习题 1 答案

1. 选择题

C　C　C　C　D　B　C

2. 简答题

（1）简述 Python 程序的执行过程。

Python 源文件的扩展名是.py。在执行时， Python 解释器先将.py 文件中的源代码翻译成中间代码，这个中间代码是一个扩展名为.pyc 的文件，再由 Python 虚拟机 (PVM)逐条将字节码翻译成机器指令执行。

（2）列举出 IDLE 编程环境 5 个以上快捷键的功能。

快捷键	功能说明
Ctrl + [	缩进代码
Ctrl +]	取消缩进代码
Alt+3	注释代码行
Alt+4	取消注释代码行
Alt+/	单词自动完成，只要文中出现过，就可以帮你自动补齐。多按几次可以循环选择
Alt+P	浏览历史命令（上一条）
Alt+N	浏览历史命令（下一条）
F1	打开 Python 帮助文档
F5	运行程序
Ctrl+F6	重启 Shell，之前定义的对象和导入的模块全部失效

（3）叙述程序的编译方式和解释方式的区别。

编译是将源程序代码转换成目标代码的过程。源代码是计算机高级语言代码，而目标代码则是机器语言代码。**解释**是将源代码逐条转换成目标代码同时逐条运行目标代码的过程。执行解释的计算机程序称为解释器(Interpreter)。

解释和编译的区别在于编译是一次性地翻译，程序被编译后，运行的时候不再需要源代码。解释则在每次程序运行时都需要解释器和源代码。这两者的区类似于外语资料的翻译和实时的同声传译。

（4）叙述程序设计的 IPO 模式的特点。

程序设计模式的 IPO 模式，即程序包括输入(Input)、处理(Process)、输出(Output)3 部分。输入是程序设计的起点，包括文件输入、网络输入、交互输入、参数输入等。输出是程序展示运算成果的方式，包括文件输出、网络输出、控制台输出、图表输出等。而处理部分则是编程的核心，包括数据处理与赋值，更重要的是算法。

3. 编程题

（1）参考例 1-3，输入三角形的底边长和高，计算并输出三角形的面积。

```
# ex0101.py
#计算三角形面积s。
import math
a=eval(input("底边长："))
h=eval(input("底边对应的高："))

s = a*h/2
print("三角形的面积是{:.2f}".format(s))
```

（2）参考例 1-6，在列表中给出若干字符串，计算并输入最长的字符串。

```
# ex0102.py
lst=['hi',"program",'school','for','''chinese''']
lst1=[]
for item in lst:
    lst1.append(len(item))
p=max(lst1)
for item in lst:
    if len(item)==p:
        print(item,end=" ")
```

（3）略。

习题 2 答案

1. 选择题

D　D　D　D　D　　A　C　A

2. 简答题

（1）简述 Python 标识符的命名规则。

- Python 的标识符可以由字母、数字和下划线“_”组成，且不能以数字开头。
- 标识符区分大小写，没有长度限制。
- 标识符不能使用计算机语言中预留有特殊作用的关键字。
- 标识符的命名尽量符合见名知意的原则，从而提高代码的可读性。

（2）整数的二进制、八进制、十六进制都用什么格式描述？将十进制数转换为二进制、八进制、十六进制的函数是什么？

、二进制(以“OB”或“Ob”开头)、八进制(以数字“0o”或“0O”开头)和十六进制(以“Ox”或“OX”开头)。

bin(),hex(),oct()

（3）Python 的数值类型数据有几种？举例说明。

数值类型（Number）是 Python 的基本数据类型，包含整型、浮点型、复数类型和布尔类型等 4 种。

3. 编程题

（1）输入三科成绩，计算平均分和总分。

```
x=eval(input("请输入A科成绩："))
y=eval(input("请输入B科成绩："))
z=eval(input("请输入C科成绩："))

print("3科成绩的平均分是{}，总分是{}".format((x+y+z)/3,x+y+z))
```

（2）编写程序，给出三角形的三边，输出三角形的面积。

```
# 输入三角形三条边，有海伦公式计算三角形面积s。
import math
a=eval(input("请输入a边长："))
b=eval(input("请输入b边长："))
c=eval(input("请输入c边长："))
p = (a + b + c) / 2
s = math.sqrt(p * (p - a) * (p - b) * (p - c))
print("三角形的面积是{:.2f}".format(s))
```

习题 3 答案

1. 选择题

B　A　B　B

2. 简答题

（1）字符串有哪 3 种表示形式?

Python 中的字符串是字符集合，它被引号所包含，引号可以是单引号、双引号或者三引号(三个连续的单引号或者双引号)。单引号和双引号包含的是单行字符串，二者的作用相同。三引号可以包含多行字符串。

（2）字符串的 format()方法中，该方法的参数有哪几种?

str.format()方法中的 str 被称为模板字符串，其中包括多个由“{}”表示的占位符，占位符接收 format()方法中的参数。str 模板字符串与 format()方法中的参数的对应关系有以下 3 种情况。

第一种，位置参数匹配。在模板字符串中，如果占位符{}为空（没有表示顺序的序号），按照参数出现的先后次序匹配。如果占位符{}指定参数的序号，按照序号对应参数替换。

第二种 ，使用键值对的关键字参数匹配。format()方法中的参数用键值对形式表示时，在模板字符串中用“键”来表示。

第三种 ，使用序列的索引作为参数匹配。如果 format()方法的参数是列表或元组，可以用其索引（序号）来匹配。

（3）字符串的合并与拆分是两种重要的运算，支持合并与拆分的函数是什么，通过示例来验证。

split(sep, num)，以 sep 为分隔符分隔字符串，如果 num 有指定值，则仅截取 num 个子字符串

join(seq)，以指定字符串作为分隔符，将 seq 中所有的元素(的字符串表示)合并为一个新的字符串

3. 编程题

（1）编写程序，给出一个英文句子，统计单词个数。

```
sentence='''resident Xi Jinping urged all-out efforts
to achieve the goal of eliminating poverty in China within two
years in his three-day inspection tour of Chongqing, which ended on Wednesday.
'''
sentence=sentence.replace(","," ")
sentence=sentence.replace("."," ")
words=sentence.split()
print(len(words))
```

（2）编写程序，给出一个字符串，将其中的字符“E“用空格替换后输出。

```
sentence='''residEnt Xi Jinping urgEd all-out Efforts'''
print(sentence.replace("E"," "))
```

