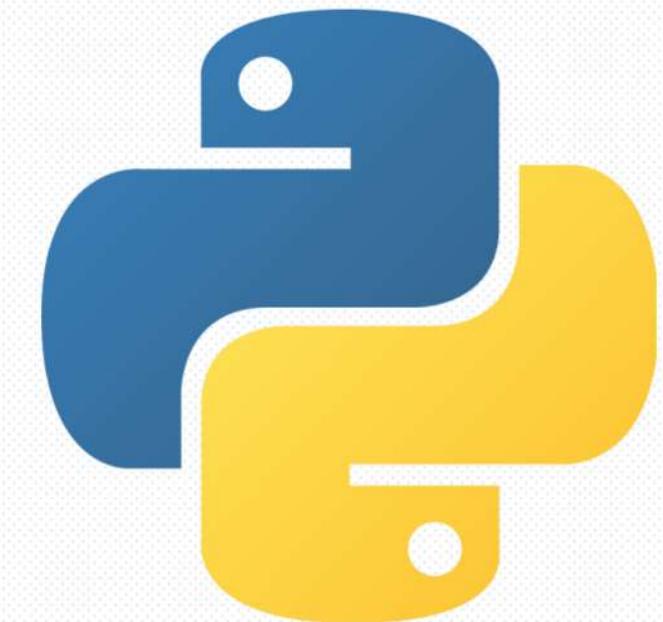


Unicode Table:
0Xffff
(0X→hex 16進位
16的4次方
=65536)

內建函式

Built-In Function



Python Fundamental



Outline

- ◆ 字串運算子
- ◆ 字串處理函式
- ◆ 數值處理函式
- ◆ 日期處理函式



Outline

- ◆ 字串運算子
- ◆ 字串處理函式
- ◆ 數值處理函式
- ◆ 日期處理函式



字串運算 - 連接運算子 & 重複運算子

- ◆ 連接運算子 + : 用來連接字串

```
In [1]: 'Happy' + 'Birthday' + 'To' + '小美'  
Out[1]: 'HappyBirthdayTo小美'
```

- ◆ 重複運算子 * : 用來重複字串 僅限正整數

```
In [1]: 3 * 'Yeah!'  
Out[1]: 'Yeah!Yeah!Yeah!'
```

```
In [2]: 'Yeah!' * 3  
Out[2]: 'Yeah!Yeah!Yeah!'
```



字串運算 - 比較運算子& in/not in 運算子

- ◆ 比較運算子：比較運算子 ($>$ 、 $<$ 、 \geq 、 \leq 、 $==$ 、 $!=$) 可以用來比較兩個字串的大小或相等與否。

```
In [1]: 'X'>'A'
```

```
Out[1]: True
```

根據Ascii table
(0-127)

```
In [2]: '123'>'456'
```

```
Out[2]: False
```

```
In [3]: 'XYZ'>'xyz'
```

```
Out[3]: False
```

- ◆ in 與 not in 運算子：in 與 not in 運算子可以確認元素是否在於字串中。

```
In [1]: 'or' in 'forever'
```

```
Out[1]: True
```

```
In [2]: 'for' not in 'forever'
```

```
Out[2]: False
```



字串運算 - 索引運算子

◆ 索引運算子：我們可以使用索引運算子（[]）取得字串中的字元。

◆ 假設變數 S 的值為 "Python程式設計"，其存放順序如下：

- 索引 0 表示從前端開始。
- 索引 -1 表示從尾端開始。
- S[0]、S[1]、...、S[9] 表示 "P"、"y"、...、"計"。
- S[-1]、S[-2]、...、S[-10] 表示 "計"、"設"、...、"P"。

索引	0	1	2	3	4	5	6	7	8	9
內容	P	y	t	h	o	n	程	式	設	計
索引	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1



字串運算 - 片段運算子

- ◆ 片段運算子：我們也可以使用片段運算子 (`[start:end]`) 指定索引範圍。

In [1]: `S="Python程式設計"`

In [2]: `print(S[2:5])`
tho

In [3]: `print(S[3:8])`
hon程式

In [4]: `print(S[5:-1])`
n程式設

索引 2 到索引 4 的字元(不含索引5)

索引 3 到索引 7 的字元(不含索引8)

