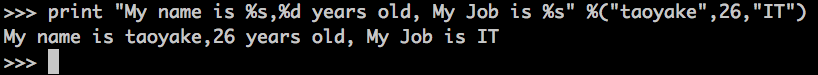
第三天知识点总结

第三天主要是复习之前的知识，把比较重要的知识点总结下来。

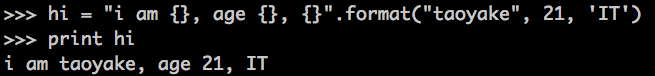
1. 字符串的两种格式化方式：

1).C语言风格：



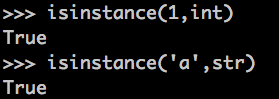


1. C#风格：通过｛｝代替% .format(a,b)

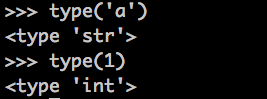


1. 判断一个python字符串的类型：

方法1: #使用isinstance内置函数方法



方法2: ##使用内置的type函数



1. 布尔值：＃要么真，要么假，（返回true或者false）

几种条件为假的情况：：

空字符串

空列表

数字0

空字典

流程控制语句：

if 条件；

xxxx #做操作

else: #否则

xxxx #做操作

多条件判断：

if 条件：

xxxx

elfi 条件：

xxxx

elfi 条件：

xxxx

......

else:

xxxx

注意缩进格式：

for循环：

一般都是结合，循环遍历一个列表或者字典，打印出来里边的值，再结合if判断语句，对打印出来的值，进行增删改查等处理操作

for I in list:

print i

for key,value in dict.items():

print keys,values

while循环：

while 条件：

xxxx

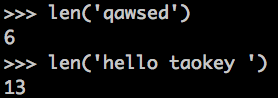
一般情况下，这个条件在执行几次以后，是会发生变化了，条件成立就执行下边的代码，不成立，就退出前前循环或者结束循环。

退出当前循环,进行下一个操作是用continue这个命令；终止整个while循环使用break整个命令，退出整个程序代码用exit.

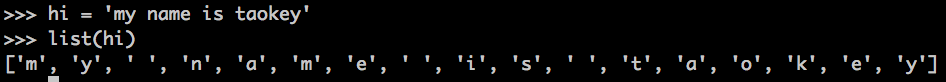
字符串：

字符串是 Python 中最常用的数据类型。我们可以使用引号('或")来创建字符串。

1. 字符串长度： #len



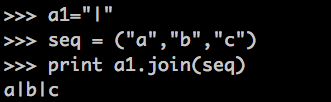
1. 字符串转换成列表： #list



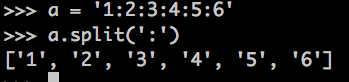
1. 字符串拼接： join(...)

S.join(iterable) -> string 将序列中的元素以指定的字符连接生成一个新的字符串。

例子:

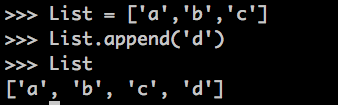


字符串的分割： #split.(‘ ’) #以什么作为分隔符，把字符串分割开

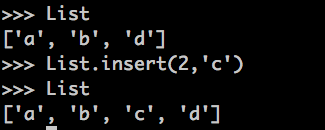


列表：

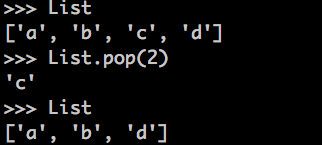
1. 往列表中添加一个元素： #list.append



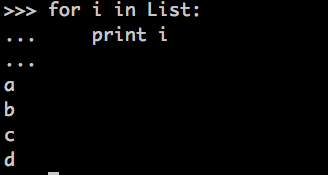
假如是往列表指定的位置添加元素，使用insert(下标，’要添加的元素’)