（3）从键盘交互式输入一个人的 18 位的身份证号，以类似于“2001 年 09 月 12 日”的形式输出该人的出生日期。

```
xx=input("IDCard Number:")
print(xx[6:10]+"年"+xx[10:12]+'月'+xx[12:14]+"日")
```

习题 4 答案

1. 选择题

D　A　A　D　A　　D　B　D

2. 简答题

（1）叙述 pass 语句的作用。

pass 语句的含义是空语句，主要是为了保持程序结构的完整性设计的。pass 语句一般用做占位语句，该语句不影响其后面语句的执行。

（2）跳转语句 break 和 continue 的区别是什么?

break 语句的作用是循环体内部跳出，即结束循环。有时也称为断路语句，就是循环被中断，不再执行循环体。

continue 语句必须用于循环结构中，它的作用是终止当前这一轮的循环，跳过本轮剩余的语句，直接进入下一轮循环。continue 语句有时也被称为短路语句，指的是只对本次循环短路，并不终止整个循环。

（3）叙述 for 循环和 while 循环的执行过程。

for 循环从序列中逐一提取元素，放在循环变量中，对于序列中的每个元素执行一次语句块。序列可以是字符串、列表、文件或 range()函数等。

3.编程题

（1）给定某一字符串 s，对其中的每一字符 c 大小写转换：如果 c 是大写字母，则将它转换成小写字母；如果 c 是小写字母，则将它转换成大写字母；如果 c 不是字母，则不转换。

```
s="abc32TT9ac"
t=""
for i in range(len(s)):
   if s[i].islower():
      t+=s[i].upper()
```

```
    elif s[i].isupper():
        t+=s[i].lower()
    else:
        t+=s[i]
print(t)
```

（2）输入一个整数，将各位数字反转后输出。

```
num=eval(input("请输入一个整数："))
s=str(num)
print(s[::-1])
```

（3）计算 $1^2-2^2+3^2-4^2+\cdots+97^2-98^2+99^2$。

```
s=0
flag=1
for i in range(1,100):
    s=s+i*i*flag
    flag*=-1

print(s)
```

（4）一个数如果恰好等于它的因子这和，这个数就称为“完数”，例如，6 的因子为 1，2，3，而 6=1+2+3，因此 6 就是“完数”。编程找出 100 内的所有完数。

```
for num in range(1,100):
    t=0
    s=""
    for i in range(1,num//2+1):
        if num%i==0:
            t+=i
            s+=str(i)+"  "
    if t==num:
        print("{}是完数，其因子为：{}".format(t,s))
```

（5）输入两个正整数 m 和 n，求其最大公约数和最小公倍数。

```
m=eval(input("请输入正整数 m:"))
n=eval(input("请输入正整数 n:"))
if m<n:m,n=n,m
t=m*n
r=m%n
while r!=0:
    m=n
    n=r
    r=m%n
print("最大公约数是：{}".format(n))
print("最小公倍数是：{}".format(int(t/n)))
```

习题 5 答案

1. 选择题

B　A　D　C　D　　C　D　C　D　D

2. 简答题

（1）列表、元组、字典都用什么标记或函数创建?

列表使用标记“[]”可以创建，元组通常使用标记“()”创建。字典可以用标记“{}”创建，字典中每个元素都包含键和值两部分，键和值用冒号分开，元素之间用逗号分隔。

列表和元组相互转换的函数是 tuple(lst)和 list(tup)，都可以创建元组和列表；dict()是用于创建字典的函数。

（2）列表和元组两种序列结构有什么区别?

列表他元组都是 Python 中常用的序列类型。创建列表时，只要把逗号分隔的元素使用方括号括起来即可。列表是可变的，可向列表中任意增加元素或删除元素，可以对列表进行遍历、排序、反转等操作。

元组是包含 0 个或多个元素的不可变序列类型。创建元组时，只要将元组的元素用小括号包围，并使用逗号隔开即可。元组中的任何元素不能替换或删除。元组与列表的区别在于元组的元素不能修改。

（3）字典有什么特点？列出任意 5 种字典的操作函数。

字典是 Python 中内置的映射类型。字典可以看成元素对构成的列表，其中一个元素是键，另一个元素是值。在搜索字典时，首先查找键，当查找到键后就可以直接获取该键对应的值。

字典中的值并没有特殊的顺序，但是都存储在一个特定的键(key)里，键可以是数字、字符串以及元组等。此外，字典中的元素（键值对）是无序的。

函数或操作	功能描述
dicts.keys()	返回所有的键信息
dicts.values()	返回所有的值信息
dicts.items()	返回所有的键值对
dicts.get(key, defualt)	键存在则返回相应值，否则返回默认值
dicts. pop(key, default)	键存在则返回相应值，同时删除键值对，否则返回默认值
dicts. popitem()	随机从字典中取出一个键值对，以元组(key,value)形式返回
dicts.clear()	删除所有的键值对
dicts.update(dicts2)	更新字典，参数 dict2 为更新的字典

（4）遍历列表和元组有哪几种方法?

列表和元组的遍历的方式相同，以列表为列说明如下。

遍历列表可以逐个处理列表中的元素，通常使用 for 循环和 while 循环来实现。for 循环可以直接访问列表中的元素，用 while 循环遍历列表，需要先获取列表的长度，将获得的长度作为循环的条件。

3. 编程题

（1）编写程序，随机生成由英文字符和数据组成的 4 位验证码。

```
import random
letters1="ABCDEFGHIJKLMN"
letters2="opqrstuvwxyz"
letters3="0123456789"
letters=letters1+letters2+letters3
```

```
    code=""
    for i in range(4):
        code+=random.choice(letters)
    print("产生的验证码是：{}".format(code))
```

（2）用字典描述学生信息，包括 no(学号)，name(姓名),score(成绩)等。使用列表存储学生信息的，并根据给定学生姓名，查找学生的信息。

```
    stu1={"no":101,"name":"Rose","address":"Changjianroad","score":92}
    stu2=dict(id=201,name="Mike",address="Huangheroad",score=83)
    stu3=dict([('id',103),('name','Kate'),('address','Xinanroad'),('pcode','116033'),('score',90)])

    lst=[stu1,stu2,stu3]
    var=input("请输入要查找学生的姓名：")
    result=False
    for item in lst:
        if var.strip() in item.values():
            print(item)
            result=True
    if result==False:
        print("信息不存在")
```

