《Python 程序设计》题库

一、 填空题

1、 Python 安装扩展库常用的是_工具。（pip）

2、 Python 标准库 math 中用来计算平方根的函数是____。（sqrt）

3、 Python 程序文件扩展名主要有__和两种，其中后者常用于 GUI 程序。（py、pyw）

4、 Python 源代码程序编译后的文件扩展名为___。（pyc）

5、 使用 pip 工具升级科学计算扩展库 numpy 的完整命令是_____。（pip install –upgrade numpy）

6、 使用 pip 工具查看当前已安装的 Python 扩展库的完整命令是___。（pip list）

7、 在 IDLE 交互模式中浏览上一条语句的快捷键是____。（Alt+P）

8、 使用 pip 工具查看当前已安装 Python 扩展库列表的完整命令是___。（pip list）

9、 在 Python 中____表示空类型。（None）

10、 列表、元组、字符串是 Python 的___(有序？无序)序列。（有序）

11、 查看变量类型的 Python 内置函数是_____。（type()）

12、 查看变量内存地址的 Python 内置函数是_____。（id()）

13、 以 3 为实部 4 为虚部，Python 复数的表达形式为_或__。（3+4j、3+4J）

14、 Python 运算符中用来计算整商的是___。（//）

15、 Python 运算符中用来计算集合并集的是_。（|）

16、使用运算符测试集合包含集合 A 是否为集合 B 的真子集的表达式可以写作_。（A < B ）

17、 表达式[1, 2, 3]*3 的执行结果为_____。（[1, 2, 3, 1, 2, 3, 1, 2, 3]）

18、 list(map(str, [1, 2, 3]))的执行结果为_____。（[‘1’, ‘2’, ‘3’]）

19、 语句 x = 3==3, 5 执行结束后，变量 x 的值为___。（(True, 5)）

20、 已知 x = 3，那么执行语句 x += 6 之后，x 的值为_____。（9）

21、 已知 x = 3，并且 id(x)的返回值为 496103280，那么执行语句 x += 6 之后，表达式 id(x) == 496103280 的值为_。（False）

22、 已知 x = 3，那么执行语句 x *= 6 之后，x 的值为_____。（18）

23、 为了提高 Python 代码运行速度和进行适当的保密，可以将 Python 程序文件编译为扩展名____的文件。（pyc）

24、 表达式“[3] in [1, 2, 3, 4]”的值为_____。（False）

25、 列表对象的 sort()方法用来对列表元素进行原地排序，该函数返回值为 。（None）

26、 假设列表对象 aList 的值为[3, 4, 5, 6, 7, 9, 11, 13, 15, 17]，那么切片 aList[3:7]得到的值是_____。（[6, 7, 9, 11]）

27、使用列表推导式生成包含 10 个数字 5 的列表，语句可以写为_____。（[5 for i in range(10)]）

28、 假设有列表 a = [‘name’, ‘age’, ‘sex’]和 b = [‘Dong’, 38, ‘Male’]，请使用一个语句将这两个列表的内容转换为字典，并且以列表 a 中的元素为“键”，以列表 b 中的元素为“值”，这个语句可以写为_____。（c = dict(zip(a, b)）

29、 任意长度的 Python 列表、元组和字符串中最后一个元素的下标为__。（-1）

30、 Python 语句”.join(list(‘hello world!’))执行的结果是_____。（’hello world!’）

31、 转义字符’\n’的含义是_____。（回车换行）

32、 Python 语句 list(range(1,10,3))执行结果为_____。（[1, 4, 7]）

33、 表达式 list(range(5)) 的值为_____。（[0, 1, 2, 3, 4]）

34、 ____命令既可以删除列表中的一个元素，也可以删除整个列表。（del）

35、 已知 a = [1, 2, 3]和 b = [1, 2, 4]，那么 id(a[1])==id(b[1])的执行结果为_。（True）

36、 表达式 int(‘123’, 16) 的值为___。（291）
37、 表达式 int(‘123’, 8) 的值为___。（83）
38、 表达式 int(‘123’) 的值为___。（123）
39、 表达式 int(‘101’,2) 的值为____。（5）
40、 表达式 abs(-3) 的值为_。（3）
41、 切片操作 list(range(6))[::2]执行结果为_____。（[0, 2, 4]）
42、 使用切片操作在列表对象 x 的开始处增加一个元素 3 的代码为___。（x[0:0] = [3]）
43、 语句 sorted([1, 2, 3], reverse=True) == reversed([1, 2, 3])执行结果为___。（False）
44、 表达式 ‘ab’ in ‘acbed’ 的值为__。（False）
45、 Python 3.x 语句 print(1, 2, 3, sep=’:’) 的输出结果为__。（1:2:3）
46、 表达式 sorted([111, 2, 33], key=lambda x: len(str(x))) 的值为_____。（[2, 33, 111]）
47、 假设 n 为整数，那么表达式 n&1 == n%2 的值为___。（True）
48、 表达式 int(4**0.5) 的值为__。（2）
49、 达式 sorted([111, 2, 33], key=lambda x: -len(str(x))) 的值为__。（[111, 33, 2]）
50、 Python 内置函数___可以返回列表、元组、字典、集合、字符串以及 range 对象中元素个数。（len()）
51、 Python 内置函数__用来返回序列中的最大元素。（max()）
52、 Python 内置函数__用来返回序列中的最小元素。（min()）
53、 Python 内置函数_____用来返回数值型序列中所有元素之和。（sum()）
54、 已知列表对象 x = [‘11’, ‘2’, ‘3’]，则表达式 max(x) 的值为_。（’3’）
55、 表达式 min([‘11’, ‘2’, ‘3’]) 的值为_____。（’11’）
56、 已知列表对象 x = [‘11’, ‘2’, ‘3’]，则表达式 max(x, key=len) 的值为_。（’11’）
57、 语句 x = (3,) 执行后 x 的值为_____。（(3,)）
58、 语句 x = (3) 执行后 x 的值为_____。（3）
59、 已知 x=3 和 y=5，执行语句 x, y = y, x 后 x 的值是__。（5）
60、 可以使用内置函数_____查看包含当前作用域内所有全局变量和值的字典。（globals()）
61、 可以使用内置函数_____查看包含当前作用域内所有局部变量和值的字典。（locals()(）
62、 字典中多个元素之间使用__分隔开，每个元素的“键”与“值”之间使用__分隔开。（逗号、冒号）
63、 字典对象的_方法可以获取指定“键”对应的“值”，并且可以在指定“键”不存在的时候返回指定值，如果不指定则返回 None。（get()）
64、 字典对象的___方法返回字典中的“键-值对”列表。（items()）
65、 字典对象的__方法返回字典的“键”列表。（keys()）
66、 字典对象的____方法返回字典的“值”列表。（values()）
67、 已知 x = {1:2}，那么执行语句 x[2] = 3 之后，x 的值为_____。（{1: 2, 2: 3}）
68、 表达式 {1, 2, 3, 4} - {3, 4, 5, 6}的值为_____。（{1, 2}）
69、 表达式 set([1, 1, 2, 3])的值为_____。（{1, 2, 3}）
70、 关键字____用于测试一个对象是否是一个可迭代对象的元素。（in）
71、 使用列表推导式得到 100 以内所有能被 13 整除的数的代码可以写作_____。（[i for i in range(100) if i%13==0]）
72、 表达式 3<5>2 的值为_____。（True）
73、 已知 x = {‘a’:’b’, ‘c’:’d’}，那么表达式 ‘a’ in x 的值为____。（True）
74、 已知 x = {‘a’:’b’, ‘c’:’d’}，那么表达式 ‘b’ in x 的值为____。（False）

75、 已知 x = { ‘a’ :’ b’ , ‘c’ :’ d’ }，那么表达式 ‘b’ in x.values() 的值为____。（True）
76、 表达式 1<2<3 的值为___。（True）
77、 表达式 3 or 5 的值为__（3）
78、 表达式 0 or 5 的值为___（5）
79、 表达式 3 and 5 的值为__。（5）
80、 表达式 3 and not 5 的值为____。（False）
81、 表达式 3 | 5 的值为____。（7）
82、 表达式 3 & 6 的值为___。（2）
83、 表达式 3 ** 2 的值为___。（9）
84、 表达式 3 * 2 的值为_。（6）
85、 已知 x = [3, 5, 7]，那么表达式 x[10:]的值为__。（[]）
86、 已知 x = [3, 5, 7]，那么执行语句 x[len(x):] = [1, 2]之后，x 的值为____。（[3, 5, 7, 1, 2]）
87、 已知 x = [3, 7, 5]，那么执行语句 x.sort(reverse=True)之后，x 的值为_____。（[7, 5, 3]）
88、 已知 x = [3, 7, 5]，那么执行语句 x = x.sort(reverse=True)之后，x 的值为___。（None）
89、 已知 x = [1, 11, 111]，那么执行语句 x.sort(key=lambda x: len(str(x)), reverse=True) 之后，x 的值为_____。（[111, 11, 1]）
90、 表达式 list(zip([1,2], [3,4])) 的值为_____。（[(1, 3), (2, 4)]）
91、 已知 x = [1, 2, 3, 2, 3]，执行语句 x.pop() 之后，x 的值为___。（[1, 2, 3, 2]）
92、 表达式 list(map(list,zip(*[[1, 2, 3], [4, 5, 6]]))) 的值为_____。（[[1, 4], [2, 5], [3, 6]]）
93、 表达式 [x for x in [1,2,3,4,5] if x<3] 的值为_____。（[1, 2]）
94、 表达式 [index for index, value in enumerate([3,5,7,3,7]) if value == max([3,5,7,3,7])] 的值为_____。（[2, 4]）
95、 已知 x = [3,5,3,7]，那么表达式 [x.index(i) for i in x if i==3] 的值为____。（[0, 0]）
96、 已知列表 x = [1, 2]，那么表达式 list(enumerate(x)) 的值为_____。（[(0, 1), (1, 2)]）
97、 已知 vec = [[1,2], [3,4]]，则表达式 [col for row in vec for col in row] 的值为_____。（[1, 2, 3, 4]）
98、 已知 vec = [[1,2], [3,4]]，则表达式 [[row[i] for row in vec] for i in range(len(vec[0]))] 的值为_____。（[[1, 3], [2, 4]]）
99、 已知 x = list(range(10))，则表达式 x[-4:] 的值为____。（[6, 7, 8, 9]）
100、 已知 path = r’c:\test.html’，那么表达式 path[:-4]+’htm’ 的值为____。（’c:\test.htm’）
101、 已知 x = [3, 5, 7]，那么执行语句 x[1:] = [2]之后，x 的值为_____。（[3, 2]）
102、 已知 x = [3, 5, 7]，那么执行语句 x[:3] = [2]之后，x 的值为_____。（[2]）
103、 已知 x 为非空列表，那么执行语句 y = x[:]之后，id(x[0]) == id(y[0])的值为____。（True）
104、 已知 x = [1, 2, 3, 2, 3]，执行语句 x.remove(2) 之后，x 的值为__。（[1, 3, 2, 3]）
105、 表达式 3<<2 的值为_____。（12）
106、 表达式 65 >> 1 的值为___。（32）
107、 表达式 chr(ord(‘a’)^32) 的值为_。（’ A’）
108、 表达式 chr(ord(‘a’)-32) 的值为_。（’ A’）
109、 表达式 abs(3+4j) 的值为__。（5.0）
110、 表达式 callable(int) 的值为_。（True）
111、 表达式 list(str([1,2,3])) == [1,2,3] 的值为____。（False）
112、 表达式 str([1, 2, 3]) 的值为_____。（’ [1, 2, 3]’）
113、 表达式 str((1, 2, 3)) 的值为_____。（’ (1, 2, 3)’）
114、 Python 中用于表示逻辑与、逻辑或、逻辑非运算的关键字分别是_、___、___。（and、