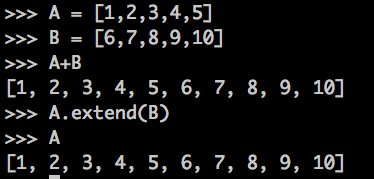
1. 删除列表中一个元素： pop(列表索引)



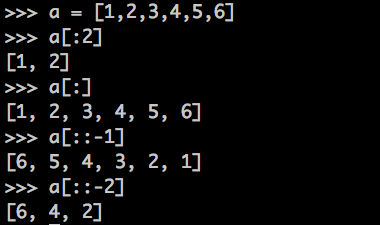
1. 查询列表：



1. 两个列表合并：



列表的切片功能：



a[::-1] #可视为反转列表的操作

>>> a[::2] #隔一个取一个元素的操作

[1, 3, 5]

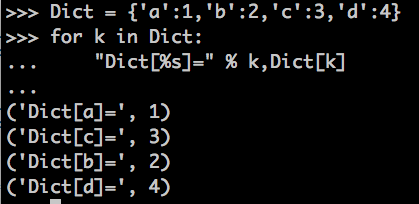
注意的是：

列表索引的下标是从0开始的，做分片的时候，获取到的几个元素是n-1。

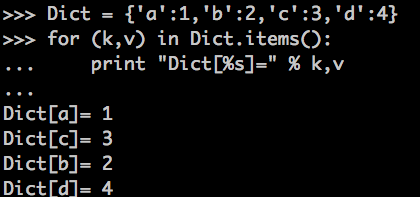
字典：

遍历字典的方法：

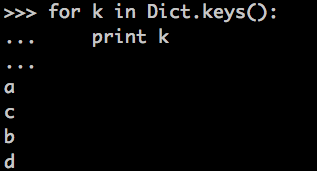
第一种方法：



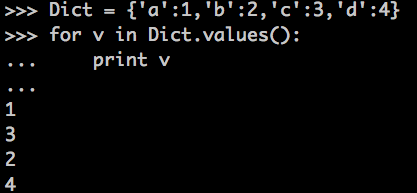
第二种方法： dict.items() keys,value形式，直接展示出来。



获取字典中所有的key 键

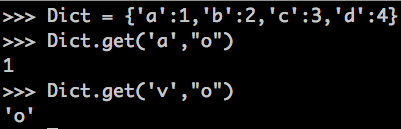


或者字典中所有的values值，



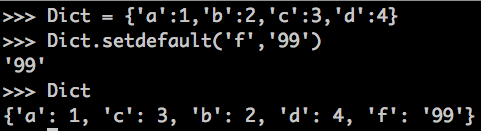
字典get，获取字典中指定键的值，如果没有则赋一个默认值

# dict.get(key, default=None)

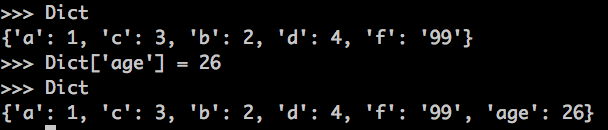


字典setdefault,类似于get,如果字典中没有键，则添加键，并将值设置成默认值

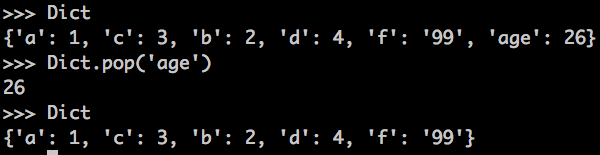
dict.setdefault(key, default=None)



添加字典



删除字典，dict.pop(key)



判断一个元素是否在字典中，两种方式：

1．用has\_key判断

2. dict.get(key, default=None) ,给字典一个默认的值例如：

>>> Dict.get('hello','apple')

'apple'

