# Python概述

1. **填空题**
2. Python是面向 的高级语言。
3. Python可以在多种平台运行，这体现了Python语言 的特性。
4. Python模块的本质是 文件。
5. 使用 关键字可以在当前程序中导入模块。
6. 使用 语句可以将指定模块中的全部内容导入当前程序。
7. **判断题**
8. 相比C++程序，Python程序的代码更加简洁、语法更加优美，但效率较低。（ ）
9. “from 模块名 import \*”语句与“import 模块名”都能导入指定模块的全部内容，相比之下，from…import \*导入的内容无须指定模块名，可直接调用，使用更加方便，因此更推荐在程序中此种方式导入指定模块的全部内容。（ ）
10. Python3.x版本完全兼容Python2.x。（ ）
11. PyCharm是Python的集成开发环境。（ ）
12. 模块文件的后缀名必定是.py。（ ）
13. **选择题**
14. 下列选项中，不是Python语言特点的是（ ）。

A.简洁 B.开源 C.面向过程 D.可移植

1. 下列哪个不是Python的应用领域（ ）。
2. Web开发
3. 科学计算
4. 游戏开发
5. 操作系统管理
6. 下列关于Python的说法中，错误的是（ ）。
7. Python是从ABC发展起来的
8. Python 是一门高级计算机语言
9. Python只能编写面向对象的程序
10. Python程序的效率比C程序的效率低
11. **简答题**
12. 简述Python的特点。
13. 简单介绍如何导入与使用模块。
14. 简述Python中模块、包和库的意义。
15. **编程题**

请在Python开发工具中输入并运行以下程序，查看程序运行结果。

1. 整数求和。输入整数n，计算1~n之和。

n = int(input("请输入一个整数："))

sum = 0

for i in range(n):

sum += n

print("1~%d的求和结果为%d"%(n,sum))

1. 整数排序。输入三个整数，把这三个数由小到大输出。

l = []

for i in range(3):

x = int(input('请输入整数：'))

l.append(x)

l.sort()

print(l)

1. 打印九九乘法表。

for i in range(1,10):

for j in range(1,i+1):

print("%d×%d=%-2d "%(j,i,i\*j),end = '')

print('')

1. 绘制多个起点相同单大小不同的五角星，如图1-29所示。

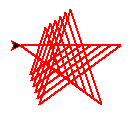


图1-29 重叠五角星

import turtle as t

def draw\_fiveStars(leng):

count = 1

while count <= 5:

t.forward(leng) #向前走50

t.right(144) #向右转144度

count += 1

leng += 10 #设置星星大小

if leng <= 100:

draw\_fiveStars(leng)

def main():

t.penup()

t.backward(100)

t.pendown()

t.pensize(2)

t.pencolor('red')

segment = 50

draw\_fiveStars(segment)

t.exitonclick()

if \_\_name\_\_ == '\_\_main\_\_':

main()

# Python基础

1. **填空题**
2. Python中建议使用\_\_\_\_\_\_个空格表示一级缩进。
3. 布尔类型的取值包括\_\_\_\_\_\_和\_\_\_\_\_\_。
4. 使用\_\_\_\_\_\_函数可查看数据的类型。
5. float()函数用于将数据转换为\_\_\_\_\_\_类型的数据。
6. 若a=3，b=-2，则a+=b的结果为\_\_\_\_\_\_。
7. **判断题**
8. Python中可以使用关键字作为变量名。（ ）
9. 变量名可以以数字开头。（ ）
10. Python标识符不区分大小写。（ ）
11. 布尔类型是特殊的浮点型。（ ）
12. 复数类型的实数部分可以为0。（ ）
13. **选择题**
14. Python中使用（ ）符号表示单行注释。
15. #
16. /
17. //
18. <!-- -->
19. 下列选项中，属于Python关键字的是（ ）。
20. name
21. if
22. is
23. and
24. 下列选项中，属于数值类型的是（ ）。
25. 0
26. 1.0
27. 1+2j
28. 以上全部
29. 若将2转换为0b10，应该使用（ ）函数。
30. oct()
31. bin()
32. hex()
33. int()
34. 下列选项中，不属于Python数据类型的是（ ）。
35. bool
36. dict
37. string
38. set
39. **简答题**
40. 请简单介绍Python中的数据类型和数字类型。
41. 请简述Python变量的命名规范。
42. 请简单介绍Python中的运算符。
43. **编程题**
44. 编写程序，要求程序能根据用户输入的数据计算圆的面积（圆的面积公式：），并分别输出圆的直径和面积。
45. 已知某煤场有29.5顿煤，先用一辆载重4顿的汽车运3次，剩下的用一辆载重为2.5顿的汽车运送，请计算还需要运送几次才能送完？编写程序，解答此问题。

