1. Python既支持函数式编程也支持面向对象编程。
2. Python规定定义在类中的方法第一个参数必须有，并且是self. 调用对象方法时, Python自动传入 self参数，我们传其他参数。[**Python自动传的这个self参数就是实例化的对象，所以在类方法可以self.attribute取得当时实例化对象的属性**。]
3. 每个对象中都存有一个类对象指针，用于表明这个对象的属于哪一个类。当调用方法的时候，将使用这个指针，然后找到类中定义的方法。
4. 当多个方法使用相同的参数时候，可以考虑将其封装成类，然后参数通过类成员的方法获取。
5. 当以类（）方法的形式构建对象时候，会自动调用类中的\_\_init\_\_方法，无参构造有参构造都可以。但是\_\_init\_\_方法定义需要多少个参数，那么以类()形式构建的时候也要传入多少个参数。
6. 类定义中使用属性成员，调用属性方法时候，要以self.作为前缀，表示是调用对象中的这些方法。
7. 面向对象三大特性：封装，继承，多态。Python的多态特性并不显著，因为Python是弱类型语言。在JAVA, C#强类型语言中多态特性显著。
8. Python支持多继承，没有接口，接口在Python中使用mixin类来类似实现。

Java不支持多继承, Java的多继承使用接口来实现，Java有接口。

1. Python子类将继承父类的方法，可以考虑成本质将父类方法完全在子类中复制一份。
2. Python的多继承中，申明子类时Class C3(C1, C2) 父类从左到右，优先级由高到低。若有重名的方法，则子类，然后父类从左到右的优先级。并且若父类中也是子类时，则继续找其父类，深度遍历找除非有父类中，有其共同父类。
3. 判断类的实例是否在list中，可以直接使用 obj in list
4. Pycharm如果可以确定变量是某个类的实例时，会自动提示方法和变量。但是如果是传入方法当中的一个类引用，由于无法确定实际传入的变量是哪个类的实例，所以在编写这个方法的时候是不会提示内容的。
5. Python中判断一个对象是否存在于列表中，Python执行的操作是这个对象的指针地址是否在list当中。
6. 当一些对象被序列化放入文件中时，再反序列化取出时，他们的指针地址将会变化，Python会创建全新的一份内存空间。
7. isinstance(a,b) 查看对象a是否是类b的实例，b只要是a的类，父类，祖父类或以上的都为True

issubclass(a,b)查看 类a是否是类b的子类。

1. 子类中如果和父类有相同的方法名，想要调用父类的方法。有两种方式，第一可以使用super关键字来调用父类的方法。第二可以是用父类名.方法名（self）
2. Python自带的字典是无序字典，但是也有基于无序字典的有序字典。有序字典的设计原理在于，如果它的key通过有序的列表保存，并且输出的时候按照列表的顺序，那么这个字典便是有序的。有序字典一般设计逻辑是，首先继承Python原生字典，然后在对字典就行初始化(\_\_init\_\_) ，添加键值对(\_\_setitem\_\_), 删除键值对(\_\_delitem), 输出字典(\_\_str\_\_) 的时候，在调用原字典即是父类的方法时，再执行一些自定义的代码（例如将key放入另外建立的有序列表中，在\_\_str\_\_的从有序列表中逐一取出）。