索引 5 到索引 -2 的字元(不包含索引-1)

索引	0	1	2	3	4	5	6	7	8	9
內容	P	y	t	h	o	n	程	式	設	計
索引	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

- ◆ 若指定範圍時省略第一個索引，表示採取預設值為 0；若省略第二個索引，表示採取預設值為字串的長度。

In [5]: `print(S[::-2])`
Python程式

In [6]: `print(S[2:])`
thon程式設計

索引 0 到索引 -3 的字元(不包含索引-2)

索引 2 到最後的字元



Exercise

- ◆ 三個字串變數如下，請寫出下列問題的語法。

```
S1 = 'HappyNewYear'  
S2 = 'happynewyear'  
S3 = 'new'
```

- S1 和 S2 是否相等。
- S3 是否存在於 S1 。
- S1 的第 5~9 個字元。



Outline

- ◆ 字串運算子
- ◆ 字串處理函式
- ◆ 數值處理函式
- ◆ 日期處理函式



基本字串函式

- ◆ `ord()`

- ◆ `chr()`

- ◆ `len(s)`

- ◆ `max(s)`

- ◆ `min(s)`

- ◆ `str(n)`



基本字串函式

- ◆ `ord(x)`：傳回字元參數 x 的 Unicode 碼(十進位)。

```
ord('A')          # 傳回大寫英文字母 A 的 Unicode 碼  
65  
ord('€')          # 傳回歐元符號的 Unicode 碼  
8364
```

- ◆ `chr(x)`：傳回整數參數 x 代表的 Unicode 字元。

```
chr(88)          # 傳回 88 所代表的 Unicode 字元  
'X'  
chr(8364)        # 傳回 8364 所代表的 Unicode 字元  
'€'
```



基本字串函式

- ◆ `len(x)`：傳回字串參數 x 的長度，也就是字串由幾個字元所組成。

```
In [1]: len('Python程式設計')  
Out[1]: 10
```

- ◆ `min(x)`、`max(x)`：傳回字串參數 x 中 Unicode 碼最小/最大的字元。

```
In [1]: min('Python程式設計')  
Out[1]: 'P'
```

- ◆ `str(x)`：傳回數值參數 x 轉換成字串的結果。

```
In [1]: str(111.2)  
Out[1]: '111.2'
```



字串處理函式

◆ Python內建許多字串處理函式如下，以下依續說明常用的函式。

- 字串轉換函式
- 字串測試函式
- 搜尋子字串函式
- 刪除指定字元或空白函式
- 格式化函式



字串處理函式

◆ 字串轉換函式

- str.upper(s) # 傳回字串參數 s 的所有字元轉換成大寫的字串
- str.lower(s) # 傳回字串參數 s 的所有字元轉換成小寫的字串
- str.swapcase(s) # 傳回字串參數 s 大小寫互換的字串
- str.capitalize(s) # 傳回字串參數 s 的字元轉換成大寫的字串
- str.title(s) # 傳回字串參數 s 每個單字第一個字元轉換成大寫的字串
- str.replace(old, new) # 傳回將字串參數 old 取代成字串參數 new 的字串



字串處理函式

◆ 字串轉換函式

```
In [1]: X='Good Afternoon! How are you?'
```

```
In [2]: print(X.upper())
GOOD AFTERNOON! HOW ARE YOU?
```

所有字元轉換成大寫

```
In [3]: print(X.lower())
good afternoon! how are you?
```

所有字元轉換成小寫

```
In [4]: print(X.swapcase())
gOOD aFTERNOON! hOW ARE YOU?
```

大小寫互換

```
In [5]: print(X.capitalize())
Good afternoon! how are you?
```

第一個字元轉換成大寫

```
In [6]: print(X.title())
Good Afternoon! How Are You?
```

每個單字第一個字元轉換成大寫

```
In [7]: print(X.replace('oo','xx'))
Gxxd Afternxxn! How are you?
```

