Python
基礎語法

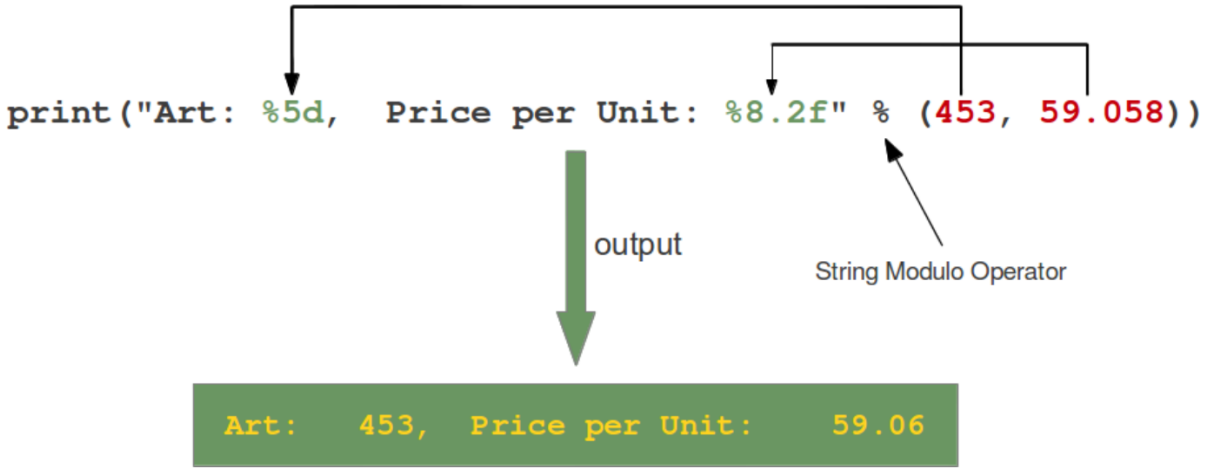
Cheng-Yu Ma

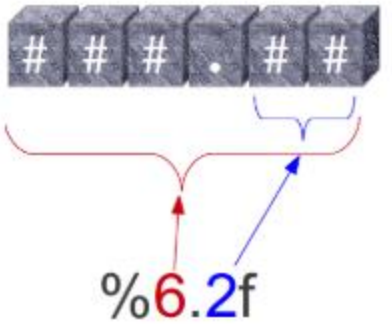
2025/3/7

基礎語法

- Print: format, end = ‘’
- If…else: if…elif….else
- Loop: for, while, continue, break
- Function: scope, pass by reference