（3）使用 input 函数，输入若干单词，按字典顺序输出单词(如果某个单词出现多次，只输出一次)。

```
    words=input("请输入若干单词，用英文逗号分割：")
    word_list=words.split(",")
    aset=set()
    for i in word_list:    #去除重复元素
        aset.add(i)

    word3=list(aset)   #将集合转换为列表后，排序
    word3.sort()
    print(word3)
```

（4）利用元组创建一个存储 Python 关键字的对象，并检测给定的单词是否是 Python 关键字。

```
    tup1=("in","break","continue","True","False","if","while","pass")
    var=input("Enter a word:")
    if var in tup1:
        print("这个单词是 Python 关键字")
    else:
        print("这个单词不是 Python 关键字")
```

习题 6 答案

1. 选择题

B　D　D　B　C　　B　C　D

2. 简答题

（1）什么是嵌套函数？举例说明。

嵌套函数通常是指函数的嵌套定义，即在函数内部定义的函数，但内嵌的函数只能在该函数内部使用，闭包应用了函数的嵌套定义。

```
def sum(n):
    def fact(a):  # 嵌套函数，求阶乘
        t = 1
        for i in range(1, a + 1):
            t *= i
        return t
    s = 0
    for i in range(1, n + 1):
        s += fact(i)  # 调用嵌套函数 fact()
    return s

n=5
print(sum(n))
```

（2）函数的可变参数有哪几种，各有什么特点？

可变参数指的是在函数定义时，该参数可以接受任意个数的参数。可变参数有两种形式，参数名称前加星号（*）或者加两个星号（**）。定义可变参数的函数语法格式如下。

```
def funname(formal_args,*args,**kwargs):
    statements
    return expression
```

在上面的函数格式定义中，formal_args 为定义的传统参数，代表一组参数，*args 和**kwargs 为可变参数。函数传入的参数个数会优先匹配 formal_args 参数的个数，*args 以元组的形式保存多余的参数，**kwargs 以字典的形式保存带有指定名称形式的参数。

（3）函数传递时，基本数据类型做参数和组合数据类型做参数，有什么区别？举例说明。

基本数据类型的变量作为实际参数时，是将常量或变量的值传递给形参，是一种值传递的过程，实参和形参是两个独立不相关的变量，因此，实参值一般是不会改变的。

列表、元组、字典等组合数据类型的变量用做函数参数时，这些变量在函数体外，是全局变量。形参和实参之间传递的只是组合数据类型变量（参数）的地址（引用），如果在函数内部修改了参数的值，参数的地址是不发生改变的，这种修改将影响到外部的全局变量。

（4）在内置函数中，列出 5 种常用的数学运算函数和字符运算函数

（略）

3. 编程题

（1）编写函数 isodd(x)，若 x 不是整数，给出提示后退出程序；如果 x 为奇数，返回 True，如果 x 为偶数，返回 False。

```
def isodd(x):
    if int(x)!=x:
        print("x 不是整数，程序退出")
        return
    elif x//2 != x/2:
        return True
    else:
        return False
```

```
print(isodd(3))
print(isodd(3.2))
print(isodd(32))
```

（2）编写函数 change(str1)，其功能是对参数 str1 进行大小写转换，其中的大写字母转换成小写字母；小写字母转换成大写字母；非英文字符不转换。

```
def change(str1):
    t=""
    for i in range(len(str1)):
        if str1[i].islower():
            t+=str1[i].upper()
        elif str1[i].isupper():
            t+=str1[i].lower()
        else:
            t+=str1[i]
    return t
print(change("iu98kLLD"))
```

（3）编写并测试函数 gcd(m, n)和 lcm(m, n)，功能是求两个整数的最大公约数和最小公倍数。

```
def gcd(m,n):
    if m<n:m,n=n,m
    r=m%n
    while r!=0:
        m=n
        n=r
        r=m%n
    return n

def lcm(m,n):
    t=m*n
    return int(t/gcd(m,n))

m=eval(input("请输入正整数m:"))
n=eval(input("请输入正整数n:"))
print("最大公约数是",gcd(m,n))
print("最小公倍数是",lcm(m,n))
```

（4）编写并测试函数 reverse(x)，输入一个整数，将各位数字反转后输出。

```
def reverse(x):
    y=str(x)
    y=y[::-1]
    return int(y)
print(reverse(12345))
```

（5）用递归方法反转一个字符串，例如“abcde”，反转为“edcba”。

```
def func(s):
    if len(s) <1:
        return s
    return func(s[1:])+s[0]

s="abcde"
result = func(s)
print(result)
```

（6）编写程序求 $1^2-2^2+3^2-4^2+\cdots+97^2-98^2+99^2$。

（略）

习题 7 答案

1. 选择题

A　A　D　C　D　A　C　B

2. 简答题

(1) 什么是对象？什么是类？类与对象的关系是什么？

对象（Object）对应客观世界的事物，将描述事物的一组数据和与这组数据有关操作封装在一起，形成一个实体，这个实体就是对象。具有相同或相似性质的对象的抽象就是**类**（Class）。因此，对象的抽象是类，类的具体化就是对象。

(2) 类属性与实例属性的区别是什么？

属性也叫成员变量，分为两种类型：一种是实例属性，另一种是类属性。

实例属性是在构造方法__init__()中定义的，定义时以 self 作为第 1 个参数；类属性是在类中方法之外定义的属性。在类的的外部，实例属性属于实例（对象），只能通过对象名访问；类属性属于类，可以通过类名访问，也可以通过对象名访问，被类的所有对象共享。

(3) 构造方法作用是什么？与实例方法有什么不同?

