

第3章

练习题参考答案

3.1 填空题参考答案

1. 解释
2. 交互式、文件式
3. 面向对象
4. py
5. import
6. PyInstaller
7. dist
8. JOLIE
9. lie
10. `s[int(len(s)/2) ]`
11. 'Not Exists'
12. -1
13. IDLE、pip
14. `seek()`、`tell()`
15. F5 键
16. Package Installer for Python
17. 命令提示符 cmd、Scripts
18. `sys.executable`
19. 基本型、组合型
20. 5
21. 17
22. 42
23. 60
24. 0.5
25. '0b11'

26. '0o12'
27. '0xc'
28. round()
29. 列表、元组、集合、字典
30. 1.23
31. a.real、a.image
32. 5.0
33. 1.5
34. 'e'
35. 'dog'
36. str()
37. 5
38. '11'
39. 8
40. 'haha'
41. 'haha'
42. 'goodidea'
43. 3
44. [1, 2, 1, 2]
45. [1, 2, 1, 3, 1, 2]
46. (1, 2, 1, 3, 1, 2)
47. ['b', 'd', 'a']
48. ['a', 'd', 'b']
49. ['d', 'a']
50. ['d']
51. ['b', 'e']
52. ['g', 'o', 'o', 'd']
53. ['amazing', 'China']
54. 'just=for=fun'
55. 'C-C-T-V'
56. tu='a'
57. ('a',)
58. ('o', 'n', 'e')
59. 2
60. 0
61. 'qq.com'
62. {'n', 'a', 'b'}
63. 3

- 64. 1
- 65. 4
- 66. hash()
- 67. 字符串
- 68. 4.6
- 69. 5
- 70. Just-for-Fun
- 71. 1
- 72. 8
- 73. 12
- 74. -13
- 75. 72
- 76. 15
- 77. g-o-o-d-
- 78. [0, 1, 2, 3, 4]
- 79. object 类
- 80. "
- 81. False
- 82. '26'
- 83. False
- 84. '---demo---
- 85. 'c'
- 86. 'g-o-o-d'
- 87. 'a blue blue rose'
- 88. 'fun'
- 89. ['one', 'two', 'three']
- 90. ['a', ", ", 'ff']
- 91. ['1', '3']
- 92. WHERE 子句
- 93. background 或 bg、foreground 或 fg
- 94. [6, 7, 8, 9]
- 95. 6,10
- 96. [2, 4]
- 97. [(0, 'a'), (1, 'b')]
- 98. 3
- 99. 5
- 100. in
- 101. 低耦合、高内聚

102. 数据约束

103. 物理模型

3.2 单选题参考答案

1-4	D A D B	5-8	B C B A	9-12	A C D C	13-16	C D D D
17-20	D D D B	21-24	D D D C	25-28	D B A B	29-32	B D B B
33-36	C B D B	37-40	A A C C				

3.3 简答题参考答案

1. 对浮点数 x 进行四舍五入,保留 d 位小数。
2. 安装 module 模块: `pip install module`;
查看本机已安装的所有模块: `pip list`;
升级 module 模块: `pip install --upgrade module`。
3. Python Package Index。PyPI 是 Python 语言的软件库。
4. `pyinstaller -F demo.py`
5. 5、6
6. 两种,分别是单行字符串和多行字符串。
单行字符串用一对单引号(')或一对双引号(")作为边界;
多行字符串用一对三单引号("''')或一对三双引号('"""')作为边界。
7. 两种,单行注释和多行注释。
8. 以一个反斜杠“\”开头的字符序列。
退格**\b**,换行符**\n**,回车符**\r**,水平制表符**\t**。
9. 反斜杠“\”也是续行符,也就是将一行代码写成两行或两行以上,一般用于代码较长的行,以增加可读性。
10. UTF-8、UTF-16 和 UTF-32。
11. 只能使用 52 个大小写英文字母、0 ~ 9 十个阿拉伯数字和一个下画线“_”;
变量名不能以数字开头;
不能用 Python 解释器使用的关键字(Keyword)。
12. 两种,`list()`和**[]**。
13. 中括号运算符**[]**。
14. 三种,分别是 `pop()`、`del()`和 `remove()`。
15. `'b'`、`['a', 'c']`。
16. 元组是不可改变的(Immutable)。
17. 两种,`tuple()`和**()**。

