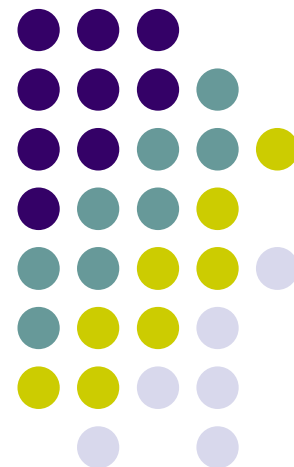


# Python程式設計入門 字串與格式化

葉難





# 大綱

- 字串型別**str**
- 字串格式化：**%運算子與方法format**
- **Unicode**：中文、日文、法文等等
- 文字編碼系統
- 檔案、檔案格式
- 位元組資料



## 型別**str**（字串、**string**）

- 序列（**Sequence**）型別，也可算是容器
- **str**物件裡的元素型別皆相同，這點與**list**、**tuple**不同
- 有人稱**str**為平坦容器（**flat container**）
- **Python**沒有「字元（**character**）」型別，可使用含一個字元的字串，如 `'a'`



## 2.x與3.x版本差異

- 2.x版，型別**str**只能含有八位元字元（例**ASCII**），**同時**也擔任位元組（二進位）資料的型別
- 2.x版，**Unicode**字串由型別**unicode**負責
- 3.x版，型別**str**就是**Unicode**字串，無**unicode**型別
- 3.x版，以型別**bytes**負責位元組（二進位）資料



# 字面值 (**literal**)

- 單引號「'」或雙引號「"」
- 跳脫 (**escape**) : 反斜線「\」

```
>>> 'abc', "def"
```

```
('abc', 'def')
```

```
>>> "I'm a boy."      # 雙引號內有單引號
```

```
"I'm a boy."
```

```
>>> 'I\'m a boy.'      # 使用跳脫字元
```

```
"I'm a boy."
```

```
>>> 'Hello '           "Python"      # 字串中間有空白字元
```

```
'Hello Python'        # 會被自動連接
```



# 特殊字元，以跳脫串來表示

跳脫串	意義
\'	單引號「'」
\"	雙引號「"」
\\	反斜線「\」
\n	ASCII的LF，插入一行（換行）
\t	ASCII的HT（TAB），水平定位
\xff	以十六進位表示字元
\uXXXX	Unicode

跳脫串	意義
\r	ASCII的CR，游標返回
\v	ASCII的VT，垂直定位
\a	ASCII的BEL，鈴響
\b	ASCII的BS，退後一格
\f	ASCII的FF，插入一頁
\ooo	以八進位表示字元



## 跳脫串範例

```
>>> 'You\'re \"so\" nice.'      # 單引號、雙引號
'You\'re \"so\" nice.'
>>> print('You\'re \"so\" nice.')
You're \"so\" nice.
>>> print('Amy\tBob\nCathy')  # TAB與換行
Amy      Bob
Cathy
>>> '\x61\x62\x63'           # 十六進位
'abc'
>>> '\u4F60\u597D'           # Unicode
'你好'
```



## 「\ 加 換行字元」延續到下一行

```
>>> 'abc def \
... ghi jkl mno \
... pqr stu vw'
'abc def ghi jkl mno pqr stu vw'
```

>>> 建議採用下列寫法，使用運算式的小括號

```
>>> ('abc def '
... 'ghi jkl mno '
... 'pqr stu vw')
'abc def ghi jkl mno pqr stu vw'
```





## 三重引號：可跨多行

- 換行與引號：毋須跳脫（除了三重引號）
- 適用於HTML、XML、JSON

```
s = '''abc def  
ghi jkl    mno  
pqr\nstuvw'''
```

```
# print(s) 輸出如下  
abc def  
ghi jkl    mno  
pqr  
stuvw
```



