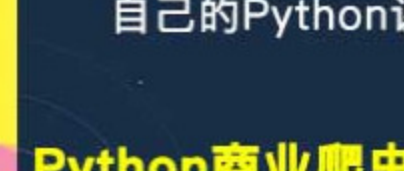


- 📄 Python教程
- 📄 Python简介
- 📄 安装Python
- 📄 第一个Python程序
- 📄 Python基础
- 📄 函数
- 📄 高级特性
- 📄 装饰器编程
- 📄 模块
- 📄 面向对象编程
- 📄 面向对象的编程
- 📄 使用__slots__
- 📄 使用@property
- 📄 多重继承
- 📄 定制类
- 📄 持久化存储
- 📄 使用元类
- 📄 错误、调试和测试
- 📄 IO编程
- 📄 进程和线程
- 📄 正则表达式
- 📄 常用内建模块
- 📄 常用第三方模块
- 📄 virtualenv
- 📄 图形界面
- 📄 网络编程
- 📄 数据库编程
- 📄 Web开发
- 📄 异步IO
- 📄 实践
- 📄 FAQ
- 📄 期末总结

关于作者



廖雪峰

自己的Python课程

Python商业爬虫全解密

让天下没有爬不到的数据!

Python爬虫 + 数据分析

Python机器学习 + 深度学习

找廖雪峰老师

廖雪峰老师

自己的Java课程

Java高级架构师

更专业 更权威

源码分析专题 + 微服务架构专题

高并发分布式专题 + 性能优化专题

找廖雪峰老师

Python内部教材+100G全套学习视频

🔍 搜索: 100G全套学习视频

python大型免费高级进阶公开课

授课模式: 在线直播+课后视频, 从零基础到中级开发工程师

Python内部教材+100G全套学习视频

🔍 搜索: 100G全套学习视频

python大型免费高级进阶公开课

授课模式: 在线直播+课后视频, 从零基础到中级开发工程师

使用枚举类

阅读: 286245

当我们需要定义常量时, 一个办法是用大写字母通过整数来定义, 例如月份:

```
JAN = 1
FEB = 2
MAR = 3
```

```
APR = 4
MAY = 5
JUN = 6
```

好处是简单, 缺点是类型是 `int`, 并且仍然是变量。

更好的方法是为这样的枚举类型定义一个 `class` 类型, 然后, 每个常量都是 `class` 的一个唯一实例。Python 提供了 `Enum` 类来实现这个功能:

```
from enum import Enum
```

```
Month = Enum('Month', ('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'))
```

这样我们就获得了 `Month` 类型的枚举类, 可以直接使用 `Month.Jan` 来引用一个常量, 或者枚举它的所有成员:

```
for name, member in Month._members.items():
```

```
    print(name, '<=', member, '>=', member.value)
```

`value` 属性则是自动附加给成员的一个 `int` 常量, 默认从 1 开始计数。

如果需要更精确地控制枚举类型, 可以从 `Enum` 类生出自定义类:

```
from enum import Enum, unique
```

```
@unique
```

```
class Weekday(Enum):
```

```
    Sun = 0 # Sun的value被设定为0
```

```
    Mon = 1
```

```
    Tue = 2
```

```
    Wed = 3
```

```
    Thu = 4
```

```
    Fri = 5
```

```
    Sat = 6
```

`@unique` 装饰器可以帮助我们检查保证没有重复值。

访问这些枚举类型可以有若干种方法:

```
>>> day = Weekday.Mon
```

```
>>> print(day)
```

```
Weekday.Mon
```

```
>>> print(Weekday.Tue)
```

```
Weekday.Tue
```

```
>>> print(Weekday('Tue'))
```

```
Weekday.Tue
```

```
>>> print(Weekday.Tue.value)
```

```
2
```

```
>>> print(day == Weekday.Mon)
```

```
True
```

```
>>> print(day == Weekday.Tue)
```

```
False
```

```
>>> print(Weekday(3))
```

```
Weekday.Wed
```

```
>>> print(day == Weekday(3))
```

```
True
```

```
>>> Weekday.__members__
```

```
{'Sun': <enum>
```

```
Weekday.Sun: 0, 'Mon': <enum>
```

```
Weekday.Mon: 1, 'Tue': <enum>
```

```
Weekday.Tue: 2, 'Wed': <enum>
```

```
Weekday.Wed: 3, 'Thu': <enum>
```

```
Weekday.Thu: 4, 'Fri': <enum>
```

```
Weekday.Fri: 5, 'Sat': <enum>
```

```
Weekday.Sat: 6}
```

可见, 既可以用成员名称引用枚举常量, 又可以直接给出 `value` 的值引用枚举常量。

练习

把 `Student` 的 `gender` 属性改造为枚举类型, 可以避免使用字符串:

```
# -*- coding: utf-8 -*-
```

```
from enum import Enum, unique
```

```
class Gender(Enum):
```

```
    Male = 0
```

```
    Female = 1
```

```
class Student(object):
```

```
    def __init__(self, name, gender):
```

```
        self.name = name
```

```
        self.gender = gender
```

```
# 测试:
```

```
bart = Student('bart', Gender.Male)
```

```
if bart.gender == Gender.Male:
```

```
    print('测试通过!')
```

```
else:
```

```
    print('测试失败!')
```

小结

`Enum` 可以把一组相关常量定义在一个 `class` 中, 且 `class` 不可变, 而且成员可以直接比较。

参考源码

use_enum.py

该后有收获可以请作者喝咖啡, 该后有疑问可加群讨论:



还可以分享给朋友:

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎

分享到豆瓣

分享到贴吧

分享到论坛

分享到博客

分享到微博

分享到微信

分享到QQ

分享到知乎