python-类

##面向对象

在编程语言中，我们将变量看成数据，它用来存储多种形式的值；我们将函数看成操作，它用来对数据进行某些处理。所有的代码都由数据和操作构
行的本质就是对数据进行各种操作。

类只是一张图纸，起到说明的作用，不占用空间内存；对象才是具体的零件，要有地方来存放，才会占用内存空间。

类

python中，**首字母大写的名称指的是类。**

1，方法—init—()，类中的函数称为**方法**

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

p1 = Person("小明", 20)

print(p1.name)
print(p1.age)

```

调用—init—()方法来创建类时，将自动传入实参self。每个与类相关联的方法调用都自动传递实参self，**它是一个指向实例本身本身的引用**

self参数是对类当前实例的引用，用于访问属于该类的变量。他不必命名self，你可以随意调用它，但它必须是类中任何函数的第一个参数

修改属性的值

1，直接修改

2，通过方法修改属性的值

替换

```
class Person:
  def __init__(mysillyobject, name, age):
    mysillyobject.name = name
    mysillyobject.age = age

  def myfunc(abc):
    print("我的名字是 " + abc.name)

p1 = Person("小明", 20)
p1.age = 21
print(p1.age)

```

3，删除对象属性

即调用del函数，通过句点表示法来删除对象

```
class Person:
  def __init__(mysillyobject, name, age):
    mysillyobject.name = name
    mysillyobject.age = age

  def myfunc(abc):
    print("我的名字是 " + abc.name)

p1 = Person("小明", 20)
del p1.age
print(p1.age)#没有了自然打印报错

```

删除对象

通过del删除对象

pass语句

class定义不能为空，如果由于某种原因class中没有内容的定义，则放入pass语句来避免出错。

其实，类属性是指包含在类中的变量，而类方法是指包含在类中的函数。也就是说，类属性和类方法分别是包含类中变量和函数的别称。

继承

继承经常用于创建和现有类功能类似的新类，又或者是在现有类基础上添加一些成员（属性和方法），通过继承，可以轻松实现类的重复使用。

父类是被继承的类，也称为基类。子类是从另一个类继承的类，也称派生类。

####创建基类

.创建子类：

创建从另一个类中继承功能的类，在创建子类时将父类作为参数发送。

```
class Student(Person):
  pass
```

现在student类具有与person类相同的属性和方法。

2，使用super()函数可以让子类继承其父类的所有方法和属性。

```
class Person:
  def __init__(self, fname, lname):
    self.firstname = fname
    self.lastname = lname

  def printname(self):
    print(self.firstname, self.lastname)

class Student(Person):
  def __init__(self, fname, lname):
    super().__init__(fname, lname)

x = Student("川川", "菜鸟")
x.printname()
```

使用super()函数，可以不用使用父类的名称，它会自动从父类中继承方法和属性。

父类重写

对于父类的方法，只要他不符合子类模拟的行为，都可以对其进行重写

```
class Bird:
    #鸟有翅膀
    def isWing(self):
        print("鸟有翅膀")
    #鸟会飞
    def fly(self):
        print("鸟会飞")

class Ostrich(Bird):
    # 重写Bird类的fly()方法
    def fly(self):
        print("鸵鸟不会飞")
```

重写，有时又称覆盖，意思是一样的。都是对类中已有的方法进行修改。

导入类

python允许将类存储在模块中，然后在主程序中导入所需的模块。

```
from car import ElectricCar
```

与导入多个函数语法类似。