## raw（原始）字串

- 取消大部分的跳脫串，但仍可跳脫引號
- 適合用於表達Windows檔案路徑

```
>>> '\n'      # 一個字元，換行
```

```
'\n'
```

```
>>> r'\n'      # 兩個字元，反斜線與n
```

```
'\\n'
```

```
>>> r'C:\Program Files\Mozilla Firefox'
```

```
'C:\\Program Files\\Mozilla Firefox'
```

```
>>> r'C:\'      # raw字串的結尾不能是「\」
```

```
SyntaxError: EOL while scanning string literal
```



# 操作動作

- **str**是不可變序列型別：索引、切片、運算子+與\*、內建函式**len**、**ord**、**chr**、方法**index**與**count**

```
>>> s = 'abcdef'
>>> s[-1], s[::-1]
('f', 'fedcba')
>>> ord('a'), hex(ord('\n'))
(97, '0xa')
>>> chr(0x61)
'a'
```



## str的方法find、index、count

```
>>> 'abc def abc ghi abc jkl'.find('abc')  
0  
# 找子字串
```

```
>>> 'abc def abc ghi abc jkl'.find('abc', 3)  
8  
# 從某索引值之後開始找
```

```
>>> 'abc def abc ghi abc jkl'.index('zyx')  
ValueError: substring not found
```

```
>>> 'abc def abc ghi abc jkl'.count('abc')  
3  
# 子字串出現次數
```



## str的方法strip、split、join

```
>>> ' Hello, how are you? \t\n'.strip()
'Hello, how are you?'      # 去除兩端空白字元
>>> 'abc,def, ghi,   jkl'.split(',')
['abc', 'def', ' ghi', '   jkl'] # 切割

>>> '/'.join(('2014', '11', '23')) # 組合
'2014/11/23'

>>> ''.join(('abc', 'def', 'ghi')) # 組合
'abcdefghi'
```



# 字串格式化

- 「%」運算子：舊式，較為簡單
  - 「format」方法：新式，功能更強但也更複雜
  - 兩種都有人用，只好都學
- 
- 學字串格式化，猶如學習一套（迷你）**語言**，  
例如點飲料：去多少冰、糖度多少
  - 作練習，記住基本常用的語法



# 「%」運算子

- 字串格式化、內插（interpolation）
- 語法：

% 映射鍵 旗標 欄寬 精確度 形式

- 範例：九九乘法表（m99\_format.py）

```
'%2d x%2d = %2d' % (x, y, x*y)
```

```
2 x 1 = 2    3 x 1 = 3    4 x 1 = 4    5 x 1 = 5
2 x 2 = 4    3 x 2 = 6    4 x 2 = 8    5 x 2 = 10
2 x 3 = 6    3 x 3 = 9    4 x 3 = 12   5 x 3 = 15
```

...省略...



# 形式

形式字元	說明
'd'或'i'或'u'	有號十進位整數
'o'	有號八進位整數
'x'	有號十六進位整數（小寫字母）
'X'	有號十六進位整數（大寫字母）
'f'	浮點數，小數格式
'F'	浮點數，小數格式
'e'	浮點數，指數格式（小寫字母）
'E'	浮點數，指數格式（大寫字母）
'g'	浮點數，若指數小於-4或不小於精確度，使用指數格式（小寫字母），否則使用小數格式
'G'	浮點數，若指數小於-4或不小於精確度，使用指數格式（大寫字母），否則使用小數格式
'c'	字元，可接受整數或僅含一個字元的字串
's'	字串，使用str()轉換
'r'	字串，使用repr()轉換
'a'	字串，使用ascii()轉換
'%'	字元「%」





## 範例

```
>>> '%d' % 999          # 只有一個
'999'

>>> '%d %x 0x%X' % (999, 999, 999) # tuple
'999 3e7 0x3E7'

>>> '%f %e %g' % (12.345, 12.345, 12.345)
'12.345000 1.234500e+01 12.345'

>>> '%c %c' % (97, 'a')
'a a'

>>> '%s %r' % ('a', 'a')
'a 'a''
```