# 流程控制

1. **填空题**
2. \_\_\_\_\_\_\_\_语句是最简单的条件语句。
3. Python中的循环语句有\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_循环。
4. 若循环条件的值变为\_\_\_\_\_\_\_\_，说明程序进入无限循环。
5. \_\_\_\_\_\_\_\_循环一般用于实现遍历循环。
6. \_\_\_\_\_\_\_\_语句可以跳出本次循环，执行下一次循环。
7. **判断题**
8. if-else语句可以处理多个分支条件。（ ）
9. if语句不支持嵌套使用。（ ）
10. elif可以单独使用。（ ）
11. break语句用于结束循环。（ ）
12. for循环只能遍历字符串。（ ）
13. **选择题**
14. 下列选项中，运行后会输出1、2、3的是（ ）。

for i in range(3):

print(i)

for i in range(2):

print(i + 1)

nums = [0, 1, 2]

for i in nums:

print(i + 1)

i = 1

while i < 3:

print(i)

i = i + 1

1. 现有如下代码：

sum = 0

for i in range(100):

if(i % 10):

continue

sum = sum + i

print(sum)

若运行代码，输出的结果为（ ）。

1. 5050
2. 4950
3. 450
4. 45
5. 已知x=10，y=20，z=30；以下代码执行后x、y、z的值分别为（ ）。

if x < y:

z = x

x = y

y = z

1. 10，20，30
2. 10，20，20
3. 20，10，10
4. 20，10，30
5. 已知x与y的关系如表3-1所示：

表3-1 x与y的关系

|  |  |
| --- | --- |
| **x** | **y** |
| x<0 | x−1 |
| x=0 | x |
| x>0 | x+1 |

以下选项中，可以正确地表达x与y之间关系的是（ ）。

y = x + 1

if x >= 0:

if x == 0:

y = x

else:

y = x - 1

y = x - 1

if x! = 0:

if x > 0:

y = x + 1

else:

y = x

if x <= 0:

if x < 0:

y = x - 1

else:

y = x

else:

y = x + 1

y = x

if x <= 0:

if x < 0:

y = x - 1

else:

y = x + 1

1. 下列语句中，可以跳出循环结构的是（ ）。
2. continue
3. break
4. if
5. while
6. **简答题**
7. 简述break和continue的区别。
8. 简述while和for语句的区别。
9. **编程题**
10. 编写程序，实现利用while循环输出100以内偶数的功能。
11. 编写程序，实现判断用户输入的数是正数还是负数的功能。
12. 编写程序，实现输出100以内质数的功能。

# 字符串

1. **填空题**
2. 定义字符串可使用 、双引号和三引号包裹。
3. 删除字符串中头部的空格，可以使用 方法。
4. 拼接字符串可以使用 方法和运算符 。
5. **判断题**
6. 字符串中不可以包含特殊字符。（ ）
7. 无论是使用单引号或双引号定义的字符串，使用print()输出的结果一致。（ ）
8. rjust()方法用于将字符串的字符以右对齐方式进行显示。（ ）
9. find()方法返回-1说明子串在指定的字符串中。（ ）
10. strip()方法默认会删除字符串头尾的空格。（ ）
11. 如果字符串中包含三引号，可以使用单引号包裹这个字符串。（ ）
12. **选择题**
13. Python中使用（ ）转义字符。
14. /
15. \
16. $
17. %
18. 下列选项中，用于格式化字符串的是（ ）。
19. %
20. format()
21. f-string
22. 以上全部
23. 关于字符串的说法中，下列描述错误的是（ ）。
24. 字符串创建后可以被修改
25. 字符串可以使用单引号、双引号和三引号定义
26. 转义字符\n表示换行
27. 格式符均由%和说明转换类型的字符组成
28. 下列方法中，可以将字符串中的字母全部转换为大写的是（ ）。
29. upper()
30. lower()
31. title()
32. capitalize()
33. 下列选项中，不属于字符串的是（ ）。
34. "1"
35. 'python'
36. """^"""
37. '1'.23
38. **简答题**
39. 请简述什么是字符串。
40. 请简述Python中格式化字符串的几种方式。
41. 请简述Python中字符串对齐的几种内置方法。
42. **编程题**
43. 编写程序，已知字符串s = 'AbcDeFGhIJ'，请计算该字符串中小写字母的数量。
44. 编写程序，检查字符串" Life is short. I use python"中是否包含字符串"python"，若包含则替换为"Python"后输出新字符串，否则输出原字符串。

