官方的说法：   
classmethod(function)  
中文说明：  
classmethod是用来指定一个类的方法为类方法，没有此参数指定的类的方法为实例方法，使用方法如下：

class C:

@classmethod

def f(cls, arg1, arg2, ...): ...

   
 看后之后真是一头雾水。说的啥子东西呢？？？  
   
自己到国外的论坛看其他的例子和解释，顿时就很明朗。 下面自己用例子来说明。  
   
看下面的定义的一个时间类：

class Data\_test(object):

day=0

month=0

year=0

def \_\_init\_\_(self,year=0,month=0,day=0):

self.day=day

self.month=month

self.year=year

def out\_date(self):

print "year :"

print self.year

print "month :"

print self.month

print "day :"

print self.day

t=Data\_test(2016,8,1)

t.out\_date()

输出：

year :

2016

month :

8

day :

1

符合期望。  
   
如果用户输入的是 "2016-8-1" 这样的字符格式，那么就需要调用Date\_test 类前做一下处理：

string\_date='2016-8-1'

year,month,day=map(int,string\_date.split('-'))

s=Data\_test(year,month,day)

先把‘2016-8-1’ 分解成 year，month，day 三个变量，然后转成int，再调用Date\_test(year,month,day)函数。 也很符合期望。  
   
那我可不可以把这个字符串处理的函数放到 Date\_test 类当中呢？  
   
那么@classmethod 就开始出场了

class Data\_test2(object):

day=0

month=0

year=0

def \_\_init\_\_(self,year=0,month=0,day=0):

self.day=day

self.month=month

self.year=year

@classmethod

def get\_date(cls,data\_as\_string):

#这里第一个参数是cls， 表示调用当前的类名

year,month,day=map(int,string\_date.split('-'))

date1=cls(year,month,day)

#返回的是一个初始化后的类

return date1

def out\_date(self):

print "year :"

print self.year

print "month :"

print self.month

print "day :"

print self.day

在Date\_test类里面创建一个成员函数， 前面用了@classmethod装饰。 它的作用就是有点像静态类，比静态类不一样的就是它可以传进来一个当前类作为第一个参数。  
   
那么如何调用呢？

r=Data\_test2.get\_date("2016-8-6")

r.out\_date()

输出：

year :

2016

month :

8

day :

1

这样子等于先调用get\_date（）对字符串进行出来，然后才使用Data\_test的构造函数初始化