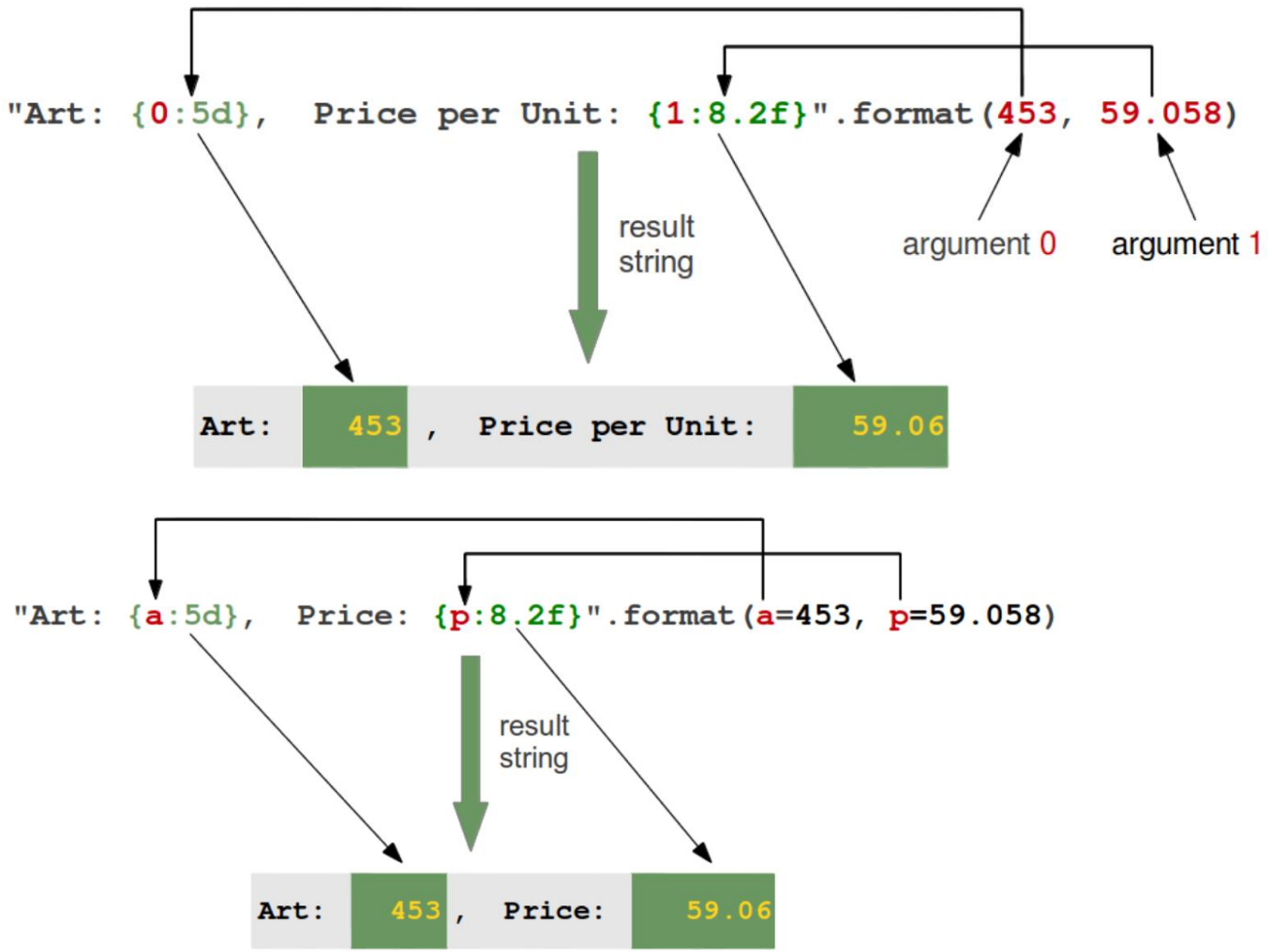
Print Format– type 1





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.

Print Format– type 2



If…else

- If…
- If… else…
- If … elif…
- If… elif….else

```
num = 5
if num == 3:            # 判断num的值
    print 'boss'
elif num == 2:
    print 'user'
elif num == 1:
    print 'worker'
elif num < 0:           # 值小于零时输出
    print 'error'
else:
    print 'roadman'     # 条件均不成立时输出
```

```
num = 9
if num >= 0 and num <= 10:    # 判断值是否在0~10之间
    print 'hello'
# 输出结果: hello
 num = 10
if num < 0 or num > 10:    # 判断值是否在小于0或大于10
    print 'hello'
else:
    print 'undefine'
# 输出结果: undefine
 num = 8
# 判断值是否在0~5或者10~15之间
if (num >= 0 and num <= 5) or (num >= 10 and num <= 15):
    print 'hello'
else:
    print 'undefine'
# 输出结果: undefine
```

Loop

- while 條件 :
- for i in range(0,10): …..
- A= [‘a’, ’b’, ‘c’, ‘d’, ‘e’]
- for a in A: …

List (array)

- a = [1, 2, 34, 5, 67]
- a[index]
- b = [
  [1,’b’,4,d,t,4],
  [2, g, ‘d’ ,e ,’h’, h],
  [d,g,hh,j,3]
  ]
- B[row][col]

```
print(len([1,2,3]))

a = [1,2,3]+[4,5,6]
print(a)

b = ['HI']*4
print(b)

c = [
    [1,2,3],
    [2,a,'ddd','h'],
    [7,8,9,10,11]
]
print(c)

print(c[2][1])
```

