

Python_property的用法

1.属性可以拦截对普通成员变量的访问和设置：

示例：

```
class TestClass(object):
```

```
    def __init__(self,age):
        self._age = age
```

`@property` #用`property`装饰器来指明该方法表示属性，随后该方法会被隐藏，下次再调用`TestClass().age()`会出错，因为`age`已经是属性了

```
    def age(self):
        return self._age
```

`@age.setter` #用`age.setter`装饰器来拦截对属性的设置，可以在这里加上对属性的设置逻辑

```
    def age(self,val):
        if val > 100:
            raise Exception("Too old")
        self._age = val
```

`@age.getter` #用`age.getter`装饰器来拦截对属性的访问，可以在这里加上对属性的访问控制逻辑(一般不需要)

```
    def age(self):
        return self._age
```

```
if __name__ == "__main__":
```

```
    test = TestClass(20)
```

```
    test.age = 150 #调用属性赋值方法
```

```
    print("age:{}".format(test.age)) #调用属性访问方法
```