

格式化输出——str.format()

1. 基本语法

format 函数可以接受不限个参数，位置可以不按顺序：

```
>>> "{} {}".format("hello", "world")      # 不设置指定位置，按默认顺序
'hello world'
>>> "{0} {1}".format("hello", "world")    # 设置指定位置
'hello world'
>>> "{1} {0} {1}".format("hello", "world") # 设置指定位置
'world hello world'
```

也可以设置参数：

```
>>> print("姓名： {name},年龄： {age}".format(name='yee',age='26'))
姓名： yee,年龄： 26

>>> # 通过字典设置参数
>>> info= {"name": "yee", "age": "26"}
>>> print("姓名： {name},年龄： {age}".format(**info))
姓名： yee,年龄： 26

>>> # 通过列表索引设置参数
>>> lst=['yee',26]
>>> print("姓名： {0[0]},年龄： {0[1]}".format(lst))# 0 是必须的，表示第一个参数
姓名： yee,年龄： 26
```

To summary:

- 字符串中大括号之间可以直接写整数，从 0 开始，对应 format 的参数，0 对应第 1 个参数，1 对应第 2 个参数，以此类推，这些整数表示的是位置。
- 如果 format 参数有列表，那么按照上面规则，一个数字对应的是列表全部，你要替换输出内容必须通过下标取值，如 {0[1]} 这种，0 表示列表，下标 [1] 表示取该列表的第 2 个元素
- 还可以利用字典来替换，大括号里可以不写位置，可以自定义任何可用名字，只要 format() 函数参数用命名关键字传入即可，或者先定义好字典，直接用 “**xxx” 的方式做参数也行
- format() 函数的参数可以是有属性对象，如果要的是对象的属性值，可以在字符串里以 {xxx.属性} 的方式，取到对应的属性值

2. 格式化进阶

语法: `{[参数名]:[填充字符][对齐][标志][最小宽度][千位分隔符][.精度][类型]}`

(1) 类型

| 符号 | 示例 | 结果 | 含义 |
|-----|------------------------------|---------------|--------|
| d | "age = ({:4d}).format(26)" | 'age = (26)' | 整数 |
| F/f | "{:2f}.format(3.1415)" | '3.14' | 浮点数 |
| % | "{:2%}.format(0.024937)" | '2.49%' | 百分比格式 |
| E/e | "{:2e}.format(314.15)" | '3.14e+02' | 科学计数法 |
| s | "age = ({:4s}).format("26")" | 'age = (26)' | 字符串 |
| b | "{:b}.format(36)" | '100100' | 二进制整数 |
| o | "{:o}.format(36)" | '44' | 八进制整数 |
| X/x | "{:x}.format(36)" | '24' | 十六进制整数 |

(2) 对齐与填充

| | |
|---|-----|
| ^ | 居中 |
| < | 左对齐 |
| > | 右对齐 |

对齐符号左边是填充字符, 注意: 只能是一个字符! 不写默认是空格填充
有填充字符必须有对齐符号, 不然中间丢失对其符号会报错!

默认, 字符串左对齐, 数值右对齐

(3) 标志: 针对数值, 用于字符串会报错

| | |
|----|--------------------|
| + | 显示正负号 |
| 空格 | 整数显示空格, 负数显示符号 |
| # | 仅用于显示前缀, 如八进制、十六进制 |

(4) 最小宽度

`format()`后, 该段格式化后整体的最小长度, 不足长度会填充填充字符, 如果替换的内容长度超过设定的最小长度, 以实际长度为准

(5) 精度

作用于字符串时, 是输出字符串的个数, 作用于整数时报错, 作用于浮点数时, 表示输出小数位数, 科学计数法表示的是底数的小数位数, 默认 6 位小数

(6) 千位分隔符

就是逗号: “,”, 只作用于数值

```
>>> "{:,d}".format(123456789)
'123,456,789'
>>> "{:,.2f}".format(12345.6789)
'12,345.68'
```