將單字幾單字第一個字元轉換成大寫

```
In [8]: print(X)
Good Afternoon! How are you?
```

輸出X



字串處理函式

◆ 字串測試函式

- str.isupper(s) # 若字串參數 s 的所有字元都是大寫英文字母，就傳回 True
- str.islower(s) # 若字串參數 s 的所有字元都是小寫英文字母，就傳回 True
- str.isidentifier(s) # 若字串參數 s 是合法的識別字(包括關鍵字)，就傳回 True

```
In [1]: print(str.isupper('abcde'))  
False
```

```
In [5]: print(str.isidentifier('a1bcde'))  
True
```

```
In [2]: print(str.isupper('ABCDE'))  
True
```

```
In [6]: print(str.isidentifier('1abcde'))  
False
```

```
In [3]: print(str.islower('Abcde'))  
False
```

數字不能作為開頭

```
In [4]: print(str.islower('abcde'))  
True
```



字串處理函式

◆ 字串測試函式

- str.isspace(s) # 若字串參數 s 的所有字元都是空白，就傳回 True
- str.istitle(s) # 若字串參數 s 的第一個字元都是大寫英文字母，且其他字母為小寫，就傳回 True

```
In [1]: print(str.isspace(' a '))  
False
```

```
In [2]: print(str.isspace(' '))  
True
```

```
In [3]: print(str.istitle('It Is ok'))  
False
```

```
In [4]: print(str.istitle('It Is Ok'))  
True
```



字符串處理函式

◆ 字串測試函式

➤ str.isalpha(s)

```
In [1]: "ABCDE".isalpha()
Out[1]: True

In [2]: "abcde".isalpha()
Out[2]: True

In [3]: "abcde1".isalpha()
Out[3]: False
```

判斷字串參數 s 是否只包含英文字母

➤ str.isdigit (s)

```
In [1]: "12345".isdigit()
Out[1]: True

In [2]: "123.45".isdigit()
Out[2]: False

In [3]: "I".isdigit()
Out[3]: False

In [4]: "-".isdigit()
Out[4]: False
```

判斷字串參數 s 是否只包含正整數，不包含羅馬數字及中文字



字符串處理函式

◆ 字串測試函式

➤ str.isdecimal(s)

```
In [1]: "12345".isdecimal()
Out[1]: True

In [2]: "123.45".isdecimal()
Out[2]: False

In [3]: "I".isdecimal()
Out[3]: False

In [4]: "-".isdecimal()
Out[4]: False
```

判斷字串參數 s 是否只包含正整數，不包含羅馬數字及中文整數

➤ str.isnumeric(s)

```
In [1]: "12345".isnumeric()
Out[1]: True

In [2]: "123.45".isnumeric()
Out[2]: False

In [3]: "I II III".isnumeric()
Out[3]: True

In [4]: "一二三".isnumeric()
Out[4]: True
```

判斷字串參數 s 是否只包含正整數，**包含**中文數字及羅馬數字

不能打英文字I



字串處理函式

◆ 搜尋子字串函式

- str.count(s) # 傳回字串中出現參數 s 的次數 (不能重疊)
- str.startswith(s) # 若字串以參數 s 開頭，就傳回 True
- str.endswith(s) # 若字串以參數 s 結尾，就傳回 True
- str.find(s) # 傳回參數 s 出現在字串的最小索引，若找不到則傳回 -1
- str.rfind(s) # 傳回參數 s 出現在字串的最大索引，若找不到則傳回 -1



字串處理函式

◆ 搜尋子字串函式

```
In [1]: Y='HiHiHiHiHi'
```

```
In [2]: print(Y.count('Hi'))          # 字串中出現 'Hi'的次數  
5
```

```
In [3]: print(Y.startswith('Hi'))      # 字串是否以 'Hi' 開頭  
True
```

```
In [4]: print(Y.startswith('HiH'))     # 字串是否以 'HiH' 開頭  
True
```

```
In [5]: print(Y.endswith('HiH'))       # 字串是否以 'HiH' 結尾  
False
```

```
In [6]: print(Y.endswith('iHi'))        # 字串是否以 'iHi' 結尾  
True
```

```
In [7]: print(Y.find('iHi'))           # 'iHi' 出現在字串中的最小索引  
1
```

```
In [8]: print(Y.rfind('iHi'))          # 'iHi' 出現在字串中的最大索引  
7
```



字符串处理函数

◆ 字符串切割函数

- str.split([chars])

◆ 删除指定的字符或空白函数

- str.strip ([chars])
- str.lstrip([chars])
- str.rstrip([chars])