类中定义的名字为__init__()的方法（以两个下画线“_”开头和结尾）被称为**构造方法**。一个类定义了__init__()方法以后，创建对象时，就会自动为新生成的对象调用该方法。构造方法一般用于完成对象数据成员设置初值或进行其他必要的初始化工作。

实例方法也叫成员方法，需要通过对象调用，实现特定的功能。

(4) 列举出 5 种重载的运算符及对应的方法。

方法名　重载说明 运算符调用方式

__add__　对象加法运算　x+y，　x+=y

__sub__　对象减法运算 x-y，　x-=y

__div__　对象除法运算 x/y，　x/=y

__mul__　对象乘法运算 x*y，　x*-=y

__mod__　对象取余运算 x%y，　x%=y

__repr__或__str__　打印或转换对象　print(x)、repr(x)、str()

__getitem__　对象对象索引运算 x[key]，x[i:j]

__setitem__　对象索引赋值 x[key]，x[i:j]=

3. 编程题

(1)设计一个 **StuGroup** 类，在该类中包括：一个数据成员 score(每个学生的分数)及两个类成员变量 total(总分)和 count(人数)。成员方法 setScore(socre)和 getScore（ ）用于设置和获得分数，成员方法 sum()用于累计总分，类方法 average()用于求平均值。交互式输入该组学生的成绩，显示该组学生的总分和平均分。

```
class StuGroup:
```

```
    total = count = 0

    def setScore(self, score):
        self.score = score

    def getScore(self):
        return self.score

    def __init__(self, score):
        self.setScore(score)
        StuGroup.count += 1
        self.sum()

    def sum(self):
        StuGroup.total+=self.getScore()

    @classmethod
    def average(cls):
        return StuGroup.total/StuGroup.count

    @classmethod
    def show(cls):
        print("总分是：{}".format(StuGroup.total))
        print("平均分是：{}".format(StuGroup.average()))

s1=StuGroup(90)
s2=StuGroup(87)
s3=StuGroup(93)

StuGroup.show()
```

(2)为二次方程式 $ax^2+bx+c=0$ 设计一个名为 Equation 的类，这个类包括：

- 代表三个系数的成员变量 a、b、c。
- 一个参数为 a、b、c 的构造方法。
- 一个名为 getDiscriminant()的方法返回判别式的值。
- 一个名为 getRoot1()和 getRoot2()的方法返回等式的两个根：如果判别式为负，这些方法返回 0。

```
import math
class Equation:
    def __init__(self,a,b,c):
        self.a=a
        self.b=b
        self.c=c

    def getDeta(self):
        return self.b*self.b-4*self.a*self.c

    def getRoot1(self):
        if self.getDeta()<0:
            return 0
        else:
            return (-self.b+math.sqrt(self.getDeta()))/(2*self.a)
```

```
    def getRoot2(self):
         if self.getDeta()<0:
            return 0
         else:
            return (-self.b-math.sqrt(self.getDeta()))/(2*self.a)
e1= Equation(1,4,5)
print(e1.getRoot1(),e1.getRoot2())
```

(3)设计一个描述自由落体运动的类，要求能获得任意时刻的速度及位移，并进行测试。已知重力加速度为 9.8m/s^2。

```
class Movement:
    acceleration=9.8

    def __init__(self,t):
        self.t=t

    def getSpeed(self):
        return Movement.acceleration*self.t

    def getDistance(self):
        return Movement.acceleration*self.t*self.t/2

m1=Movement(3)
print("{:.1f}秒后，速度是{:.2f}，位移是{:.2f}".
     format(m1.t,m1.getSpeed(),m1.getDistance()))
```

(4)设计一个整形数组的封装类。

（略）

习题 8 答案

1. 选择题

B　D　A　D　C

2. 简答题

（1）Python 导入模块时一般采用什么搜索顺序？

Python 模块的路径搜索顺序如下：

当前目录，环境变量 pythonpath 包含的目录，python 标准库目录和第三方包目录。

（2）Python 的内置属性__name__ 有什么作用？

__name__ 是 Python 的内置属性，用于表示当前模块的名字，也能反映一个包的结构。如果.py 文件作为模块被调用，__name__的属性值为模块文件的主名，如果模块独立运行，__name__属性值为__main__。

语句 if __name__ == 'main' 的作用就是控制这两种不同情况执行代码的过程，当 __name__ 值为“main”时，文件作为脚本直接执行，而使用 import 或 from 语句导入到其他程序中时，模块中的代码是不会被执行的。

（3）Python 的第三方库如何安装？如何查看当前计算机中已经安装的第三方库？

用户可以使用 pip 工具安装第三方库。pip 工具由 Python 官方提供并维护，是常用且高效的在线第三方库安装工具。pip3 是 Python 的内置命令，用于 Python3 版本安装第三方库，需要在命令行下执行。

pip3 list 命令用于列出当前系统中已安装的第三方库。

（4）叙述用 Python 的第三方库 pyinstaller 打包文件的过程和注意事项。

pyinstaller 是用于源文件打包的第三方库，它能够在 Windows、Linux、Mac OS X 等操作系统下将 Python 源文件打包。打包后的 Python 文件可以在没有安装 Python 的环境中运行，也可以作为一个独立文件方便传递和管理。

使用 pyinstaller 命令打包文件，需要注意以下几个问题。

- 文件路径中不能出现空格和英文句号(.)，如果存在，需要修改 Python 源文件的名字。
- 源文件必须是 UTF-8 编码。采用 IDLE 编写的源文件均保存为 UTF-8 格式，可以直接使用。
- 如果命令提示符前的路径提示符是 d:\python，生成的打包生成文件的位置与与'>'提示符前的路径是一致的。

3. 编程题

（1）使用 random 库，产生 10 个 100 到 200 之间的随机数，并求其最大值、平均值、标准差和中位数。

```
#平均数
def mean(numlist):
    s = 0.0
    for num in numlist:
        s = s + num
    return s/len(numlist)


##标准差=sqrt(((x1-x)^2 +(x2-x)^2 +......(xn-x)^2)/n)。
def dev(numlist,mean):
    sdev = 0.0
    for num in numlist:
        sdev = sdev + (num - mean)**2
    return (sdev /(len(numlist)-1) )** 0.5

def median(numlist):
    numlist.sort()
    half = len(numlist) // 2
    return (numlist[half] + numlist[~half]) / 2

#函数调用
ls1 = [1,3,-6,5,8]
ls2=[3.2,9,-32,56,0,1]

v1=max(ls2)
v2=mean(ls2)
v3=dev(ls2,v2)
v4=median(ls2)
```

```
print("最大值:{:.2f}, 平均值:{:.2f},标准差:{:.2f},中位数:{:.2f}".format(v1,v2,v3,v4))
```

（2）使用 datetime 库，对一个日期（含时间），输出不少于 8 种 日期格式。

（略）

（3）使用 turtle 库绘制一个叠加三角形。

```
import turtle as t
t.setup(260, 260, None,None)
t.pu()
t.fd(-60)
t.pensize(2)
t.width(2)
t.pencolor("darkgreen")
t.pd()
t.fd(100)
t.seth(120)
t.pencolor("black")
t.fd(100)
t.seth(-120)
t.pencolor("blue")
t.fd(100)
t.pencolor("purple")
t.fd(100)
t.seth(0)
t.pencolor("green")
t.fd(100)
t.pencolor("gold")
t.fd(100)
t.seth(120)
t.pencolor("red")
t.fd(100)
t.seth(-120)
t.pencolor("grey")
t.fd(100)
t.seth(120)
t.pencolor("violet")
t.fd(100)
```