18. 一种, `set()`。
19. 两种, `dict()`和`{}`。
20. `{'g', 'f'}`。
21. `['b', 'a', 'c']`。
22. `input()`函数。
23. $3/2=1.5$
24. (1) 3.5 (6) 1
(2) 3 (7) 5.0
(3) -4 (8) True
(4) (3, 1) (9) False
(5) 9 (10) False
25. 10001101、11110010、11110011。
26. 两个, 分别是 `is` 和 `is not`。
27. 两种, 分别是选择结构和循环结构。
28. 两种, 分别是 `for` 循环和 `while` 循环。
29. `for` 和 `while` 循环分别适用于循环次数已知和循环次数未知的情况。
30. 1 2 3 4。
31. 4
6
7
else 分支
32. 两种, 分别是局部变量和全局变量。
33. 两种, 分别由单星 `*` 操作符和双星 `**` 操作符定义。
34. `class`。
35. 两种, 分别是类方法和实例方法。
36. 两种, 分别是类属性和实例属性。
37. 多重继承(Multiple Inheritance)。
38. 三种, 分别是私有成员、保护成员和公共成员。
公共成员能够在类的外部访问;保护成员可以在该成员所在类及其子类中访问;私有成员只能在该成员所在类中使用。
39. 如果一个父类及其子类使用了完全相同的方法名, 但是却有不同的实现方式, 这种现象叫作多态(Polymorphism)。
40. `__new__()`构造方法, 创建实例时自动调用;
`__init__()`实例的初始化方法;
`__del__()`析构方法, 释放实例时自动调用。
41. 这两种方法几乎是等价的。唯一的区别: 如果找不到子串, `index()`方法抛出异常, 而 `find()`方法返回-1。
42. 字符串 `s` 的值保持不变, 即 `s='fun\t\n'`。

43. 点“.”、匹配行首“^”、匹配行尾“\$”。

44. [' ', 'a', 'b', 'c']。

注意：结果列表的第一个元素为空。

45. try-except 语句和 raise 语句。

46. finally 子句和 else 子句。

47. Base Exception 类。

48. 两种,分别是单元测试(Unit Test)和集成测试(Integration Test)。

49. 无论是否发生异常情况,都需要执行一些清理工作的场合,如关闭文件。

50. open('test.dat', 'wb+')。

51. 空字符串"。注意不是空格',也不是空行\n。

52. close()、read()、write()、tell()。

53. open()是 Python 语言的一个内置函数,用于打开文件。

54. r 为只读模式(默认)、w 为只写模式、a 为追加模式。

55. 使用 try...finally 异常处理结构或者使用 with 语句。

56. pickle 模块和 shelve 模块。

57. tempfile 模块。

58. os 模块、shutil 模块和 pathlib 模块。

```
59. import os
    print(os.getcwd())
```

60. SQL 是 Structured Query Language 的缩写,它是一种结构化查询语言。使用 SQL 可以实现与数据库的交互。

61. 一般流程包括五个步骤:

- (1) 创建 SQLite 数据库连接 connection;
- (2) 获取游标 cursor;
- (3) 执行相关操作;
- (4) 关闭游标 cursor;
- (5) 关闭数据库连接 connection。

```
62. import sqlite3
    from sqlite3 import Error
    try:
        connection=sqlite3.connect('test.sqlite')
    except Error as e:
        print(e)
    finally:
        connection.close()
```

```
63. create_users_table="""CREATE TABLE IF NOT EXISTS users(
    ID INTEGER PRIMARY KEY AUTOINCREMENT,
```

```
NAME TEXT NOT NULL,
GENDER TEXT NOT NULL,
AGE INTEGER NOT NULL
);"""
```

64. cursor()创建并返回一个游标对象、commit()提交当前事务、close()关闭连接。

65. execute()执行一条 SQL 语句、fetchone()获取查询结果集的下一行、fetchall()获取查询结果集的所有行、close()关闭游标对象。

66. 主键 PRIMARY KEY;外键 FOREIGN KEY。

67. DB Browser (SQLite)。

68. 连接操作。

69. Update 和 Delete。

70. SELECT * FROM users

71. 四个步骤:

(1) 加载 tkinter 模块;

(2) 创建主窗口;

(3) 在应用程序中添加一个或多个组件,如按钮 Button;

