

海子

不经历风雨，怎能见彩虹？做一个快乐的程序员。

博客园    首页    新随笔    联系    订阅    管理

随笔 - 184   文章 - 0   评论 - 1287

昵称：海子  
园龄：6年1个月  
粉丝：3750  
关注：6  
+加关注

< 2013年3月 >						
日	一	二	三	四	五	六
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

搜索

找找看

谷歌搜索

最新随笔

1. 金山快盘+ TortoiseSVN构建版本控制仓库
2. 在windows下安装配置Ulipad
3. Java并发编程：线程间协作的两种方式：wait、notify、notifyAll和Condition
4. JAVA多线程和并发基础面试题答（转载）
5. Java线程面试题 Top 50（转载）
6. Java并发编程：Timer和TimerTask（转载）
7. Java并发编程：Callable、Future和FutureTask
8. Java并发编程：CountDownLatch、CyclicBarrier和Semaphore
9. Java并发编程：线程池的使用
10. Java并发编程：阻塞队列

随笔分类(193)

- Android(14)
- C/C++(30)
- Java NIO(2)
- Java并发编程(19)
- Java代码之美
- Java基础(18)

Python 面向对象编程（一）

Python 面向对象编程（一）

虽然Python是解释性语言，但是它是面向对象的，能够进行对象编程。下面就来了解一下如何在Python中进行对象编程。

一.如何定义一个类

在进行python面向对象编程之前，先来了解几个术语：类，类对象，实例对象，属性，函数和方法。

类是对现实世界中一些事物的封装，定义一个类可以采用下面的方式来定义：

```
class className:
    block
```

注意类名后面有个冒号，在block块里面就可以定义属性和方法了。当一个类定义完之后，就产生了一个类对象。类对象支持两种操作：引用和实例化。引用操作是通过类对象去调用类中的属性或者方法，而实例化是产生出一个类对象的实例，称作实例对象。比如定义了一个people类：

```
class people:
    name = 'jack'           #定义了一个属性
    #定义了一个方法
    def printName(self):
        print self.name
```

people类定义完成之后就产生了一个全局的类对象，可以通过类对象来访问类中的属性和方法了。当通过people.name（至于为什么可以直接这样访问属性后面再解释，这里只要理解类对象这个概念就行了）来访问时，people.name中的people称为类对象，这点和C++中的有所不同。当然还可以进行实例化操作，p=people()，这样就产生了一个people的实例对象，此时也可以通过实例对象p来访问属性或者方法了(p.name).

理解了类、类对象和实例对象的区别之后，我们来了解一下Python中属性、方法和函数的区别。

在上面代码中注释的很清楚了，name是一个属性，printName()是一个方法，与某个对象进行绑定的函数称作为方法。一般在类里面定义的函数与类对象或者实例对象绑定了，所以称作为方法；而在类外定义的函数一般没有同对象进行绑定，就称为函数。

二.属性

在类中我们可以定义一些属性，比如：

```
class people:
    name = 'jack'
    age = 12

p = people()
print p.name,p.age
```

定义了一个people类，里面定义了name和age属性，默认值分别为'jack'和12。在定义了类之后，就可以用来产生实例化对象了，这句p = people()实例化了一个对象p，然后就可以通过p来读取属性了。这里的name和age都是公有的，可以直接在类外通过对象名访问，如果想定义成私有的，则需在前面加2个下划线 '\_\_'。

```
class people:
    __name = 'jack'
```

Java集合
Java网络编程
Java虚拟机
JS/CSS/Jquery
Linux/Shell(4)
Mysql
OJ(14)
PHP
Python/Web框架(16)
Redis
Web开发(2)
XML(1)
计算机网络
计算机系统(3)
开发工具(7)
设计模式(1)
数据结构(11)
数据库(2)
数据挖掘(1)
算法(27)
无线传感器网络(1)
信息检索
业余娱乐(6)
转载(14)
自然语言处理

常用链接
C++ Reference
MSDN 主页
SOJ
北大OJ
并发编程网
杭电OJ

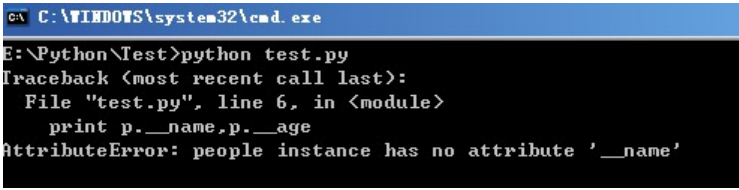
积分与排名
积分 - 423583
排名 - 286

最新评论
1. Re:Java并发编程：深入剖析Threa dLocal

```
__age = 12

p = people()
print p.__name,p.__age
```

