目录
- Python教程
  - Python简介
  - 安装Python
  - 第一个Python程序
  - Python语法
  - 函数
  - 调用函数
  - 函数的定义
  - 模块
  - 面向对象编程
    - 类和实例
    - 访问限制
    - 继承和多态
    - 获取对象信息
    - 实例属性和类属性
  - 面向对象高级编程
  - 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
  - 电子邮件
  - 访问数据库
  - Web开发
  - 异步IO
  - 实战
  - FAQ
  - 廖雪峰说我

实例属性和类属性

阅读: 315021

由于Python是动态语言，根据类创建的实例可以任意绑定属性。

给实例绑定属性的方法是通过实例变量，或者通过self变量：

```
class Student(object):
    def __init__(self, name):
        self.name = name

s = Student('Bob')
s.score = 90
```

但是，如果Student实例有很多很多个属性怎了？可以直接在class中定义属性，这种属性是类属性，归Student类所有：

```
class Student(object):
    name = 'Student'
```

当给Bob定义了一个同名属性，这个属性会屏蔽掉类的同名属性，但实例对象本身还可以访问到。来测试一下：

```
>>> class Student(object):
...     name = 'Student'
...
>>> s = Student()# 创建实例：
>>> print(s.name)# 打印name，从实例或者类定义name属性：
Student
>>> print(Student.name)# 打印类的name属性：
Student
>>> s.name = 'Michael'# 给实例绑定name属性：
>>> print(s.name)# 打印实例的name属性，现在，它会屏蔽类的name属性：
Michael
>>> print(Student.name)# 但是类的name属性并没有被改变：
Student
>>> del s.name# 删除实例的name属性：
>>> print(s.name)# 再次打印name，由于实例的name属性没有了，但Student的name属性还在：
Student
```

从上面的例子可以看出，在编写程序的过程中，千万不要对实例属性和同名的类属性使用相同的名字，因为相同名字的实例属性将屏蔽掉类属性，但是在调用实例属性的时候，会优先访问实例属性。

练习

为了给学生人数，可以给Student类增加一个类属性，每创建一个实例，属性数加1即可：

```
# -*- coding: utf-8 -*-
class Student(object):
    count = 0

    def __init__(self, name):
        self.name = name

# 测试:
if Student.count != 0:
    print('测试失败!')
else:
    bart = Student('Bart')
    if Student.count != 1:
        print('测试失败!')
    else:
        lisa = Student('Bart')
        if Student.count != 2:
            print('测试失败!')
        else:
            print('测试成功!')
            print('Students:', Student.count)
            print('测试成功!')
```

▶ Run

小结

实例属性属于各个实例所有，互不干扰；

类属性属于类所有，所有实例共享一个属性；

不要对实例属性和类属性使用相同的名字，否则将产生意想不到的结果。

读后有感记记录你学者嘴嘴呀，读后有福利送配图好公：

关注公众号抽奖：

关注公众号抽奖

廖雪峰
自己的Python课程
Python商业爬虫全解
让天下没有难不到的数据！
Python爬虫 + 数据分析
Python数学化 + 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
源码分析专家 微服务架构专家
海外支付专家 性能优化专家
找廖雪峰老师

python免费公开课
编程不已限
授课模式：在线直播+课后辅导，从零基础到中级高级开发工程师
编程不已限
查看详情

1 python免费公开课

2 视频通话SDK
声网Agora.io，API接口，时代领跑人，每日17万钟免费。
www.agora.io

学生服务器年终盘点
10元/月
腾讯云服务器高性能计算能力 高性价比优惠套餐公 助你成功开端冒险。

校园专享服务器
套餐10元/月
腾讯云服务器高性能计算能力 高性价比优惠套餐公 助你成功开端冒险。

关于作者

廖雪峰 V 分享 编辑过
关注作者

廖雪峰
自己的Python课程
Python商业爬虫全解
让天下没有难不到的数据！
Python爬虫 + 数据分析
Python数学化 + 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
源码分析专家 微服务架构专家
海外支付专家 性能优化专家
找廖雪峰老师

