2016年7月2日星期六

Python字典

1：在Python中列表类似于java中的list但是又不同，

1. Python列表index只能是数字
2. 使用append()方法增加

2：python提供了字典，类似java中Map

1. 必须在最外边使用 {}

Eg：

stuff = {'name':'gyt','age':26,'height':178}

print "stuff[\'name\']: " , stuff['name']

print "stuff[\'age\']: " , stuff['age']

1. 添加更简单，直接赋值就可以

Eg：

#这样实现了添加

stuff[1] = "Wow"

1. 删除也很简单：

Eg：

#direct 删除的实现!!!

del stuff[1]

1. 不能使用[]来定义

Eg：

#注意：使用中括号是不对的

#注意：单引号和双引号都是可以的，为了美观和方便，使用单引号

states = {

'Oregon':'OR',

'Florida':'FL',

'New York':'NY',

'California':'CA',

'Michigan':'MI',

"Huadong":"HD"

}

1. 其中k-v对可以不是固定的类型，随意增加
2. 遍历：

Eg:

#遍历 k v

print '\*' \* 40

for astr,up in states.items():

print "%s Up is %s ." %(astr,up)

1. 安全获取：

Eg：

#安全获取

print '\*' \* 40

state = states.get('Texas',None)

if not state:

print "Sorry ,no Texas. "

1. 获取默认值

Eg:

#获取默认值

city = cities.get('TX','Does Not Exist')

print "The　city for the state \'TX\' is : %s" %city

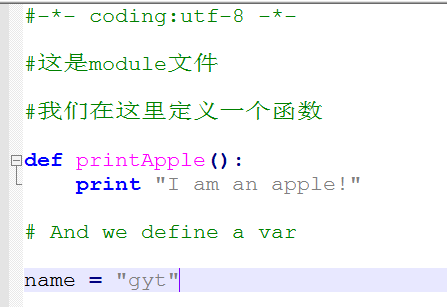
Python模块 类 和对象

1：模块

1. 模块就是在一个py文件中定义函数，和变量，然后作为模块可以在其他py文件中import和调用

Eg：

在一个No40\_ModulePy.py中定义如下：



然后在另一个py文件中这样使用：

import No40\_ModulePy

No40\_ModulePy.printApple()

print "My name is %s . " %No40\_ModulePy.name

2：类和对象

1. 关于init初始化函数：

**def \_\_init\_\_ init前后是两个英文下划线，一个会出错**

Eg：

class MyStuff(object):

#开始的时候我以为attr 必须声明，因为在后边取thing.attr的时候会报错，声明之后不会报错，但是现在还是取不到，这个我赋的值

attr = "th" #说明构造函数中的赋值不起作用。。。

#最后终于发现init函数没有写对，来自stackOverFlow

#init前后不是一个下划线，而是两个连在一起的，要有两个下划线

#这也说明 attr是可以不用声明，直接就能使用的

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

self.attr = "I am gyt."

def apple(self):

print "I am a red apple!!!"

thing = MyStuff()

thing.apple()

print "thing.attr: ", thing.attr

1. 在函数中使用的变量不用在类中声明，就可以使用

Eg：

class Song(object):

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

self.lyrics = lyrics

def sing\_me\_a\_song(self):

for line in self.lyrics:

print line

1. **关于创建init或者其他函数中的self参数问题**：

这个self参数是必须的，因为self.xxx = \*\*\*如果不加这个参数，那么赋值代码的含义就会不明确，可能指的是实例的一个属性xxx，也可能是一个叫xxx的局部变量。加上self参数之后能明确这个指的是实例的属性self.xxx。