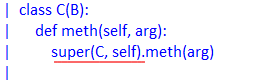
### Super用法

类的三大特点：封装、继承、多态

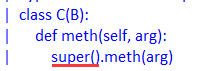
而当我们写派生类时想要在继承基类方法的基础上又想要增加自己的特性时，super就派上用场了，它让类既实现继承又实现了多态。

super是一个类，在python2和python3中，用法有点区别。而这种变化其实也算是python从2到3的一种优化。

Python2中括号里面是有参数，参数为类名和self。

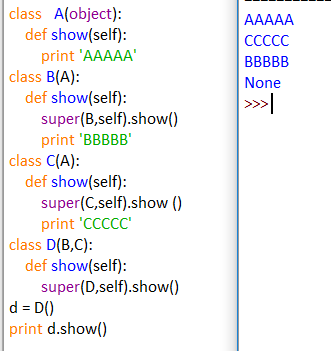


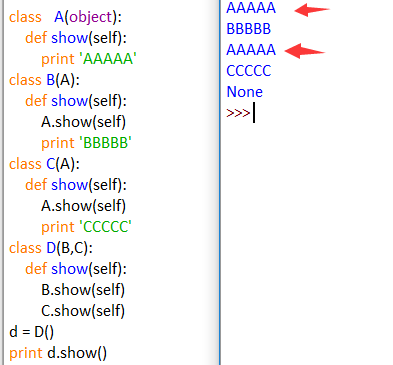
Python3中括号里没有参数。



还有一种方法来实现继承和重写的是通过基类名调用其方法的形式，而这种方法与super的的区别在多继承的时候体现的更明显。

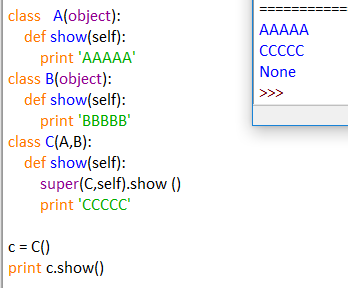
例一：

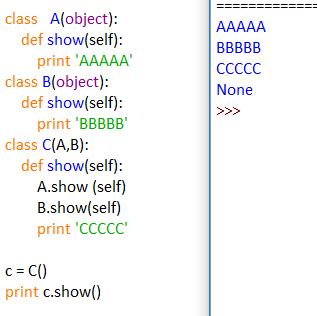




上图例子中，D同时继承B,C，而B,C类的继承类都是A，这时候使用super的方法来继承时，A类中的方法中调用了一次，而使用类名的方法继承时，A被调用了两次。在这个例子中使用super方法就要好一点了。但也不代表我们就应该把所有继承类都使用super，接下来的例子我们可以看到类名继承的方法的用处。

例二：





如上图所示，C类同时继承A,B，使用super方法，类会自动继承只第一个基类的方法，而通过基类名调用其方法的形式就可以两个基类的方法都继承过来了。

所以通过上面的两个例子我们可以知道，两种对于方法重写和继承的方法并没有好坏之分，主要看继承的结构和个人的需求是怎样的。