(4) 进入主事件循环。

72. import tkinter 或 from tkinter import *。

73. Button 按钮组件、Canvas 画布组件、Checkbutton 复选按钮、Label 标签组件、Text 文本组件。

74. borderwidth 或 bd、height、width。

75. 两种,分别为

(1) 可以使用字符串,指定十六进制数字中 RGB 的比例。如"#fff"表示白色、"#000000"表示黑色、"#000fff000"表示绿色。

(2) 也可以使用任何本地定义的标准颜色名称,如白色"white"。

76. 两种,分别为

(1) 作为字体属性使用。

首先生成一个字体实例:

```
ft=tkFont.Font(family="隶书", size=14, weight=tkFont.BOLD, slant=tkFont.ITALIC)
```

然后将其作为字体属性使用:

```
label=Label(root, text="字体设置", font=ft)
```

在上述代码中,字体实例 ft 被作为标签组件的字体属性 font 使用。

(2) 以字体三元组的方式使用。

其实第一种使用方式,完全可以用一行代码代替:

```
label=Label(root, text="字体设置", font=("隶书", 14, "bold italic"))
```

上述代码中的 `font=("隶书", 14, "bold italic")` 就是一个字体三元组。此处 `font` 是一个元组, 由三个元素构成。

77. `anchor`; 用于定义文字放置位置的参照点。

78. `relief` 属性。常用的边框样式有 `FLAT`、`SUNKEN`、`RAISED`。

79. 三种, 分别是 `pack` 类、`grid` 类和 `place` 类。

80. 三个属性, `side` 属性、`expand` 属性和 `fill` 属性。

81. `sum=3`。

82. 2。

```
83. one:two:three
    one-two-three
```

`sep` 是可选参数/关键字参数。

84. 4

24

36

85. 10

5

6

86. 2。

87. 3。

88. `else` 分支

89. `__name__` 变量是 Python 的内置变量。每个 Python 模块都有一个 `__name__` 变量。如果程序被作为模块加载, 则其 `__name__` 变量的值被自动设置为模块名; 如果程序是独立运行的, 则其 `__name__` 属性值被自动设置为 `__main__`。利用 `__name__` 变量可控制 Python 程序的运行方式。

90. 两种功能, 分别是位运算和集合的交集运算。

举例如下。

```
>>> 0b101 & 0b011
1
>>> 0b101 & 0b110
4
>>> st1={1, 5, 3}
>>> st2={3, 2, 4}
>>> st1 & st2
{3}
```

91. (1) 二叉树如图 3-1 所示。

(2) 后序序列: ABCDEFGH。

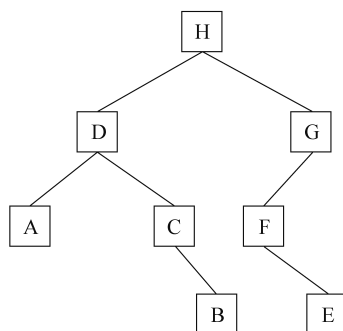


图 3-1 二叉树

3.4 编程题参考答案

```
1. a, b=b, a

2. user_name=input("输入您的姓名:")
   print("Hello,", user_name)

3. num1=int(input("输入第一个数:"))
   num2=int(input("输入第二个数:"))
   print(num1+ num2)

4. num1=int(input("输入第一个数:"))
   num2=int(input("输入第二个数:"))
   if num1>num2:
       print("第一个数大:", num1)
   else:
       print("第二个数大:", num2)

5. n1=int(input("输入第一个数:"))
   n2=int(input("输入第二个数:"))
   n3=int(input("输入第三个数:"))
   if n1 >=n2 and n1 >=n3:
       print("第一个数最大:", n1)
   elif n2 >=n1 and n2 >=n3:
       print("第二个数最大:", n2)
   else:
       print("第三个数最大:", n3)
```

或者

```
n1=int(input("输入第一个数:"))
n2=int(input("输入第二个数:"))
n3=int(input("输入第三个数:"))
print(max(n1, n2, n3))
```

```
6. for num in range(2, 100, 2):
    print(num)
```

```
7. for num in range(50, 0, -1):
    if num%2==1:
        print(num)
```

```
8. import math
```

```
def is_prime(num):
```

```

        divisor=int(math.sqrt(num))+1

    for i in range(2, divisor):
        if num%i==0:
            return False
    return True

for num in range(2, 100):
    if is_prime(num):
        print(num)

9. def mul_or_sum(num1, num2):          #mul 代表 multiplication
    product=num1 * num2
    if product <=1000:
        return product
    else:
        return num1+num2
num1=30
num2=20
result=mul_or_sum(num1, num2)
print("The result is:", result)