✓ 0.0s

```
3
[1, 2, 3, 4, 5, 6]
['HI', 'HI', 'HI', 'HI']
[[1, 2, 3], [2, [1, 2, 3, 4, 5, 6], 'ddd', 'h'], [7, 8, 9, 10, 11]]
8
```

```
print(a)
print(a[3])
print(a[-2])
print(a[1:3])
print(a[1:])
print(a[:3])
print(a[:])
print(a[::-1])
print(a[1:5:2])
```

✓ 0.0s

```
[1, 2, 3, 4, 5, 6]
4
5
[2, 3]
[2, 3, 4, 5, 6]
[1, 2, 3]
[1, 2, 3, 4, 5, 6]
[6, 5, 4, 3, 2, 1]
[2, 4]
```

enumerate

- a = [‘ha’, ’hi’, ‘hh’, ‘kk’, ‘dd’]

- for i, v in enumerate(a):

    print(‘index: ’ + i + ’ v: ’ + v)

Function

- def functionName(para1, para2,…):

  ….

  ….

  return a, b, c

I/O

- Input()
- open(filename, ’mode’)
- write()

List function

Python包含以下方法:

序号	方法
1	list.append(obj) 在列表末尾添加新的对象
2	list.count(obj) 统计某个元素在列表中出现的次数
3	list.extend(seq) 在列表末尾一次性追加另一个序列中的多个值（用新列表扩展原来的列表）
4	list.index(obj) 从列表中找出某个值第一个匹配项的索引位置
5	list.insert(index, obj) 将对象插入列表
6	list.pop([index=-1]) 移除列表中的一个元素（默认最后一个元素），并且返回该元素的值
7	list.remove(obj) 移除列表中某个值的第一个匹配项
8	list.reverse() 反向列表中元素
9	list.sort(cmp=None, key=None, reverse=False) 对原列表进行排序

Dictionary

- {key1:value1, key2:value2, key3:value3, …}

- Value = d[‘key’]

