
图解
Python
可能是迄今为止
最易懂的Python视频教程

```
"""
    类对象的特殊方法之__str__()和__repr__()用于自定义并返回实例对象的字符串表示形式。
"""

"""
1. 当在交互式命令行中直接打印一个实例对象时，
    如果在实例对象对应的类对象中实现了特殊方法__repr__()，会自动调用该方法；
    否则，会打印实例对象对应的类对象和实例对象在内存中的地址。

2. 当调用内置函数print打印一个实例对象时，
    如果在实例对象对应的类对象中实现了特殊方法__str__()，会自动调用该方法；
    否则，如果在实例对象对应的类对象中实现了特殊方法__repr__()，会自动调用该方法；
    否则，会打印实例对象对应的类对象和实例对象在内存中的地址。

3. 当调用内置函数str创建字符串并且实参是一个实例对象时，
    如果在实例对象对应的类对象中实现了特殊方法__str__()，在内置函数str的内部会自动调用该方法；
    否则，如果在实例对象对应的类对象中实现了特殊方法__repr__()，在内置函数str的内部会自动调用该方法；
    否则，会打印实例对象对应的类对象和实例对象在内存中的地址。

4. 当调用内置函数repr创建字符串并且实参是一个实例对象时，
    如果在实例对象对应的类对象中实现了特殊方法__repr__()，在内置函数repr的内部会自动调用该方法；
    否则，会打印实例对象对应的类对象和实例对象在内存中的地址。
"""
```

```
"""
    通常情况下，类对象的特殊方法__str__()和__repr__()的实现代码是一样的，因此，
当实现了其中一个后，可以把其方法名赋值给另一个的方法名。
"""
class MyClass(object):
    def __str__(self):
        return "xxx"

    __repr__ = __str__

"""
    内置函数str()和repr()都返回对象的字符串表示，其区别在于：
str()的返回值是给用户看的，更加用户友好；
repr()的返回值是给程序开发者看的，是为调试服务的。

>>> str("Hello,\nworld!")
'Hello,\nworld!'
>>> repr("Hello,\nworld!")
"'Hello,\\nworld!'"
"""
```