（4）编写程序统计《水浒转》中前 10 位出场最多的人物。

（略）

习题 9 答案

1. 选择题

A　D　C　D　B

2. 简答题

（1）常用的文本文件编码方式有哪几种？汉字在不同的编码中各占几个字节？

- ASCII 码即美国标准信息交换码，仅对 10 个数字、26 个大写英文字母、26 个小写英文字母及一些其它常用符号进行了编码。 ASCII 采用 8 位（1 字节）编码.
- UTF-8 编码是国际通用的编码方式，用 8 位（1 字节）表示英语（兼容 ASCII 码），以

24 位（3 字节）表示中文及其他语言，UTF-8 对全世界所有国家的字符进行了编码。

- GB2312 编码是中国制定的中文编码，用 1 字节表示英文字符，用 2 字节表示汉字字符。GBK 是对 GB2312 的扩充。
- Unicode 是国际组织制定的可以容纳世界上所有文字和符号的字符编码方案，它是编码转换的基础。

采用不同的编码方式，写入文件的内容可能是不同的。就汉字编码而言，GBK 编码的 1 个汉字占 2 个字符，UTF-8 编码的 1 个汉字占 3 个字符，Unicode 编码中的 1 个汉字占 1 个字符。

（2）列出任意 4 种文件访问模式，说明其含义。

r　只读模式打开，默认值。该模式打开的文件必须存在，如果不存在，将报异常

r+　读写模式打开。该模式打开的文件必须存在，如果不存在，将报异常

w　写模式打开。文件如果存在，清空内容后重新创建文件

w+　读写模式打开。文件如果存在，清空内容后重新创建文件

a　追加的方式打开，写入的内容追加到文件尾。该模式打开的文件如果已经存在，不会清空，否则新建一个文件

（3）文本文件和二进制文件在读写时有什么区别？举例说明。

当文件被打开后，根据文件的访问模式可以对文件进行读写操作。以文本文件方式打开的文件，默认的，程序按照当前操作系统的编码方式来读写文件，也可以指定编码方式来读写文件；如果文件是以二进制文件方式打开的，按字节流方式读写。

read()和 write()方法可以读写二进制文件，但二进制文件只能读写 bytes 字符串。例如，

```
>>> fileb=open(r"d:\pythonfile36\\ch7\\mydata.dat",'wb')   #以'wb'方式打开二进制文件
>>> fileb.write(b"Hello Python")                  #写 bytes 字符串
>>> n=123
>>> fileb.write(bytes(str(n),encoding='utf-8'))  #将整数转换为 bytes 字符串写入文件
>>> fileb.write(b"\n3.14")
>>> fileb.close()
#以'rb'方式打开二进制文件
>>> file=open(r" d:\pythonfile36\\ch7\\mydata.dat",'rb')
>>> print(file.read())
b'Hello Python123\n3.14'
>>> file.close()
```

（4）readlines()方法和 readline()方法读取文本文件时，主要区别是什么？

readline([size]) 主要用于读取文件一行内容，可以使用参数 size，读取指定长度的字符或字节。

readlines([hint])用于读取文件的所有行，返回行所组成的列表。参数 hint 指定读入行数。

3. 编程题

（1）将一个文件中的所有英文字母转换成大写，复制到另一文件中。

```
fi=open("ex0701.py",'r')
fo=open("f2.txt",'w')
for line in fi:
   line=line.upper()
```

```
        fo.write(line)

    fi.close()
    fo.close()
```

（2）将一个文件中的指定单词删除后，复制到另一个文件中。

```
    fi=open("ex0702.py",'r')
    fo=open("f2.txt",'w')
    deleteword="line"
    for line in fi:
        line1=line.replace(deleteword,"")
        #print(line1)
        fo.write(line1)
    fi.close()
    fo.close()
```

（3）接收用户从键盘输入的一个文件名，然后判断该文件是否存在于当前目录。若存在，则输出以下信息：文件是否可读和可写、文件的大小、文件是普通文件还是目录。

```
    import os,os.path

    filename=input("请输入文件的全名:")
    if os.path.exists(filename):
        print("该文件存在于当前目录下")
        print("-----下面是文件信息-------")
        print("文件大小是: ",os.path.getsize(filename))
        if os.path.isfile(filename):
            print(filename,"是一个文件")
        else:
            print(filename,"是一个目录")

    else:
        print("该文件不存在! ")
```

（4）将一文本文件加密后输出，规则如下：大写英文字符 A 变换为 C，B 变换为 D，……，Y 变换为 A，Z 变换为 B，小写英文字符规则同上，其他字符不变。

```
    s="abc123^&**KJFDFDLKy"
    s2=""
    for c in s:
        if c.islower():
            c=(ord(c)-ord('a')+3)%26+ord('a')
            #print(chr(c),end="")
        elif c.isupper():
            c=(ord(c)-ord('A')+3)%26+ord('A')
            #print(chr(c),end="")
        else:
            c=ord(c)
        s2+=chr(c)
    print(s2)
```

习题 10 答案

1. 选择题

D　B　B　C　C　　D　B　A　B　D

2. 简答题

(1) 什么叫异常?简述 Python 的异常处理机制。

异常（Exception）就是程序在运行过程中发生的，由于硬件故障、软件设计错误、运行环境不满足等原因导致的程序错误事件，比如除 0 溢出、引用序列中不存在的索引、文件找不到等，这些事件的发生将阻止程序的正常运行。

异常处理机制如下。

- 程序执行过程中如果出现异常，会自动生成一个异常对象，该异常对象被提交给 Python 解释器，这个过程称为抛出异常。抛出异常也可以由用户程序自行定义。
- 当 Python 解释器接收到异常对象时，会寻找处理这一异常的代码并处理，这一过程叫捕获异常。
- 如果 Python 解释器找不到可以处理异常的方法，则运行时系统终止，应用程序退出。

Python 通过 try-except 语句处理异常。

(2) 除了教材上列出的 7 种常见异常外，查文档，列举 3 种其他的内置异常类。

KeyboardInterrupt，TimeoutError，EOFError，MemoryError，ImportError 等

(3) 如何创建用户自定义异常?

