凡是在类中定义了这个\_\_getitem\_\_ 方法，那么它的实例对象（假定为p），可以像这样

p[key] 取值，当实例对象做p[key] 运算时，会调用类中的方法\_\_getitem\_\_。

一般如果想使用索引访问元素时，就可以在类中定义这个方法（\_\_getitem\_\_(self, key) ）。

先用一个夸张一点的例子来说明这个问题。

class DataBase:

def \_\_init\_\_(self,id,address):

#初始化方法

self.id=id

self.address=address

self.d={self.id:1,self.address:"192.168.1.1"}

def \_\_getitem\_\_(self,key):

return self.d.get(key,"default")

data=DataBase(1,"192.168.1.1")

print(data['hi'])

print(data[data.id])

print(data[data.address])

当实例对象通过[] 运算符取值时，会调用它的方法\_\_getitem\_\_，从上面代码中输出二个笑脸可以看出来。一般这样用

还可以用在对象的迭代上

class STgetitem:

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

self.text=text

def \_\_getitem\_\_(self,index):

result=self.text[index].upper()

return result

p=STgetitem("你是谁")

print(p[0])

for char in p:

print(char,end=' ')