```

一个更简洁的答案如下:

```

a, b=int(input("输入第一个整数:")), int(input("输入第二个整数:"))
prod=a * b
print(prod if prod<1000 else a+b)

10. user_str='hello'
    for i in range(0, len(user_str), 2):
        print(user_str[i])

```

一个更简洁的答案如下:

```

user_str=input("输入一个字符串:")
print(user_str[0::2])

11. def remove_chars(str, n):
    return str[n+1:]
print(remove_chars('hello', 2))

12. def first_last_is_same(number_list):
    first_element=number_list[0]
    last_element=number_list[-1]
    if first_element==last_element:
        return True
    else:
        return False

```

```

result=first_last_is_same([10, 20, 30, 40, 10])
print("result is:", result)

13. def divided_by_five(num):
    if num%5==0:
        return True
    else:
        return False
number_list=[10, 33, 25, 55, 46]
for num in number_list:
    if divided_by_five(num):
        print(num)

14. sample_str="中华民族是一个伟大的民族。"
count=sample_str.count('族')
print(count)

15. for num in range(1, 6):
    for i in range(num):
        print(num, end=' ')
    print()

16. number=input("输入一个整数:")
if number==number[::-1]:
    print("是回文数")
else:
    print("不是回文数")

17. def merge_list(list1, list2):
    list3=[]
    for num in list1:
        if num%2 !=0:
            list3.append(num)
    for num in list2:
        if num%2==0:
            list3.append(num)
    return list3
list_one=[5, 20, 23, 10]
list_two=[12, 15, 7]
print("第一个列表:", list_one)
print("第二个列表:", list_two)
print("新列表:", merge_list(list_one, list_two))

18. def power(base, exp):
    result=base**exp
    return result

```

```

base=int(input("输入底数:"))
exp=int(input("输入指数:"))
print("结果等于:", power(base, exp))

```

```

19. for i in range(5):
    for j in range(i+1):
        print("* ", end="")
    print()

```

一个更简洁的答案如下:

```

for i in range(1, 6):
    print(" * " * i)

```

```

20. print('大', '国', '重', '器', sep=' * ')

```

```

21. print("%o"%8)

```

```

22. import math
    print("%.2f"%math.pi)

```

```

23. float_numbers=[]
    for i in range(5):
        print("输入下标为", i, "的浮点数:")
        num=float(input())
        float_numbers.append(num)

```

```

print(float_numbers)

```

```

24. with open("test.txt") as fin:
    lines=fin.readlines()

    count=0
    with open("newfile.txt", 'w') as fout:
        for line in lines:
            count+=1
            if count!=3:
                fout.write(line)
            print(line, end='')

```

```

25. num=1
    while num <=10:
        print(num)
        num+=1

```

```

26. for row in range(1, 6):
    for column in range(1, row+1):
        print(column, end=' ')
    print()

```

```

27. num=int(input("输入一个正整数:"))
    sum=0
    for i in range(1, num+1):
        sum+=i
    print("The sum is:", sum)

28. list1=[11, 15, 28, 42, 55, 100]
    for num in list1:
        if num>50:
            break
        if num%2==0:
            print(num)

29. list1=[1, 2, 3, 4, 5]
    start=len(list1)-1
    stop=-1
    for i in range(start, stop, -1):
        print(list1[i])

30. def demo(name, age):
        print(name, age)

    demo('Mike', 20)

31. def func1(*args):
        for arg in args:
            print(arg, end=' ')

    func1(5, 3, 2)
    func1('a', 'nice', 'day')

32. def calc(a, b):
        return a+b, a-b

    print(calc(5, 3))

33. lt=[4, 5, 20, 7]
    print(max(lt))

34. print(list(range(4, 12, 2)))

35. def f(name, age):
        print(name, age)

    f('Kate', 21)
    g=f
    g('Kate', 21)

```

```

36. def calc_sum(num):
    if num:
        return num+calc_sum(num-1)
    else:
        return 0

    result=calc_sum(10)
    print(result)

37. import string

    str1="/ * 科技 @强 & 国!!"
    for ch in str1:
        if ch==' ' or ch in string.punctuation:
            str1=str1.replace(ch, '#')
    print(str1)

38. str1="GluOcODk"
    lower=[]
    upper=[]
    for ch in str1:
        if ch.islower():
            lower.append(ch)
        else:
            upper.append(ch)
    str2=''.join(upper+lower)
    print(str2)

39. str1="HTTPS://www.163.com"
    lower, upper, digit, other=0, 0, 0, 0
    for ch in str1:
        if ch.islower():
            lower+=1
        elif ch.isupper():
            upper+=1
        elif ch.isdigit():
            digit+=1
        else:
            other+=1
    print("Lower chars=", lower)
    print("Upper chars=", upper)
    print("Digit chars=", digit)
    print("Other chars=", other)

40. str1="god"
    print(str1[::-1])