or、not）
115、 Python 3.x 语句 for i in range(3):print(i, end=’,’) 的输出结果为_____。（0,1,2,）
116、 Python 3.x 语句 print(1, 2, 3, sep=’,’) 的输出结果为_____。（1,2,3）
117、 对于带有 else 子句的 for 循环和 while 循环，当循环因循环条件不成立而自然结束时__(会？不会？)执行 else 中的代码。（会）
118、 在循环语句中，____语句的作用是提前结束本层循环。（break）
119、 在循环语句中，_语句的作用是提前进入下一次循环。（continue）
120、 表达式 sum(range(1, 10, 2)) 的值为__。（25）
121、 表达式 sum(range(1, 10)) 的值为___。（45）
122、 表达式 ‘%c’%65 的值为___。（’A’）
123、 表达式 ‘%s’%65 的值为____。（’65’）
124、 表达式 ‘%d,%c’ % (65, 65) 的值为__。（’65,A’）
125、 表达式 ‘The first:{1}, the second is {0}’.format(65,97) 的值为_____。（’The first:97, the second is 65’）
126、 表达式 ‘{0:#d},{0:#x},{0:#o}’.format(65) 的值为___。（’65,0x41,0o101’）
127、 表达式 isinstance(‘abcdefg’, str) 的值为__。（True）
128、 表达式 isinstance(‘abcdefg’, object) 的值为___。（True）
129、 表达式 isinstance(3, object) 的值为___。（True）
130、 表达式 ‘abcabcabc’.rindex(‘abc’) 的值为__。（6）
131、 表达式 ‘:’.join(‘abcdefg’.split(‘cd’)) 的值为____。（’ab:efg’）
132、 表达式 ‘Hello world. I like Python.’.rfind(‘python’) 的值为__。（-1）
133、 表达式 ‘abcabcabc’.count(‘abc’) 的值为___。（3）
134、 表达式 ‘apple.peach,banana,pear’.find(‘p’) 的值为____。（1）
135、 表达式 ‘apple.peach,banana,pear’.find(‘ppp’) 的值为__。（-1）
136、 表达式 ‘abcdefg’.split(‘d’) 的值为_____。（[‘abc’, ‘efg’]）
137、 表达式 ‘:’.join(‘1,2,3,4,5’.split(‘,’)) 的值为_____。（’1:2:3:4:5’）
138、 表达式 ‘,’.join(‘a b ccc\n\n\nddd ‘.split()) 的值为____。（’a,b,ccc,ddd’）
139、 表达式 ‘Hello world’.upper() 的值为_。（’HELLO WORLD’）
140、 表达式 ‘Hello world’.lower() 的值为___。（’hello world’）
141、 表达式 ‘Hello world’.lower().upper() 的值为_。（’HELLO WORLD’）
142、 表达式 ‘Hello world’.swapcase().swapcase() 的值为____。（’Hello world’）
143、 表达式 r’c:\windows\notepad.exe’.endswith(‘.exe’) 的值为___。（True）
144、 表达式 r’c:\windows\notepad.exe’.endswith((‘.jpg’, ‘.exe’)) 的值为_。（True）
145、 表达式 ‘C:\Windows\notepad.exe’.startswith(‘C:’) 的值为___。（True）
146、 表达式 len(‘Hello world!’.ljust(20)) 的值为___。（20）
147、 表达式 len(‘abcdefg’.ljust(3)) 的值为___。（7）
148、 表达式 len([i for i in range(10)]) 的值为____。（10）
149、 表达式 len(range(1,10)) 的值为___。（9）
150、 表达式 range(10)[-1] 的值为__。（9）
151、 表达式 range(10,20)[4] 的值为____。（14）
152、 表达式 round(3.4) 的值为_。（3）
153、 表达式 round(3.7) 的值为___。（4）
154、 表达式 ‘a’ + ‘b’ 的值为___。（’ab’）
155、 已知 x = ‘123’ 和 y = ‘456’，那么表达式 x + y 的值为____。（’123456’）

156、 表达式 ‘a’.join(‘abc’.partition(‘a’)) 的值为_____。(’aaabc’)
157、 表达式 re.split(‘.+’,‘alpha.beta…gamma..delta’) 的值为_____。([‘alpha’, ‘beta’, ‘gamma’, ‘delta’])
158、 已知 x = ‘a234b123c’，并且 re 模块已导入，则表达式 re.split(‘\d+’,x) 的值为_____。([‘a’, ‘b’, ‘c’])
159、 表达式 ”.join(‘asdssfff’.split(‘sd’)) 的值为__。(’assfff’)
160、 表达式 ”.join(re.split(‘[sd]’,’asdssfff’)) 的值为_____。(’afff’)
161、 假设 re 模块已导入，那么表达式 re.findall(‘(\d)\1+’, ‘33abcd112’) 的值为_____。([‘3’, ‘1’])
162、 语句 print(re.match(‘abc’, ‘defg’)) 输出结果为___。(None)
163、 表达式 ‘Hello world!’[-4] 的值为_____。(’r’)
164、 表达式 ‘Hello world!’[-4:] 的值为_____。(’rld!’)
165、 表达式 ‘test.py’.endswith((‘.py’, ‘.pyw’)) 的值为____。(True)
166、 已知 x = (3)， 那么表达式 x * 3 的值为____。(9)
167、 已知 x = (3,)，那么表达式 x * 3 的值为___。((3, 3, 3))
168、 表达式 len(‘abc’.ljust(20)) 的值为____。(20)
169、 代码 print(re.match(‘^[a-zA-Z]+$’,’abcDEFG000’)) 的输出结果为___。(None)
170、 当在字符串前加上小写字母或大写字母表示原始字符串，不对其中的任何字符进行转义。(r、R)
171、 在设计正则表达式时，字符_紧随任何其他限定符(*、+、?、{n}、{n,}、{n,m})之后时，匹配模式是“非贪心的”，匹配搜索到的、尽可能短的字符串。(?)
172、 假设正则表达式模块 re 已导入，那么表达式 re.sub(‘\d+’, ‘1’, ‘a12345bbbb67c890d0e’) 的值为_____。(’a1bbbb1c1d1e’)
173、 假设列表对象 x = [1, 1, 1]，那么表达式 id(x[0]) == id(x[2])的值为___。(True)
174、 已知列表 x = list(range(10))，那么执行语句 del x[::2]之后，x 的值为___。([1, 3, 5, 7, 9])
175、 已知列表 x = [1, 2, 3, 4]，那么执行语句 del x[1] 之后 x 的值为__。([1, 3, 4])
176、 表达式 [1] * 2 的值为_____。([1, 1])
177、 表达式 [1, 2] * 2 的值为____。([1, 2, 1, 2])
178、 已知列表 x = [1, 2, 3]，那么执行语句 x.insert(1, 4) 只有，x 的值为_。([1, 4, 2, 3])
179、 已知列表 x = [1, 2, 3]，那么执行语句 x.insert(0, 4) 只有，x 的值为_。([4, 1, 2, 3])
180、 已知列表 x = [1, 2, 3]，那么执行语句 x.pop(0) 之后，x 的值为___。([2, 3])
181、 已知 x = [[1]] * 3，那么执行语句 x[0][0] = 5 之后，变量 x 的值为_____。([[5], [5], [5]])
182、 表达式 list(map(lambda x: x+5, [1, 2, 3, 4, 5])) 的值为_____。([6, 7, 8, 9, 10])
183、 表达式 {1, 2, 3, 4, 5} ^ {4, 5, 6, 7} 的值为_____。({1, 2, 3, 6, 7})
184、 表达式 5 if 5>6 else (6 if 3>2 else 5) 的值为___。(6)
185、 已知 x = [1, 2, 3]，那么执行语句 x[len(x)-1:] = [4, 5, 6]之后，变量 x 的值为_____。([1, 2, 4, 5, 6])
186、 表达式 len(range(1, 10)) 的值为_。(9)
187、 表达式 len(‘中国’.encode(‘utf-8’)) 的值为_。(6)
188、 表达式 len(‘中国’.encode(‘gbk’)) 的值为__。(4)
189、 表达式 chr(ord(‘A’)+2) 的值为____。(’C’)
190、 已知 x 是一个列表对象，那么执行语句 y = x[:] 之后表达式 id(x) == id(y) 的值为_。

（False）
191、 表达式 sorted([13, 1, 237, 89, 100], key=lambda x: len(str(x))) 的值为_____。（[1, 13, 89, 237, 100]）
192、 Python 中定义函数的关键字是_____。（def）
193、 在函数内部可以通过关键字_____来定义全局变量。（global）
194、 如果函数中没有 return 语句或者 return 语句不带任何返回值，那么该函数的返回值为_____。（None）
195、 表达式 sum(range(10)) 的值为_____。（45）
196、 表达式 sum(range(1, 10, 2)) 的值为__。（25）
197、 表达式 ‘abcab’.replace(‘a’,’yy’) 的值为_。（’yybcyyb’）
198、 已知 table = ”.maketrans(‘abcw’, ‘xyzc’)，那么表达式 ‘Hellow world’.translate(table) 的值为_____。（’Helloc corld’）
199、 表达式‘hello world, hellow every one’.replace(‘hello’,‘hi’) 的值为_____。（’hi world, hiw every one’）
200、 已知字符串 x = ‘hello world’，那么执行语句 x.replace(‘hello’, ‘hi’) 之后，x 的值为__。（’hello world’）
201、 正则表达式元字符__用来表示该符号前面的字符或子模式 1 次或多次出现。（+）
202、 已知 x = ‘a b c d’，那么表达式 ‘,’.join(x.split()) 的值为_。（’a,b,c,d’）
203、 正则表达式元字符__用来表示该符号前面的字符或子模式 0 次或多次出现。（*）
204、 表达式 ‘abcab’.strip(‘ab’) 的值为____。（’c’）
205、 表达式 [str(i) for i in range(3)] 的值为__。（[‘0’, ‘1’, ‘2’]）
206、 表达式 ‘abc.txt’.endswith((‘.txt’, ‘.doc’, ‘.jpg’)) 的值为_。（True）
207、 表达式 list(filter(None, [0,1,2,3,0,0])) 的值为_____。（[1, 2, 3]）
208、 表达式 list(filter(lambda x:x>2, [0,1,2,3,0,0])) 的值为___。（[3]）
209、 表达式 list(range(50, 60, 3)) 的值为_____。（[50, 53, 56, 59]）
210、 表达式 list(filter(lambda x: x%2==0, range(10))) 的值为_____。（[0, 2, 4, 6, 8]）
211、 表达式 list(filter(lambda x: len(x)>3, [‘a’, ‘b’, ‘abcd’])) 的值为_。（[‘abcd’]）
212、 Python 使用_____关键字来定义类。（class）
213、 表达式 isinstance(‘abc’, str) 的值为_____。（True）
214、 表达式 isinstance(‘abc’, int) 的值为_____。（False）
215、 表达式 isinstance(4j, (int, float, complex)) 的值为___。（True）
216、 表达式 isinstance(‘4’, (int, float, complex)) 的值为___。（False）
217、 表达式 type(3) in (int, float, complex) 的值为__。（True）
218、 表达式 type(3.0) in (int, float, complex) 的值为__。（True）
219、 表达式 type(3+4j) in (int, float, complex) 的值为__。（True）
220、 表达式 type(‘3’) in (int, float, complex) 的值为__。（False）
221、 表达式 type(3) == int 的值为____。（True）
222、 代码 print(1,2,3,sep=’:’) 的执行结果为_____。（1:2:3）
223、 代码 for i in range(3):print(i, end=’,’) 的执行结果为_____。（0,1,2,）
224、 表达式 eval(”’import(‘math’).sqrt(9)”’) 的值为____。（3.0）
225、 表达式 eval(”’import(‘math’).sqrt(3**2+4**2)”’) 的值为___。（5.0）
226、 表达式 eval(‘3+5’) 的值为_____。（8）
227、 表达式 eval(‘[1, 2, 3]’) 的值为_____。（[1, 2, 3]）