这段程序运行会报错：



提示找不到该属性，因为私有属性是不能在类外通过对象名来进行访问的。在Python中没有像C++中public和private这些关键字来区别公有属性和私有属性，它是以属性命名方式来区分，如果在属性名前面加了2个下划线'\_\_'，则表明该属性是私有属性，否则为公有属性（方法也是一样，方法名前面加了2个下划线的话表示该方法是私有的，否则为公有的）。

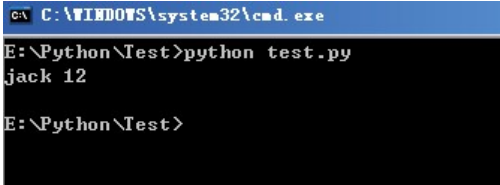
三.方法

在类中可以根据需要定义一些方法，定义方法采用def关键字，在类中定义的方法至少会有一个参数，，一般以名为'self'的变量作为该参数（用其他名称也可以），而且需要作为第一个参数。下面看个例子：

```
class people:
    __name = 'jack'
    __age = 12

    def getName(self):
        return self.__name
    def getAge(self):
        return self.__age

p = people()
print p.getName(),p.getAge()
```



如果对self不好理解的话，可以把它当做C++中类里面的this指针一样理解，就是对象自身的意思，在用某个对象调用该方法时，就将该对象作为第一个参数传递给self。

四.类中内置的方法

在Python中有一些内置的方法，这些方法命名都有比较特殊的地方（其方法名以2个下划线开始然后以2个下划线结束）。类中最常用的就是构造方法和析构方法。

构造方法\_\_init\_\_(self,...)在生成对象时调用，可以用来进行一些初始化操作，不需要显示去调用，系统会默认去执行。构造方法支持重载，如果用户自己没有重新定义构造方法，系统就自动执行默认的构造方法。

析构方法\_\_del\_\_(self)在释放对象时调用，支持重载，可以在里面进行一些释放资源的操作，不需要显示调用。

还有其他的一些内置方法：

比如 \_\_cmp\_\_( ), \_\_len\_\_( )等，具体的用法可以参考这篇博文：

<http://www.cnblogs.com/simayixin/archive/2011/05/04/2036295.html>

因为ThreadLocal在每个线程中对该变量会创建一个副本，即每个线程内部都会有一个该变量，且在线程内部任何地方都可以使用，线程之间互不影响，这样一来就不存在线程安全问题，也不会严重影响程序执行性能.....

--晴雪夏天

2. Re:Java：类与继承

从测试结果来看，“动态绑定”应该只是针对方法，而不是针对属性

--漂泊一剑客

3. Re:浅谈Java中的equals和==

一定不能用==运算符检测两个字符串是否相等！这个运算符只能确定两个字符串是否放在同一位置上。如果虚拟机始终将相同的字符串共享，就可以使用==运算符检测是否相等。但是实际上只有字符串常量是共享的，+.....

--sqy123

阅读排行榜

1. Java并发编程：线程池的使用(194 955)

2. 深入理解Java的接口和抽象类(176 673)

3. Dijkstra算法（单源最短路径）(15 7557)

4. 浅析Java中的final关键字(15467 5)

5. 将Sublime Text 2搭建成一个好用的IDE(139194)

6. 二叉树的非递归遍历(126120)

7. Java并发编程：Callable、Future和FutureTask(110703)

8. 堆排序(103407)

9. Java ConcurrentModificationException异常原因和解决方法(98302)

10. Java中的static关键字解析(9633 3)

五.类属性、实例属性、类方法、实例方法以及静态方法

在了解了类基本的东西之后，下面看一下python中这几个概念的区别。

先来谈一下类属性和实例属性

在前面的例子中我们接触到的就是类属性，顾名思义，类属性就是类对象所拥有的属性，它被所有类对象的实例对象所共有，在内存中只存在一个副本，这个和C++中类的静态成员变量有点类似。对于公有的类属性，在类外可以通过类对象和实例对象访问。

```
class people:
    name = 'jack' #公有的类属性
    __age = 12    #私有的类属性

p = people()

print p.name      #正确
print people.name #正确
print p.__age     #错误，不能在类外通过实例对象访问私有的类属性
print people.__age #错误，不能在类外通过类对象访问私有的类属性
```

