

PYTHON

编程基础

类中普通方法 定义及调用

概述

- 类中的方法实际上就是执行某种数据处理功能的函数。
- 与普通函数定义一样，类中的方法在定义时也需要使用def 关键字。
- 类中的方法分为两类：普通方法和内置方法。
 - 普通方法需要通过类的实例对象根据方法名调用；
 - 内置方法是在特定情况下由系统自动执行。

普通方法定义和调用

在定义类的普通方法时，要求第一个参数需要对应调用方法时所使用的实例对象（一般命名为`self`，但也可以改为其他名字）。

当使用一个实例对象调用类的普通方法时，其语法格式为：

实例对象名.方法名(实参列表)

提示：在通过类的实例对象调用类中的普通方法时，并不需要传入`self`参数的值，`self`会自动对应调用该方法时所使用的对象。

普通方法定义和调用



例：类中普通方法定义及调用示例

```
1      class Student: #定义Student类
2          name='Unknown' #定义Student类中有一个name属性
3          def SetName(self, newname): #定义类的普通方法SetName
4              self.name=newname #将self对应实例对象中的name属性值赋为newname
5          def PrintName(self): #定义类的普通方法PrintName
6              print('姓名：%s'%self.name) #输出self对应实例对象中的name属性值
7      if __name__=='__main__':
8          stu1=Student() #定义Student类对象stu1
9          stu2=Student() #定义Student类对象stu2
10         stu1.SetName('李晓明') #通过stu1对象调用SetName方法
11         stu2.SetName('马红') #通过stu2对象调用SetName方法
12         stu1.PrintName() #通过stu1对象调用PrintName方法
13         stu2.PrintName() #通过stu2对象调用PrintName方法
```

普通方法定义和调用



提示

类的普通方法必须通过实例对象调用，而不能通过类名直接调用。

例如，如果运行

```
Student.SetName('未知')  
Student.PrintName()
```

系统会分别给出
如下报错信息

```
TypeError: SetName() missing 1 required  
positional argument: 'newname'  
TypeError: PrintName() missing 1 required  
positional argument: 'self'
```

即都缺少了一个
位置参数

这是因为通过实例对象调用时会自动将该实例对象传给self，而通过类调用时则不会有这个隐含的参数传递。