228、 假设 math 标准库已导入，那么表达式 eval（‘math.sqrt(4)’）的值为___。（2.0）
229、 已知 x 为非空列表，那么表达式 random.choice(x) in x 的值为_。（True）
230、 表达式 ‘abc10’.isalnum() 的值为____。（True）
231、 表达式 ‘abc10’.isalpha() 的值为_____。（False）
232、 表达式 ‘abc10’.isdigit() 的值为_____。（False）
233、 表达式 [1,2,3].count(4) 的值为_____。（0）
234、 Python 标准库 random 中的_方法作用是从序列中随机选择 1 个元素。（choice()）
235、 表达式 ‘C:\windows\notepad.exe’.endswith(‘.exe’) 的值为__。（True）
236、 Python 标准库 random 中的 sample(seq, k)方法作用是从序列中选择__(重复？不重复？)的 k 个元素。（不重复）
237、 random 模块中___方法的作用是将列表中的元素随机乱序。（shuffle()）
238、 Python 关键字 elif 表示__和___两个单词的缩写。（else、if）
239、 执行代码 x, y, z = sorted([1, 3, 2]) 之后，变量 y 的值为__。（2）
240、 已知 x = {1:2, 2:3}，那么表达式 x.get(3, 4) 的值为__。（4）
241、 已知 x = {1:2, 2:3}，那么表达式 x.get(2, 4) 的值为__。（3）
242、 表达式 {1, 2, 3} | {3, 4, 5} 的值为_____。（{1, 2, 3, 4, 5}）
243、 表达式 {1, 2, 3} | {2, 3, 4} 的值为_____。（{1, 2, 3, 4}）
244、 表达式 {1, 2, 3} & {3, 4, 5} 的值为____。（{3}）
245、 表达式 {1, 2, 3} & {2, 3, 4} 的值为_。（{2, 3}）
246、 表达式 {1, 2, 3} - {3, 4, 5} 的值为___。（{1, 2}）
247、 表达式 {1, 2, 3} < {3, 4, 5} 的值为___。（False）
248、 表达式 {1, 2, 3} < {1, 2, 4} 的值为_。（False）
249、 表达式 ‘%s’%[1,2,3] 的值为___。（’ [1, 2, 3]’）
250、 在 Python 定义类时，与运算符“**”对应的特殊方法名为 ___。（pow()）
251、 在 Python 中定义类时，与运算符“//”对应的特殊方法名为_____。（floordiv()）
252、 对文件进行写入操作之后，_____方法用来在不关闭文件对象的情况下将缓冲区内容写入文件。（flush()）
253、 Python 内置函数___用来打开或创建文件并返回文件对象。（open()）
254、 使用上下文管理关键字____可以自动管理文件对象，不论何种原因结束该关键字中的语句块，都能保证文件被正确关闭。（with）
255、 Python 标准库 os 中用来列出指定文件夹中的文件和子文件夹列表的方式是__。（listdir()）
256、 Python 标准库 os.path 中用来判断指定文件是否存在的方法是____。（exists()）
257、 Python 标准库 os.path 中用来判断指定路径是否为文件的方法是_____。（isfile()）
258、 Python 标准库 os.path 中用来判断指定路径是否为文件夹的方法是____。（isdir()）
259、 Python 标准库 os.path 中用来分割指定路径中的文件扩展名的方法是____。（splitext()）
260、 Python 内建异常类的基类是_____。（BaseException）
261、 Python 扩展库___支持 Excel 2007 或更高版本文件的读写操作。（openpyxl）
262、 Python 标准库__中提供了计算 MD5 摘要的方法 md5()。（hashlib）
263、 表达式 len（‘SDIBT’）的值为____。（5）
264、 表达式 ‘Hello world!’.count（‘l’）的值为_。（3）
265、 表达式 (1, 2, 3)+(4, 5) 的值为_____。（(1, 2, 3, 4, 5)）
266、 表达式 dict(zip([1, 2], [3, 4])) 的值为_____。（{1: 3, 2: 4}）
267、 已知 x = ‘abcdefg’，则表达式 x[3:] + x[:3] 的值为_____。（’ defgabc’）

268、 一直 g = lambda x, y=3, z=5: x*y*z，则语句 print(g(1)) 的输出结果为_。（15）
269、 表达式 list(map(lambda x: len(x), [‘a’, ‘bb’, ‘ccc’])) 的值为____。（[1, 2, 3]）
270、 语句 x, y, z = [1, 2, 3] 执行后，变量 y 的值为____。（2）
271、 Python 标准库____对 Socket 进行了二次封装，支持 Socket 接口的访问，大幅度简化了网络程序的开发。（socket）
272、 Python 扩展库____中封装了 Windows 底层几乎所有 API 函数。（pywin32）
273、 线程对象的_方法用来阻塞当前线程，指定线程运行结束或超时后继续运行当前线程。（join()）
274、 Python 用来访问和操作内置数据库 SQLite 的标准库是___。（sqlite3）
275、 用于删除数据库表 test 中所有 name 字段值为’10001’的记录的 SQL 语句为_____。（delete from test where name=’10001’）
276、 Python 扩展库_____完美封装了图形库 OpenGL 的功能。（pyopengl）
277、 Python 扩展库____和____提供了图像处理功能。（PIL、pillow）
278、 已知 x = [[1,3,3], [2,3,1]]，那么表达式 sorted(x, key=lambda item:item[0]+item[2]) 的值为_____。（[[2, 3, 1], [1, 3, 3]]）
279、 已知 x = [[1,3,3], [2,3,1]]，那么表达式 sorted(x, key=lambda item:(item[1],item[2])) 的值为_____。（[[2, 3, 1], [1, 3, 3]]）
280、 已知 x = [[1,3,3], [2,3,1]]，那么表达式 sorted(x, key=lambda item:(item[1], -item[2])) 的值为_____。（[[1, 3, 3], [2, 3, 1]]）
281、 已知 x = {1, 2, 3}，那么执行语句 x.add(3) 之后，x 的值为____。（{1, 2, 3}）
282、 已知 x = {1:1}，那么执行语句 x[2] = 2 之后，len(x)的值为__。（2）
283、 已知 x = {1:1, 2:2}，那么执行语句 x[2] = 4 之后，len(x)的值为__。（2）
284、 假设已从标准库 functools 导入 reduce()函数，那么表达式 reduce(lambda x, y: x-y, [1, 2, 3]) 的值为__。（-4）
285、 假设已从标准库 functools 导入 reduce()函数，那么表达式 reduce(lambda x, y: x+y, [1, 2, 3]) 的值为__。（6）
286、 已知有函数定义 def demo(*p):return sum(p)，那么表达式 demo(1, 2, 3) 的值为_、表达式 demo(1, 2, 3, 4) 的值为____。（6、10）
287、 已知列表 x = [1, 2]，那么连续执行命令 y = x 和 y.append(3) 之后，x 的值为__。（[1, 2, 3]）
288、 已知列表 x = [1, 2]，那么连续执行命令 y = x[:] 和 y.append(3) 之后，x 的值为__。（[1, 2]）
289、 已知列表 x = [1, 2]，执行语句 y = x[:] 后，表达式 id(x) == id(y) 的值为_。（False）
290、 已知列表 x = [1, 2]，执行语句 y = x 后，表达式 id(x) == id(y) 的值为_。（True）
291、 已知列表 x = [1, 2]，执行语句 y = x 后，表达式 x is y 的值为_。（True）
292、 已知列表 x = [1, 2]，执行语句 y = x[:] 后，表达式 x is not y 的值为_。（True）
293、 表达式 sorted(random.sample(range(5), 5)) 的值为_____。（[0, 1, 2, 3, 4]）
294、 表达式 [i for i in range(10) if i>8] 的值为_____。（[9]）
295、 已知有列表 x = [[1, 2, 3], [4, 5, 6]]，那么表达式 [[row[i] for row in x] for i in range(len(x[0]))] 的值为_____。（[[1, 4], [2, 5], [3, 6]]）
296、 执行语句 x,y,z = map(str, range(3)) 之后，变量 y 的值为_。（’1’）
297、 已知列表 x = [1, 2]，那么执行语句 x.extend([3]) 之后， x 的值为__。（[1, 2, 3]）
298、 已知列表 x = [1, 2]，那么执行语句 x.append([3]) 之后，x 的值为___。（[1, 2, [3]]）
299、 表达式 ‘aaasdf’.lstrip(‘as’) 的值为_____。（’df’）

300、 表达式 ‘aaasdf’.lstrip(‘af’) 的值为_____。(’sdf’)
301、 表达式 ‘aaasdf’.strip(‘af’) 的值为____。(’sd’)
302、 表达式 ‘aaasdf’.rstrip(‘af’) 的值为_____。(’aaasd’)
303、 已知 f = lambda x: x+5，那么表达式 f(3) 的值为__。(8)
304、 表达式 print(0b10101) 的值为__。(21)
305、 表达式 ‘\x41’ == ‘A’ 的值为___。(True)
306、 已知 x = [1, 2, 3, 4, 5]，那么执行语句 del x[:3] 之后，x 的值为____。([4, 5])
307、 表达式 sorted([‘abc’, ‘acd’, ‘ade’], key=lambda x:(x[0],x[2])) 的值为___。([‘abc’, ‘acd’, ‘ade’])
308、 已知 x = range(1,4) 和 y = range(4,7)，那么表达式 sum([i*j for i,j in zip(x,y)]) 的值为____。(32)
309、 表达式 [5 for i in range(3)] 的值为_____。([5, 5, 5])
310、 表达式 {1, 2, 3} == {1, 3, 2} 的值为_。(True)
311、 表达式 [1, 2, 3] == [1, 3, 2] 的值为__。(False)
312、 已知 x = [1, 2, 1]，那么表达式 id(x[0]) == id(x[2]) 的值为_____。(True)
313、 表达式 3 not in [1, 2, 3]的值为____。(False)
314、 已知 x = [1, 2]，那么执行语句 x[0:0] = [3, 3]之后，x 的值为_。([3, 3, 1, 2])
315、 已知 x = [1, 2]，那么执行语句 x[0:1] = [3, 3]之后，x 的值为_。([3, 3, 2])
316、 已知 x = [1, 2, 3, 4, 5]，那么执行语句 del x[1:3] 之后，x 的值为__。([1, 4, 5])
317、 已知 x = [[1, 2, 3,], [4, 5, 6]]，那么表达式 sum([i*j for i,j in zip(*x)]) 的值为__。(32)
318、 已知列表 x = [1, 2, 3] 和 y = [4, 5, 6]，那么表达式 [(i,j) for i, j in zip(x,y) if i==3] 的值为__。([(3, 6)])
319、 已知列表 x = [1.0, 2.0, 3.0]，那么表达式 sum(x)/len(x) 的值为_。(2.0)
320、 表达式 ‘abc’ in (‘abcdefg’) 的值为____。(True)
321、 表达式 ‘abc’ in [‘abcdefg’] 的值为____。(False)
322、 已知 x = {1:2, 2:3, 3:4}，那么表达式 sum(x) 的值为_。(6)
323、 已知 x = {1:2, 2:3, 3:4}，那么表达式 sum(x.values()) 的值为___。(9)
324、 已知 x = [3, 2, 3, 3, 4]，那么表达式 [index for index, value in enumerate(x) if value==3] 的值为__。([0, 2, 3])
325、 表达式 1234%1000//100 的值为_。(2)
326、 正则表达式模块 re 的____方法用来编译正则表达式对象。(compile())
327、 正则表达式模块 re 的____方法用来在字符串开始处进行指定模式的匹配。(match())
328、 正则表达式模块 re 的____方法用来在整个字符串中进行指定模式的匹配。(search())
329、 表达式 re.search(r’\w*?(?P\b\w+\b)\s+(?P=f)\w*?’, ‘Beautiful is is better than ugly.’).group(0) 的值为_。(’is is’)
330、 已知 g = lambda x, y=3, z=5: x+y+z，那么表达式 g(2) 的值为__。(10)
331、 假设有 Python 程序文件 abc.py，其中只有一条语句 print(name)，那么直接运行该程序时得到的结果为___。(main)
332、 表达式 3 in {1, 2, 3} 的值为___。(True)
333、 表达式 ‘ac’ in ‘abce’ 的值为___。(False)
334、 表达式 not 3 的值为_____。(False)
335、 表达式 3 // 5 的值为_____。(0)
336、 表达式 [1, 2] + [3] 的值为_____。([1, 2, 3])
337、 表达式 (1,) + (2,) 的值为___。((1, 2))

