

字符串的format方法

1. 概念:

- `str.format()` 方法通过字符串中的大括号 `{}` 来识别替换字段 `replacement field`，从而完成字符串的格式化。
- 替换字段 由字段名 `field name` 和转换字段 `conversion field` 以及格式说明符 `format specifier` 组成，即一般形式为 `{字段名!转换字段:格式说明符}`。
- 字段名分为简单字段名 `simple field name` 和复合字段名 `compound field name`。而转换字段和格式说明符都是可选的。

2. 字段名

form的完整格式是`{字段名!转换字符:格式说明符}`。其中字段名是必须的，而且可以分为简单字段名和复合字段名。

2.1 简单字段名

简单字段名由三中写法:

- 省略字段名: `{}`
- 使用非负十进制整数 `{0}`
- 变量名`{name}`

2.1.1 省略字段名

大括号内省略字段名，传递位置参数。

- 替换字段形式: `{}`
- 注意：大括号个数可以少于位置参数的个数，反之不然。

```
# 省略字段名传递位置参数
print('我叫{}, 今年{}岁。'.format('小明', 18))
"""
我叫小明, 今年18岁。
"""

# 大括号个数可以少于位置参数的个数
print('我爱吃{}和{}'.format('香蕉', '苹果', '大鸭梨'))
"""
我爱吃香蕉和苹果。
"""

# 大括号个数多于位置参数的个数则会报错
# print('我还吃{}和{}'.format('西红柿'))
"""
IndexError: tuple index out of range
"""
```

2.1.2 数字字段名

可以通过数字形式的简单字段名传递位置参数。

- 数字必须是大于等于 0 的整数。
- 带数字的替换字段可以重复使用。
- 数字形式的简单字段名相当于把 format 中的所有位置参数整体当作一个元组，通过字段名中的数字进行取值。即 {0} 等价于 tuple[0]，所以大括号内的数字不能越界。

```

# 通过数字形式的简单字段名传递位置参数
print('身高{0}，家住{1}。'.format(1.8, '铜锣湾'))
"""
身高1.8，家住铜锣湾
"""

# 数字形式的简单字段名可以重复使用。
print('我爱{0}。\\n她今年{1}。\\n我也爱{0}。'.format('阿香', 17))
"""
我爱阿香。
她今年17。
我也爱阿香。
"""

# 体会把所有位置参数整体当成元组来取值
print('阿香爱吃{1}、{3}和{0}。'.format(
    '榴莲', '臭豆腐', '皮蛋', '鲱鱼罐头', '螺狮粉'))
"""
阿香爱吃臭豆腐、鲱鱼罐头和榴莲。
"""

# 尝试一下越界错误
# print('{1}'.format('错误用法'))
"""
IndexError: tuple index out of range
"""

```

2.1.3 变量字段名

使用**变量名形式**的简单字段名传递**关键字参数**。

- 关键字参数的位置可以随意调换。

```

# 使用变量名形式的简单字段名传递关键字参数
print('我大哥是{name}，今年{age}岁。'.format(name='阿飞', age=20))
"""
我大哥是阿飞，今年20岁。
"""

# 关键字参数的顺序可以随意调换
print('我大哥是{name}，今年{age}岁。'.format(age=20, name='阿飞'))
"""
我大哥是阿飞，今年20岁。
"""

```

2.1.4 简单字段名的混合使用

- 混合使用数字形式和变量名形式的字段名，可以同时传递位置参数和关键字参数。
- 关键字参数必须位于位置参数之后。
- 混合使用时可以省略数字。
- 省略字段名 `{}` 不能和数字形式的字段名 `{非负整数}` 同时使用。

```
# 混合使用数字形式和变量名形式的字段名
# 可以同时传递位置参数和关键字参数
print('这是一个关于{0}、{1}和{girl}的故事.'.format(
    '小明', '阿飞', girl='阿香'))
"""
这是一个关于小明、阿飞和阿香的故事。
"""

# 但是关键字参数必须位于位置参数之后
# print('这是一个关于{0}、{1}和{girl}的故事.'.format(
#     # '小明', girl='阿香' , '阿飞'))
"""
SyntaxError: positional argument follows keyword argument
"""

# 数字也可以省略
print('这是一个关于{}、{}和{girl}的故事.'.format(
    '小明', '阿飞', girl='阿香'))

# 但是省略字段名不能和数字形式的字段名同时出现
# print('这是一个关于{}、{1}和{girl}的故事.'.format(
#     # '小明', '阿飞', girl='阿香'))
"""
ValueError: cannot switch from automatic field numbering to manual field specification
"""
```