创建用户自定义异常的基本步骤如下。

- 声明一个新的异常类，使之以 Exception 类或其他某个已经存在的系统异常类或用户异常类为父类。
- 为新的异常类定义属性和方法，或重载父类的属性和方法，使这些属性和方法能够体现该类所对应的错误的信息

(4) 已知学生类 Student 及年龄异常类 AgeException 如下，分析程序的执行结果。

19　　#创建 z3 对象时正常执行。

年龄错误　#创建 li 对象时报异常

3. 编程题

(1)编程实现索引超出范围异常 IndexError 类型。

```
try:
    chars=['a','b','c','d','e']
    chars[4]=1
    print(chars)
    chars[5]='xx'
except IndexError:
    print("索引超过范围")
```

(2)设计一个一元二次方程类，并为这个类添加异常处理。

```
import math
class EquationException(Exception):
```

```
    def __init__(self,eid,message):    #异常描述
        self.eid=eid
        self.message=message

class ExceptionDemo:                   #业务逻辑
    def computing(self,a,b,c):
        print("called computing()");
        deta=b*b-4*a*c
        if deta<0:
            raise EquationException(101,"deta 值小于零")
        else:
            x1=(-b+math.sqrt(deta))/(2*a)
            x2=(-b-math.sqrt(deta))/(2*a)
            print("一元二次方程的根是{}, {}".format(x1,x2))
            print("normal exit")


myobject=ExceptionDemo()           #功能测试
try:
    myobject.computing(4,4,-2)
    myobject.computing(2,3,2)
except EquationException as e:
    print("Exception caught,id:{},message:{}".format(e.eid,e.message))
```

(3)定义一个 Circle 类,其中有求面积的方法,当半径小于 0 时,抛出一个用户自定义异常。(略)

(4)从键盘输入一个整数,求 100 除以它的商,并显示。要求对从键盘输入的数值进行异常处理。

```
n=eval(input("请输入数值: "))
try:
    s=100/n
    print(s)
except Exception as ee:
    print(ee)
```

习题 11 答案

1. 选择题

A　C　D　D

2. 简答题

(1) 使用 grid()方法的布局有什么特点?

使用 grid()方法的布局被称为网格布局,它按照二维表格的形式,将容器划分为若干行和若干列,组件的位置由行列所在位置确定。

(2) Radiobutton 组件和 Checkbutton 组件的区别是什么?

Radiobutton 组件用于创建单选按钮组。按钮组由多个单选按钮组成,选中按钮组中的一项时,其他选项则取消取中。

Checkbutton 组件用于创建复选框,Checkbutton 组件与 Radiobutton 组件的功能类似，但 Radiobutton 组件实现的是单选功能，而 Checkbutton 在系列选项中可以选择 0 个或多个，实现复选功能。

(3) 列举出 Label 组件 6 种以上的属性。

属性	说明
text	设置标签显示的文本
bg 和 fg	指定组件的背景色和前景色
width 和 height	指定组件的宽度和高度
padx 和 pady	组件内文本 左右和上下 的预留空白宽度，默认值为 1(像素)
anchor	设置文本在组件内部的位置，取值为 N、S、W、E、NW、SW、NE、SE
justify	设置文本对齐方式，取值为 LEFT(左对齐)、 RGHT(右对齐)或 CENTER(居中对齐)
font	设置字体

(4) Python 的 GUI 编程中，组件和容器的概念有什么区别?

组件是指标签、按钮、列表框等对象，需要将其放在容器中显示。**容器**是指可放置其他组件或容器的对象，例如，窗口或 Frame（框架），容器也可以叫做容器组件。

3. 编程题

(1) 编制求两个正整形数最小公倍数程序。要求：两个输入框 txt、txt2，用来输入整形数据；一个按钮；一个不可编辑的输入组件 txt3。当单击按钮时，在 txt3 中显示两个整形数的最小公倍数的值。

```
from tkinter import *
def computing():
    n1 = int(number1.get())
    n2=int(number2.get())
    if n1<n2:n1,n2=n2,n1
    t=n1*n2
    temp=n1%n2
    while temp!=0:
        n1=n2
        n2=temp
        temp=n1%n2


    result="最小公倍数是："+ str(t/n2)
    label3.config(text=result)

win=Tk()
win.title("最小公倍数")
win.geometry("300x350")
label1=Label(win,text='请输入数值1：  ')
label1.config(width=16,height=3)
label1.config(font=('宋体',12))
label1.grid(row=0,column=0)
```

```
number1 = StringVar()
txt1 = Entry(win,textvariable = number1,width=16)
txt1.grid(row=0,column=1)

label0=Label(win,text='请输入数值 2： ')
label0.config(width=16,height=3)
label0.config(font=('宋体',12))
label0.grid(row=1,column=0)
number2 = StringVar()
txt2 = Entry(win,textvariable = number2,width=16)
txt2.grid(row=1,column=1)

label2=Label(win,text='请单击确认：')
label2.config(width=14,height=3)
label2.config(font=('宋体',12))
label2.grid(row=2,column=0)

button1=Button(win,text="计算")
button1.config(justify=CENTER)          #设置按钮文本居中
button1.config(width=14,height=2)       #设置按钮的宽和高
button1.config(bd=3,relief=RAISED)      #设置边框宽度和样式
button1.config(anchor=CENTER)           #设置内容在按钮内部居中
button1.config(font=('隶书',12))
button1.config(command=computing)
button1.grid(row=2,column=1)
label3=Label(win,text='显示结果 ')
label3.config(width=22,height=3)
label3.config(font=('宋体',12))
label3.place(x=50,y=230)
win.mainloop()
```