338、 表达式 (1) + (2) 的值为__。（3）
339、 已知 x, y = map(int, [‘1’, ‘2’])，那么表达式 x + y 的值为_。（3）
340、 已知列表 x = list(range(5))，那么执行语句 x.remove(3) 之后，表达式 x.index(4) 的值为__。（3）
341、 已知列表 x = [1, 3, 2]，那么执行语句 x.reverse() 之后，x 的值为__。（[2, 3, 1]）
342、 已知列表 x = [1, 3, 2]，那么执行语句 x = x.reverse() 之后，x 的值为__。（None）
343、 已知 x 为非空列表，那么表达式 x.reverse() == list(reversed(x)) 的值为____。（False）
344、 已知 x 为非空列表，那么表达式 x.sort() == sorted(x) 的值为____。（False）
345、 已知列表 x = [1, 3, 2]，那么执行语句 y = list(reversed(x)) 之后，x 的值为____。（[1, 3, 2]）
346、 已知列表 x = [1, 3, 2]，那么执行语句 y = list(reversed(x)) 之后，y 的值为____。（[2, 3, 1]）
347、 表达式 ‘Beautiful is better than ugly.’.startswith(‘Be’, 5) 的值为___。（False）
348、 已知列表 x 中包含超过 5 个以上的元素，那么表达式 x == x[:5]+x[5:] 的值为__。（True）
349、 已知字典 x = {i:str(i+3) for i in range(3)}，那么表达式 sum(x) 的值为__。（3）
350、 已知字典 x = {i:str(i+3) for i in range(3)}，那么表达式 ”.join(x.values()) 的值为__。（’345’）
351、 已知字典 x = {i:str(i+3) for i in range(3)}，那么表达式 sum(item[0] for item in x.items()) 的值为_。（3）
352、 已知字典 x = {i:str(i+3) for i in range(3)}，那么表达式 ”.join([item[1] for item in x.items()]) 的值为___。（’345’）
353、 已知列表 x = [1, 3, 2]，那么表达式 [value for index, value in enumerate(x) if index==2] 的值为_____。（[2]）
354、 已知列表 x = [1, 3, 2]，那么执行语句 a, b, c = sorted(x) 之后，b 的值为__。（2）
355、 已知列表 x = [1, 3, 2]，那么执行语句 a, b, c = map(str,sorted(x)) 之后，c 的值为__。（’3’）
356、 表达式 set([1,2,3]) == {1, 2, 3} 的值为__。（True）
357、 表达式 set([1,2, 2,3]) == {1, 2, 3} 的值为__。（True）
358、 表达式 ‘%c’%65 == str(65) 的值为_。（False）
359、 表达式 ‘%s’%65 == str(65) 的值为___。（True）
360、 表达式 chr(ord(‘b’)^32) 的值为___。（’B’）
361、 表达式 ‘abc’ in ‘abdcefg’ 的值为___。（False）
362、 已知函数定义 def func(*p):return sum(p)，那么表达式 func(1,2,3) 的值为__。（6）
363、 已知函数定义 def func(*p):return sum(p)，那么表达式 func(1,2,3, 4) 的值为__。（10）
364、 已知函数定义 def func(**p):return sum(p.values())，那么表达式 func(x=1, y=2, z=3) 的值为__。（6）
365、 已知函数定义 def func(**p):return ”.join(sorted(p))，那么表达式 func(x=1, y=2, z=3) 的值为____。（’xyz’）
366、 已知 x 为整数变量，那么表达式 int(hex(x), 16) == x 的值为___。（True）
367、 已知 f = lambda x: 5，那么表达式 f(3)的值为___。（5）
368、 已知 x, y = 3, 5，那么执行 x, y = y, x 之后，x 的值为____。（5）
369、 已知 x = ‘abcd’ 和 y = ‘abcde’，那么表达式 [i==j for i,j in zip(x,y)] 的值为_____。（[True, True, True, True]）

370、 表达式 16**0.5 的值为_____。（4.0）
371、 表达式 type({3})的值为_____。（set）
372、 表达式 isinstance(‘Hello world’, str)的值为_____。（True）
373、 已知 x = list(range(20))，那么表达式 x[-1]的值为_____。（19）
374、 已知 x = 3+4j 和 y = 5+6j，那么表达式 x+y 的值为_____。（8+10j）
375、 已知 x = [3]，那么执行 x += [5]之后 x 的值为_____。（[3, 5]）
376、 已知 x = [3, 3, 4]，那么表达式 id(x[0])==id(x[1])的值为_____。（True）
377、 表达式 int(‘11’, 2)的值为_____。（3）
378、 表达式 int(‘11’, 8)的值为_____。（9）
379、 表达式 int(bin(54321), 2)的值为_____。（54321）
380、 表达式 chr(ord(‘A’)+1)的值为_____。(’B’)
381、 表达式 int(str(34)) == 34 的值为_____。（True）
382、 表达式 list(str([3, 4])) == [3, 4]的值为_____。（False）
383、 表达式{1, 2, 3, 4, 5, 6} ^ {5, 6, 7, 8}的值为_____。（{1, 2, 3, 4, 7, 8}）
384、 表达式 15 // 4 的值为_____。（3）
385、 表达式 sorted({‘a’:3, ‘b’:9, ‘c’:78})的值为_____。([‘a’, ‘b’, ‘c’])
386、 表达式 sorted({‘a’:3, ‘b’:9, ‘c’:78}.values())的值为___。([3, 9, 78])
387、 已知 x = [3, 2, 4, 1]，那么执行语句 x = x.sort()之后，x 的值为__。（None）
388、 表达式 list(filter(lambda x: x>5, range(10)))的值为_____。（[6, 7, 8, 9]）
389、 已知 x = list(range(20))，那么语句 print(x[100:200])的输出结果为_____。（[]）
390、 已知 x = list(range(20))，那么执行语句 x[:18] = []后列表 x 的值为____。（[18, 19]）
391、 已知 x = [1, 2, 3]，那么连续执行 y = x[:]和 y.append(4)这两条语句之后，x 的值为_____。（[1, 2, 3]）
392、 已知 x = [1, 2, 3]，那么连续执行 y = x 和 y.append(4)这两条语句之后，x 的值为_____。（[1, 2, 3, 4]）
393、 已知 x = [1, 2, 3]，那么连续执行 y = [1, 2, 3]和 y.append(4)这两条语句之后，x 的值为_____。（[1, 2, 3]）
394、 已知 x = [[]] * 3，那么执行语句 x[0].append(1)之后，x 的值为_____。（[[1], [1], [1]]）
395、 已知 x = [[] for i in range(3)]，那么执行语句 x[0].append(1)之后，x 的值为_____。（[[1], [], []]）
396、 已知 x = ([1], [2])，那么执行语句 x[0].append(3)后 x 的值为_____。（([1, 3], [2])）
397、 已知 x = {1:1, 2:2}，那么执行语句 x.update({2:3, 3:3})之后，表达式 sorted(x.items())的值为_____。（[(1, 1), (2, 3), (3, 3)]）
398、 已知 x = {1:1, 2:2}，那么执行语句 x[3] = 3 之后，表达式 sorted(x.items())的值为_____。（[(1, 1), (2, 2), (3, 3)]）
399、 表达式 type({}) == dict 的值为_____。（True）
400、 表达式 type({}) == set 的值为_____。（False）
401、 已知 x = [1, 2, 3]，那么表达式 not (set(x*100)-set(x))的值为_____。（True）
402、 已知 x = [1, 2, 3]，那么表达式 not (set(x*100)&set(x))的值为_____。（False）
403、 表达式{‘x’: 1, **{‘y’: 2}}的值为_____。({‘x’: 1, ‘y’: 2})
404、 表达式{range(4), 4, (5, 6, 7)}的值为_____。（{0, 1, 2, 3, 4, 5, 6, 7}）
405、 在 Python 中，不论类的名字是什么，构造方法的名字都是_____。（init）
406、 如果在设计一个类时实现了 contains ()方法，那么该类的对象会自动支持___运算符。

（in）

407、 已知函数定义 def demo(x, y, op):return eval(str(x)+op+str(y))，那么表达式 demo(3, 5, ‘+’)的值为_____。（8）

408、 已知函数定义 def demo(x, y, op):return eval(str(x)+op+str(y))，那么表达式 demo(3, 5, ‘*’)的值为_____。（15）

409、 已知函数定义 def demo(x, y, op):return eval(str(x)+op+str(y))，那么表达式 demo(3, 5, ‘-‘)的值为_____。(-2）

410、 字符串编码格式 UTF8 使用_____个字节表示一个汉字。（3）

411、 字符串编码格式 GBK 使用_____个字节表示一个汉字。（2）

412、 已知字符串编码格式 utf8 使用 3 个字节表示一个汉字、1 个字节表示英语字母，那么表达式 len(‘abc 你好’)的值为_____。（5）

413、 已知字符串编码格式 utf8 使用 3 个字节表示一个汉字、1 个字节表示英语字母，那么表达式 len(‘abc 你好’.encode())的值为_____。（9）

414、 已知字符串编码格式 gbk 使用 2 个字节表示一个汉字、1 个字节表示英语字母，那么表达式 len(‘abc 你好’.encode(‘gbk’))的值为_____。（7）

415、 已知 ord(‘A’)的值为 65 并且 hex(65)的值为’0x41’，那么表达式’\x41b’的值为_____。(’Ab’）

416、 已知 formatter = ‘good{0}’.format，那么表达式 list(map(formatter, [‘morning’]))的值为_____。([‘good morning’]）

417、 已知 x = ‘hello world.’，那么表达式 x.find(‘x’)和 x.rfind(‘x’)的值都为___。(-1）

418、 表达式’:’.join(‘hello world.’.split())的值为_____。(’hello:world.’）

419、 表达式’:’.join(‘a b c d’.split(maxsplit=2))的值为_____。(’a:b:c d’）

420、 已知 x = ‘hello world’，那么表达式 x.replace(‘l’, ‘g’)的值为_____。(’heggo worgd’）

421、 假设已成功导入 Python 标准库 string，那么表达式 len(string.digits)的值为_____。（10）

422、 表达式’aaaassddf’.strip(‘af’)的值为_____。(’ssdd’）

423、 表达式 len(‘aaaassddf’.strip(‘afds’))的值为_____。(0）

424、 表达式 len(‘hello world’[100:])的值为_____。(0）

425、 表达式 chr(ord(‘a’)^32^32)的值为_____。(’a’）

426、 表达式 chr(ord(‘a’)^32)的值为_____。(’A’）

427、 已知 x = ‘aa b ccc dddd’，那么表达式”.join([v for i,v in enumerate(x[:-1]) if v==x[i+1]])的值为_____。(’accddd’）

428、 已知当前文件夹中有纯英文文本文件 readme.txt，请填空完成功能把 readme.txt 文件中的所有内容复制到 dst.txt 中，with open(‘readme.txt’) as src, open(‘dst.txt’, __) as dst:dst.write(src.read())。(’w’）

429、 假设正则表达式模块 re 已正确导入，那么表达式”.join(re.findall(‘\d+’, ‘abcd1234’))的值为_____。(’1234’）

430、 假设正则表达式模块 re 已正确导入，那么表达式 re.findall(‘\d+?’, ‘abcd1234’)的值为_____。([‘1’, ‘2’, ‘3’, ‘4’]）

431、 假设正则表达式模块 re 已正确导入，那么表达式 re.sub(‘(.\s)\1+’, ‘\1’,’a a a a a bb’)的值为_____。(’a bb’）

432、 Python 标准库_____提供了对 SQLite 数据库的访问接口。(sqlite3）

433、

二、 判断题

1、 Python 是一种跨平台、开源、免费的高级动态编程语言。（对）

2、 Python 3.x 完全兼容 Python 2.x。（错）

3、 Python 3.x 和 Python 2.x 唯一的区别就是：print 在 Python 2.x 中是输出语句，而在 Python 3.x 中是输出函数。（错）