## 問題：印出**ASCII**的十六進位值

- 請印出從0x20到0x7E的**ASCII**，模樣如下：

20	21	!	22	"	23	#	24	\$	25	%	26	&	27	'		
28	(		29	)	2A	*	2B	+	2C	,	2D	-	2E	.	2F	/
30	0		31	1	32	2	33	3	34	4	35	5	36	6	37	7
38	8		39	9	3A	:	3B	;	3C	<	3D	=	3E	>	3F	?
40	@		41	A	42	B	43	C	44	D	45	E	46	F	47	G
48	H		49	I	4A	J	4B	K	4C	L	4D	M	4E	N	4F	O
50	P		51	Q	52	R	53	S	54	T	55	U	56	V	57	W
58	X		59	Y	5A	Z	5B	[	5C	\	5D	]	5E	^	5F	_
60	`		61	a	62	b	63	c	64	d	65	e	66	f	67	g
68	h		69	i	6A	j	6B	k	6C	l	6D	m	6E	n	6F	o
70	p		71	q	72	r	73	s	74	t	75	u	76	v	77	w
78	x		79	y	7A	z	7B	{	7C		7D	}	7E	~		



# 旗標

旗標字元	說明
'#'	使用另一種形式
'0'	數值前會補「0」
'-'	靠左對齊；會蓋過旗標'0'，如果同時出現的話
' '	（空格字元），正數（或空字串）之前會留一個空格
'+'	加上正負號「+、-」；會蓋過旗標' '，如果同時出現的話



## 範例

```
>>> '%X %#X' % (200, 200)    # 井字號，
'C8 0XC8'                     # 另一種形式
>>> '% d %+d %+d' % (123, 456, -789)
' 123 +456 -789'              # 空格、正負號

>>> '%8d %08d' % (123, 456) # 欄寬，補0
'      123 00000456'
>>> '%-+8d %-+8d' % (-123, -456)
'-123      -456'              # 靠左對齊
```



## 問題：產生檔名

- 請印出如下的一連串檔名

img\_001.jpg

img\_002.jpg

img\_003.jpg

...

img\_999.jpg



## 精確度：浮點數

- 以小數點「.」和一個數字表示，代表想表示小數點後幾個位數，預設為6

```
>>> '%f %.2f' % (12.345, 12.345)
```

```
'12.345000 12.35'
```

```
>>> '%.0f %#.0f' % (12.345, 12.345)
```

```
'12 12.'
```

```
>>> '%8.4f %8.4f' % (12.345, 67.89)
```

```
' 12.3450  67.8900'
```

```
>>> '%08.4f %08.4f' % (12.345, 67.89)
```

```
'012.3450 067.8900'
```



# 映射鍵

- 以小括號「(、)」指定鍵，然後由映射型別的物件（dict）提供值

```
>>> data = {'name': 'Amy', 'age': 33}
```

```
>>> '%(name)s is %(age)d years old.' % data  
'Amy is 33 years old.'
```

```
>>> 'T: %(temp)4.2f C, H: %(humid)4.2f %%' %  
    {'temp': 31.25, 'humid': 35}  
'T: 31.25 C, H: 35.00 %'
```



# 內建函式**format**

- 把值轉成指定格式化後的字串
- 只能格式化單一項目
- 對應到特殊方法\_\_format\_\_

```
>>> format(123.456, '10.2f')  
'      123.46'  
>>> format(-123.456, '#10.2g')  
' -1.2e+02'
```





## 「format」方法

- 格式字串中的置換處，以「{、}」括起來
  - 若想印出大括號，使用「{{」與「}}」
- ```
>>> '{} {} {}'.format(12, 3.4, 'hi')
'12 3.4 hi'
>>> '{2} {1} {0}'.format(12, 3.4, 'hi')
'hi 3.4 12'                                # 指定取用
>>> template = '{0} is {1} years old.'
>>> template.format('Amy', 33)
'Amy is 33 years old.'
```



# format語法

- 語法  
{ 指名 成分 !轉換 :格式指定子 }
- 「指名」：整數、關鍵字
- 「成分」：`[index]`取用容器元素，或`.name`取用物件屬性項
- 「轉換」以「!」相隔，可以是「`s`」、「`r`」或「`a`」，分別代表以`str`、`repr`、`ascii`取得字串



