 代表中間。
- 有個選用的符號（*sign*）供數字使用。在前面加上負號（'-'）代表負數，' ' 代表在負數前面加上負號，空格（' '）代表正數。

- 選用的最小寬度（*minwidth*）。用選用的句點（'.'）來隔開最小寬度（*minwidth*）與最大字元（*maxchars*）。
- 選用的最大字元（*maxchars*）。
- 轉換型態。

```
>>> thing = 'wraith'
>>> place = 'window'
>>> 'The {} is at the {}'.format(thing, place)
'The wraith is at the window'
>>> 'The {:10s} is at the {:10s}'.format(thing, place)
'The wraith     is at the window    '
>>> 'The {:<10s} is at the {:<10s}'.format(thing, place)
'The wraith     is at the window    '
>>> 'The {:^10s} is at the {:^10s}'.format(thing, place)
'The   wraith   is at the   window  '
>>> 'The {:>10s} is at the {:>10s}'.format(thing, place)
'The     wraith is at the     window'
>>> 'The {:!^10s} is at the {:!^10s}'.format(thing, place)
'The !!wraith!! is at the !!window!!'
```

```
>>> d = {'thing': 'duck', 'place': 'bathtub'}
```

在下面的範例中，{0} 是 format() 的第一個引數（字典 d）：

```
>>> 'The {0[thing]} is in the {0[place]}.'.format(d)
'The duck is in the bathtub.'
```

f-string

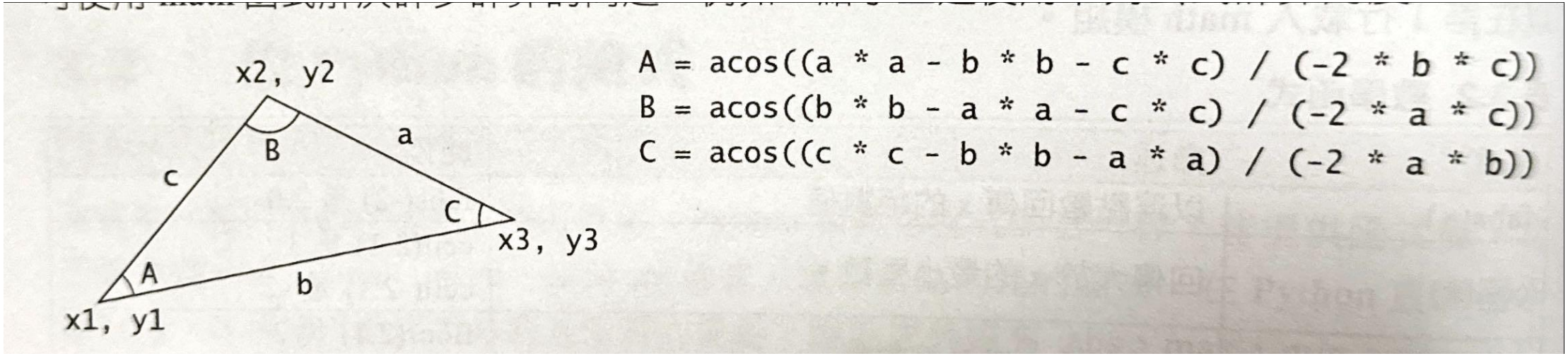
```
>>>thing = ‘wereduck’
>>>place = ‘werepond’
>>>f’The {thing} is in the {place}’
‘The wereduck is in the werepond’
>>>f’The {thing.capitalize()} is in the {place.rjust(20)}’
‘The Wereduck is in the                werepond’
>>>f’The {thing:>20} is in the {place:.^20}’
The                wereduck is in the ……werepond……
```

f-string (python 3.8之後新增)

```
>>>f’{thing=}, {place=}’ #可印出變數名稱與它們的值
thing = ‘wereduck’, place = ‘werepond’
>>>f’{thing[-4:]=}, {place.title()=}’ #可印出變數名稱與它們的值
Thing[-4:] = ‘duck’, place.title() = ‘Werepond’
>>>f’{thing= :>4.4}’
thing = ‘were’
```

Practice

1.



提示使用者輸入在三角形中三個角的 x 與y座標然後顯示三角形的三個角度
x1, y1, x2, y2, x3,y3 = eval(input('Enter three points: '))

輸出 "The three angles are xx.xx xx.xx xx.xx" (round取到小數點後兩位)

2. (幾何：大圓距離) 所謂的大圓距離即為球面上兩點間的距離。假設(x1, y1)與(x2, y2)為兩點地理上的緯度(latitude)與經度(longitude)的座標。兩點間的大圓距離可藉由以下公式計算之：

$$d = radius \times \text{acos}(\sin(x_1) \times \sin(x_2) + \cos(x_1) \times \cos(x_2) \times \cos(y_1 - y_2))$$

請撰寫一程式，提示使用者輸入以度數表示地球上兩點的緯度與經度，接著顯示其大圓距離。地球平均半徑為 6,371.01 km。公式裡的緯度與經度代表北與西。

3. 用舊式格式化來寫出下面的詩。將字串 'roast beef'、'ham'、'head' 與 'clam' 代入這個字串：

```
My kitty cat likes %s,
My kitty cat likes %s,
My kitty cat fell on his %s
And now thinks he's a %s.
```

4. 使用新式格式化來寫一封公式化信件。將下列的字串存為 letter（下一個習題會用到）：

```
Dear {salutation} {name},

Thank you for your letter. We are sorry that our {product}
{verbed} in your {room}. Please note that it should never
be used in a {room}, especially near any {animals}.

Send us your receipt and {amount} for shipping and handling.
We will send you another {product} that, in our tests,
is {percent}% less likely to have {verbed}.

Thank you for your support.

Sincerely,
{spokesman}
{job_title}
```

將值指派給名為 'salutation'、'name'、'product'、'verbed'（過去式動詞）、'room'、'animals'、'percent'、'spokesman' 與 'job_title' 的變數字串。用 letter.format() 印出使用這些值的信件。

5.

將 m 開頭的單字改為首字大寫：

```
>>> song = """When an eel grabs your arm,
... And it causes great harm,
... That's - a moray!"""
```