4、 在 Windows 平台上编写的 Python 程序无法在 Unix 平台运行。（错）

5、 不可以在同一台计算机上安装多个 Python 版本。（错）

6、 已知 x = 3，那么赋值语句 x = ‘abcedfg’ 是无法正常执行的。（错）

7、 继承自 threading.Thread 类的派生类中不能有普通的成员方法。（错）

8、 扩展库 os 中的方法 remove()可以删除带有只读属性的文件。（错）

9、 使用内置函数 open()且以”w”模式打开的文件，文件指针默认指向文件尾。（错）

10、 使用内置函数 open()打开文件时，只要文件路径正确就总是可以正确打开的。（错）

11、Python 变量使用前必须先声明，并且一旦声明就不能再当前作用域内改变其类型。（错）

12、 Python 采用的是基于值得自动内存管理方式。（对）

13、 在任何时刻相同的值在内存中都只保留一份（错）

14、 Python 不允许使用关键字作为变量名，允许使用内置函数名作为变量名，但这会改变函数名的含义。（对）

15、 在 Python 中可以使用 if 作为变量名。（错）

16、 在 Python 3.x 中可以使用中文作为变量名。（对）

17、 Python 变量名必须以字母或下划线开头，并且区分字母大小写。（对）

18、 加法运算符可以用来连接字符串并生成新字符串。（对）

19、 9999**9999 这样的命令在 Python 中无法运行。（错）

20、 3+4j 不是合法的 Python 表达式。（错）

21、 0o12f 是合法的八进制数字。（错）

22、 Python 2.x 和 Python 3.x 中 input()函数的返回值都是字符串。（错）

23、 pip 命令也支持扩展名为.whl 的文件直接安装 Python 扩展库。（对）

24、 只有 Python 扩展库才需要导入以后才能使用其中的对象，Python 标准库不需要导入即可使用其中的所有对象和方法。（错）

25、 在 Python 中 0xad 是合法的十六进制数字表示形式。（对）

26、 3+4j 是合法 Python 数字类型。（对）

27、 在 Python 中 0oa1 是合法的八进制数字表示形式。（错）

28、 Python 使用缩进来体现代码之间的逻辑关系。（对）

29、 Python 代码的注释只有一种方式，那就是使用#符号。（错）

30、 调用函数时，在实参前面加一个型号*表示序列解包。（对）

31、 放在一对三引号之间的任何内容将被认为是注释。（错）

32、 Python 支持使用字典的“键”作为下标来访问字典中的值。（对）

33、 列表可以作为字典的“键”。（错）

34、 元组可以作为字典的“键”。（对）

35、 字典的“键”必须是不可变的。（对）

36、 尽管可以使用 import 语句一次导入任意多个标准库或扩展库，但是仍建议每次只导入一个标准库或扩展库。（对）

37、 为了让代码更加紧凑，编写 Python 程序时应尽量避免加入空格和空行。（错）

38、 在 Python 3.5 中运算符+不仅可以实现数值的相加、字符串连接，还可以实现列表、元

组的合并和集合的并集运算。（错）
39、 已知 x 为非空列表，那么表达式 sorted(x, reverse=True) == list(reversed(x)) 的值一定是 True。（错）
40、 已知 x 为非空列表，那么 x.sort(reverse=True)和 x.reverse()的作用是等价的。（错）
41、 生成器推导式比列表推导式具有更高的效率，推荐使用。（对）
42、 Python 集合中的元素不允许重复。（对）
43、 Python 集合可以包含相同的元素。（错）
44、 Python 字典中的“键”不允许重复。（对）
45、 Python 字典中的“值”不允许重复。（错）
46、 Python 集合中的元素可以是元组。（对）
47、 Python 集合中的元素可以是列表。（错）
48、 Python 字典中的“键”可以是列表。（错）
49、 Python 字典中的“键”可以是元组。（对）
50、 Python 列表中所有元素必须为相同类型的数据。（错）
51、 Python 列表、元组、字符串都属于有序序列。（对）
52、 在 Python 3.x 中语句 print(*[1,2,3]) 不能正确执行。（错）
53、 已知 A 和 B 是两个集合，并且表达式 A < B 的值为 False，那么表达式 A > B 的值一定为 True。（错）
54、 列表对象的 append()方法属于原地操作，用于在列表尾部追加一个元素。（对）
55、 对于列表而言，在尾部追加元素比在中间位置插入元素速度更快一些，尤其是对于包含大量元素的列表。（对）
56、 假设有非空列表 x，那么 x.append(3)、x = x+[3]与 x.insert(0,3)在执行时间上基本没有太大区别。（错）
57、 使用 Python 列表的方法 insert()为列表插入元素时会改变列表中插入位置之后元素的索引。（对）
58、 假设 x 为列表对象，那么 x.pop()和 x.pop(-1)的作用是一样的。（对）
59、 使用 del 命令或者列表对象的 remove()方法删除列表中元素时会影响列表中部分元素的索引。（对）
60、 带有 else 子句的循环如果因为执行了 break 语句而退出的话，则会执行 else 子句中的代码。（错）
61、 对于带有 else 子句的循环语句，如果是因为循环条件表达式不成立而自然结束循环，则执行 else 子句中的代码。（对）
62、 已知列表 x = [1, 2, 3]，那么执行语句 x = 3 之后，变量 x 的地址不变。（错）
63、 在 UTF-8 编码中一个汉字需要占用 3 个字节。（对）
64、 在 GBK 和 CP936 编码中一个汉字需要 2 个字节。（对）
65、 如果仅仅是用于控制循环次数，那么使用 for i in range(20)和 for i in range(20, 40)的作用是等价的。（对）
66、 使用列表对象的 remove()方法可以删除列表中首次出现的指定元素，如果列中不存在要删除的指定元素则抛出异常。（对）
67、 元组是不可变的，不支持列表对象的 inset()、remove()等方法，也不支持 del 命令删除其中的元素，但可以使用 del 命令删除整个元组对象。（对）
68、 Python 字典和集合属于无序序列。（对）
69、 无法删除集合中指定位置的元素，只能删除特定值的元素。（对）
70、 元组的访问速度比列表要快一些，如果定义了一系列常量值，并且主要用途仅仅是对

其进行遍历二不需要进行任何修改，建议使用元组而不使用列表。（对）
71、 当以指定“键”为下标给字典对象赋值时，若该“键”存在则表示修改该“键”对应的“值”，若不存在则表示为字典对象添加一个新的“键-值对”。（对）
72、 假设 x 是含有 5 个元素的列表，那么切片操作 x[10:]是无法执行的，会抛出异常。（错）
73、 只能对列表进行切片操作，不能对元组和字符串进行切片操作。（错）
74、 只能通过切片访问列表中的元素，不能使用切片修改列表中的元素。（错）
75、 只能通过切片访问元组中的元素，不能使用切片修改元组中的元素。（对）
76、 字符串属于 Python 有序序列，和列表、元组一样都支持双向索引。（对）
77、 Python 字典和集合支持双向索引。（错）
78、 使用 print()函数无法将信息写入文件。（错）
79、 Python 集合不支持使用下标访问其中的元素。（对）
80、 相同内容的字符串使用不同的编码格式进行编码得到的结果并不完全相同。（对）
81、 删除列表中重复元素最简单的方法是将其转换为集合后再重新转换为列表。（对）
82、 已知列表 x 中包含超过 5 个以上的元素，那么语句 x = x[:5]+x[5:] 的作用是将列表 x 中的元素循环左移 5 位。（错）
83、 对于生成器对象 x = (3 for i in range(5))，连续两次执行 list(x)的结果是一样的。（错）
84、 在循环中 continue 语句的作用是跳出当前循环。（错）
85、 在编写多层循环时，为了提高运行效率，应尽量减少内循环中不必要的计算。（对）
86、 在 Python 中，任意长的字符串都遵守驻留机制。（错）
87、 Python 运算符%不仅可以用来求余数，还可以用来格式化字符串。（对）
88、 Python 字符串方法 replace()对字符串进行原地修改。（错）
89、 如果需要连接大量字符串成为一个字符串，那么使用字符串对象的 join()方法比运算符+具有更高的效率。（对）
90、 对于大量列表的连接，extend()方法比运算符+具有更高的效率。（对）
91、 表达式 {1, 3, 2} > {1, 2, 3} 的值为 True。（错）
92、 列表对象的 extend()方法属于原地操作，调用前后列表对象的地址不变。（对）
93、 正则表达式模块 re 的 match()方法是从字符串的开始匹配特定模式，而 search()方法是在整个字符串中寻找模式，这两个方法如果匹配成功则返回 match 对象，匹配失败则返回空值 None。（对）
94、 函数是代码复用的一种方式。（对）
95、 定义函数时，即使该函数不需要接收任何参数，也必须保留一对空的圆括号来表示这是一个函数。（对）
96、 编写函数时，一般建议先对参数进行合法性检查，然后再编写正常的功能代码。（对）
97、 一个函数如果带有默认值参数，那么必须所有参数都设置默认值。（错）
98、 定义 Python 函数时必须指定函数返回值类型。（错）
99、 定义 Python 函数时，如果函数中没有 return 语句，则默认返回空值 None。（对）
100、 如果在函数中有语句 return 3，那么该函数一定会返回整数 3。（错）
101、 函数中必须包含 return 语句。（错）
102、 函数中的 return 语句一定能够得到执行。（错）
103、 不同作用域中的同名变量之间互相不影响，也就是说，在不同的作用域内可以定义同名的变量。（对）
104、 全局变量会增加不同函数之间的隐式耦合度，从而降低代码可读性，因此应尽量避免过多使用全局变量。（对）
105、 函数内部定义的局部变量当函数调用结束后被自动删除。（对）

106、 在函数内部，既可以使用 global 来声明使用外部全局变量，也可以使用 global 直接定义全局变量。（对）
107、 在函数内部没有办法定义全局变量。（错）
108、 对于数字 n，如果表达式 0 not in [n%d for d in range(2, n)] 的值为 True 则说明 n 是素数。（对）
109、 表达式 ‘a’+1 的值为’b’。（错）
110、 在函数内部直接修改形参的值并不影响外部实参的值。（对）
111、 在函数内部没有任何方法可以影响实参的值。（错）
112、 调用带有默认值参数的函数时，不能为默认值参数传递任何值，必须使用函数定义时设置的默认值。（错）
113、 创建只包含一个元素的元组时，必须在元素后面加一个逗号，例如(3,)。（对）
114、 在同一个作用域内，局部变量会隐藏同名的全局变量。（对）
115、 形参可以看做是函数内部的局部变量，函数运行结束之后形参就不可访问了。（对）
116、 假设已导入 random 标准库，那么表达式 max([random.randint(1, 10) for i in range(10)]) 的值一定是 10。（错）
117、 Python 标准库 random 的方法 randint(m,n)用来生成一个[m,n]区间上的随机整数。（对）
118、 Python 中一切内容都可以称为对象。（对）
119、 栈和队列的都具有先入后出的特点。（错）
120、 在一个软件的设计与开发中，所有类名、函数名、变量名都应该遵循统一的风格和规范。（对）
121、 定义类时所有实例方法的第一个参数用来表示对象本身，在类的外部通过对象名来调用实例方法时不需要为该参数传值。（对）
122、 在面向对象程序设计中，函数和方法是完全一样的，都必须为所有参数进行传值。（错）
123、 Python 中没有严格意义上的私有成员。（对）
124、 在 Python 中定义类时，运算符重载是通过重写特殊方法实现的。例如，在类中实现了 mul()方法即可支持该类对象的**运算符。（对）
125、 在 IDLE 交互模式下，一个下划线“_”表示解释器中最后一次显示的内容或最后一次语句正确执行的输出结果。（对）
126、 对于 Python 类中的私有成员，可以通过“对象名.类名_私有成员名”的方式来访问。（对）
127、 运算符 / 在 Python 2.x 和 Python 3.x 中具有相同的功能。（错）
128、 运算符“-”可以用于集合的差集运算。（对）
129、 如果定义类时没有编写析构函数，Python 将提供一个默认的析构函数进行必要的资源清理工作。（对）
130、 已知 seq 为长度大于 10 的列表，并且已导入 random 模块，那么[random.choice(seq) for i in range(10)]和 random.sample(seq,10)等价。（错）
131、 在派生类中可以通过“基类名.方法名()”的方式来调用基类中的方法。（对）
132、 Python 支持多继承，如果父类中有相同的方法名，而在子类中调用时没有指定父类名，则 Python 解释器将从左向右按顺序进行搜索。（对）
133、 对文件进行读写操作之后必须显式关闭文件以确保所有内容都得到保存。（对）
134、 Python 标准库 os 中的方法 startfile()可以启动任何已关联应用程序的文件，并自动调用关联的程序。（对）

135、 程序中异常处理结构在大多数情况下是没必要的。（错）
136、 在 try…except…else 结构中，如果 try 块的语句引发了异常则会执行 else 块中的代码。（错）
137、 Python 标准库 threading 中的 Lock、RLock、Condition、Event、Semaphore 对象都可以用来实现线程同步。（对）
138、 异常处理结构中的 finally 块中代码仍然有可能出错从而再次引发异常。（对）
139、 在 GUI 设计中，复选框往往用来实现非互斥多选的功能，多个复选框之间的选择互不影响。（对）
140、 在 GUI 设计中，单选按钮用来实现用户在多个选项中的互斥选择，在同一组内多个选项中只能选择一个，当选择发生变化之后，之前选中的选项自动失效。（对）
141、 在 Python 中定义类时实例方法的第一个参数名称必须是 self。（错）
142、 在 Python 中定义类时实例方法的第一个参数名称不管是什么，都表示对象自身。（对）
143、 Python 代码可以内嵌在 asp 文件中。（对）
144、 无法配置 IIS 来支持 Python 程序的运行。（错）
145、 Python 标准库 os 中的方法 startfile()可以用来打开外部程序或文件，系统会自动关联相应的程序来打开或执行指定的文件。（对）
146、 在编写应用程序时，应合理控制线程数量，线程并不是越多越好。（对）
147、 在多线程编程时，当某子线程的 daemon 属性为 False 时，主线程结束时会检测该子线程是否结束，如果该子线程尚未运行结束，则主线程会等待它完成后再退出。（对）
148、 Python 只能使用内置数据库 SQLite，无法访问 MS SQLServer、ACCESS 或 Oracle、MySQL 等数据库。（错）
149、 使用 OpenGL 画图时，画点是最基本的操作，具体生成的图形由 glBegin()函数指定的 mode 来决定。例如，mode 值为 GL_TRIANGLES 时表示将要绘制三角形。（对）
150、 OpenGL 采用的“状态机”工作方式，一旦设置了某种状态以后，除非显式修改该状态，否则该状态将一直保持。（对）
151、 假设 os 模块已导入，那么列表推导式 [filename for filename in os.listdir('C:\Windows') if filename.endswith('.exe')] 的作用是列出 C:\Windows 文件夹中所有扩展名为.exe 的文件。（对）
152、 表达式 list('[1, 2, 3]') 的值是[1, 2, 3]。（错）
153、 在函数内部没有任何声明的情况下直接为某个变量赋值，这个变量一定是函数内部的局部变量。（对）
154、 定义类时如果实现了 contains()方法，该类对象即可支持成员测试运算 in。（对）
155、 定义类时如果实现了 len()方法，该类对象即可支持内置函数 len()。（对）
156、 定义类时实现了 eq()方法，该类对象即可支持运算符==。（对）
157、 定义类时实现了 pow()方法，该类对象即可支持运算符**。（对）
158、 二进制文件不能使用记事本程序打开。（错）
159、 使用普通文本编辑器软件也可以正常查看二进制文件的内容。（错）
160、 二进制文件也可以使用记事本或其他文本编辑器打开，但是一般来说无法正常查看其中的内容。（对）
161、 Python 标准库 os 中的方法 isfile()可以用来测试给定的路径是否为文件。（对）
162、 Python 标准库 os 中的方法 exists()可以用来测试给定路径的文件是否存在。（对）
163、 Python 标准库 os 中的方法 isdir()可以用来测试给定的路径是否为文件夹。（对）
164、 Python 标准库 os 中的方法 listdir()返回包含指定路径中所有文件和文件夹名称的列

