第九课: Python 创建类

魏守科 博士, 教授

目录页 CONTENTS PAGE







类和对象

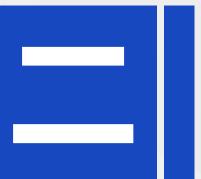
- 对象

类和对象

Python 是一种面向对象的编程语言。

Python 中的几乎所有东西都是对象,拥有**属性和方法**。

类(Class)类似对象构造函数,或者是用于创建对象的"蓝图"。一类:拥有相同**属性和方法**的不同对象



创建类

- 创建类
- 创建对象
- 创建实例
- 执行类

创建类

class person(): pass

创建对象

__init__() 函数:

- 所有类都有一个名为 __init__() 的函数,它始终在启动 类时执行。
- 使用 __init__() 函数将值赋给对象属性,或者在创建对 象时需要执行的其他操作:

class person():
 def __init__(name, gender, age, origin, phone):
 pass

创建对象

self 参数

- self 参数是对类的当前实例的引用,用于访问属于该类的变量。
- 它不必被命名为 self, 您可以随意调用它, 但它必须是 类中任意函数的首个参数:

class person(): def __init__(self, name, gender, age, origin, phone): self.name = name self.gender = gender self.age = age self.origin = origin self.phone = phone

创建对象

定义更函数:

```
class person():
  def init (self, name, gender, age, origin, phone):
    self.name = name
    self.gender = gender
    self.age = age
    self.origin = origin
    self.phone = phone
  def info func(self):
     print(f'这是 {self.name}, 性别: {self.gender}, 年龄: {self.age} years old.')
     if self.gender == '男':
       print(f'他来自 {self.origin},他的电话为 {self.phone}.')
     else:
       print(f'她来自{self.origin}, 她的电话为{self.phone}.')
```

创建实例

实例 (instance) : 真实的对象

person1 = person('小张', '女', 23, '北京', 1525551888) person2 = person('小李', '男', 21, '山东', 1572221666)

执行类

```
#访问实例的属性
person1.name
print(person1.name)
print(person2.age)
#访问实例的方法
person2.info_func()
```



类继承

- 类的继承
- 创建父类
- 创建子类
- 创建实例
- 运行子类

类的继承

类继承: 创建一个类,它可以使用现有类的所有功能,并在无需重新编

写原来的类的情况下对这些功能进行扩展。

子类: 通过继承创建的新类称为"子类"或"派生类"

父类:被继承的类称为"父类","基类"、或"超类"

创建父类

```
class person():
  def init (self, name, gender, age, origin, phone):
    self.name = name
    self.gender = gender
    self.age = age
    self.origin = origin
    self.phone = phone
  def info func(self):
     print(f'这是 {self.name}, 性别: {self.gender}, 年龄: {self.age} years old.')
     if self.gender == '男':
        print(f'他来自 {self.origin},他的电话为 {self.phone}.')
     else:
        print(f'她来自{self.origin}, 她的电话为{self.phone}.')
```

创建子类

注意: 那就是两个类之间的关系应该是"属于"关系。例如,students、teachers、都是人,因此这两个类都可以继承 person 类。

class student(person): pass

pass: 我们不想加入任何属性和方法

一一创建实例

student1 = student('小明', '男', 21, '上海', 557889999)

运行实例

print(student1.phone)

student1.info_func()

总结



展示创建类的过程, 定义类、创建对象、创建实例、执行类

学习了类继承,有一个父类,创建子类、创建 子类实例、执行子类

