

# 傲娇的草履虫

博客园

首页

新随笔

联系

订阅

管理

随笔 - 89 文章 - 2 评论 - 10

## Python单例模式的四种方法

在这之前，先了解super()和\_\_new\_\_()方法

super()方法：

返回一个父类或兄弟类类型的代理对象，让你能够调用一些从继承过来的方法。

它有两个典型作用：

- a. 在单继承的类层次结构中，super()可用于引用父类而不显式父类名称，从而使代码更易于维护。
- b. 在多重继承中，可以保证公共父类仅被执行一次。

\_\_new\_\_方法：

- a.它是一个类级别的静态方法。通常用于控制生成一个新实例的过程。
- b.返回的是一个实例化出来的实例

下面为四种实现单例模式的方法

### 1. 使用\_\_new\_\_方法



```
class Singleton(object):
    def __new__(cls, *args, **kwargs):
        if not hasattr(cls, '_instance'):
            orig = super(Singleton, cls)
            cls._instance = orig.__new__(cls, *args, **kwargs)
        return cls._instance

class A(Singleton):
    pass

# 类A即为单例类
```



### 2. 共享属性



```
# 创建实例时把所有实例的__dict__指向同一个字典,这样它们都具有相同的属性和方法 (类的__dict__存储对象属性)
class Singleton(object):
    _state = {}
    def __new__(cls, *args, **kwargs):
        ob = super(Singleton,cls).__new__(cls, *args, **kwargs)
        ob.__dict__ = cls._state
        return ob

class B(Singleton):
    pass

# 类B即为单例类
```



## 公告

昵称：傲娇的草履虫  
园龄：2年11个月  
粉丝：16  
关注：10  
[+加关注](#)

2020年9月						
日	一	二	三	四	五	六
30	31	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	1	2	3
4	5	6	7	8	9	10

## 搜索

## 常用链接

[我的随笔](#)  
[我的评论](#)  
[我的参与](#)  
[最新评论](#)  
[我的标签](#)


## 我的标签

[Python\(33\)](#)  
[Redis\(9\)](#)  
[MySQL\(7\)](#)  
[Shell\(7\)](#)  
[Others\(6\)](#)  
[Django\(6\)](#)  
[Linux\(3\)](#)  
[GIT\(3\)](#)  
[Golang\(3\)](#)  
[Java\(3\)](#)  
[更多](#)

## 随笔档案

[2020年5月\(1\)](#)  
[2020年4月\(4\)](#)  
[2019年12月\(1\)](#)  
[2019年8月\(2\)](#)  
[2019年7月\(2\)](#)  
[2019年6月\(1\)](#)


3.使用装饰器



```
def singleton(cls):
    instance = {}
    def wapper():
        if cls not in instance:
            instance[cls] = cls(*args, **kwargs)
        return instance[cls]
    return wapper

@singleton
class C:
    pass

# 类C即为单例类
```



4.import方法



```
# 作为Python模块时是天然的单例模式

#创建一个sington.py文件，内容如下：
class Singleton(object):
    def foo(self):
        pass

mysingleton = Singleton()

# 运用
from sington import mysingleton
mysingleton.foo()
```



标签: Python

好文要顶

关注我

收藏该文





傲娇的草履虫

关注 - 10

粉丝 - 16

+加关注

« 上一篇: Python虚拟环境virtualenv的使用  
» 下一篇: MySQL数据库引擎MyISAM与InnoDB的区别  
posted @ 2018-07-18 10:44 傲娇的草履虫 阅读(5554) 评论(2) 编辑 收藏

评论列表

#1楼

2019-05-10 19:46

潇洒坤

写得很好，一下就看懂了。

支持(0) 反对(0)

回复 引用

#2楼

2019-08-13 19:41

凯帅

老铁写的不错，铺盖仔

支持(0) 反对(0)

回复 引用

发表评论

- 2019年5月(3)
- 2019年4月(2)
- 2019年3月(2)
- 2019年2月(1)
- 2019年1月(1)
- 2018年12月(4)
- 2018年11月(10)
- 2018年10月(4)
- 2018年9月(8)
- 2018年8月(6)
- 2018年7月(15)
- 2018年6月(6)
- 2018年5月(1)
- 2018年3月(3)
- 2018年1月(3)
- 2017年12月(2)
- 2017年11月(4)
- 2017年10月(3)

相册

delav(1)

最新评论

1. Re:Python替换字符串中的反斜杠  
@Leoaxis  
  
怎么解决的，兄弟  
  
--百川东流水
2. Re:Shell脚本重启Python程序  
我在： 2020年 4月 29日 16:8:8 看过本篇博客！  
  
--努力变胖-HWP
3. Re:什么是 DevOps ?  
写的不错，点个赞  
  
--zzh1995
4. Re:Python替换字符串中的反斜杠  
@ 傲娇的草履虫好的谢谢，已经解决了。...  
--Leoaxis
5. Re:Python替换字符串中的反斜杠  
@ Leoaxis如果能手动改，就直接在字符串的前面加个 r，让它变为原始字符串，像这样：s = r'cdp\fd'result = s.replace('\f','@')print(resul...  
--傲娇的草履虫

阅读排行榜

1. git 更新本地代码(38774)
2. 超详细：Python(wordcloud+jieba)生成中文词云图(23996)
3. Python替换字符串中的反斜杠(18726)
4. Python os模块常用函数详解(6887)
5. Git无法删除文件问题：fatal: pathspec 'readme.txt' did not match any files(6833)

评论排行榜

1. Python替换字符串中的反斜杠(4)
2. Python单例模式的四种方法(2)
3. 什么是 DevOps ?(1)
4. Shell脚本重启Python程序(1)

推荐排行榜

编辑预览

B

支持 Markdown

- 1. 超详细：Python(wordcloud+jieba)生成中文词云图(6)
- 2. 什么是 DevOps ?(2)
- 3. 关于 [lambda x: x\*i for i in range(4)] 理解(2)
- 4. git 更新本地代码(1)
- 5. Python单例模式的四种方法(1)

提交评论退出 订阅评论

[Ctrl+Enter快捷键提交]