# 组合数据类型

1. **填空题**
2. 使用内置的 函数可创建一个列表。
3. Python中列表的元素可通过 或 两种方式访问。
4. 使用内置的 函数可创建一个元组。
5. 字典元素由 和 组成。
6. 通过Python的内置方法 可以查看字典键的集合。
7. 调用items()方法可以查看字典中的所有 。
8. **判断题**
9. 列表只能存储同一类型的数据。（ ）
10. 元组支持增加、删除和修改元素的操作。（ ）
11. 列表的索引从1开始。（ ）
12. 字典中的键唯一。（ ）
13. 集合中的元素无序。（ ）
14. 字典中的元素可通过索引方式访问。（ ）
15. **选择题**
16. 下列方法中，可以对列表元素排序的是（ ）。
17. sort()
18. reverse()
19. max()
20. list()
21. 阅读下面的程序：

li\_one = [2, 1, 5, 6]

print(sorted(li\_one[:2]))

运行程序，输出结果是（ ）。

1. [1 ,2]
2. [2 ,1]
3. [1 ,2 ,5 ,6]
4. [6 ,5 ,2 ,1]
5. 下列方法中，默认删除列表最后一个元素的是（ ）。
6. del
7. remove()
8. pop()
9. extend()
10. 阅读下面程序：

lan\_info = {'01': 'Python', '02': 'Java', '03': 'PHP'}

lan\_info.update({'03': 'C++'})

print(lan\_info)

运行程序，输出结果是（ ）。

1. {'01': 'Python', '02': 'Java', '03': 'PHP'}
2. {'01': 'Python', '02': 'Java', '03': 'C++'}
3. {'03': 'C++','01': 'Python', '02': 'Java'}
4. {'01': 'Python', '02': 'Java'}
5. 阅读下面程序：

set\_01 = {'a', 'c', 'b', 'a'}

set\_01.add('d')

print(len(set\_01))

运行程序，输出结果是（ ）。

1. 5
2. 3
3. 4
4. 2
5. **简答题**
6. 列举Python中常用的组合数据类型，简单说明它们的异同。
7. 简单介绍删除字典元素的几种方式。
8. **编程题**
9. 已知列表li\_num1 = [4, 5, 2, 7]和li\_num2 = [3, 6]，请将这两个列表合并为一个列表，并将合并后的列表中的元素按降序排列。
10. 已知元组tu\_num1 = ('p', 'y', 't', ['o', 'n'])，请向元组的最后一个列表中添加新元素“h”。
11. 已知字符串str= 'skdaskerkjsalkj'，请统计该字符串中各字母出现的次数。
12. 已知列表li\_one = [1,2,1,2,3,5,4,3,5,7,4,7,8]，编写程序实现删除列表li\_one中重复数据的功能。

# 函数

1. **填空题**
2. \_\_\_\_\_\_\_\_是组织好的、实现单一功能或相关联功能的代码段。
3. 匿名函数是一类无须定义\_\_\_\_\_\_\_\_的函数。
4. 若函数内部调用了自身，则这个函数被称为\_\_\_\_\_\_\_\_。
5. Python使用\_\_\_\_\_\_\_\_关键字可以将局部变量声明为全局变量。
6. 全局变量是指在函数\_\_\_\_\_\_\_\_定义的变量。
7. **判断题**
8. 函数在定义完成后会立刻执行。（ ）
9. 变量在程序的任意位置都可以被访问。（ ）
10. 函数可以提高代码的复用性。（ ）
11. 任何函数内部都可以直接访问和修改全局变量。（ ）
12. 函数的位置参数有严格的位置关系。（ ）
13. **选择题**
14. 下列关于函数的说法中，描述错误的是（ ）。
15. 函数可以减少重复的代码，使得程序更加模块化
16. 不同的函数中可以使用相同名字的变量
17. 调用函数时，实参的传递顺序与形参的顺序可以不同
18. 匿名函数与使用关键字def定义的函数没有区别
19. Python使用（ ）关键字定义一个匿名函数。
20. function
21. func
22. def
23. lambda
24. Python使用（ ）关键字自定义一个函数。
25. function
26. func
27. def
28. lambda
29. 请阅读下面的代码：

num\_one = 12

def sum(num\_two):

global num\_one

num\_one = 90

return num\_one + num\_two

print(sum(10))