← 上一篇 下一篇 →

python免费公开课
编程不已限
授课模式：在线直播+课后辅导，从零基础到中级高级开发工程师
编程不已限
查看详情

ACM金牌得主
世界顶尖高校一线教学家亲授公开课
人工智能与自然语言/计算机视觉课程培训
北京·上海·广州·深圳·杭州·成都·南京·武汉·西安·长沙·重庆
无offer退全款

廖雪峰推荐
JAVA进阶教程
原价199元
0元领取

爆款云产品拼购2折起
1核云主机低至19元/年，海量上云门槛
立即查看

Python全栈工程师特训营
Python全栈工程师特训营，全面掌握Python语言，全栈开发工具，数据分析，爬虫工程师，Web开发等核心技术
立即报名

评论

登录/注册 | 使用Markdown格式 | 按Ctrl+Enter直接提交

梦沙罗人民 在Student类增加一个类属性，每创建一个实例，属性数自动增加。
Jeffrey0 created at May 24, 2019 10:24 PM, Last updated at May 24, 2019 10:24 PM

```
class Student(object):
    count = 0

    def __init__(self, name):
        self.name = name
        Student.count+=1
```

回复(0) 举报

关于自增切换
Jeffrey0 created at May 27, 2019 10:26 PM, Last updated at 10 hours ago

哪位有知道如何实现类似于a.b这样可以实现每次调用时自增？而不是每次调用都加1的count = 0+1操作？如果是get的话，那会每次初始化为0然？那不是因为python是动态语言？

王小飞 Created at May 28, 2019 7:07 PM

count 是类属性，应该初始是类对象初始化 self 的属性

MOCK_ Created at May 31, 2019 9:19 PM

重定义时调用了__init__方法？而且不是初始化count = 1的初始化时属性count的属性没了，所以每次实例化都增1

李春
Created at June 1, 2019 7:42 AM

Java中属性会不同修改实例，每次都是初始化count = 0 ?

cuiwu001 Created 20 hours ago

Java中也可以做count=0吧？

回复(0) 举报

为什么不能使用 nonlocal 呢
王小飞 created at May 28, 2019 7:09 PM, Last updated at May 30, 2019 7:12 PM

a

王小飞 Created at May 28, 2019 7:12 PM

我已解决：

1. 在局变量：在所有函数和闭包外面
2. 局部变量：1）在函数内
   2）在闭包的范围内，并且 self 赋值
3. 静态变量：在class内，并且是在方法外
4. 实例变量：是class的实例，用self赋值

总型的 count 不是在class的范围内，不是局部变量，是静态变量，所以不能用 nonlocal，静态变量使用 类名.属性 访问

回复(0) 举报

哈哈，自作聪明了一回
李春 created at May 31, 2019 2:37 PM, Last updated at May 31, 2019 4:27 PM

```
class Student(object):
    count = 0

    def __init__(self, name):
        self.name = name
        Student.count+=1
```

回复(0) 举报

学不完，每天学习
Jeffrey0 created at May 16, 2019 3:06 PM, Last updated at May 16, 2019 3:06 PM

```
class Student(object):
    count = 0

    def __init__(self, name):
        self.name = name
        Student.count+=1
```

回复(0) 举报

求问这个能实现，如何写
Jeffrey created at May 11, 2019 10:32 AM, Last updated at May 11, 2019 10:32 AM

```
class Student(object):
    count = 0

    def __init__(self, name):
        self.name = name
        Student.count+=1
```

回复(0) 举报

学习
Jeffrey created at May 6, 2019 8:09 PM, Last updated at May 6, 2019 8:09 PM

```
class Student(object):
    count = 0

    def __init__(self, name):
        self.name = name
        Student.count=Student.count+1
```

回复(0) 举报

实例属性属于各个实例所有，互不干扰；类属性属于类所有，所有实例共享一个属性；
梦沙罗人民 created at May 24, 2019 10:17 PM, Last updated at May 24, 2019 10:17 PM

