**Python知识总结**

**1、.什么是lambda函数？它有什么好处?**

答：lambda 表达式，通常是在需要一个函数，但是又不想费神去命名一个函数的场合下使用，也就是指匿名函数

lambda [arguments]:expression

>>> a=lambdax,y:x+y

>>> a(3,11)

1. **获取指定路径下的所有文件名**

**os.listdir() os.path.join() os.path.isdir**

"""

这个函数接受文件夹的名称作为输入参数，

返回该文件夹中文件的路径，

以及其包含文件夹中文件的路径。

"""

import os

def print\_directory\_contents(sPath):

#指定路径下的文件和文件夹名['.ipynb\_checkpoints', 'python1.ipynb']

for sChild in os.listdir(sPath):

sChildPath = os.path.join(sPath,sChild)

if os.path.isdir(sChildPath):

#如果path路径下是文件夹，则继续找出该文件夹下的所有文件名

print\_directory\_contents(sChildPath)

else:

print(sChildPath)

#print\_directory\_contents('D:\daacheng\Python\PythonCode\pythontest')

path='D:\daacheng\Python\PythonCode\pythontest'

print\_directory\_contents(path)

1. **python中list和tuple转换**

#python中list和tuple转换

a=[1,2,3,4]

b=tuple(a)

c=list(b)

print(type(b))#tuple

print(type(c))#list

**4、请写出一段Python代码实现删除一个list里面的重复元素**

a=[1,1,1,2,3,4,4]

#b=set(a)#{1,2,3,4}#方法一：使用set()方法

b=dict.fromkeys(a)#方法二：dict.fromkeys()方法转换成dict，然后取key

single=list(b.keys())#[1,2,3,4]

print(single)

**5.介绍一下except的用法和作用？**

答：try…except…except…[else…][finally…]

执行try下的语句，如果引发异常，则执行过程会跳到except语句。对每个except分支顺序尝试执行，如果引发的异常与except中的异常组匹配，执行相应的语句。如果所有的except都不匹配，则异常会传递到下一个调用本代码的最高层try代码中。

try下的语句正常执行，则执行else块代码。如果发生异常，就不会执行

如果存在finally语句，最后总是会执行

**5.Python中pass语句的作用是什么？**

答：pass语句不会执行任何操作，一般作为占位符或者创建占位程序，whileFalse:pass

**6.如何用Python来进行查询和替换一个文本字符串？**

答：可以使用re模块中的sub()函数或者subn()函数来进行查询和替换，

格式：sub(replacement, string[,count=0])（replacement是被替换成的文本，string是需要被替换的文本，count是一个可选参数，指最大被替换的数量）

>>> import re

>>>p=re.compile(‘blue|white|red’)

>>>print(p.sub(‘colour’,'blue socks and red shoes’))

colour socks and colourshoes

>>>print(p.sub(‘colour’,'blue socks and red shoes’,count=1))

colour socks and redshoes

subn()方法执行的效果跟sub()一样，不过它会返回一个二维数组，包括替换后的新的字符串和总共替换的数量

**7.Python里面match()和search()的区别？**

答：re模块中match(pattern,string[,flags]),检查string的开头是否与pattern匹配。

re模块中research(pattern,string[,flags]),在string搜索pattern的第一个匹配值。

>>>print(re.match(‘super’, ‘superstition’).span())

(0, 5)

>>>print(re.match(‘super’, ‘insuperable’))

None

>>>print(re.search(‘super’, ‘superstition’).span())

(0, 5)

>>>print(re.search(‘super’, ‘insuperable’).span())

(2, 7)