运行代码，输出结果为（ ）。

1. 102
2. 100
3. 22
4. 12
5. 请阅读下面的代码：

def many\_param(num\_one, num\_two, \*args):

print(args)

many\_param(11, 22, 33, 44, 55)

运行代码，输出结果为（ ）。

1. (11,22,33)
2. (22,33,44)
3. (33,44,55)
4. (11,22)
5. **简答题**
6. 简述位置参数、关键字参数、默认参数传递的区别。
7. 简述函数参数混合传递的规则。
8. 简述局部变量和全局变量的区别。
9. **编程题**
10. 编写函数，输出1~100中偶数之和。
11. 编写函数，计算20\*19\*18\*…\*3的结果。
12. 编写函数，判断用户输入的整数是否为回文数。回文数是一个正向和逆向都相同的整数，如123454321、9889。
13. 编写函数，判断用户输入的三个数字是否能构成三角形的三条边。
14. 编写函数，求两个正整数的最小公倍数。

# 文件与数据格式

1. **填空题**
2. 打开文件对文件进行读写后，应调用 方法关闭文件以释放资源。
3. seek()方法用于指定文件读写位置，该方法的 参数表示要偏移的字节数。
4. readlines()方法读取整个文件内容后会返回一个 。
5. os模块中的mkdir()函数用于 。
6. 在读写文件的过程中， 方法可以获取当前的读写位置。
7. **判断题**
8. 文件打开的默认方式是只读。（ ）
9. 以读写方式打开一个文件，若文件已存在，文件内容会被清空。（ ）
10. 使用write()方法写入文件时，数据会追加到文件的末尾。（ ）
11. 实际开发中，目录操作需要使用os模块中的函数。（ ）
12. read()方法只能一次性读取文件中的所有数据。（ ）
13. **选择题**
14. 打开一个已有文件，在文件末尾添加信息，正确的打开方式为（ ）。
15. r
16. w
17. a
18. w+
19. 假设文件不存在，如果使用open()方法打开文件会报错，那么该文件的打开方式是下列哪种？（ ）
20. r
21. w
22. a
23. w+
24. 假设file是文本文件对象，下列哪个选项可读取file的一行内容？（ ）
25. file.read()
26. file.read(200)
27. file.readline()
28. file.readlines()
29. 下列选项中，用于向文件中写入数据的是（ ）。
30. open()
31. write()
32. close()
33. read()
34. 下列选项中，用于获取当前目录的是（ ）。
35. open()
36. write()
37. getcwd()
38. read()
39. 下列代码要打开的文件应该在（ ）。

f = open('itheima.txt', 'w')

1. C盘根目录
2. D盘根目录
3. Python安装目录
4. 程序所在目录
5. 若文本文件abc.txt中的内容如下：

abcdef

阅读下面的程序：

file = open('abc.txt', 'r')

data = file.readline()

data\_list = list(data)

print(data\_list)

以上程序的执行结果为（ ）。

1. ['abcdef']
2. ['abcdef\n']
3. ['a', 'b', 'c', 'd', 'e', 'f']
4. ['a', 'b', 'c', 'd', 'e', 'f', '\n']
5. **简答题**
6. 请简述文本文件和二进制文件的区别。
7. 请简述读取文件3种方法read()、readline()、readlines()的区别。
8. **编程题**
9. 读取一个文件，打印除了以#开头的行之外的所有行。
10. 编写程序，实现文件备份功能。
11. 编写程序，读取存储若干数字的文件，对其中的数字进行排序后输出。

