# class类为什么要继承object

近期项目中发现class类很多都是继承object，而之前的项目中好像没有继承object，带着疑惑找了一下原因，总结如下。

**1 新式类和经典类**

在python 2中，继承object的类叫新式类，不继承object的类叫经典类。  
资料来源：<https://www.cnblogs.com/attitudeY/p/6789370.html>

#经典类

class A:

def \_\_init\_\_(self):

print 'this is A'

def save(self):

print 'come from A'

class B(A):

def \_\_init\_\_(self):

print 'this is B'

class C(A):

def \_\_init\_\_(self):

print 'this is C'

def save(self):

print 'come from C'

class D(B,C):

def \_\_init\_\_(self):

print 'this is D'

d1=D()

d1.save() #结果为'come from A

a1 = A()

print dir(a1) #结果为 ['\_\_doc\_\_', '\_\_module\_\_']

#新式类

class A(object):

def \_\_init\_\_(self):

print 'this is A'

def save(self):

print 'come from A'

class B(A):

def \_\_init\_\_(self):

print 'this is B'

class C(A):

def \_\_init\_\_(self):

print 'this is C'

def save(self):

print 'come from C'

class D(B,C):

def \_\_init\_\_(self):

print 'this is D'

d1=D()

d1.save() #结果为'come from C'

a2 = A()

print dir(a2)

#结果为 ['\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_doc\_\_', '\_\_format\_\_', '\_\_getattribute\_\_',

'\_\_hash\_\_', '\_\_init\_\_', '\_\_module\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_',

'\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_']

**（1） 区别1**

* 在经典类中 调用D的save方法 搜索按深度优先 路径B-A-C， 执行的为A中save
* 在新式类的 调用D的save方法 搜索按广度优先 路径B-C-A， 执行的为C中save
* 由于D是直接继承C的，调用C中的save显得更合理，所以推荐用新式类。
* 在 Python 3.x 中的新式类貌似已经兼容了经典类，无论 A 是否继承 object 类， D 实例中的save() 都会执行 C 中的 save() 。但是在 Python 2.7 中这种差异仍然存在，因此还是推荐使用新式类，要继承 object 类.

**（2）区别2**

* 不继承object的类，只有 ['**doc**', '**module**']
* 继承object类，有 ['**class**', '**delattr**', '**dict**', '**doc**', '**format**', '**getattribute**',  
  '**hash**', '**init**', '**module**', '**new**', '**reduce**', '**reduce\_ex**', '**repr**',  
  '**setattr**', '**sizeof**', '**str**', '**subclasshook**', '**weakref**']有更多对象可操作。
* 无论是否继承object，都是属于object对象，print isinstance(A, object)， 结果都为true

1人点赞

[Python基础](https://www.jianshu.com/nb/11925442)