2021年Python面试题及答案

找工作是每个学习Python人员的目标，为了更好的找到工作，刷面试题是必不可少的，了解最新企业招聘试题，可以让你面试更加的顺利。小编整理了一些2021年的Python面试题，来测测你技术的掌握度吧。

**1、Python里面如何拷贝一个对象？（赋值，浅拷贝，深拷贝的区别）**

答：赋值（=），就是创建了对象的一个新的引用，修改其中任意一个变量都会影响到另一个。

浅拷贝：创建一个新的对象，但它包含的是对原始对象中包含项的引用（如果用引用的方式修改其中一个对象，另外一个也会修改改变）{1,完全切片方法；2，工厂函数，如list()；3，copy模块的copy()函数}

深拷贝：创建一个新的对象，并且递归的复制它所包含的对象（修改其中一个，另外一个不会改变）{copy模块的deep.deepcopy()函数}

**2、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)

**3、有没有一个工具可以帮助查找python的bug和进行静态的代码分析？**

答：PyChecker是一个python代码的静态分析工具，它可以帮助查找python代码的bug, 会对代码的复杂度和格式提出警告

Pylint是另外一个工具可以进行codingstandard检查

**4、简要描述Python的垃圾回收机制（garbage collection）。**

答案

这里能说的很多。你应该提到下面几个主要的点：

Python在内存中存储了每个对象的引用计数（reference count）。如果计数值变成0，那么相应的对象就会小时，分配给该对象的内存就会释放出来用作他用。

偶尔也会出现引用循环（reference cycle）。垃圾回收器会定时寻找这个循环，并将其回收。举个例子，假设有两个对象o1和o2，而且符合o1.x == o2和o2.x == o1这两个条件。如果o1和o2没有其他代码引用，那么它们就不应该继续存在。但它们的引用计数都是1。

Python中使用了某些启发式算法（heuristics）来加速垃圾回收。例如，越晚创建的对象更有可能被回收。对象被创建之后，垃圾回收器会分配它们所属的代（generation）。每个对象都会被分配一个代，而被分配更年轻代的对象是优先被处理的。

**5、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函数：首要用途是指点短小的回调函数

lambda [arguments]:expression

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

>>> a(3,11)

**6、请写出一段Python代码实现删除一个list里面的重复元素**

答：

1,使用set函数，set(list)

2，使用字典函数，

>>>a=[1,2,4,2,4,5,6,5,7,8,9,0]

>>> b={}

>>>b=b.fromkeys(a)

>>>c=list(b.keys())

>>> c

**7、用Python匹配HTML tag的时候，<.\*>和<.\*?>有什么区别？**

答：术语叫贪婪匹配( <.\*> )和非贪婪匹配(<.\*?> )

例如:

test

<.\*> :

test

<.\*?> :

8、如何在一个function里面设置一个全局的变量？

答：解决方法是在function的开始插入一个global声明：

def f()

global x

9、编程用sort进行排序，然后从最后一个元素开始判断

a=[1,2,4,2,4,5,7,10,5,5,7,8,9,0,3]

a.sort()

last=a[-1]

for i inrange(len(a)-2,-1,-1):

if last==a[i]:

del a[i]

else:last=a[i]

print(a)

10、下面的代码在Python2中的输出是什么？解释你的答案

def div1(x,y):

 print "%s/%s = %s" % (x, y, x/y)

def div2(x,y):

 print "%s//%s = %s" % (x, y, x//y)

div1(5,2)

div1(5.,2)

div2(5,2)

div2(5.,2.)

另外，在Python3中上面的代码的输出有何不同(假设代码中的print语句都转化成了Python3中的语法结构)？

在Python2中，代码的输出是：

5/2 = 2

5.0/2 = 2.5

5//2 = 2

5.0//2.0 = 2.0

默认情况下，如果两个操作数都是整数，Python2默认执行整数运算。所以，5/2 结果是2，而5./2结果是2.5

注意你可以通过下面的import语句来覆盖Python2中的这一行为

from \_\_future\_\_ import division

还要注意“双斜杠”(//)操作符将会一直执行整除，忽略操作数的类型。这就是为什么5.0//2.0即使在Python2中结果也是2.0

但是在Python3并没有这一行为。两个操作数都是整数时，也不执行整数运算。在Python3中，输出如下：

5/2 = 2.5

5.0/2 = 2.5

5//2 = 2

5.0//2.0 = 2.0