表。（对）
165、 Python 扩展库 xlwt 支持对 Excel 2003 或更低版本的 Excel 文件进行写操作。（对）
166、 Python 扩展库 xlrd 支持对 Excel 2003 或更低版本的 Excel 文件进行读操作。（对）
167、 带有 else 子句的异常处理结构，如果不发生异常则执行 else 子句中的代码。（对）
168、 异常处理结构也不是万能的，处理异常的代码也有引发异常的可能。（对）
169、 在异常处理结构中，不论是否发生异常，finally 子句中的代码总是会执行的。（对）
170、 在 Python 中定义函数时不需要声明函数参数的类型。（对）
171、 在 Python 中定义函数时不需要声明函数的返回值类型。（对）
172、 在函数中没有任何办法可以通过形参来影响实参的值。（错）
173、 已知 x = 3，那么执行语句 x+=6 之后，x 的内存地址不变。（错）
174、 已知 x 为非空字符串，那么表达式 ”.join(x.split()) == x 的值一定为 True。（错）
175、 已知 x 为非空字符串，那么表达式 ‘,’.join(x.split(‘,’)) == x 的值一定为 True。（对）
176、 在 Python 中可以使用 for 作为变量名。（错）
177、 在 Python 中可以使用 id 作为变量名，尽管不建议这样做。（对）
178、 Python 关键字不可以作为变量名。（对）
179、 一个数字 5 也是合法的 Python 表达式。（对）
180、 同一个列表对象中的元素类型可以各不相同。（对）
181、 同一个列表对象中所有元素必须为相同类型。（错）
182、 已知 x 为非空列表，那么执行语句 x[0] = 3 之后，列表对象 x 的内存地址不变。（对）
183、 列表可以作为集合的元素。（错）
184、 集合可以作为列表的元素。（对）
185、 元组可以作为集合的元素。（对）
186、 集合可以作为元组的元素。（对）
187、 字典可以作为集合的元素。（错）
188、 集合可以作为字典的键。（错）
189、 集合可以作为字典的值。（对）
190、 可以使用 del 删除集合中的部分元素。（错）
191、 标准库 os 的 rename()方法可以实现文件移动操作。（对）
192、 标准库 os 的 listdir()方法默认只能列出指定文件夹中当前层级的文件和文件夹列表，而不能列出其子文件夹中的文件。（对）
193、 当作为条件表达式时，[]与 None 等价。（对）
194、 表达式 [] == None 的值为 True。（错）
195、 当作为条件表达式时，{}与 None 等价。（对）
196、 表达式 {}==None 的值为 True。（错）
197、 表达式 pow(3,2) == 3**2 的值为 True。（对）
198、 当作为条件表达式时，空值、空字符串、空列表、空元组、空字典、空集合、空迭代对象以及任意形式的数字 0 都等价于 False。（对）
199、 在定义函数时，某个参数名字前面带有一个*符号表示可变长度参数，可以接收任意多个普通实参并存放于一个元组之中。（对）
200、 在定义函数时，某个参数名字前面带有两个*符号表示可变长度参数，可以接收任意多个关键参数并将其存放于一个字典之中。（对）
201、 定义函数时，带有默认值的参数必须出现在参数列表的最右端，任何一个带有默认值的参数右边不允许出现没有默认值的参数。（对）

202、 在调用函数时，可以通过关键参数的形式进行传值，从而避免必须记住函数形参顺序的麻烦。（对）
203、 在调用函数时，必须牢记函数形参顺序才能正确传值。（错）
204、 调用函数时传递的实参个数必须与函数形参个数相等才行。（错）
205、 正则表达式对象的 match()方法可以在字符串的指定位置开始进行指定模式的匹配。（对）
206、 使用正则表达式对字符串进行分割时，可以指定多个分隔符，而字符串对象的 split()方法无法做到这一点。（对）
207、 在编写函数时，建议首先对形参进行类型检查和数值范围检查之后再编写功能代码，或者使用异常处理结构，尽量避免代码抛出异常而导致程序崩溃。（对）
208、 执行语句 from math import sin 之后，可以直接使用 sin() 函数，例如 sin(3)。（对）
209、 列表对象的 pop()方法默认删除并返回最后一个元素，如果列表已空则抛出异常。（对）
210、 在 Python 中定义类时，如果某个成员名称前有 2 个下划线则表示是私有成员。（对）
211、 在类定义的外部没有任何办法可以访问对象的私有成员。（错）
212、 可以使用 py2exe 或 pyinstaller 等扩展库把 Python 源程序打包成为 exe 文件，从而脱离 Python 环境在 Windows 平台上运行。（对）
213、 Python 程序只能在安装了 Python 环境的计算机上以源代码形式运行。（错）
214、 不同版本的 Python 不能安装到同一台计算机上。（错）
215、 一般来说，Python 扩展库没有通用于所有版本 Python 的，安装时应选择与已安装 Python 的版本对应的扩展库。
216、 表达式 {1, 2} * 2 的值为 {1, 2, 1, 2}。（错）
217、 Python 变量名区分大小写，所以 student 和 Student 不是同一个变量。（对）
218、 正则表达式元字符“^”一般用来表示从字符串开始处进行匹配，用在一对方括号中的时候则表示反向匹配，不匹配方括号中的字符。（对）
219、 正则表达式元字符“\s”用来匹配任意空白字符。（对）
220、 正则表达式 元字符“\d”用来匹配任意数字字符。（对）
221、 lambda 表达式中可以使用任意复杂的表达式，但是必须只编写一个表达式。（对）
222、 Python 类的构造函数是 init()。（对）
223、 定义类时，在一个方法前面使用@classmethod 进行修饰，则该方法属于类方法。（对）
224、 定义类时，在一个方法前面使用@staticmethod 进行休息，则该方法属于静态方法。（对）
225、 通过对象不能调用类方法和静态方法。（错）
226、 在 Python 中可以为自定义类的对象动态增加新成员。（对）
227、 Python 类不支持多继承。（错）
228、 属性可以像数据成员一样进行访问，但赋值时具有方法的优点，可以对新值进行检查。（对）
229、 文件对象是可以迭代的。（对）
230、 文件对象的 tell()方法用来返回文件指针的当前位置。（对）
231、 以写模式打开的文件无法进读操作。（对）
232、 假设已成功导入 os 和 sys 标准库，那么表达式 os.path.dirname(sys.executable) 的值为 Python 安装目录。（对）
233、 只可以动态为对象增加数据成员，而不能为对象动态增加成员方法。（错）
234、 Python 字典支持双向索引。（错）

235、 Python 集合支持双向索引。（错）
236、 Python 元组支持双向索引。（对）
237、 假设 re 模块已成功导入，并且有 pattern = re.compile(‘^’+’.’.join([r’\d{1,3}’ for i in range(4)])+’$’)，那么表达式 pattern.match(‘192.168.1.103’) 的值为 None。（错）
238、 假设 random 模块已导入，那么表达式 random.sample(range(10), 20) 的作用是生成 20 个不重复的整数。（错）
239、 假设 random 模块已导入，那么表达式 random.sample(range(10), 7) 的作用是生成 7 个不重复的整数。（对）
240、 在 Python 3.x 中 reduce()是内置函数。（错）
241、 以读模式打开文件时，文件指针指向文件开始处。（对）
242、 以追加模式打开文件时，文件指针指向文件尾。（对）
243、 已知 x = (1, 2, 3, 4)，那么执行 x[0] = 5 之后，x 的值为(5, 2, 3, 4)。（错）
244、 已知 x = 3，那么执行 x += 6 语句前后 x 的内存地址是不变的。（错）
245、 成员测试运算符 in 作用于集合时比作用于列表快得多。（对）
246、 在 Python 3.x 中，使用内置函数 input()接收用户输入时，不论用户输入的什么格式，一律按字符串进行返回。（对）
247、 安装 Python 扩展库时只能使用 pip 工具在线安装，如果安装不成功就没有别的办法了。（错）
248、 使用 random 模块的函数 randint(1, 100)获取随机数时，有可能会得到 100。（对）
249、 如果只需要 math 模块中的 sin()函数，建议使用 from math import sin 来导入，而不要使用 import math 导入整个模块。（对）
250、 已知列表 x = [1, 2, 3, 4]，那么表达式 x.find(5)的值应为-1。（错）
251、 列表对象的排序方法 sort()只能按元素从小到大排列，不支持别的排序方式。（错）
252、 g = lambda x: 3 不是一个合法的赋值表达式。（错）
253、 内置函数 len()返回指定序列的元素个数，适用于列表、元组、字符串、字典、集合以及 range、zip 等迭代对象。（对）
254、 已知 x 和 y 是两个等长的整数列表，那么表达式 sum((i*j for i, j in zip(x, y)))的作用是计算这两个列表所表示的向量的内积。（对）
255、 已知 x 和 y 是两个等长的整数列表，那么表达式[i+j for i,j in zip(x,y)]的作用时计算这两个列表所表示的向量的和。（对）
256、 表达式 int(‘1’*64, 2)与 sum(2**i for i in range(64))的计算结果是一样的，但是前者更快一些。（对）
257、 已知 x = list(range(20))，那么语句 del x[::2]可以正常执行。（对）
258、 已知 x = list(range(20))，那么语句 x[::2] = []可以正常执行。（错）
259、 已知 x = list(range(20))，那么语句 print(x[100:200])无法正常执行。（错）
260、 已知 x 是个列表对象，那么执行语句 y = x 之后，对 y 所做的任何操作都会同样作用到 x 上。（对）
261、 已知 x 是个列表对象，那么执行语句 y = x[:]之后，对 y 所做的任何操作都会同样作用到 x 上。（错）
262、 在 Python 中，变量不直接存储值，而是存储值的引用，也就是值在内存中的地址。（对）
263、 表达式(i**2 for i in range(100))的结果是个元组。（错）
264、 在 Python 中元组的值是不可变的，因此，已知 x = ([1], [2])，那么语句 x[0].append(3)是无法正常执行的。（错）