实例属性是不需要在类中显示定义的，比如：

```
class people:
    name = 'jack'

p = people()
p.age = 12
print p.name #正确
print p.age  #正确

print people.name #正确
print people.age  #错误
```

在类外对类对象people进行实例化之后，产生了一个实例对象p，然后p.age = 12这句给p添加了一个实例属性age，赋值为12。这个实例属性是实例对象p所特有的，注意，类对象people并不拥有它（所以不能通过类对象来访问这个age属性）。当然还可以在实例化对象的时候给age赋值。

```
class people:
    name = 'jack'

    #__init__()是内置的构造方法，在实例化对象时自动调用
    def __init__(self,age):
        self.age = age

p = people(12)
print p.name #正确
print p.age  #正确

print people.name #正确
print people.age  #错误
```

如果需要在类外修改类属性，必须通过类对象去引用然后进行修改。如果通过实例对象去引用，会产生一个同名的实例属性，这种方式修改的是实例属性，不会影响到类属性，并且之后如果通过实例对象去引用该名称的属性，实例属性会强制屏蔽掉类属性，即引用的是实例属性，除非删除了该实例属性。

```
class people:
    country = 'china'
```

```

print people.country
p = people()
print p.country
p.country = 'japan'
print p.country #实例属性会屏蔽掉同名的类属性
print people.country
del p.country #删除实例属性
print p.country

```



```

C:\WINDOWS\system32\cmd.exe - PAUSE
china
china
japan
china
china
请按任意键继续. . .

```

下面来看一下类方法、实例方法和静态方法的区别。

类方法：是类对象所拥有的方法，需要用修饰器"@classmethod"来标识其为类方法，对于类方法，第一个参数必须是类对象，一般以"cls"作为第一个参数（当然可以用其他名称的变量作为其第一个参数，但是大部分人都习惯以'cls'作为第一个参数的名字，就最好用'cls'了），能够通过实例对象和类对象去访问。



```

class people:
    country = 'china'

    #类方法，用classmethod来进行修饰
    @classmethod
    def getCountry(cls):
        return cls.country

p = people()
print p.getCountry() #可以用过实例对象引用
print people.getCountry() #可以通过类对象引用

```



类方法还有一个用途就是可以对类属性进行修改：



```

class people:
    country = 'china'

    #类方法，用classmethod来进行修饰
    @classmethod
    def getCountry(cls):
        return cls.country

    @classmethod
    def setCountry(cls, country):
        cls.country = country

p = people()
print p.getCountry() #可以用过实例对象引用
print people.getCountry() #可以通过类对象引用

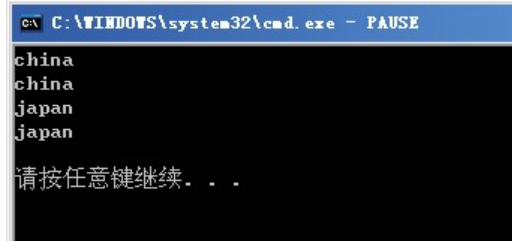
p.setCountry('japan')

print p.getCountry()
print people.getCountry()

```



运行结果：



结果显示在用类方法对类属性修改之后，通过类对象和实例对象访问都发生了改变。

实例方法：在类中最常定义的成员方法，它至少有一个参数并且必须以实例对象作为其第一个参数，一般以名为'self'的变量作为第一个参数（当然可以以其他名称的变量作为第一个参数）。在类外实例方法只能通过实例对象去调用，不能通过其他方式去调用。

```
class people:
    country = 'china'

    #实例方法
    def getCountry(self):
        return self.country

p = people()
print p.getCountry()      #正确，可以用过实例对象引用
print people.getCountry() #错误，不能通过类对象引用实例方法
```

静态方法：需要通过修饰器"@staticmethod"来进行修饰，静态方法不需要多定义参数。

```
class people:
    country = 'china'

    @staticmethod
    #静态方法
    def getCountry():
        return people.country

print people.getCountry()
```

对于类属性和实例属性，如果在类方法中引用某个属性，该属性必定是类属性，而如果在实例方法中引用某个属性（不作更改），并且存在同名的类属性，此时若实例对象有该名称的实例属性，则实例属性会屏蔽类属性，即引用的是实例属性，若实例对象没有该名称的实例属性，则引用的是类属性；如果在实例方法更改某个属性，并且存在同名的类属性，此时若实例对象有该名称的实例属性，则修改的是实例属性，若实例对象没有该名称的实例属性，则会创建一个同名称的实例属性。想要修改类属性，如果在类外，可以通过类对象修改，如果在类里面，只有在类方法中进行修改。