2.1.5 使用元组和字典传参

`str.format()` 方法还可以使用 `*元组` 和 `**字典` 的形式传参，两者可以混合使用。位置参数、关键字参数、`*元组` 和 `**字典` 也可以同时使用，但是要注意，位置参数要在关键字参数前面，`*元组` 要在 `**字典` 前面。

```

# 使用元组传参
infos = '钢铁侠', 66, '小辣椒'
print('我是{}, 身价{}亿。'.format(*infos))
"""
我是钢铁侠, 身家66亿。
"""

print('我是{2}, 身价{1}亿。'.format(*infos))
"""
我是小辣椒, 身家66亿。
"""

# 使用字典传参
venom = {'name': '毒液', 'weakness': '火'}
print('我是{name}, 我怕{weakness}'.format(**venom))
"""
我是毒液, 我怕火。
"""

# 同时使用元组和字典传参
hulk = '绿巨人', '拳头'
captain = {'name': '美国队长', 'weapon': '盾'}
print('我是{}, 我怕{weapon}'.format(*hulk, **captain))
print('我是{name}, 我怕{1}'.format(*hulk, **captain))

"""
我是绿巨人, 我怕盾。
我是美国队长, 我怕拳头。
"""

# 同时使用位置参数、元组、关键字参数、字典传参
# 注意:
# 位置参数要在关键字参数前面
# *元组要在**字典前面
tup = '鹰眼',
dic = {'weapon': '箭'}
text = '我是{1}, 我怕{weakness}。我是{0}, 我用{weapon}。'
text = text.format(
    *tup, '黑寡妇', weakness='男人', **dic)
print(text)
"""
我是黑寡妇, 我怕男人。我是鹰眼, 我用箭。
"""

```

2.2 复合字段名

- 同时使用了数字和变量名两种形式的字段名就是复合字段名。

- 复合字段名

支持两种操作符：

- `.` 点号
- `[]` 中括号

2.2.1 使用 `.` 点号

传递位置参数

- 替换字段形式： `{数字.属性名}`
- 只有一个替换字段的时候可以省略数字

```
class Person(object):
    def __init__(self, name, age, gender):
        self.name = name
        self.age = age
        self.gender = gender
p = Person('zhangsan', 18, 'female')
print('姓名是{0.name}, 年龄是{0.age}, 性别是{0.gender}'.format(p))
print('姓名是{.name}'.format(p)) # 只有一个替换字段时，可以省略数字
```

2.2. 使用 `[]` 中括号

- 用列表传递位置参数
- 用元组传递位置参数
- 用字典传递位置参数

```

# 中括号用法：用列表传递位置参数
infos = ['阿星', 9527]
food = ['霸王花', '爆米花']
print('我叫{0[0]}, 警号{0[1]}, 爱吃{1[0]}。'.format(
    infos, food))
"""
我叫阿星, 警号9527, 爱吃霸王花。
"""

# 中括号用法：用元组传递位置参数
food = ('僵尸', '脑子')
print('我叫{0[0]}, 年龄{1}, 爱吃{0[1]}。'.format(
    food, 66))
"""
我叫僵尸, 年龄66, 爱吃脑子。
"""

# 中括号用法：用字典传递位置参数
dic = dict(name='阿星', pid=9527)
print('我是{[name]}!'.format(
    dic))
# 多个替换字段, 不能省略数字
print('我是{0[name]}, 警号{0[pid]}。'.format(
    dic))
"""
我是阿星!
我是阿星, 警号9527。
"""

```

3. 转换字段

转换字段 `conversion field` 的取值有三种, 前面要加 `!` :

- `s` : 传递参数之前先对参数调用 `str()`
- `r` : 传递参数之前先对参数调用 `repr()`
- `a` : 传递参数之前先对参数调用 `ascii()`

`ascii()` 函数类似 `repr()` 函数, 返回一个可以表示对象的字符串。但是对于非 `ASCII` 字符, 使用 `\x`, `\u` 或者 `\U` 转义。

```
# 转换字段
print('I am {!s}!'.format('Bruce Lee 李小龙'))
print('I am {!r}!'.format('Bruce Lee 李小龙'))
print('I am {!a}!'.format('Bruce Lee 李小龙'))
"""
I am Bruce Lee 李小龙!
I am 'Bruce Lee 李小龙'!
I am 'Bruce Lee \u674e\u5c0f\u9f99'!
"""
```

4. 格式说明符

格式说明符使用过于复杂，且实际使用场景不多，暂不讨论。

千锋Python人工智能学院