Dictionary function

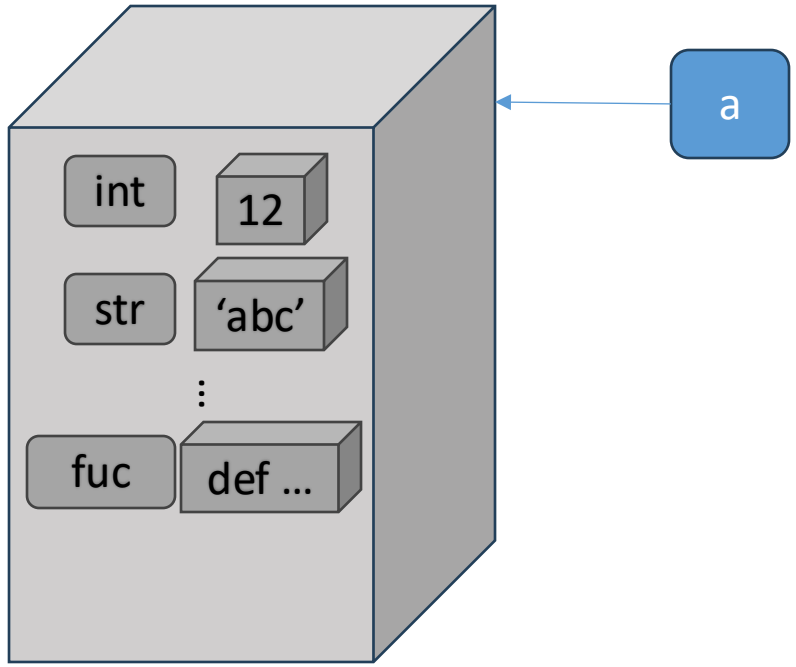
Python字典包含了以下内置方法：

序号	函数及描述
1	dict.clear() 删除字典内所有元素
2	dict.copy() 返回一个字典的浅复制
3	dict.fromkeys(seq[, val]) 创建一个新字典，以序列 seq 中元素做字典的键，val 为字典所有键对应的初始值
4	dict.get(key, default=None) 返回指定键的值，如果值不在字典中返回default值
5	dict.has_key(key) 如果键在字典dict里返回true，否则返回false
6	dict.items() 以列表返回可遍历的(键, 值) 元组数组
7	dict.keys() 以列表返回一个字典所有的键
8	dict.setdefault(key, default=None) 和get()类似, 但如果键不存在于字典中，将会添加键并将值设为default
9	dict.update(dict2) 把字典dict2的键/值对更新到dict里
10	dict.values() 以列表返回字典中的所有值
11	pop(key[,default]) 删除字典给定键 key 所对应的值，返回值为被删除的值。key值必须给出。 否则，返回default值。
12	popitem() 随机返回并删除字典中的一对键和值。

Data Types, Variables, and Operators

Python 的資料都是物件

- Object
  - 定義它可以做什麼的型態
  - Unique id, 用來區別它與其他物件
  - 與型態一致的值
  - 參考計數, 追蹤該物件被使用的次數



基本資料型態

名稱	型態	可變？	範例
布林	bool	否	True, False
整數	int	否	47, 25000, 25_000
浮點數	float	否	3.14, 2.7e5
複數	complex	否	3j, 5 + 9j
字串	str	否	'alas', "alack", '''a verse attack'''
串列	list	是	['Winken', 'Blinken', 'Nod']
tuple	tuple	否	(2, 4, 8)
bytes	bytes	否	b'ab\xff'
ByteArray	bytearray	是	bytearray(...)
集合	set	是	set([3, 5, 7])
不可變集合	frozenset	否	frozenset(['Elsa', 'Otto'])
字典	dict	是	{'game': 'bingo', 'dog': 'dingo', 'drummer': 'Ringo'}

Variable

- 命名規則：
  - 只能包含以下字元
    - 大小寫字母(a-zA-Z)
    - 數字(0-9)
    - 底線(_)
  - 區分大小寫
  - 開頭必須是字母或底線，不可用數字
  - 底線開頭的名稱會被特殊對待
  - 不可是python保留字（關鍵字）

保留字

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Ex: 有效

a
a1
a_B_C_11
_abc
_1a

Ex: 無效

1
1a
1_
Name!
another-name

賦值 (=)

Ex: y = x +13

Ex:

x = 5

y = x + 13

print(y)

Ex:

y = x + 13

x = 5

print(y)

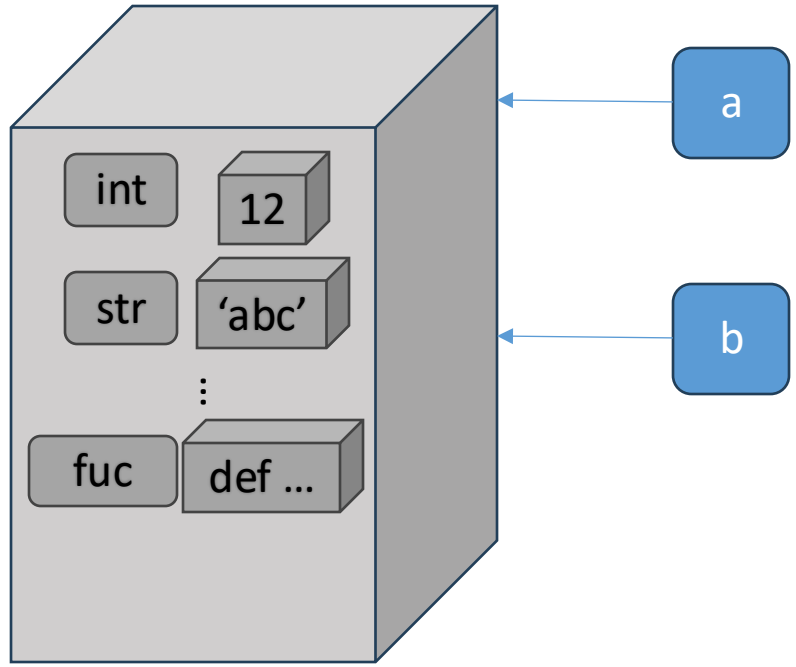
Ex:

a = 12

b = a

Ex:

c = d = e = 1



複製

```
x = 5
y = x
print(x)
print(y)
x = 29
print(x)  => 印出 29
print(y)  =>印出???
```