(2) 设计 GUI 界面，模拟 QQ 登录界面，用户输入用户名和密码，如果正确提示登录成功；否则提示登录失败。

请参考例 11-17 完成。

(3) 例 11-17 使用 Button 组件的 command 参数实现事件处理，将事件处理的方法使用 bind() 方法实现。

```
import tkinter
import tkinter.messagebox
#创建应用程序窗口
win = tkinter.Tk()
varName = tkinter.StringVar()
varName.set('')
varPwd = tkinter.StringVar()
varPwd.set('')
#创建标签
labelName = tkinter.Label(text='User Name:', justify=tkinter.RIGHT,width=80)
labelName.place(x=10, y=5, width=80, height=20)
#创建文本框，同时设置关联的变量
entryName = tkinter.Entry(win, width=80,textvariable=varName)
```

```
entryName.place(x=100, y=5, width=80, height=20)
labelPwd = tkinter.Label(win, text='User Pwd:', justify=tkinter.RIGHT, width=80)
labelPwd.place(x=10, y=30, width=80, height=20)
#创建密码文本框
entryPwd = tkinter.Entry(win, show='*',width=80, textvariable=varPwd)
entryPwd.place(x=100, y=30, width=80, height=20)
users={"zhang3":"a12","admin":"123456","li4":"abc"}
def login(event):    #登录按钮事件处理函数
    #获取用户名和密码
    name = entryName.get()
    pwd = entryPwd.get()
    flag=False
    for item in users:
        if item==name and users[item]==pwd:
            flag=True
    if flag==True:
        tkinter.messagebox.showinfo(title='Python tkinter',message='OK')
    else:
        tkinter.messagebox.showerror('Python tkinter', message='Error')
def cancel(event):     #取消按钮的事件处理函数
    varName.set('')
    varPwd.set('')
#创建按钮组件，同时设置按钮事件处理函数
buttonOk = tkinter.Button(win, text='Login')
buttonOk.bind("<Button-1>",login)
buttonOk.place(x=30, y=70, width=50, height=20)
buttonCancel = tkinter.Button(win, text='Reset')
buttonCancel.bind("<Button-1>",cancel)
buttonCancel.place(x=90, y=70, width=50, height=20)
win.mainloop()   #启动消息循环
```

习题 12 答案

1. 选择题

B　C　A　B　B　B　B

2. 简答题

(1) Python 中访问 SQLite 数据库主要使用哪些对象，功能是什么？

- sqlite3.version：常量，返回 sqlite3 模块的版本号。
- sqlite3.sqlite_version：常量，返回 sqlite 数据库的版本号。
- sqlite3.Connection ：数据库连接对象。
- sqlite3.Cursor：游标对象。
- sqlite3.Row：行对象。
- sqlite3.connect(dbname)：函数，链接到数据库，返回 Connection 对象。

(2)列举出 SQLite 数据库支持的 5 种数据类型？SQLite 数据库的动态数据类型有什么特点？

类型	说明

smallint	16 位整数
integer	32 位整数
decimal(p,s)	小数。p 是数字的位数，s 是小数位数
float	32 位浮点数
double	64 位浮点数
char(n)	固定长度字符串，n 不能大于 254
varchar(n)	不固定长度字符串，n 不能大于 4000
graphic(n)	和 char(n)一样，单位是两个字节。n 不能大于 127
vargraphic(n)	长度可变且最大长度不能大于 4000 的双字节字符串
date	日期，包含年、月、日
time	时间，包含时、分、秒
datetime	日期和时间

SQLite3 使用动态的数据类型，数据库管理系统会根据列值自动判断列的数据类型。这与多数 SQL 数据库管理系统使用静态数据类型是不同的。SQLite3 的动态数据类型能够向后兼容其他数据库普遍使用的静态类型。

(3) 游标对象的 fetchone(),fetchall(),fetchmang()系列方法有什么区别?

fetch 系列方法用于获取游标的查询结果集。具体如下。

- cur.fetchone()：返回结果集的下一行（Row 对象），无数据时，返回 None。
- cur.fetchall()：返回结果集的剩余行（Row 对象列表），无数据时，返回空 List。
- cur.fetchmany()：返回结果集的多行（Row 对象列表），无数据时，返回空 List。

3. 编程题

(1) 基于书中创建的 test.db 数据库和 employee 表,完成下列 SQL 命令,表中初始数据如下。

[1132，李四，男，部门经理，7548.6，11；

1443，王五， 男， 职员，6656，14；

1036，高七，女，经理，7600，10]

①inser into 命令向表中任意插入 2 条记录。

```
    insert into employee(emp_id,emp_name,sex,title,wage,dep_id) values(101,'John','男','经理',7000,11)
```

②用 delete from 命令删除 emp_id 为 1443 的雇员记录。

```
    delete from employee where emp_id=1443
```

③用 update 命令为职称为部门经理的雇员工资增加 10%。

```
    update employee set wage=wage*1.1 where title="部门经理"
```

④查询工资大于 7000 的部门经理信息。

```
    select * from employee where wage>7000 and title="部门经理"
```

⑤查询不同性别的雇员人数。

```
    select sex,count(*) as 雇员人数  from employee group by sex
```

⑥查询 employee 表中工资在 7000 元以上的雇员信息，并将查询的结果按工资降序排序。

```
    select * from employee where wage>7000 order by wage
```

⑦查询 employee 表中男女雇员人数、平均工资（显示：性别、人数，平均工资）。

```
select sex as 性别,count(*) as 人数,average(*)  as 平均工资  from employee group by sex
```

(2) 设计 GUI 界面，模拟用户登录功能，用户输入用户名和密码，如果正确提示登录成功；否则提示登录失败，用户的密码信息保存在 SQLite 数据库中。

注意：有 sqlite 数据库 d:/sqlite/test.db，其中有表 users，其结构为 (username,pwd)，并在该表中保存了用户登录信息。

```
import tkinter
import tkinter.messagebox
import sqlite3

#创建应用程序窗口
win = tkinter.Tk()
varName = tkinter.StringVar()
varName.set('')
varPwd = tkinter.StringVar()
varPwd.set('')
#创建标签
labelName = tkinter.Label(text='User Name:', justify=tkinter.RIGHT,width=80)
labelName.place(x=10, y=5, width=80, height=20)
#创建文本框，同时设置关联的变量
entryName = tkinter.Entry(win, width=80,textvariable=varName)
entryName.place(x=100, y=5, width=80, height=20)
labelPwd = tkinter.Label(win, text='User Pwd:', justify=tkinter.RIGHT, width=80)
labelPwd.place(x=10, y=30, width=80, height=20)
#创建密码文本框
entryPwd = tkinter.Entry(win, show='*',width=80, textvariable=varPwd)
entryPwd.place(x=100, y=30, width=80, height=20)

# 连接数据库的通用函数
def getConnection():
    dbstring="d:/sqlite/test.db"
    conn=sqlite3.connect(dbstring)
    return conn
#print(getConnection())   测试连接是否成功

def login():    #登录按钮事件处理函数
    #获取用户名和密码
    name = entryName.get()
    pwd = entryPwd.get()
    #获取数据中数据
    dbinfo=getConnection()
    cur=dbinfo.cursor()
    sqlstr="select * from user where username=? and pwd=?"
    cur.execute(sqlstr,(name,pwd))
    if cur.fetchone() !=None:  #如果取得记录
        tkinter.messagebox.showinfo(title='Python tkinter',message='OK')
    else:
        tkinter.messagebox.showerror('Python tkinter', message='Error')

def cancel():     #取消按钮的事件处理函数
```

```
    varName.set('')
    varPwd.set('')
#创建按钮组件，同时设置按钮事件处理函数
buttonOk = tkinter.Button(win, text='Login', command=login)
buttonOk.place(x=30, y=70, width=50, height=20)
buttonCancel = tkinter.Button(win, text='Reset', command=cancel)
buttonCancel.place(x=90, y=70, width=50, height=20)
win.mainloop()   #启动消息循环
```

