

## python\_类里的方法及变量

在类中主要搞清楚几个概念：类变量，实例变量，**self**，实例方法，类方法，静态方法，属性方法。

### 1、类变量

在类里定义的变量，在类中使用，不需要实例化，通过**self.xx**就可使用，实例对象可通过对象**.xx**进行使用。

```
class Person(): #定义类

    name='zwj' #

    def info(self):

        print(self.name) #本类中使用通过self.name

    def run(self):

        print("我会跑")

    def fly(self):

        print("我会飞")

    def walk(self):

        print("我会走")

zwj=Person()

print(zwj.name) #实例化对象，通过对象.name进行访问
```

### 2、实例变量

如果实例变量写在构造函数中，不需要手动实例，即可使用，如果在其他类方法中，需要先调用类方法，声明该实例变量，才能使用

```
class Person(): #定义类

    def __init__(self):

        self.name='xiaoniao' #实例变量在析构函数方法中

    def info(self):

        print(self.name) #本类中可直接使用

    def run(self):

        self.time=5 #实例变量在实例方法中

        print("我会跑")

    def fly(self):

        print(self.time) #本类中直接使用

        print("我会飞")

    def walk(self):

        print("我会走")


zwj=Person()

print(zwj.name) #实例化对象，通过对象.name进行访问

zwj.run()

print(zwj.time) #调用这个变量之前需要先运行run方法进行声明该变量才可以
```

### 3、self的含义

self就是本类对象的意思

### 4、实例方法

实例方法的是类中最常用的方法，方法本身带self参数，非本类中实例化后才能使用该方法，本类中调用实例方法通过self.xx()

```
class Person(): #定义类

    def info(self): #实例方法

        print('ddd')

    def run(self):

        self.info() #本类中调用实例方法

        print("我会跑")


zwj=Person() #实例化对象

zwj.run() #实例化后才可使用

zwj.info() #实例化后才可使用
```

### 5、类方法

注意几点：

- 类方法需要加@classmethod装饰器
- 类方法无需实例化就可调用
- 在类中类方法调用实例方法和实例变量，需要先实例化类对象，然后才能调用
- 在类中类方法调用类变量，可直接通过cls.xx的方式直接调用

```
class Person(): #定义类
```

```
    ino='xxx'
```

```
    def info(self): #实例方法
```

```
        self.name='xiaoming'
```

```
        print('ddd')
```

```
    def run(self):
```

```
        print("我会跑")
```

```
@classmethod #类方法
```

```
def shuoming(cls):
```

```
    print(cls.ino) #类方法可直接使用类变量
```

```
    b=cls() #cls本类的意思，这句话相当于b=Person()
```

```
    b.run() #类方法使用实例方法需要先实例化才可调用
```

```
    print(b.name) #类方法使用类实例变量，需要先实例化才可调用
```

```
Person.shuoming() #类方法无需实例化就可调用
```

## 6、静态方法

需注意几点：

- 静态方法需要加@staticmethod装饰器
- 静态方法不能调用类变量，类方法，实例变量及实例方法，但是实例方法可以调用静态方法，通过self.XX() ,类方法调用静态方法也需要实例化对象后才能使用
- 静态方法调用类变量，类方法，实例变量及实例方法，需要实例化对象，然后通过对象去调取
- 外部调用静态方法，需实例化对象

```
class Person(): #定义类
```

```
    ino='xxx'
```

```
    def info(self): #实例方法
```

```
        self.name='xiaoming'
```

```
        print('ddd')
```

```
        self.stact_method() #静态方法可通过self.XX()进行类中使用
```

```
@classmethod #类方法
```

```
def shuoming(cls):
```

```
    b = cls() # cls本类的意思，这句话相当于b=Person()
```

```
    b.stact_method()
```

```
@staticmethod #静态方法
```

```
def stact_method():
```

```
    X=Person() #类中的类变量，类方法，实例变量及实例方法均不可调用，需要实例化对象后才能使用
```

```
    print(X.ino)
```

```
zwj=Person()
```

```
zwj.stact_method() #类静态方法实例化对象后可调用
```

## 7、属性方法

类属性方法和类实例方法相似，唯一的区别是类属性方法无传参，实例化对象调用的时候通过调取属性的方式进行调用

```
class Person(): #定义类

    ino='xxx'

    def info(self): #实例方法

        self.name='xiaoming'

        print("shi li fang fa")

    @classmethod # 类方法

    def shuoming(cls):

        print("lei fang fa")

    @staticmethod #静态方法

    def stact_method():

        print("jing tai fang fa ")

    @property

    def pro_mothod(self):

        print("kkk")

        print(self.name) #调用实例变量

        self.stact_method() #调用静态方法

        self.shuoming() #调用类方法

        print(self.ino) #调用类变量

        self.info() #调用实例方法

zwj=Person()

zwj.info()

zwj.pro_mothod #通过调用属性的方式，调用类属性方法
```

## 8、类构造函数

构造函数，实例化就自动调用该方法，一般该方法会放一些属性或者初始化的东西



```
class Person(): #定义类
```

```
    def __init__(self): #构造函数，在类实例化后会自动执行该方法
```

```
        print("实例化我就会自动运行")
```

```
    def info(self): #实例方法
```

```
        self.name='xiaoming'
```

```
        print("shi li fang fa")
```

```
zwj=Person() #实例化对象，不调取任何函数
```

```
C:\Users\zhaow\AppData\Local\Programs\Python\Python37\python.exe D:/study/python/test/yc.py
```

```
实例化我就会自动运行
```

## 9、类析构函数

析构函数，类实例被销毁的时候自动执行

```
class Person(): #定义类

    def info(self): #实例方法

        self.name='xiaoming'

        print("shi li fang fa")

    def __del__(self): #析构函数，类销毁自动执行

        print("类销毁的时候就执行")
```

```
zwj=Person()

zwj.info()

del zwj #销毁类

print("=====")
```

C:\Users\zhaow\AppData\Local\Programs\Python\Python37\python.exe D:/study/python/test/yc.py

```
shi li fang fa

类销毁的时候就执行

=====
```