实例属性属于各个实例所有，互不干扰；

类属性属于类所有，所有实例共享一个属性；

回复(0) 举报

DCSO
Mike created at April 20, 2019 10:29 PM, Last updated at April 20, 2019 10:29 PM

```
class Student(object):
    count = 0

    def __init__(self, name):
        self.name = name
        Student.count=Student.count+1
```

回复(0) 举报

为什么这种中一定要用Student.count去调用count呢，内容改动后内存动态改变是会有相应的活动吗
梦沙罗人民 created at April 20, 2019 10:30 PM, Last updated at April 20, 2019 7:32 PM

class Student(): count = 0 def init(self, name): self.name = name Student.count += 1 问题是用实例方法直接调用count？notice a问题

张春雷同学
Created at April 20, 2019 7:32 PM

notice a的调用是用实例调用的吗？回答最后提问的人应该是编程思维问题？notice a问题，个人理解，不知道是否正确

回复(0) 举报

小小
Jeffrey created at April 20, 2019 3:45 PM, Last updated at April 20, 2019 3:45 PM

```
class Student(object):
    count = 0

    def __init__(self, name):
        self.name = name
        Student.count=Student.count+1
```

回复(0) 举报

好的
Jeffrey created at April 20, 2019 12:30 PM, Last updated at April 20, 2019 4:06 PM

```
class Student(object):
    count = 0
    # Student
    count = 1

    def __init__(self, name):
        self.name = name
        Student.count = 2
```

梦春雷同学回复 created at April 20, 2019 4:06 PM

count = Student.count 你在调用count = 0 这句工夫要你在类Student中调用了 和实例对 Student 的实例调用的属性

回复(0) 举报

春雷
Jeffrey created at April 20, 2019 2:57 PM, Last updated at April 20, 2019 2:57 PM

```
class Student(object):
    count = 0

    def __init__(self, name):
        self.name = name
        Student.count = 1
```

回复(0) 举报

春雷
梦沙罗人民 created at April 20, 2019 2:37 PM, Last updated at April 20, 2019 2:37 PM

```
class Student(object):
    count = 0

    def __init__(self, name):
        self.name = name
        Student.count = 1
```

回复(0) 举报

春雷
Jeffrey 编辑 created at April 20, 2019 2:35 PM, Last updated at April 20, 2019 2:35 PM

```
class Student(object):
    count = 0

    def __init__(self, name):
        self.name = name
        Student.count=1
```

回复(0) 举报

最明最好了，你最知其所不知其所以然
Fishbone created at April 20, 2019 12:06 AM, Last updated at April 20, 2019 12:06 PM

```
class Student(object):
    count = 0

    def __init__(self, name):
        self.name = name
        Student.count = self.count +1
```

王小飞回复
Created at April 21, 2019 3:33 PM

哪哪了，只要这么一问就会的已一遍 Student.count = self.count +1这部分，就这这里的self.count等同于student.self，因为都是使用的类属性

回复(0) 举报

空空空
Fishbeforeme created at March 20, 2019 11:52 AM, Last updated at April 21, 2019 2:11 AM

```
class Student(object):
    count = 0 def init(self, name): self.name = name Student.count+=1
```

空空空回复 Created at March 21, 2019 10:44 AM

为什么要加student.count+=1,不能ctrl+c+v??

王小飞回复 Created at April 21, 2019 2:11 AM

count 是默认的类属性由外层定义不能解释成默认为全局的实例属性的小白个人理论，仅供参考

回复(0) 举报

初学一天，就点打脸了
小明 created at April 2, 2019 8:06 PM, Last updated at April 2, 2019 8:40 PM

```
class Student(object):
    count = 0

    def __init__(self, name):
        self.name = name
        Student.count += 1
```

回复(0) 举报

测试小妹 空空空，加油，大家一起努力。
小明 编辑过 created at April 2, 2019 8:08 AM, Last updated at April 2, 2019 8:08 AM

```
# 实例属性
count = 0

class Student(object):
    # 类属性
    count = 0

    def __init__(self, name):
        self.name = name
        Student.count += 1
```

回复(0) 举报

发表评论

登录后发表评论