

## format 格式化输出

### 1. 常用方法 %

举例：

```
>>>print("今天是%d 年%d 月%d 日"%(2019,5,18))
今天是 2019 年 5 月 18 日
>>>print('第一个数字是%.5f,第二个数字是%d'%(0.678,10))
第一个数字是 0.67800,第二个数字是 10
```

### 2. 字符串的 format()方法

- 使用大括号 “{}” 代替 “%”
- str.format()

#### 2.1 基本用法

(1) 不带编号

```
>>>print('{} {}'.format('hello','world'))
hello world
>>>print("今天是{}年{}月{}日".format(2019,5,18))
今天是 2019 年 5 月 18 日
```

(2) 带数字编号： 参数个数没有限制，可以多次使用，顺序可以任意放置

```
>>>print('{0} {1}'.format('hello','world'))
hello world
>>> print('{0} {1} {0}'.format('hello','world'))
hello world hello
>>> print('{1} {1} {0}'.format('hello','world'))
world world hello
>>>print('她叫{0},今年{1}岁,她的朋友 lisa,今年也{1}岁了'.format('Lucy',10))
她叫 Lucy,今年 10 岁,她的朋友 lisa,今年也 10 岁了
```

(3) 带关键字

```
>>>print('她叫{name},今年{age}岁.'.format(age=10,name='Lucy'))
她叫 Lucy,今年 10 岁.
```

#### 2.2 进阶用法

{ :填充字符 对齐方式 数据宽度 数据类型 }

(1) 填充：默认使用空格填充。=（只用于数字）在小数点后补齐

(2) 对齐：

符号	含义
<	左对齐（默认）
>	右对齐
^	中间对齐

举例：

宽度为 8，不够 8 位用！填充，右对齐，

```
>>>print('第一个数字是{:!>8}'.format(123.98))
第一个数字是!!123.98
```

宽度为 8，不够 8 位默认用空格填充，右对齐，

```
print('第一个数字是{:>8}'.format(123.98)) >>第一个数字是 123.98
```

(3) 数据类型

符号	含义
f	浮点数
b	二进制
d	十进制
o	八进制
x	十六进制

举例：

第一个数字是浮点类型，小数点后保留 5 位有效数字；第二个数字是十进制整数

```
>>>print('第一个数字是{:.5f},第二个数字是{:d}'.format(0.678,10))
第一个数字是 0.67800,第二个数字是 10
```

综合示例：

浮点类型，宽度为 8，小数点后保留 3 位有效数字，右对齐，不够 8 位宽度用 “！” 填充

```
>>>print('第一个数字是{:!>8.3f}'.format(123.98))
>>第一个数字是!123.980
```