## 指名、成分

```
>>> li = ['a', 'b', 'c']
>>> t = [1, 2, 3]
>>> '{0[0]} {1[2]}'.format(li, t)
'a 3'
```

```
>>> '{} {fruit} {animal}'.format(
    'Hi', fruit='apple', animal='dog')
'Hi apple dog'
```

```
>>> c = 5+6j
>>> 'real {0.real}, imag {0.imag}'.format(c)
'real 5.0, imag 6.0'
```



# 轉換

```
>>> x = '你好'
```

```
>>> '{!s}'.format(x)
'你好'
```

```
>>> '{!r}'.format(x)
"'你好'"
```

```
>>> '{!a}'.format(x)
"'\\u4f60\\u597d'"
```



# 轉換

```
>>> from decimal import Decimal as D
>>> d = D('3.14')
>>> d
Decimal('3.14')
>>> '{!s}'.format(d)
'3.14'
>>> '{!r}'.format(d)
"Decimal('3.14')"
>>> '{!a}'.format(d)
"Decimal('3.14')"
```

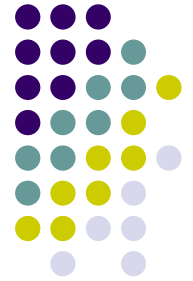


## : 格式指定子

- 語法

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

- **fill**：任何填補字元，除了大括號以外
- **align**：對齊，「<」靠左、「>」靠右、「=」正負號後填補、「^」置中
- **sign**：「+」或「-」或空格「 ' '」，同「%」
- **#**：同「%」格式指定子
- **0**：同「%」格式指定子



# 格式指定子

- **width**：欄寬，同「%」格式指定子
- **,**：每一千（三位數）就加上逗號
- **.precision**：精確度，同「%」格式指定子
- **typecode**：同「%」格式指定子（幾乎），還能以「b」代表二進位



## 範例

```
>>> '{:d} {:#X} {:#b}'.format(200, 200, 200)
'200 0XC8 0b11001000'      # 進位
```

```
>>> '{:8.3f} {:08.3f}'.format(12.3456, 3.14)
' 12.346 0003.140'        # 欄寬、精確度、補零
```

```
>>> '{:<8.2f} {:>+8.2f}'.format(12.3456, -3.14)
'12.35          -3.14'    # 對齊，正負號
```

```
>>> '{:??>8s} {:-->8s}'.format('abc', 'def')
'?????abc  -----def'  # 指定填補字元
```





## 範例

```
>>> '{:,} {:10.6f}'.format(123456789, 123456.789)
'123,456,789 123456.789000'      # 加逗點
>>> '{:^08} {:^8}'.format(3.14, 12.3456)
'003.1400 12.3456 '           # 置中

>>> '{:0=+10}'.format(3.14)
'+0000003.14'                  # 在正負號後填補
>>> '{:>10.2e}'.format(12.3456)
' 1.23e+01'                     # 指數形式
```



# 大括號裡有大括號

- 取用後面的參數

```
>>> '{0:>{2}.2e} {1:>{2}.2e}'.  
      format(3.14, 12.3456, 10)  
' 3.14e+00 1.23e+01'  
>>> '{0:{1}>8s}'.format('abc', '!')  
'!!!!!!abc'
```



# 問題

- 請改寫九九乘法表（`m99_format.py`）
- 改用`format`方法



# 問題

- 你手中有下列資料

```
cities = ['Taipei', 'Tokyo', 'London']
```

```
temps = [34.52, 26.25, 15.5]
```

```
humids = [70, 42, 56]
```

- 請寫程式印出如下的樣子

|   | City   | Temp   | Humid |
|---|--------|--------|-------|
| 0 | Taipei | 34.5 C | 70 %  |
| 1 | Tokyo  | 26.2 C | 42 %  |
| 2 | London | 15.5 C | 56 %  |

# Q&A