**python中文件读写**

1.文件基本读写操作

fo = open(“/tmp/user.txt”) # 打开文件file＝open,默认只读

fo #查看文件句柄

fo.read() #读取，一次全部读完，结果为一个整体的字符串

fo.write('文件路径以及文件名') #写入

fo.close() #关闭文件，关闭后无法打开

2.打开文件的几种模式：

open('path') #默认只读打开

open('path','r+') #读写打开,如果有内容,就会从头覆盖相应字符串的内容

open('path','w') #写入,先删除原文件重新写入,没有文件自己创建

open(‘path’,'w+') #读写,同上

open('path','a') #写入,在文件末尾追加新内容,文件不存在自动创建

open('path','a+') #读写,同上,最常用的

open('path','b') #打开二进制文件,要上述模式结合使用,读取图片

open('path','u') #支持所有的换行符号 \r \n \r\n

例子：

bogon:day2 playcrab$ python test.py

name : taoyake

passwd: 123.com

bogon:day2 playcrab$ cat test.py

F = open('user.txt')

print F.read()

readline 一次之读取一行：

bogon:day2 playcrab$ python test.py

name : taoyake

bogon:day2 playcrab$ cat test.py

F = open('user.txt','r')

print F.readline()

readlines: 把文件一行一行读取出来

bogon:day2 playcrab$ python test.py

name : taoyake

passwd: 123.com

bogon:day2 playcrab$ cat test.py

F = open('user.txt','r')

line = F.readlines()

for i in line:

print i

第三种打开文件的方法：加了容错，而且更加优雅

with open(‘user.txt’) as f:

data = f.read()

do something

函数介绍：

函数是组织好的，可重复使用的，用来实现单一，或相关联功能的代码段。

函数能提高应用的模块性，和代码的重复利用率。你已经知道Python提供了许多内建函数，比如print()。但你也可以自己创建函数，这被叫做用户自定义函数。

定义一个函数：

你可以定义一个由自己想要功能的函数，以下是简单的规则：

函数代码块以 def 关键词开头，后接函数标识符名称和圆括号()。

任何传入参数和自变量必须放在圆括号中间。圆括号之间可以用于定义参数。

函数的第一行语句可以选择性地使用文档字符串—用于存放函数说明。

函数内容以冒号起始，并且缩进。

return [表达式] 结束函数，选择性地返回一个值给调用方。不带表达式的return默认返回None。

语法：

def functionname( parameters ):

"函数\_文档字符串"

function\_suite

return [expression]

默认情况下，参数值和参数名称是按函数声明中定义的的顺序匹配起来的。

实例：

以下为一个简单的Python函数，它将一个字符串作为传入参数，再打印到标准显示设备上。

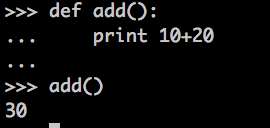
def fo(str):

“打印传入的字符串到标准显示设备上”

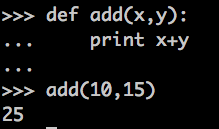
print “str”

return

定义一个函数add， 打印一个加法，默认不调用，是不会显示打印的，调用方法很简单，add()



例子：



return和print的区别：

return 可以赋值给变量

print不可以赋值给变量

用函数写一个计算器：

#-\*- coding:utf-8 -\*-

def add(x,y):

return x+y

def jian(x,y):

return x-y

def cheng(x,y):

return x\*y

def chu(x,y):

return x/y

def start(x,y):

x = int(raw\_input("请您输入一个数字 ").strip())

y = int(raw\_input("请您输入另一个数字 ").strip())

cctv = raw\_input("请您输入运算方法:add/jian/cheng/chu:").strip()

if cctv == "add":

print add(x,y)

elif cctv == "jian":

print jian(x,y)

elif cctv == "cheng":

print cheng(x,y)

elif cctv == "chu":

print chu(x,y)

print start()

return返回值的两层意思：

a．跳出函数。return后面的代码将不再执行，函数退出

b. 返回函数结果。没有return(默认输出为None),或者return后面不跟东西，

则函数被调用执行也没有任何输出。