python进阶之类常用魔法方法和魔法属性 - 天宇之游 - 博客园

cnblogs.com/cwp-bg/p/9856291.html

前言

前面我们总结过了python的关键字、运算符、内置函数、语法糖等与python魔法方法之间的关系，现在我们更细一点，看看python的面向对象编程有哪些常用的魔法属性和魔法方法。

魔法属性

对于一个类，python定义了许多可用的魔法属性，有些每个类都默认存在，有些需要用户手动定义。

__dict__

__dict__属性可以说是一个类最常用的属性之一了，它又分为类的__dict__属性和实例的__dict__属性。

```
class Person(object):
    eye = 2
    hand = 2
    def __init__(self, name):
        self.name = name

    def run(self):
        print('run')

    @classmethod
    def eat(cls):
        print('eat')

if __name__ == "__main__":
    person = Person('cai')
    print(Person.__dict__)
    print(person.__dict__)
```

1. 类的__dict__属性存储了类定义的所有类属性、类方法等组成的键值对，但不包括继承而来的属性和方法

2. 实例的__dict__属性存储了所有的实例属性的键值对，如果没有就为空；__init__方法其实就是对__dict__属性的初始化赋值；

__doc__

该属性记录了类的说明文档，用类和实例引用指向的都是类的__doc__属性,如果没有默认为None。

```
class Person(object):
    """person"""
    pass

per = Person()
print(per.__doc__)
```

__module__

该属性记录类定义的位置，如果定义的位置正好是主程序，那么该值为"_main_",否则是类属于的模块的名字；

```
class Person(object):
    """person"""
    pass

per = Person()
print(per.__module__)
```

__class__

该属性指向该实例的类，即实例指向类对象，类对象指向元类；

```
class Person(object):
    """person"""
    pass

per = Person()
print(per.__class__)
print(per.__class__())
print(Person.__class__)
```

__slots__

该属性起到限制动态绑定属性和方法的作用，该属性是一个元组，默认是不存在的，需要手动定义并且只对当前的类起作用，只有添加到元组中的名字才能被动态添加属性，否则报错！

```
class Person(object):
    __slots__ = ('name','age','run')

    def __init__(self):
        self.height = 100

def run(self):
    print('run')

if __name__ == "__main__":
    from types import MethodType
    person = Person()
    person.name = 'cai'
    person.run = MethodType(run,person)
    person.run()
```

1. **__slots__属性定义好后，限制了一个类的实例的属性以及可以动态添加的属性和方法；**
2. **__slots__属性定义好后，不得在类中定义元组中已有的同名的方法；**

魔法方法

__new__

该方法是类创建实例调用的第一个方法，返回一个实例；这是一个实例从无到有必须调用的方法，在单例模式中常用，其他不常用。

```
class Person(object):
    def __new__(cls, *args, **kwargs):
        print(args)
        return object.__new__(cls)

if __name__ == "__main__":
    person = Person('cai')
```

创建实例时会将参数传入new方法，但new方法中无法更改参数。

__init__

该方法可以说是类最常用的方法了，python在调用new方法后会紧接着调用init方法,我们将实例的一些初始化操作放在该方法中，即对__dict__属性进行操作；

```
class Person(object):

    def __init__(self, name):
        self.name = name
    def __setattr__(self, key, value):
        print(key,value)
        super().__setattr__(key,value)
if __name__ == "__main__":
    person = Person('cai')
    print(person.__dict__)
```

- 所有的“self.name = name”这种语法糖，python会先调用setattr魔法方法，该魔法方法对__dict__属性中添加键值对；
- self一定是__new__方法的返回值，如果返回的是1，那么self就是1.

__del__

该方法在实例对象引用计数变为0或del关键字调用的时候触发执行。

__repr__

该方法在print()调用或repr()调用时执行，用来定义对类的信息的描述，每个类都应该定义这个方法。

总结

1. 类的常用魔法属性有__dict__,__doc__,__mould__,__slots__,其中slots属性需要自定义，其他属性默认存在；
2. 构造类常用init,new,del方法，它们在类创造、初始化、销毁时触发；

参考

- https://blog.csdn.net/u012332571/article/details/70141438
- https://docs.python.org/3/