字串處理函式

◆ 字串切割函式

➤ str.split([chars]) :

- 字串會根據參數的設定(chars)進行切割。
- 參數 chars 可以省略不寫，表示切割的字元為預設值(空白)，即以一格空白進行字串的切割。

In [1]: X='A B C D E'

In [2]: Y=X.split()

In [3]: print(Y)
['A', 'B', 'C', 'D', 'E']

In [1]: X='1,3,5,7,9'

In [2]: Y=X.split(',')
['1', '3', '5', '7', '9']



字符串處理函式

◆ 刪除指定的字元或空白函式

➤ str.strip([chars]) :

- 從字串**兩側**刪除參數 chars 所指定的字元，一旦碰到不是指定的字元就停止刪除，然後傳回剩下的字串。
- 參數 chars 可以省略不寫，表示指定的字元為空白，即刪除字串兩側的空白。

```
In [1]: print(' abcde '.strip())          # 刪除字串兩側的空白  
abcde
```

```
In [2]: print('abcdxyzd ef'.strip('abcdef')) # 刪除字串兩側的 abcdef字元  
xyzd
```



字串處理函式

◆ 刪除指定的字元或空白函式

➤ str.lstrip([chars]) :

- 從字串**左側**刪除選擇性參數**chars**所指定的字元，一旦碰到不是指定的字元就停止刪除，然後傳回剩下的字串。
- 參數**chars**可以省略不寫，表示指定的字元為空白，即刪除字串左側的空白。

➤ str.rstrip([chars]) :

- 從字串**右側**刪除選擇性參數**chars**所指定的字元，一旦碰到不是指定的字元就停止刪除，然後傳回剩下的字串。
- 參數**chars**可以省略不寫，表示指定的字元為空白，即刪除字串右側的空白。

```
In [1]: z='tw.yahoo.com'
```

```
In [2]: print(z.lstrip('.tomw'))  
yahoo.com
```

```
In [3]: print(z.rstrip('.tomw'))  
tw.yahoo.c
```

```
In [4]: print(z.strip('.tomw'))  
yahoo.c
```



字串處理函式

◆ 格式化函式

- `str.center(width)` # 傳回欄位寬度為參數 width 所指定的字元數、置中的字串
- `str.ljust(width)` # 傳回欄位寬度為參數 width 所指定的字元數、靠左的字串
- `str.rjust(width)` # 傳回欄位寬度為參數 width 所指定的字元數、靠右的字串
- `str.zfill(width)` # 傳回欄位寬度為參數 width 所指定的字元數、左側補 0
- `str.format(spec)` # 根據參數 spec 所指定的格式將字串格式化，然後回傳結果



字串處理函式

◆ 格式化函式

In [1]: X='Hello Python'

In [2]: print(X.center(20))
Hello Python

In [3]: print(X.ljust(20))
Hello Python

In [4]: print(X.rjust(20))
Hello Python

In [5]: print(X.zfill(6))
Hello Python

In [6]: print(X.zfill(20))
00000000Hello Python

傳回欄位寬度為 6 個字元，左側填上 0，保留正負符號的字串

傳回欄位寬度為 20 個字元，左側填上 0，保留正負符號的字串



字串處理函式

◆ 格式化函式

- `format()` 函式可以設定字串/數字的欄位寬度與對齊方式 (字串預設靠左，數字預設靠右)

In [1]: K='Hello Python'

In [2]: print(format(K,'20'))
Hello Python

In [3]: print(format(K,'>20'))
Hello Python

In [4]: print(format(K,'<20'))
Hello Python

In [5]: K=12345

In [6]: print(format(K, '20'))
12345

In [7]: print(format(K, '>20'))
12345

In [8]: print(format(K, '<20'))
12345



Outline

- ◆ 字串運算子
- ◆ 字串處理函式
- ◆ 數值處理函式
- ◆ 日期處理函式



數值處理函式

◆ Python 已內建許多數值函式，常見如下：

- `abs(x)`
- `min(x1, x2 [, x3...])`
- `max(x1, x2 [, x3...])`
- `pow(x, y)`
- `divmod(x, y)`
- `bin(x)`
- `oct(x)`
- `hex(x)`
- `int(x)`
- `float(x)`
- `round(x [, precision])`