265、 包含 yield 语句的函数一般成为生成器函数，可以用来创建生成器对象。（对）
266、 在函数中 yield 语句的作用和 return 完全一样。（错）
267、 Python 内置的字典 dict 中元素是按添加的顺序依次进行存储的。（错）
268、 Python 内置的集合 set 中元素顺序是按元素的哈希值进行存储的，并不是按先后顺序。（对）
269、 已知 x = {1:1, 2:2}，那么语句 x[3] =3 无法正常执行。（错）
270、 Python 内置字典是无序的，如果需要一个可以记住元素插入顺序的字典，可以使用 collections.OrderedDict。（对）
271、 语句 pass 仅起到占位符的作用，并不会做任何操作。（对）
272、 在条件表达式中不允许使用赋值运算符“=”，会提示语法错误。（对）
273、 任何包含 call()方法的类的对象都是可调用的。（对）
274、 在 Python 中函数和类都属于可调用对象。（对）
275、 无法使用 lambda 表达式定义有名字的函数。（错）
276、 已知 x 是一个列表，那么 x = x[3:] + x[:3]可以实现把列表 x 中的所有元素循环左移 3 位。（对）
277、 已知 x 和 y 是两个字符串，那么表达式 sum((1 for i,j in zip(x,y) if i==j))可以用来计算两个字符串中对应位置字符相等的个数。（对）
278、 函数和对象方法是一样的，内部实现和外部调用都没有任何区别。（错）
279、 在设计派生类时，基类的私有成员默认是不会继承的。（对）
280、 如果在设计一个类时实现类 len()方法，那么该类的对象会自动支持 Python 内置函数 len()。（对）
281、 Python 3.x 中字符串对象的 encode()方法默认使用 utf8 作为编码方式。（对）
282、 已知 x = ‘hellow world.’.encode()，那么表达式 x.decode(‘gbk’)的值为’hellow world.’。（对）
283、 已知 x = ‘Python 是一种非常好的编程语言’.encode()，那么表达式 x.decode(‘gbk’)的值为’Python 是一种非常好的编程语言’。（错）
284、 正则表达式’^http’只能匹配所有以’http’开头的字符串。（对）
285、 正则表达式’^\d{18}|\d{15}$’只能检查给定字符串是否为 18 位或 15 位数字字符，并不能保证一定是合法的身份证号。（对）
286、 二进制文件也可以使用记事本程序打开，只是无法正确阅读和理解其中的内容。（对）
287、 正则表达式’[^abc]’可以一个匹配任意除’a’、’b’、’c’之外的字符。（对）
288、 正则表达式’python|perl’或’p(ython|erl)’都可以匹配’python’或’perl’。（对）
289、 文本文件是可以迭代的，可以使用 for line in fp 类似的语句遍历文件对象 fp 中的每一行。（对）
290、 Python 的主程序文件 python.exe 属于二进制文件。（对）
291、 使用记事本程序也可以打开二进制文件，只不过无法正确识别其中的内容。（对）
292、 对字符串信息进行编码以后，必须使用同样的或者兼容的编码格式进行解码才能还原本来的信息。（对）
293、 使用 pickle 进行序列化得到的二进制文件使用 struct 也可以正确地进行反序列化。（错）
294、 已知当前文件夹中有一个文件 readme.txt 具有只读属性，假设标准库 os 已正确导入，那么可以通过语句 os.chmod(‘readme.txt’, 0o777)来删除该文件的只读属性。（对）
295、 Python 标准库 os 的函数 remove()不能删除具有只读属性的文件。（对）
296、 字节串 b’hello world’和 b’hello world.’的 MD5 值相差很小。（错）

297、 由于异常处理结构 try…except…finally…中 finally 里的语句块总是被执行的，所以把关闭文件的代码放到 finally 块里肯定是万无一失，一定能保证文件被正确关闭并且不会引发任何异常。（错）

298、 使用 TCP 协议进行通信时，必须首先建立连接，然后进行数据传输，最后再关闭连接。（对）

299、 TCP 是可以提供良好服务质量的传输层协议，所以在任何场合都应该优先考虑使用。（错）

300、 在 4 核 CPU 平台上使用多线程编程技术可以很轻易地获得 400%的处理速度提升。（错）

301、 多线程编程技术主要目的是为了提高计算机硬件的利用率，没有别的作用了。（错）

302、

三、 简答题

1、 简单解释 Python 基于值的自动内存管理方式？（Python 采用的是基于值得内存管理方式，在 Python 中可以为不同变量赋值为相同值，这个值在内存中只有一份，多个变量指向同一个内存地址；Python 具有自动内存管理功能，会自动跟踪内存中所有的值，对于没有任何变量指向的值，Python 自动将其删除。）

2、 写出 Python 运算符&的两种功能？（1）数字位运算；2）集合交集运算。）

3、 在 Python 中导入模块中的对象有哪几种方式？（1)import 模块名 [as 别名]；2)from 模块名 import 对象名[as 别名]；3)from math import *）

4、 解释 Python 脚本程序的“name”变量及其作用？（每个 Python 脚本在运行时都有一个“name”属性。如果脚本作为模块被导入，则其“name”属性的值被自动设置为模块名；如果脚本独立运行，则其“name”属性值被自动设置为“main”。利用“name”属性即可控制 Python 程序的运行方式。）

5、 为什么应尽量从列表的尾部进行元素的增加与删除操作？（当列表增加或删除元素时，列表对象自动进行内存扩展或收缩，从而保证元素之间没有缝隙，但这涉及到列表元素的移动，效率较低，应尽量从列表尾部进行元素的增加与删除操作以提高处理速度。）

6、 分析逻辑运算符“or”的短路求值特性？（假设有表达式“表达式 1 or 表达式 2”，如果表达式 1 的值等价于 True，那么无论表达式 2 的值是什么，整个表达式的值总是等价于 True。因此，不需要再计算表达式 2 的值。）

7、 简单解释 Python 中短字符串驻留机制？（对于短字符串，将其赋值给多个不同的对象时，内存中只有一个副本，多个对象共享改副本。）

8、 异常和错误有什么区别？（异常是指因为程序执行过程中出错而在正常控制流以外采取的行为。严格来说，语法错误和逻辑错误不属于异常，但有些语法错误往往会导致异常，例如由于大小写拼写错误而访问不存在的对象，或者试图访问不存在的文件，等等。）

9、 使用 pdb 模块进行 Python 程序调试主要有哪几种用法？（1）在交互模式下使用 pdb 模块提供的功能可以直接调试语句块、表达式、函数等多种脚本。2）在程序中嵌入断点来实现调试功能。在程序中首先导入 pdb 模块，然后使用 pdb.set_trace()在需要的位置设置断点。如果程序中存在通过该方法调用显式插入的断点，那么在命令提示符环境下执行该程序或双击执行程序时将自动打开 pdb 调试环境，即使该程序当前不处于调试状态。3）使用命令行调试程序。在命令行提示符下执行“python －m pdb 脚本文件名”，则直接进入调试环境；当调试结束或程序正常结束以后，pdb 将重启该程序。）

10、 阅读下面的代码，并分析假设文件“D:\test.txt”不存在的情况下两段代码可能发生的问题。

代码 1：

```
try:
    fp = open(r'd:\test.txt')
    print('Hello world!', file=fp)
finally:
    fp.close()
```

代码 2：

```
try:
    fp = open(r'd:\test.txt', 'a+')
    print('Hello world!', file=fp)
finally:
    fp.close()
```

答：

假设文件“D:\test.txt”不存在，那么第一段代码会抛出异常，提示 fp 没有定义；第二段代码执行正常。原因是第二段代码使用内置函数 open()打开指定文件时如果不存在则会创建该文件，从而不会抛出异常。

11、

四、 编程题

1、编写程序，在 D 盘根目录下创建一个文本文件 test.txt，并向其中写入字符串 hello world。

答：

```
fp = open(r'D:\test.txt', 'a+')
print('hello world', file=fp)
fp.close()
```

2、写出下面代码的优化版本，提高运行效率。

```
x = list(range(500))
for item in x:
    t = 5**5
print(item+t)
```

答：

```
x = list(range(500))
t = 5**5
for item in x:
    print(item+t)
```

3、编写程序，生成一个包含 20 个随机整数的列表，然后对其中偶数下标的元素进行降序排列，奇数下标的元素不变。（提示：使用切片。）

答：

```
import random
x = [random.randint(0,100) for i in range(20)]
print(x)
y = x[::2]
```

```
y.sort(reverse=True)
x[::2] = y
print(x)
```

4、写出下面代码的执行结果。

```
def Join(List, sep=None):
    return (sep or ‘,’).join(List)
print(Join([‘a’, ‘b’, ‘c’]))
print(Join([‘a’, ‘b’, ‘c’],’:’))
```

答：

```
a,b,c
a:b:c
```

5、写出下面代码的运行结果。

```
def Sum(a, b=3, c=5):
    return sum([a, b, c])
print(Sum(a=8, c=2))
print(Sum(8))
print(Sum(8,2))
```

答：

```
13
16
15
```

6、写出下面代码的运行结果。

```
def Sum(*p):
    return sum(p)
print(Sum(3, 5, 8))
print(Sum(8))
print(Sum(8, 2, 10))
```

答：

```
16
8
20
```

7、编写函数，判断一个数字是否为素数，是则返回字符串 YES，否则返回字符串 NO。

答：

```
import math
def IsPrime(v):
    n = int(math.sqrt(v)+1)
    for i in range(2,n):
        if v%i==0:
            return ‘No’
    else:
        return ‘Yes’
```

8、编写函数，模拟 Python 内置函数 sorted()。

答：

```
def Sorted(v):
```

```
    t = v[::]
    r = []
    while t:
        tt = min(t)
        r.append(tt)
        t.remove(tt)
    return r
```

9、编写程序，生成包含 20 个随机数的列表，然后将前 10 个元素升序排列，后 10 个元素降序排列，并输出结果。

答：

```
import random
x = [random.randint(0,100) for i in range(20)]
print(x)
y = x[0:10]
y.sort()
x[0:10] = y
y = x[10:20]
y.sort(reverse=True)
x[10:20] = y
print(x)
```

10、编写程序，运行后用户输入 4 位整数作为年份，判断其是否为闰年。如果年份能被 400 整除，则为闰年；如果年份能被 4 整除但不能被 100 整除也为闰年。

答：

```
x = input('Please input an integer of 4 digits meaning the year:')
x = eval(x)
if x%400==0 or (x%4==0 and not x%100==0):
    print('Yes')
else:
    print('No')
```

11、编写程序，实现分段函数计算，如下表所示。

x	y
x<0	0
0<=x<5	x
5<=x<10	3x-5
10<=x<20	0.5x-2
20<=x	0

答：

```
x = input('Please input x:')
x = eval(x)
if x<0 or x>=20:
    print(0)
```

```
elif 0<=x<5:
    print(x)
elif 5<=x<10:
    print(3*x-5)
elif 10<=x<20:
print(0.5*x-2)
```

12、阅读下面的程序，判断其是否可以正常运行，如果可以运行则写出执行结果，如果不能运行则写出理由。

```
class Test:
    def init(self, value):
        self.__value = value
    @property
    def value(self):
        return self.__value

t = Test(3)
t.value = 5
print(t.value)
```

答：

不能运行。程序中定义的是只读属性，不能修改属性的值。

13、下面代码的功能是，随机生成 50 个介于[1,20]之间的整数，然后统计每个整数出现频率。请把缺少的代码补全。

```
import random
x = [random.__(1,20) for i in range(_)]
r = dict()
for i in x:
    r[i] = r.get(i, _)+1
for k, v in r.items():
print(k, v)
```

答：

分别填写 randint、50、0

14、假设有 Python 程序文件 demo.py，代码如下：

```
def main():
    if name == ‘main‘:
        print(1)
    else:
        print(2)

main()
```

将该程序文件直接运行时输出结果为_，作为模块导入时得到结果___-。（1、2）

15、下面程序的执行结果是_____。（1）

```
s = 0
for i in range(1,101):
    s += i
```

```
else:
    print(1)
```
16、下面程序的执行结果是____。（1275）
```
s = 0
for i in range(1,101):
    s += i
    if i == 50:
        print(s)
        break
else:
    print(1)
```
17、下面的程序是否能够正常执行，若不能，请解释原因；若能，请分析其执行结果。
```
from random import randint

result = set()
while True:
    result.add(randint(1,10))
    if len(result)==20:
        break
print(result)
```
答：无法正确执行，因为该程序的功能是从[1,10]区间中选择 20 个不同的随机整数，而该区间并没有这么多整数，所以程序死循环。

18、下面的代码是否能够正确运行，若不能请解释原因；若能，请分析其执行结果。

```
x = list(range(20))
    for i in range(len(x)):
        del x[i]
```
答：无法正确执行，因为删除列表元素时会影响其他元素在列表中的索引，上面的代码会抛出下标越界的异常。

19、阅读下面的代码，解释其功能。
```
x = list(range(20))
for index, value in enumerate(x):
    if value == 3:
        x[index] = 5
```
答：将列表 x 中值为 3 的元素修改为 5。

20、阅读下面的代码，解释其功能。
```
x = [range(3*i, 3*i+5) for i in range(2)]
x = list(map(list, x))
x = list(map(list, zip(*x)))
```
答：首先生成一个包含列表的列表，然后模拟矩阵转置。

21、阅读下面的代码，解释其功能。

```
import string
x = string.ascii_letters + string.digits
import random
print(''.join(random.sample(x, 10)))
```

答：输出由英文字母大小写或数字组成的长度为 10 且不重复的随机字符串。

22、阅读下面的代码，分析其执行结果。

```
def demo(*p):
    return sum(p)
print(demo(1,2,3,4,5))
print(demo(1,2,3))
```

答：输出结果为

15

6

23、阅读下面的代码，分析其执行结果。

```
def demo(a, b, c=3, d=100):
    return sum((a,b,c,d))
print(demo(1, 2, 3, 4))
print(demo(1, 2, d=3))
```

