

[学习交流] Python 面试必须要看的 16 个问题

(一). Python 是如何进行内存管理的？

答:从三个方面来说,一对象的引用计数机制,二垃圾回收机制,三内存池机制

1.对象的引用计数机制

Python 内部使用引用计数, 来保持追踪内存中的对象, 所有对象都有引用计数。

引用计数增加的情况:

- ❶一个对象分配一个新名称
- ❷将其放入一个容器中（如列表、元组或字典）

引用计数减少的情况:

- ❶使用 del 语句对对象别名显示的销毁
- ❷引用超出作用域或被重新赋值

sys.getrefcount() 函数可以获得对象的当前引用计数

多数情况下, 引用计数比你猜测得要大得多。对于不可变数据（如数字和字符串）, 解释器会在程序的不同部分共享内存, 以便节约内存。

2.垃圾回收

- ❶当一个对象的引用计数归零时, 它将被垃圾收集机制处理掉。
- ❷当两个对象 a 和 b 相互引用时, del 语句可以减少 a 和 b 的引用计数, 并销毁用于引用底层对象的名称。然而由于每个对象都包含一个对其他对象的应用, 因此引用计数不会归零, 对象也不会销毁。（从而导致内存泄露）。为解决这一问题, 解释器会定期执行一个循环检测器, 搜索不可访问对象的循环并删除它们。

3.内存池机制

Python 提供了对内存的垃圾收集机制, 但是它将不用的内存放到内存池而不是返回给操作系统。

- ❶Pymalloc 机制。为了加速 Python 的执行效率, Python 引入了一个内存池机制, 用于管理对小块内存的申请和释放。
- ❷Python 中所有小于 256 个字节的对象都使用 pymalloc 实现的分配器, 而大的对象则使用系统的 malloc。
- ❸对于 Python 对象, 如整数, 浮点数和 List, 都有其独立的私有内存池, 对象间不共享他们的内存池。也就是说如果你分配又释放了大量的整数, 用于缓存这些整数的内存就不能再分配给浮点数。

(二)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)
```

(三)Python 里面如何实现 tuple 和 list 的转换？

答：直接使用 tuple 和 list 函数就行了，type() 可以判断对象的类型

(四)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
```

(五)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 in range(len(a)-2,-1,-1):
```

```
if last==a:
```

```
del a
```

```
else:last=a
```

```
print(a)
```

(\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.copy() 函数}

(\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。

(\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，while False: pass

(\介绍一下 Python 下 range() 函数的用法？

答：列出一组数据，经常用在 for in range() 循环中

(\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

(十一) 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 re.search(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)
```

(十二) 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
```

```
<.*?> :
```

(十三) Python 里面如何生成随机数?

答: random 模块

随机整数: random.randint(a, b): 返回随机整数 x, a<=x<=b

random.randrange(start, stop, [, step]): 返回一个范围在 (start, stop, step) 之间的随机整数, 不包括结束值。

随机实数: random.random(): 返回 0 到 1 之间的浮点数

random.uniform(a, b): 返回指定范围内的浮点数。

(十四) 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

（十五）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
```

（十六）单引号，双引号，三引号的区别

答：单引号和双引号是等效的，如果要换行，需要符号(\)，三引号则可以直接换行，并且可以包含注释

如果要表示 Let' s go 这个字符串

单引号：s4 = 'Let\' s go'

双引号：s5 = "Let' s go"

s6 = 'I realy like "python" !'

这就是单引号和双引号都可以表示字符串的原因了