數值處理函式

- ◆ `abs(x)`：傳回數值參數 x 的絕對值

```
In [1]: print(abs(50))  
50
```

```
In [2]: print(abs(-50))  
50
```

- ◆ `pow(x,y)`：傳回數值參數 x 的 y 次方值

```
In [1]: print(pow(2,5))  
32
```

```
In [2]: print(pow(3,4))  
81
```

- ◆ `divmod(x,y)`：傳回數值參數 x 除以 y 的商數與餘數

```
In [1]: divmod(100,8)  
Out[1]: (12, 4)
```

```
In [2]: divmod(125,9.5)  
Out[2]: (13.0, 1.5)
```



數值處理函式

- ◆ max/min(x1, x2, x3...) : 傳回參數中的最大/最小值

```
In [1]: print(max(10,20,30))  
30
```

```
In [4]: print(min(1,3,False))  
False
```

```
In [2]: print(max(-10,-20,-30))  
-10
```

```
In [5]: print(max('a','b','c'))  
c
```

```
In [3]: print(max(True,-20,-30))  
True
```

```
In [6]: print(min('A','b','c'))  
A
```

- ◆ bin(x) : 傳回整數參數 x 由十進位轉換成二進位的字串，前面加上 0b
- ◆ oct(x) : 傳回整數參數 x 由十進位轉換成八進位的字串，前面加上 0o
- ◆ hex(x) : 傳回整數參數 x 由十進位轉換成十六進位的字串，前面加上 0x

```
In [1]: print(bin(100))  
0b1100100
```

```
In [2]: print(oct(100))  
0o144
```

```
In [3]: print(hex(100))  
0x64
```



數值處理函式

- ◆ `int(x)`：傳回數值參數 x 的整數部份或是字串參數轉換成整數，小數部份直接捨去。

```
In [1]: print(int(666.666))  
666
```

```
In [2]: print(int('666'))  
666
```

```
In [3]: print(int('666.666'))  
Traceback (most recent call last):
```

```
In [4]: print(int('六六六'))  
Traceback (most recent call last):
```

```
File "<ipython-input-4-7daa9798bef2>", line 1, in <module>  
    print(int('六六六'))
```

```
ValueError: invalid literal for int() with base 10: '六六六'  
  
File "<ipython-input-3-b9389e6933fa>", line 1, in <module>  
    print(int('666.666'))
```

```
ValueError: invalid literal for int() with base 10: '666.666'
```

- ◆ `float(x)`：傳回字串參數 x 轉換成浮點數。

```
In [1]: print(float(666))  
666.0
```

```
In [2]: print(float("666"))  
666.0
```

```
In [3]: print(float("666.666"))  
666.666
```

```
In [4]: print(float("六六六"))  
Traceback (most recent call last):
```

```
File "<ipython-input-4-d6923ce15ad4>", line 1, in <module>  
    print(float("六六六"))
```

```
ValueError: could not convert string to float: '六六六'
```



數值處理函式

- ◆ `round(x, [y])`：傳回與數值參數 x 最接近的整數；若要設定精確度為小數幾位，可以加上選擇性參數 y 。
- ◆ 小數點一位時，若是.5的情況，會視前一位的值，「奇數進位，偶數不進位元」方式進行處理。負數情況相反。

```
In [1]: print(round(123.123))  
123
```

```
In [5]: print(round(5.5))  
6
```

```
In [2]: print(round(123.123,2))  
123.12
```

```
In [6]: print(round(6.5))  
6
```

```
In [3]: print(round(-567.567))  
-568
```

```
In [7]: print(round(-5.5))  
-6
```

```
In [4]: print(round(-567.567,2))  
-567.57
```

```
In [8]: print(round(-6.5))  
-6
```

```
In [9]: print(round("999"))  
Traceback (most recent call last):
```

```
  File "<ipython-input-9-c5f93e3370d9>", line 1, in <module>  
    print(round("999"))
```

```
TypeError: type str doesn't define __round__ method
```