从类方法和实例方法以及静态方法的定义形式就可以看出来，类方法的第一个参数是类对象cls，那么通过cls引用的必定是类对象的属性和方法；而实例方法的第一个参数是实例对象self，那么通过self引用的可能是类属性、也有可能是实例属性（这个需要具体分析），不过在存在相同名称的类属性和实例属性的情况下，实例属性优先级更高。静态方法中不需要额外定义参数，因此在静态方法中引用类属性的话，必须通过类对象来引用。

关于面向对象编程暂时就讲这么多了，其他关于类的继承和方法重载这些内容将在后面继续讲解。

作者：[海子](#)

出处：<http://www.cnblogs.com/dolphin0520/>

本博客中未标明转载的文章归作者[海子](#)和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

分类: [Python/Web框架](#)

标签: [Python](#)

好文要顶

关注我

收藏该文



[海子](#)

[关注 - 6](#)

[粉丝 - 3750](#)

[+加关注](#)

16

0

« 上一篇: [关于raw\\_input\(\)和sys.stdin.readline\(\)的区别](#)

» 下一篇: [Python面向对象编程（二）](#)

posted @ 2013-03-29 16:59 海子 阅读(46153) 评论(12) 编辑 收藏

## 评论列表

#1楼 2013-03-29 19:03 Highflyer

请楼主条件一款 Python IDE.

支持(0) 反对(0)

#2楼[楼主 ] 2013-03-29 19:10 海子

@ Highflyer

引用

请楼主条件一款 Python IDE.

我自己在用的是Pyscripter和Notepad++，写一些小脚本比较方便，如果做工程的话推荐用Pycharm。

支持(0) 反对(0)

#3楼 2013-03-29 21:17 多多晖

博主应该谈谈类方法、实例方法和静态方法的适用情况，具体作用

支持(0) 反对(0)

#4楼[楼主 ] 2013-03-31 17:09 海子

@ 多多晖

引用

博主应该谈谈类方法、实例方法和静态方法的适用情况，具体作用

呵呵，目前只是讲解一些基础知识，这些可以在后面慢慢给出来。

支持(0) 反对(0)

#5楼 2013-04-15 12:24 单曲尊环

深入浅出，写的很好，我也觉得应该谈谈使用场景

支持(1) 反对(0)

#6楼 2013-11-24 14:38 Mr.eleven

实例属性是不需要在类中显示定义的，类属性也不需要类中显示定义。

支持(0) 反对(0)

#7楼 2014-03-20 14:50 dingkong

@ Mr.eleven  
就是 感觉好混乱

支持(0) 反对(0)

#8楼 2015-02-06 20:20 夏目家的猫

引用  
如果对self不好理解的话，可以把它当做C++中类里面的this指针一样理解，就是对象自身的意思  
self只是指向了当前实例，是一个引用，不是对象本身吧。

支持(0) 反对(0)

#9楼 2015-05-22 09:31 温柔的鲨鱼

Python的继承就这么点么????????????????  
Python面向对象就这么点??????????????

支持(0) 反对(0)

#10楼 2017-02-17 11:16 小5\_5

写得简明易懂

支持(0) 反对(0)

#11楼 2017-02-18 15:06 风\_feng

看后顿时对这些感念有了清晰的认识。为了点赞，留言，专门注册了账号。为博主的分享点赞。  
支持(0) 反对(0)

#12楼 2017-03-16 16:09 一out

写的简明易懂 博主多多写点

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库
- 【推荐】中铁、中石油等大型企业的复杂报表解决方案
- 【福利】阿里云免费套餐升级，更多产品，更久时长



最新IT新闻:

- 手机号为何是11位？到底该怎么读？这是答案
- 新建快捷方式：唤出Windows 10上的UWP版资源管理器
- Debian GNU/Linux 8.8正式发布：90个安全更新+68个Bug修复
- 科学家成功治愈一只艾滋病活体白鼠
- 席勒谈Swift语言：它的发展远远超出预期
- » 更多新闻...



最新知识库文章:

- 唱吧DevOps的落地，微服务CI/CD的范本技术解读
- 程序员，如何从平庸走向理想？
- 我为什么鼓励工程师写blog
- 怎么轻松学习JavaScript
- 如何打好前端游击战
- » 更多知识库文章...

---

Copyright ©2017 海子