PYTHON
编程基础

类方法和静态方法

类方法

01

类方法是指使用@classmethod修饰的方法，其第一个参数是类本身（而不是类的实例对象）。

02

类方法的特点是既可以通过类名直接调用，也可以通过类的实例对象调用。

类方法

```
1       class Complex: #定义Complex类
2           def __init__(self,real=0,image=0): #定义构造方法
3               self.real=real #初始化一个复数的实部值
4               self.image=image #初始化一个复数的虚部值
5           @classmethod
6           def add(cls,c1,c2): #定义类方法add，实现两个复数的加法运算
7               print(cls) #输出cls
8               c=Complex() #创建Complex类对象c
9               c.real=c1.real+c2.real #实部相加
10              c.image=c1.image+c2.image #虚部相加
11              return c
```

类方法

```
12      if __name__=='__main__':
13          c1=Complex(1,2.5)
14          c2=Complex(2.2,3.1)
15          c=Complex.add(c1,c2) #直接使用类名调用类方法add
16          print('c1+c2的结果为%.2f+%.2fi'%(c.real,c.image))
```


```
<class '__main__.Complex'>
c1+c2的结果为3.20+5.60i
```

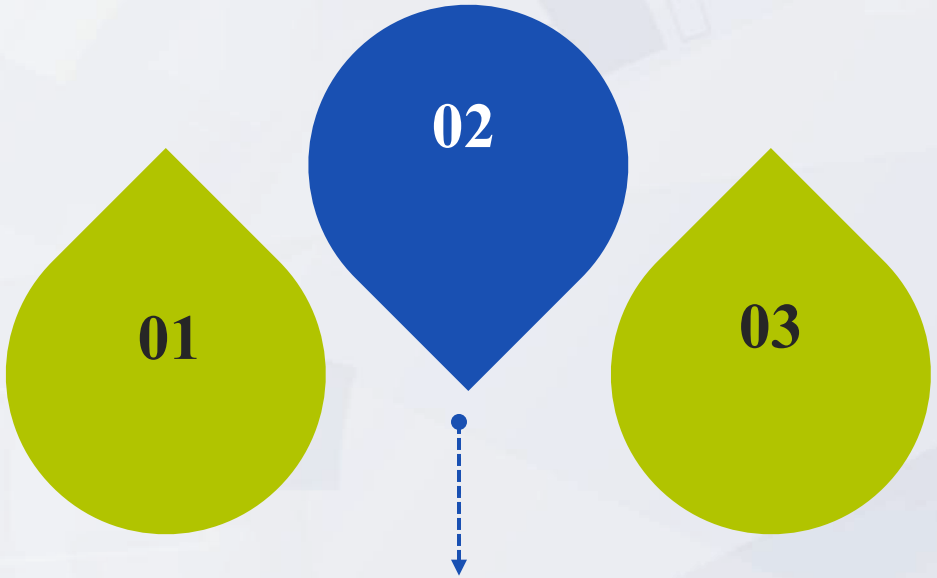
类方法

提示：

将第15行的“c=Complex.add(c1,c2)”改为“c=c1.add(c1, c2)”或“c=c2.add(c1, c2)”或“c=Complex().add(c1, c2)”，程序运行后可得到相同的输出结果，即类方法也可以使用实例对象调用。

通过“print(cls)”输出类方法add的第一个参数，从输出结果中可以看到cls是Complex类。

静态方法



静态方法是指使用@staticmethod修饰的方法。

与类方法相同，静态方法既可以直接通过类名调用，也可以通过类的实例对象调用。

与类方法不同的地方在于，静态方法中没有类方法中的第一个类参数。

静态方法

```
1       class Complex: #定义Complex类
2           def __init__(self,real=0,image=0): #定义构造方法
3               self.real=real #初始化一个复数的实部值
4               self.image=image #初始化一个复数的虚部值
5           @staticmethod
6           def add(c1,c2): #定义类方法add，实现两个复数的加法运算
7               c=Complex() #创建Complex类对象c
8               c.real=c1.real+c2.real #实部相加
9               c.image=c1.image+c2.image #虚部相加
10              return c
```

静态方法

```
11      if __name__=='__main__':
12          c1=Complex(1,2.5)
13          c2=Complex(2.2,3.1)
14          c=Complex.add(c1,c2) #直接使用类名调用类方法add
15          print('c1+c2的结果为%.2f+%.2fi'%(c.real,c.image))
```

c1+c2的结果为3.20+5.60i