# 面向对象

1. **填空题**
2. Python中使用\_\_\_\_\_\_\_\_关键字来声明一个类。
3. 类的成员包括\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_。
4. Python可以通过在类成员名称之前添加\_\_\_\_\_\_\_\_的方式将公有成员改为私有成员。
5. 被继承的类称为\_\_\_\_\_\_\_\_，继承其他类的类称为\_\_\_\_\_\_\_\_。
6. 子类中使用\_\_\_\_\_\_\_\_函数可以调用父类的方法。
7. **判断题**
8. Python通过类可以创建对象，有且只有一个对象。（ ）
9. 实例方法可以由类和对象调用。（ ）
10. 子类能继承父类全部的属性和方法。（ ）
11. 创建类的对象时，系统会自动调用构造方法进行初始化。（ ）
12. 子类中不能重新实现从父类继承的方法。（ ）
13. **选择题**
14. 下列关于类的说法，错误的是（ ）。
15. 类中可以定义私有方法和属性
16. 类方法的第一个参数是cls
17. 实例方法的第一个参数是self
18. 类的实例无法访问类属性
19. 下列方法中，只能由对象调用的是（ ）。
20. 类方法
21. 实例方法
22. 静态方法
23. 析构方法
24. 下列方法中，负责初始化属性的是（ ）。
25. \_\_del\_\_()
26. \_\_init\_\_()
27. \_\_init()
28. \_\_add\_\_()
29. 下列选项中，不属于面向对象三大重要特性的是（ ）。
30. 抽象
31. 封装
32. 继承
33. 多态
34. 请阅读下面的代码：

class Test:

count = 21

def print\_num(self):

count = 20

self.count += 20

print(count)

test= Test()

test.print\_num()

运行代码，输出结果为（ ）。

1. 20
2. 40
3. 21
4. 41
5. **简答题**
6. 简述实例方法、类方法、静态方法的区别。
7. 简述构造方法和析构方法的特点。
8. 简述面向对象的三大特性。
9. **编程题**
10. 设计一个Circle（圆）类，该类中包括属性radius（半径），还包括\_\_init\_\_()、get\_perimeter()（求周长）和get\_area()（求面积）共三个方法。设计完成后，创建Circle类的对象求圆的周长和面积。
11. 设计一个Course（课程）类，该类中包括number（编号）、name（名称）、teacher（任课教师）、location（上课地点）共4个属性，其中location是私有属性；还包括\_\_init\_\_()、show\_info()（显示课程信息）共两个方法。设计完成后，创建Course类的对象显示课程的信息。

# 异常

1. **填空题**
2. Python中所有异常都是\_\_\_\_\_\_的子类。
3. 当程序中使用了一个未定义的变量时会引发\_\_\_\_\_\_异常。
4. 自定义异常需要继承\_\_\_\_\_\_类。
5. 若不满足assert语句中的表达式会引发\_\_\_\_\_\_异常。
6. **判断题**
7. try-except语句中只能有一个except子句。（ ）
8. finally子句在任何情况下都会被执行。（ ）
9. raise语句可以抛出指定的异常。（ ）
10. 断言语句中表达式的值为True时会触发AssertionError异常。（ ）
11. try-except语句可以有多个finally子句。（ ）
12. **选择题**
13. 下列选项中，关于异常描述错误的是（ ）。
14. 错误就是异常，异常就是错误
15. 异常是程序运行时产生的
16. IndexError是Exception的子类
17. except子句一定位于else和finally子句之前
18. 当try子句中的代码没有任何错误时，一定不会执行（ ）子句。
19. try
20. except
21. else
22. finally
23. 若执行代码1/0，会引发什么异常？（ ）
24. ZeroDivisionError
25. NameError
26. KeyError
27. IndexError
28. 在完整的异常捕获语句中，各子句的顺序为（ ）。
29. try→except→else→finally
30. try→else→except→finally
31. try→except→finally→else
32. try→else→finally→except
33. 下列代码运行后会引发（ ）异常。

num\_li = [1, 2, 3]

print(num\_li[3])

1. SyntaxError
2. IndexError
3. KeyError
4. NameError
5. **简答题**
6. 请简述什么是异常。
7. 请简述本章介绍的4种Exception类异常并说明其产生的原因。
8. 请写出raise语句抛出异常的3种格式，并简单介绍每种格式的功能。
9. **编程题**
10. 编写程序，计算圆的面积，若半径为负值则抛出异常（圆的面积公式：）。
11. 编写程序，输入三角形三条边判断能否构成直角三角形，若能构成则计算三角形的面积和周长，否则引发异常。

