# 字符串转字典

将一个 python 的字符串转为字典，比如字符串：

user\_info = '{"name" : "john", "gender" : "male", "age": 28}'

想把它转为下面的字典：

user\_dict = {"name" : "john", "gender" : "male", "age": 28}

1. **通过 json 来转换**

>>> import json>>> user\_info= '{"name" : "john", "gender" : "male", "age": 28}'

>>> user\_dict = json.loads(user\_info)>>> user\_dict

{u'gender': u'male', u'age': 28, u'name': u'john'}

但是使用 json 进行转换存在一个潜在的问题。

由于 json 语法规定 **数组或对象之中的字符串必须使用双引号，不能使用单引号**

1. **通过 eval**

>>> user\_info = '{"name" : "john", "gender" : "male", "age": 28}'

>>> user\_dict = eval(user\_info)>>> user\_dict

{'gender': 'male', 'age': 28, 'name': 'john'}>>> user\_info = "{'name' : 'john', 'gender' : 'male', 'age': 28}"

>>> user\_dict = eval(user\_info)>>> user\_dict

{'gender': 'male', 'age': 28, 'name': 'john'}

通过 eval 进行转换就不存在上面使用 json 进行转换的问题。但是，使用 eval 却存在安全性的问题，比如下面的例子:

# 让用户输入 `user\_info`

user\_info = input('input user info: ')

# 输入 {"name" : "john", "gender" : "male", "age": 28}，没问题

user\_dict = eval(user\_info)

# 输入 \_\_import\_\_('os').system('dir'),user\_dict 会列出当前的目录文件！

# 再输入一些删除命令，则可以把整个目录清空了！

user\_dict = eval(user\_info)

print(user\_dict)

**3、通过 literal\_eval**

>>> import ast>>> user = '{"name" : "john", "gender" : "male", "age": 28}'

>>> user\_dict = ast.literal\_eval(user)>>> user\_dict

{'gender': 'male', 'age': 28, 'name': 'john'}

user\_info = "{'name' : 'john', 'gender' : 'male', 'age': 28}"

>>> user\_dict = ast.literal\_eval(user)>>> user\_dict

{'gender': 'male', 'age': 28, 'name': 'john'}

使用 ast.literal\_eval 进行转换既不存在使用 json 进行转换的问题，也不存在使用 eval 进行转换的 安全性问题，因此推荐使用 ast.literal\_eval。