答：输出结果为

10

9

24、下面的代码输出结果为___。（3）

```
def demo():
    x = 5
x = 3
demo()
print(x)
```

25、下面函数的功能为_____。（将序列循环左移 k 位，得到新序列并返回）

```
def demo(lst, k):
    if k
```

第一部分:

1、python 属于_____语言（解释型、强、动态）。

2、Python 的内置函数在_____文件中。（buildins.py）

3.python 提供了两个对象身份比较的操作符_____和_____，测试一两个变量是否指向同一个对象，也可以通过内建函数_____来测试对象的内存地址。（is is not id）

4、字典对象的_____方法可以获取指定“键”对应的“值”，并且可以在指定“键”不存在的时候返回指定值，如果不指定则返回 None。（get()）

5、已知 x = {‘a’:‘b’, ‘c’:‘d’}，那么表达式 ‘b’ in x 的值为_____。（False）

6、表达式 [x for x in [1,2,3,4,5] if x<3] 的值为_____。（[1, 2]）
7、表达式 isinstance(‘abcdefg’, str) 的值为_____。（True）表达式 isinstance(‘abcdefg’, object) 的值为_____。（True）
8、表达式 ‘,’.join('a b ccc\n\n\nddd '.split()) 的值为_____。（‘a,b,ccc,ddd’）
9、产生 type 的类是_____（type）
10、假设列表对象 x = [1, 1, 1]，那么表达式 id(x[0]) == id(x[2])的值为_____。（True）
11、表达式 sorted([13, 1, 237, 89, 100], key=lambda x: len(str(x))) 的值为_____。（[1, 13, 89, 237, 100]）
12、类的成员包含_____、_____、实例方法、类属性、类方法。（实例属性、静态方法）

第二部分:

1、Python 安装扩展库常用的是_____工具。（pip）
2、Python 标准库 math 中用来计算取余的操作是_____，规则是_____（fmod，正数时商向下取正，负数时商向上取正）
3.python 提供了两个操作符，当指向内容是否一致是使用_____，当指向是否是同一个对象时使用_____。（= = is）
4、字典对象的_____方法可以获取指定“键”对应的“值”，并且可以在指定“键”不存在的时候返回指定值，如果不指定则返回 None。（get()）
5、已知 x = {‘a’:‘b’, ‘c’:‘d’}，那么表达式 ‘b’ in x 的值为_____。（False）
6、表达式 [x for x in [1,2,3,4,5] if x<3] 的值为_____。（[1, 2]）
7、表达式 isinstance(‘abcdefg’, str) 的值为_____。（True）表达式 isinstance(‘abcdefg’, object) 的值为_____。（True）
8、表达式 ‘,’.join(‘a b ccc\n\n\nddd ‘.split()) 的值为_____。（‘a,b,ccc,ddd’）
9、假设正则表达式模块 re 已导入，那么表达式 re.sub(’\d+’, ‘1’, ‘a12345bbbb67c890d0e’) 的值为_____。（‘a1bbbb1c1d1e’）
10、假设列表对象 x = [1, 1, 1]，那么表达式 id(x[0]) == id(x[2])的值为_____。（True）
11、表达式 sorted([13, 1, 237, 89, 100], key=lambda x: len(str(x))) 的值为_____。（[1, 13, 89, 237, 100]）
12、Python 标准库 os.path 中用来判断指定文件是否存在的方法是_____。（exists()）

判断

第一部分:

1、Python 支持多继承，如果父类中有相同的方法名，而在子类中调用时没有指定父类名，则 Python 解释器将从左向右按顺序进行搜索。（对）

2、当将位置参数放到可变参数的后面时，位置参数自动变成关键字参数（对）

3、python 的字符串支持的字符集是 ascii 字符集（错）

4、元组的元素是不可修改的。（错）

5、在 Python 中定义类时实例方法的第一个参数名称必须是 self。（错）

6、elif 条件可以匹配多次（错）

7、通过对象不能调用类方法和类属性（错）

8、在类的外部无论如何也无法调用到私有成员（错）

9、通过对象不能调用类方法和静态方法。（错）

10、在 Python 中可以为自定义类的对象动态增加新成员。（对）

11、Python 类不支持多继承。（错）

12、属性可以像数据成员一样进行访问，但赋值时具有方法的优点，可以对新值进行检查。（对）

13、Python 程序只能在安装了 Python 环境的计算机上以源代码形式运行。（错）

14、定义函数时，带有默认值的参数必须出现在参数列表的最右端，任何一个带有默认值的参数右边不允许出现没有默认值的参数。（对）

15、当作为条件表达式时，空值、空字符串、空列表、空元组、空字典、空集合、空迭代对象以及任意形式的数字 0 都等价于 False。（对）

第二部分:

1、Python 支持多继承，如果父类中有相同的方法名，而在子类中调用时没有指定父类名，则 Python 解释器将从左向右按顺序进行搜索。（对）

2、对文件进行读写操作之后必须显式关闭文件以确保所有内容都得到保存。（对）

3、Python 标准库 threading 中的 Lock、RLock、Condition、Event、Semaphore 对象都可以用来实现线程同步。（对）

4、异常处理结构中的 finally 块中代码仍然有可能出错从而再次引发异常。（对）

5、在 Python 中定义类时实例方法的第一个参数名称必须是 self。（错）

6、在多线程编程时，当某子线程的 daemon 属性为 False 时，主线程结束时会检测该子线程是否结束，如果该子线程尚未运行结束，则主线程会等待它完成后再退出。（对）

7、定义类时实现了__ eq __()方法，该类对象即可支持运算符==。（对）

8、使用正则表达式对字符串进行分割时，可以指定多个分隔符，而字符串对象的 split()方法无法做到这一点。（对）

9、通过对象不能调用类方法和静态方法。（错）

10、在 Python 中可以为自定义类的对象动态增加新成员。（对）

11、Python 类不支持多继承。（错）

12、属性可以像数据成员一样进行访问，但赋值时具有方法的优点，可以对新值进行检查。（对）

13、Python 程序只能在安装了 Python 环境的计算机上以源代码形式运行。（错）

14、定义函数时，带有默认值的参数必须出现在参数列表的最右端，任何一个带有默认值的参数右边不允许出现没有默认值的参数。（对）

15、当作为条件表达式时，空值、空字符串、空列表、空元组、空字典、空集合、空迭代对象以及任意形式的数字 0 都等价于 False。（对）

问答题

第一部分:

1.说明 lambda 表达式是什么？如何使用？

Lambda 创建函数的一种简洁方式，用来定义匿名函数的
语法：lambda 参数: 表达式
使用的时候要注意实现的是简单函数功能。
使用场合，当高阶函数的关键字参数指向函数名，可以直接使用 lambda 来实现。

2.解释逻辑运算符的短路现象。

被短路：就是不执行当前的表达式
And 第一个表达式为 False，第二个表达式被短路
Or 第一个表达式为 True，第二个表达式被短路

3.解释 LEGB 原则并说明对于可变对象和不可变对象的不同

在访问某一个名称的时候，python 解释器 会从当前的命名空间中查找，如果查找不到则遵循 LEGB 原则
LEGB 是一个访问顺序

先从局部命名空间中查找，如果找到，则停止，否则到外围命名空间中查找，如果找到，则停止，则到全局命名空间中查找，如果找到，则停止，否则到内建命名空间中查找，如果找不到则报错。
对于可变类型来说，在局部命名空间中访问全局变量 或者外围变量不需要使用 global 和 nonlocal 关键字就可以访问到变量名称。

4.阐述一下 python 中列表的深拷贝与浅拷贝

浅拷贝：列表下的 copy 方法，copy 包下的 copy 方法，列表的切片
只复制当前列表对象的第一层。

深拷贝：copy 包下的 deepcopy
一直复制到不可变类型对象为止。

5.python 中 nonlocal 与 global 关键字的用法

对于不可变类型的数据来说，
当访问外围变量时，需要用到 nonlocal，在名称使用之前，先使用 nonlocal
当访问全局变量时，需要用到 global，在名称使用之前，使用 global

6.描述一下__new__,init,__call__方法分别在元类和类中的执行顺序。

(1) 当使用元类创建类的时候，会执行：元类的__new__,init
(2) 当使用类创建对象的时候，会执行：元类的__call__
继续执行类的__new__,init
(3) 当把对象当函数名一样使用的时候，对象()，会执行类的__call__

第二部分:

1.python 中导入模块有几种方式，分别写出来并比较一下使用方式的利弊。

import 模块名 as 别名

可以导入一个模块下的所有内容，引用的时候，可以通过模块名+名称来引用
不会引发跟内部变量同名问题

from 模块名 import 对象名 as 别名

按名字引入一个模块下的内容，使用*可以全部引入
引用的时候，可以通过名称直接引用
可能引发跟内部变量同名问题
所以可以使用别名来避免重名情况。

2.写出 python 异常处理的结构

try：
可能出现异常的代码段
except 异常类型 1：
捕获异常
except 异常类型 2：
捕获异常
except：
捕获异常
Else:
Try 正常执行没有抛出异常时走的代码段
Finally：
一直会走的代码段

3.解释一下 python 以下划线开头变量名的特点

4. 阐述协程对象的 daemon 属性的作用

5.说明一下线程中的 run 方法和 start 的不同

单选

1.下面属于合法变量名的是（A）
A X_xy B 12anc C and D X-Y

2.下面不合法的整数是（D）
A 100 B 0B101 C 0X1A2 D 0O12A

3.表示 n 是 m 的倍数的表达式正确的是（A）
A n%m0 B n/m0 C n*m0 D n//m0

4.下列代码运行结果是？ C
a = ‘a’
print a > ‘b’ or ‘c’
A. a B. b C. c D. True

5.下列哪种不是 Python 元组的定义方式？ A
A. (1) B. (1,) C. (1, 2) D. (1, 2, (3, 4))

6. 下列哪种函式参数定义不合法? C
A. def myfunc(*args): B. def myfunc(arg1=1):
C. def myfunc(*args, a=1): D. def myfunc(a=1, **args):

7.a 与 b 定义如下，下列哪个是正确的？ B
a = ‘123’
b = ‘123’
A. a != b B. a is b C.a == 123 D.a + b = 246

8.一个段代码定义如下，下列调用结果正确的是？A
def bar(multiple):
def foo(n):
return multiple ** n
return foo
A. bar(2)(3) == 8 B. bar(2)(3) == 6 C. bar(3)(2) == 8 D. bar(3)(2) == 6

9.下面代码运行后，a、b、c、d 四个变量的值，描述错误的是？ D
import copy
a = [1, 2, 3, 4, [‘a’, ‘b’]]
b = a
c = copy.copy(a)
d = copy.deepcopy(a)
a.append(5)
a[4].append(‘c’)
A. a ==[1,2, 3, 4, [‘a’, ‘b’, ‘c’], 5] B. b == [1,2, 3, 4, [‘a’, ‘b’, ‘c’], 5]
C. c ==[1,2, 3, 4, [‘a’, ‘b’, ‘c’]] D. d == [1,2, 3, 4, [‘a’, ‘b’, ‘c’]]

10.有如下类定义，下列描述错误的是？ D
class A(object):
pass
class B(A):
pass
b = B()
A. isinstance(b, A) == True B. isinstance(b, object) == True
C.issubclass(B, A) == True D. issubclass(b, B) == True

多选题

1.Python 中函数是对象，描述正确的是? ABCD
A.函数可以赋值给一个变量 B.函数可以作为元素添加到集合对象中
C.函数可以作为参数值传递给其它函数 D.函数可以当做函数的返回值

2.若 a = range(100)，以下哪些操作是合法的? ABCD
A. a[-3] B. a[2:13] C. a[::3] D. a[2-3]

3.若 a = (1, 2, 3)，下列哪些操作是合法的? ABD
A. a[1:-1] B. a*3 C. a[2] = 4 D. list(a)

4.Python 中单下划线_foo 与双下划线__foo 与__foo__的成员，下列说法正确的是？ABC
A. _foo 不能直接用于’from module import *’
B. __foo 解析器用_classname__foo 来代替这个名字，以区别和其他类相同的命名
C. __foo__代表 python 里特殊方法专用的标识
D. __foo 可以直接用于’from module import *’

5.__new__和__init__的区别，说法正确的是： ABCD
A. new__是一个静态方法，而__init__是一个实例方法
B. new__方法会返回一个创建的实例，而__init__什么都不返回
C.只有在__new__返回一个 cls 的实例时，后面的__init__才能被调用
D.当创建一个新实例时调用__new，初始化一个实例时用__init

作者：朔落南缘 Sevier
来源：CSDN
原文：https://blog.csdn.net/qq_42442369/article/details/84073784