數值處理函式

- ◆ `round(x, [y])`：傳回與數值參數 x 最接近的整數；若要設定精確度為小數幾位，可以加上選擇性參數 y 。
- ◆ 小數值二位以上時，情形如下，**沒有特殊準則可遵循**。

```
print(round(55.005,2))      55.01
print(round(55.015,2))      55.02
print(round(55.025,2))      55.02
print(round(55.035,2))      55.03
print(round(55.045,2))      55.05
print(round(55.055,2))      55.05
print(round(55.065,2))      55.06
print(round(55.075,2))      55.08
print(round(55.085,2))      55.09
print(round(55.095,2))      55.09

print(round(-55.005,2))     -55.01
print(round(-55.015,2))     -55.02
print(round(-55.025,2))     -55.02
print(round(-55.035,2))     -55.03
print(round(-55.045,2))     -55.05
print(round(-55.055,2))     -55.05
print(round(-55.065,2))     -55.06
print(round(-55.075,2))     -55.08
print(round(-55.085,2))     -55.09
print(round(-55.095,2))     -55.09
```



數值處理函式

◆ 數學函式：Python 的 math 模組有許多常用的數學函式，常用的如下。

◆ 在使用 math 模組之前，必須使用 import 指令進行匯入：import math

- math.pi、math.e、math.nan、math.inf
- math.ceil(x)
- math.fabs(x)
- math.factorial(x)
- math.floor(x)
- math.gcd(x, y)
- math.exp(x)
- math.log(x[, base])
- math.sqrt(x)
- math.isfinite(x)
- math.isinf(x)
- math.isnan(x)
- math.radians(x)
- math.degrees(x)
- 三角函式
 - math.cos(x)、math.sin(x)
 - math.tan(x)、math.acos(x)
 - math.asin(x)、math.atan(x)



數值處理函式

◆ 亂數函式：Python 的 random 模組提供了可以用來產生亂數的函式，常用的如下。

◆ 在使用 random 模組之前，必須使用 import 語法進行匯入：import random

- random.randint(x, y) 傳回一個大於等於整數參數 x、小於等於整數 y 的隨機整數。每次傳回亂數不一定相同。
少數都包的情況(大部分是包前不包後)
- random.random() 傳回一個大於等於 0.0、小於 1.0 的隨機浮點數。每次傳回亂數不一定相同。

```
import random  
  
num = random.randint(1, 3)  
  
answer = eval(input('請猜數字1 ~ 3 :'))  
  
print(num, "==" , answer, "is", num == answer)
```

請猜數字1 ~ 3 : 2
2 == 2 is True



數值處理函式

- ◆ 我們可以使用 `format()` 函式，根據選擇性參數 `spec` 所提定的格式，將參數 `value`(數值或字串) 進行格式化。

`format(value, [spec])`

- ◆ 參數 `spec` 可以使用的格式如下。

`[[fill][align][sign][#][0][width][,][.precision][type]]`



數值處理函式

◆ 整數格式化

- 設定欄位元寬度與對齊方式：

```
In [1]: print(format(123, '^10'))  
123
```

```
In [2]: print(format(123, '!^10'))  
!!!123!!!!
```

- 設定加上千分位符號：

```
In [3]: print(format(123456789, ',', ))  
123,456,789
```



數值處理函式

◆ 浮點數格式化

- 設定欄位寬度與標記法，數值預設為靠右與小數點幾位符號，例如：

```
In [1]: print(format(1234.5678, '10.2f'))  
1234.57
```

```
In [2]: print(format(1234.5678, '<10.2f'))  
1234.57
```



Outline

- ◆ 字串運算子
- ◆ 字串處理函式
- ◆ 數值處理函式
- ◆ 日期處理函式



日期處理函式

```
import datetime  
  
nowtime=datetime.datetime.now();  
  
print("Now:",nowtime)  
print("Year:",nowtime.year)  
print("Month:",nowtime.month)  
print("Day:",nowtime.day)  
print("Hour:",nowtime.hour)  
print("Minute:",nowtime.minute)  
print("Second:",nowtime.second)  
print("Weekday:",nowtime.weekday())  
print("Weekday:",nowtime.isoweekday())  
print("Weekday:",nowtime.strftime("%A"))
```

!!有括號是函數、沒括號是屬性attribute

```
Now: 2023-10-26 21:10:23.585593  
Year: 2023  
Month: 10  
Day: 26  
Hour: 21  
Minute: 10  
Second: 23  
Weekday: 3  
Weekday: 4  
Weekday: Thursday
```

Q & A