可變 v.s. 不可變

a = [2,4,6]

b = a

print(a) => [2,4,6]

print(b) => [2,4,6]

a[0] = 99

print(a) =>[99,4,6]

print(b) =>[99,4,6]

好的變數名稱？

- 在”簡潔”&”清楚”之間取得平衡
- 大寫駝峰 (UpperCaseCamel) for class
- 小寫駝峰 (lowerCaseCamel) for variable
- 蛇形命名 (snake_case) for function
- 大寫蛇形 (SNAKE_CASE) for constant

型態轉換

- int()
- float()
- str()
- bool()

Ex:

```
a = 1000
print(type(a))
print('我有'+ a +'元') => error…
print('我有'+ str(a) +'元')
```

Ex:

```
>>>int('99')
99
>>>int('1_000_000')
1000000
>>>int('10',2)# base 2
2
>>>int('10',8) #base 8
8
```

Ex:

```
>>>int('98.6')
Error…
>>>int('1.0e4')
Error…
```

bool (Boolean)

- 非零數字視為True

bool(True)

bool(1)

bool(45)

bool(-45)

- 零值數字視為False

bool(False)

bool(0)

bool(0.0)

int (Integer)

Ex:

5

05 => SyntextError: invalid token

123

+123

-123

1,000,000 => (1,0,0)

million = 1_000_000

print(million) => 1000000

Ex:

>>>4 + 7.0

11.0

>>>True + 2

3

>>>False+5.0

5.0

float(浮點數)

```
>>>5.
5.0
>>>05.0
5.0
>>>5e0
5.0
>>>5e1
50.0
```

```
>>>float('98.5')
98.5
>>>float('1.0e4')
10000.0
```

運算子1

運算子	說明	範例	結果
+	加法	a = 1+2	a = 3
-	減法	a = 20-12	a = 8
*	乘法	a = 2*3	a = 6
/	除法	a = 9/2	a = 4.5
//	除法取整數（無條件捨去）	a = 9//2	a = 4
%	餘數	a = 9%2	a = 1
**	次方	a = 2**3	a = 8

Ex:
>>>5/0
ZeroDivisionError: division by zero
>>>7//0
ZeroDivisionError: division by zero
>>>5%0
ZeroDivisionError: division by zero

>>> a = 96
>>> a
96
>>> a – 3
93
>>> a
96
>>> a = a – 3
>>> a
93

Ex:
>>> a = 95
>>> a -= 3
>>> a
92
>>> a += 8
>>> a
100
>>> a *= 2
>>> a
200
>>> a /= 3
>>> a
66.6666666667
>>> a = 13
>>> a //= 4
>>> a
3
>>>divmod(9,5)
(1,4)
>>>2**3
8
>>>2.0**3
8.0

運算子2

運算子	範例	等同於
+=	a += 1	a = a + 1
-=	a -= 1	a = a - 1
*=	a *= 2	a = a * 2
/=	a /= 2	a = a / 2
//=	a //= 3	a = a // 3
%=	a %= 3	a = a % 3
**=	a **= 2	a = a ** 2

比較運算子

運算子	說明	範例
>	大於 (a 是否大於 b)	a > b
<	小於 (a 是否小於 b)	a < b
>=	大於等於 (a 是否大於等於 b)	a >= b
<=	小於等於 (a 是否小於等於 b)	a <= b
==	等於 (a 是否等於 b)	a == b
!=	等於 (a 是否不等於 b)	a != b

邏輯運算子

a and b	a = True	a = False
b = True	**True**	**False**
b = False	**False**	**False**

a or b	a = True	a = False
b = True	**True**	**True**
b = False	**True**	**False**

	not a
a = True	**False**
a = False	**True**

Ex:

a = 1

b = 2

c = 3

```
print((a>b)&(c>b)) # False
print((a>b)|(c>b)) # True
print(not ((a>b)&(c>b))) # True ( 因為 (a>b)&(c>b) 為 True )
```

in 與 is 運算子

```
a = 2
b = 4
c = [1,2,3]
print(a in c) # True
print(b in c) # False
x = [1,2,3]
y = [1,2,3]
z = x
print(x is y) # False
print(x is z) # True
```

