选择题一：

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、Python 3.x支持使用中文作为变量名。（对）

87、在Python中，任意长的字符串都遵守驻留机制。（错）

88、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）

89、Python字符串方法replace()对字符串进行原地修改。（错）

90、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）

91、对于大量列表的连接，extend()方法比运算符+具有更高的效率。（对）

92、表达式 {1, 3, 2} > {1, 2, 3} 的值为True。（错）

93、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）

94、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）

95、函数是代码复用的一种方式。（对）

96、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）

97、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）

98、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）

99、定义Python函数时必须指定函数返回值类型。（错）

100、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）

101、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）

102、函数中必须包含return语句。（错）

103、函数中的return语句一定能够得到执行。（错）

104、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）

105、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）

106、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）

107、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）

108、在函数内部没有办法定义全局变量。（错）

109、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）

110、表达式 ‘a’+1的值为’b’。（错）

111、在函数内部直接修改形参的值并不影响外部实参的值。（对）

112、在函数内部没有任何方法可以影响实参的值。（错）

113、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）

114、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）

115、在同一个作用域内，局部变量会隐藏同名的全局变量。（对）

116、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）

117、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）

118、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）

119、Python中一切内容都可以称为对象。（对）

120、栈和队列的都具有先入后出的特点。（错）

121、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）

122、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）

123、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）

124、Python中没有严格意义上的私有成员。（对）

125、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）

126、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）

127、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）

128、运算符 / 在Python 2.x和Python 3.x中具有相同的功能。（错）

129、运算符“-”可以用于集合的差集运算。（对）

130、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）

131、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）

132、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）

133、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）

134、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）

135、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）

136、程序中异常处理结构在大多数情况下是没必要的。（错）

137、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）

138、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）

139、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）

140、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）

141、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）

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

143、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）

144、Python代码可以内嵌在asp文件中。（对）

145、无法配置IIS来支持Python程序的运行。（错）

146、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）

147、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）

148、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）

149、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）

150、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）

151、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）

152、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）

153、表达式 list('[1, 2, 3]') 的值是[1, 2, 3]。（错）

154、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）

155、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）

156、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）

157、定义类时实现了\_\_eq\_\_()方法，该类对象即可支持运算符==。（对）

158、定义类时实现了\_\_pow\_\_()方法，该类对象即可支持运算符\*\*。（对）

159、二进制文件不能使用记事本程序打开。（错）

160、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）

161、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）

162、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）

163、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）

164、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）

165、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）

166、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）

167、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）

168、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）

169、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）

170、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）

171、在Python中定义函数时不需要声明函数参数的类型。（对）

172、在Python中定义函数时不需要声明函数的返回值类型。（对）

173、在函数中没有任何办法可以通过形参来影响实参的值。（错）

174、已知 x = 3，那么执行语句 x+=6 之后，x的内存地址不变。（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。（错）

176、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）

177、在Python中可以使用 for 作为变量名。（错）

178、在Python中可以使用 id 作为变量名，尽管不建议这样做。（对）

179、Python关键字不可以作为变量名。（对）

180、一个数字5也是合法的Python表达式。（对）

181、同一个列表对象中的元素类型可以各不相同。（对）

182、同一个列表对象中所有元素必须为相同类型。（错）

183、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）

184、列表可以作为集合的元素。（错）

185、集合可以作为列表的元素。（对）

186、元组可以作为集合的元素。（对）

187、集合可以作为元组的元素。（对）

188、字典可以作为集合的元素。（错）

189、集合可以作为字典的键。（错）

190、集合可以作为字典的值。（对）

191、可以使用del删除集合中的部分元素。（错）

192、标准库os的rename()方法可以实现文件移动操作。（对）

193、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）

194、当作为条件表达式时，[]与None等价。（对）

195、表达式 [] == None 的值为True。（错）

196、当作为条件表达式时，{}与None等价。（对）

197、表达式 {}==None 的值为True。（错）

198、表达式 pow(3,2) == 3\*\*2 的值为True。（对）

199、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）

200、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）

201、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）

202、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）

203、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）

204、在调用函数时，必须牢记函数形参顺序才能正确传值。（错）

205、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）

206、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）

207、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）

208、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）

209、执行语句 from math import sin 之后，可以直接使用 sin() 函数，例如 sin(3)。（对）

210、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）

211、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）

212、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）

213、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）

214、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）

215、不同版本的Python不能安装到同一台计算机上。（错）

216、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。

217、表达式 {1, 2} \* 2 的值为 {1, 2, 1, 2}。（错）

218、Python变量名区分大小写，所以student和Student不是同一个变量。（对）

219、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）

220、正则表达式元字符“\s”用来匹配任意空白字符。（对）

221、正则表达式 元字符“\d”用来匹配任意数字字符。（对）

222、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）

223、Python类的构造函数是\_\_init\_\_()。（对）

224、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）

225、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）

226、通过对象不能调用类方法和静态方法。（错）

227、在Python中可以为自定义类的对象动态增加新成员。（对）

228、Python类不支持多继承。（错）

229、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）

230、文件对象是可以迭代的。（对）

231、文件对象的tell()方法用来返回文件指针的当前位置。（对）

232、以写模式打开的文件无法进读操作。（对）

233、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）

234、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）

235、Python字典支持双向索引。（错）

236、Python集合支持双向索引。（错）

237、Python元组支持双向索引。（对）

238、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）

239、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）

240、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）

选择题二：

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）