（3）在 SQLite 数据库管理系统中创建数据库 library，在 library 数据库中建立图书表 books(书号 bookno，书名 bookname，出版社 publish，价格 price)。建立简单的图书管理系统，完成图书信息的增加、删除、修改、条件查询等功能。

略，参考例 12-13

习题 13 答案

1. 选择题

C　A　D　D　A

2. 简答题

（1）numpy 库创建数组有哪几种方法?

- 使用 np.array()函数从常规的 Python 列表或元组创建数组。
- 函数 zeros()可创建一个全为 0 的数组，函数 ones()可创建一个全为 1 的数组，函数 empty()创建一个内容随机并且依赖于内存状态的数组。
- arange()函数和 linspace()函数，用于返回一个数列形式的数组。arange()函数类似于 Python 的 range()函数，通过指定开始值，终值和步长来创建一维数组。
- linspace()函数通过指定开始值、终值和元素个数（默认为 50）来创建一维数组。

（2）用于 numpy 数组的形状操作有哪些函数，通过例子说明。

数组的形状（Shape）取决于其每个轴上的元素个数，可以使用 reval()、reshape()、transpose()等方法修改数组的形状。

reval()函数用于降低数组的维度，reshape()用于改变数组的维度，transpose()用于转置数组。

（3）叙述使用 matplotlib.pyplot 模块绘图的过程。

- 使用 import matplotlib.pyplot as plt 语句导入 matplotlib.pyplot 模块。
- 创建绘图对象，通常使用 plt.figure()方法。
- 使用 plot()函数绘图。
- 设置绘图对象的各个属性。
- 图形保存和输出，使用 plt.savefig()方法将当前的 figure 对象保存成图像文件。

3. 编程题

（1）编写绘制余弦三角函数 y=cos(2x)的程序。

```
import matplotlib.pyplot as plt
import numpy as np
plt.figure(figsize=(6,3))      #创建绘图对象
```

```
x=np.arange(0,np.pi*4,0.01)
y=np.cos(2*x)
plt.plot(x,y,color="b",ls='--',linewidth=2.0)
plt.xlabel("x")              #x 轴文字
plt.ylabel("cos(2x)")         #y 轴文字
plt.ylim(-1,1)               #y 轴范围
plt.title("y=cos(2x)")         #图表标题
```

（2）绘制函数曲线。

```
import numpy as np
import matplotlib.pyplot as plt
def f(t):
    return np.exp(-t)*np.sin(2*np.pi*t)
t1=np.arange(0.0,5.0,0.1)
t2=np.arange(0.0,5.0,0.02)
plt.figure(1)
plt.xlim(0,5)
plt.subplot(211)
plt.plot(t1,f(t1),'bo',t2,f(t2),'k')

plt.subplot(212)
plt.plot(t2,np.sin(2*np.pi*t2),'r--')
plt.show()
```

（3）绘制一个散点图。

提示：使用 import matplotlib.pyplot as plt 和 help(plt.scatter)查看绘制散点图的帮助信息。

```
import matplotlib.pyplot as plt
import numpy as np
x=np.arange(0,np.pi*4,0.2)
y=np.sin(x)
plt.xlabel("y=sin(x)")
plt.ylabel("sin(x)")
plt.scatter(x,y,c='b',marker='x')
plt.show()
```

习题 14 答案

1. 选择题

D　C　D　C

2. 简答题

（1）略

（2）列举出 BeautifulSoup4 库操作解析文档树的主要方法和属性。

BeautifulSoup4 库解析文档树时主要用到下面属性。

contents 属性和 children 属性可以获取标记 Tag 的直接子结点；

descendants 属性可以获取所有子结点；

string 属性返回标记中的内容；

strings 属性用于获取多个内容，需要遍历获取；

parent 属性用于获取父结点；

next_sibling 属性用于获取当前结点的下一个兄弟结点；

previous_sibling 属性用表获取当前结点的上一个兄弟结点。

下面是 BeautifulSoup4 库中的方法，其中，soup 是个 BeautifulSoup4 对象

soup.find_all()方法搜索当前 Tag 的所有子结点，并可以设置过滤器的条件。

soup.find()方法返回找到的第一个结点。

soup.select()方法使用选择器选择结点。

（3）urllib.request.urlopen(req)方法中，参数 req 可以有哪几种情况，通过例子说明。

urllib 是用于访问网页的 Python 标准库中的模块。其中的 urllib.request 模块中的 urllib.request.urlopen()函数可以很方便地打开一个网站，读取并打印网页信息。

urlopen()函数中 req 参数可以是一个字符串（‘http://www.shou.com’），也可以是一个 Request 对象。例如：

```
res=urllib.request.urlopen("http://www.sohu.com")  #字符串参数

req=urllib.request.Request("http://www.sohu.com")
res=urllib.request.urlopen(req)  #Request 对象做参数
```

3. 编程题

（1）分别使用 urllib 库和 requests 库爬取 focus.tianya.cn 首页的内容。

- 使用 urllib 库

```
import urllib.request
req=urllib.request.Request("http://focus.tianya.cn")
res=urllib.request.urlopen(req)
html=res.read(320)
print(html.decode("utf-8"))
res.close()
```

- 使用 requests 库

```
import requests
def getHTMLText(url):
    r=requests.get(url,timeout=15)
    r.raise_for_status()
    r.encoding='utf-8'    #如果中文字符不能正常显示，更改编码方式为 utf-8
    return r.text[:320]

url = "http://focus.tianya.cn"
text=getHTMLText(url)
print(text)
```

（2）编写程序获取 http://www.lnzsks.com/listinfo/NewsList_1002_1.html 的招考要闻的信息。

略，参考例 14-5 完成。