位元運算子

運算子	說明	範例	結果
&	位元 且，二進位數字「完全相同」的部分，不同的部分以 0 取代。	4&5，使用 0100 和 0101 比較後，回傳 0100	4
|	位元 或，二進位數字「只要有一個為 1 」的部分都為 1。	4&5，使用 0100 和 0101 比較後，回傳 0101	5
^	位元 互斥，二進位數字「完全相同」的部分都為 0，不同的部分以 1 取代。	4^5，使用 0100 和 0101 比較後，回傳 0001	1
~	位元 相反，二進位數字 0 變成 1，1 變成 0	~4，0100 相反為 1011	-5
>>	位元 右移，將二進位數字往右移動指定位數，左側補 0	4>>2，0100 往右移動兩位 0001	1
<<	位元 左移，將二進位數字往左移動指定位數，右側補 0	5<<2，0101 往左移動兩位 10100	20

```
print(4&5) # 4
print(4|5) # 5
print(4^5) # 1
print(~4) # -5
print(4>>2) # 1
print(5<<2) # 20
```

跨列運算子

```
a = ( 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 )

b = 1 + 2 + 3 + \
        4 + 5 + 6 + \
        7 + 8 + 9
```

！！！跨列運算子後不得加上空格或其它字元！！！

優先順序？ (...) > 乘除 > 加減

運算子	說明與範例
[v, …], {v1, …}, {k1:v1, …}, (…)	建立串列 / 集合 / 字典 / 產生器，或生成式、帶括號的運算式
seq[n], seq[n:m], func(args…), obj.attr	索引、切片、函式呼叫、屬性參考
**	次方
+n, -n, ~n	正數、負數、位元 not
*, /, //, %	乘法、浮點除法、整數除法、餘數
+, -	加法、減法
<<, >>	位元左移、右移
&	位元 and
|	位元 or
in, not in, is, is not, <, <=, >, >=, !=, ==	成員與相等測試
not *x*	布林（邏輯）not
and	布林 and
or	布林 or
if ... else	條件運算式
lambda ...	lambda 運算式

```
>>>-5**2
?
```

底數

- Base 2 : 0b or 0B
- Base 8 : 0o or 0O
- Base 16: 0x or 0X

Ex:

```
>>>10
10
>>>0b10
2
>>>0o10
8
>>>0x10
16
```

Ex:

```
>>>value = 65
>>>bin(value)
‘0b1000001’
>>>oct(value)
‘0o101’
>>>hex(value)
‘0x41’
```

chr() & ord() & type() & id()

```
>>>chr(65)
‘A’
>>>ord(‘A’)
65
```

Practice

1. 撰寫程式 ，顯示Welcome to Python、Welcome to Computer 、Welcome to AI、Programming is fun。
2. 請撰寫一程式，顯示如下的表單：

a	a^2	a^3
1	1	1
2	4	8
3	9	27
4	16	64

Practice

3. Compute Area With Console input:
   - Radius = eval(input('Please input the radius:'))
   - Pi = 3.14159
   - Output format: "The area for the circle of radius [radius] is [area]"
4. 美國人口普查局根據以下假設來推算人口：
   - 每7秒有一個小孩出生
   - 每13秒有一個人死亡
   - 每45秒有一個新移民入境

請撰寫一個程式，顯示接下來五年的人口數。假設目前人口為312,033,422，每年為365天。

Practice

5. 撰寫一個以格林威治標準時間(GMT)為基準所顯示的目前時間,

顯示的格式為，小時：分鐘：秒數，Ex:11:13:29

- 使用import time 下的 time.time()會回傳從GMT 1970年1月1號的00:00:00開始計算到目前的時間所經過的秒數ex: time.time()取得 1203183068.328 sec