# Python计算生态与常用库

1. **填空题**
2. 是一种按照一定的规则，自动从网络上抓取信息的程序或者脚本。
3. 指用适当的统计分析方法对收集来的大量数据进行汇总与分析，以求最大化地发挥数据的作用。
4. Python计算生态通过 、 、 库为数据分析领域提供支持。
5. random是Python的 库，pygame是 库。
6. 通过Python的 函数可以初始化所有子模块。
7. **判断题**
8. Python开发人员可以使用内置库，也可以使用第三方库。（ ）
9. Python程序中使用内置库与第三方库的方式相同，但使用第三方库之前需要先将库导入程序。（ ）
10. 自定义库只能由自己在本地使用。（ ）
11. 时间差的计算没有意义。（ ）
12. jieba是一个中文分词库，但该库同时也可以对英文进行分词。（ ）
13. pygame库中的init()函数可以初始化所有子模块。（ ）
14. time模块是Python的内置模块，可以在程序中直接使用。（ ）
15. **选择题**
16. 下列选项中，用于判断文件是作为脚本执行还是被导入其他程序的是（ ）。
17. \_\_init\_\_
18. \_\_name\_\_
19. \_\_exce\_\_
20. \_\_main\_\_
21. 下列选项中，不会在发布自定义库时用到的是（ ）。
22. python setup.py build
23. python setup.py sdist
24. python setup.py install
25. 以上全部
26. 下列方法中，返回结果是时间戳的是（ ）。
27. time.sleep()
28. time.localtime()
29. time.strftime()
30. time.ctime()
31. 阅读下面的程序：

gmtime = time.gmtime()

time.asctime(gmtime)

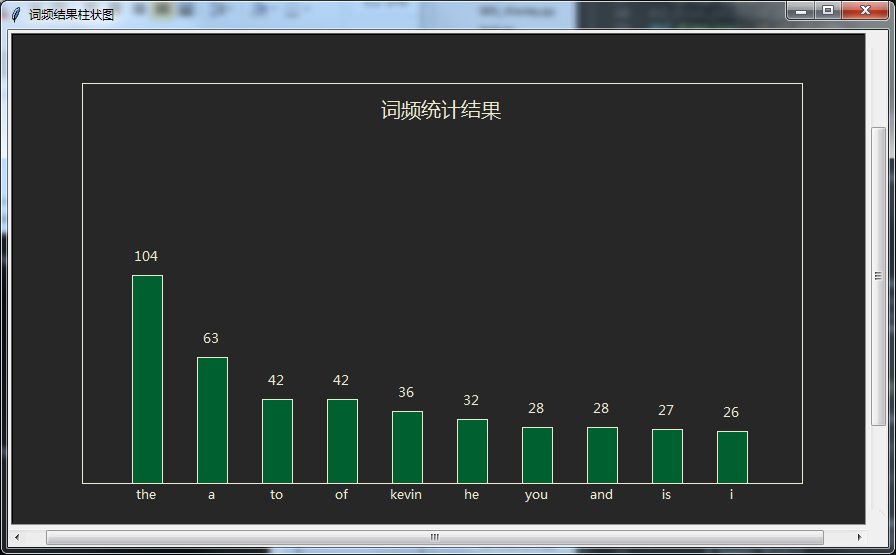
下列选项中，可能为以上程序输出结果的是（ ）。

1. 'Mon Apr 13 02:05:38 2020'
2. time.struct\_time(tm\_year=2020, tm\_mon=4, tm\_mday=11, tm\_hour=11, tm\_min=54, tm\_sec=42, tm\_wday=5, tm\_yday=102, tm\_isdst=-1)
3. 3173490635.1554217
4. '11:54:42'
5. 阅读下面程序：

random.randrange(1,10,2)

下列选项中，不可能为以上程序输出结果的是（ ）。

1. 1
2. 4
3. 7
4. 9
5. **简答题**
6. 简单列举Python计算生态覆盖的领域（至少5个）。
7. 简述介绍Python中库、包和模块的概念。
8. 若想对两个表示时间的变量进行计算，应将时间转换为什么格式？为什么？
9. **编程题**
10. 读取存储《哈姆雷特》英文剧本的文件，分析统计其中单词出现的频率，使用turtle模块绘制词频统计结果，以柱状图的形式展示统计结果。统计效果如图10-24所示。



1. 统计结果参考图示
2. 编写程序，实现一个根据可根据指定文本文件和图片文件生成不同形状词云的程序。