```

```

41. str1="Welcome to China. china is awesome, isn't it?"
    count=str1.lower().count("china")
    print("The China count is:", count)

42. import re
    str1="Chinese=90 English=80 Math=78 Art=70"
    score_list=[int(num) for num in re.findall(r'\b\d+\b', str1)]
    print("The sum is:", sum(score_list))
    print("The average is:", sum(score_list)/len(score_list))

43. str_list=["I", "am", "", "a", None, "teacher", ""]
    new_str_list=list(filter(None, str_list))
    print(new_str_list)

44. str1="apple"
    count_dict=dict()
    for ch in str1:
        count=str1.count(ch)
        count_dict[ch]=count
    print(count_dict)

45. for row in range(2):
        for col in range(3):
            print("{}{}{}".format(row, col), end=' ')
        print()

46. {i * 2 for i in range(5)}

47. lt1=['a', 'b', 'm', 'n']
    lt2=[4, 3, 2, 1]
    dict1={k:v for k, v in zip(lt1, lt2)}

48. def fib(n):
        '''Print a Fibonacci series up to n.'''
        a, b=0, 1
        while a<n:
            print(a, end=' ')
            a, b=b, a+b
        print()
    fib(20)                                     #输出 20 以内的斐波那契数列

49. f=lambda x, y: x+y
    print(f(2, 1))

50. def f(x):
        if x<=1:
            return x * x
        else:

```

```

        return 2 * x + 1

    if __name__ == '__main__':
        print(f(1))

51. class Rational:
    """ 定义一个 Rational 有理数类
    属性:分子 x, 分母 y,
    """
    def __init__(self, x=0, y=1):
        self.x=x
        self.y=y
    def __str__(self):
        return "%d/%d"%(self.x, self.y)
    __repr__ = __str__
    def __mul__(self, other):
        return Rational(self.x*other.x, self.y*other.y)
    if __name__ == '__main__':
        r1=Rational(1, 2)
        r2=Rational(1, 3)
        print(r1 * r2)

52. import re
    s='good123luck'
    word_list=re.findall('[a-zA-Z]+', s)
    for word in word_list:
        print(word)

53. with open('test.txt') as fp:
        fp.read()

54. with open('test.txt') as fp:
        for line in fp:
            print(line)

55. import os
    try:
        os.makedirs('2020/8/9')
    except Exception as ex:
        print(ex)

56. import tkinter

    root=tkinter.Tk()
    root.mainloop()

57. from tkinter import Tk, Label

```

#加载 tkinter 模块中的 Tk 和 Label 类

```

root=Tk()
root.title("欢迎界面")
lb=Label(root, text="Hello, World!")

lb.pack()
root.mainloop()

58. lt1=[4, -1, 5, -8, 0.2]
    lt2=list(filter(lambda x: x<0, lt1))
    print(lt2)

59. lt1=[0, 5, 10, 15]
    lt2=list(map(lambda x: x+5, lt1))
    print(lt2)

60. lt1=[1, 2, 3]
    lt2=['Monday', 'Tuesday', 'Wednesday']
    lt3=list(zip(lt1, lt2))
    print(lt3)

61. import random

    lt=[]
    count=0
    while True:
        num=random.randint(10, 20)
        if num in lt:
            continue
        lt.append(num)
        count+=1
        if count >=5:
            break
    print(lt)

```

#创建主窗体 root

#设置窗口的标题

#调用 Label() 方法创建标签 lb, root 是 lb 的父容器

#调用布局管理器 pack() 方法

#进入主循环, 显示主窗体 root

或者

```

import random

lt=list(range(10, 21))
lt=random.sample(lt, 5)
print(lt)

62. import turtle

```

```

t=turtle.Turtle()
t.pensize(2)
t.pencolor("blue")

```

```

t.circle(100)
t.penup()
t.forward(100)
t.pendown()
t.circle(100)
t.ht()

```

63. import turtle

```

t=turtle.Turtle()
t.penup()
t.goto(-20, 40)
t.pendown()
t.pensize(10)
t.pencolor("pink")
t.forward(100)
t.backward(100)
t.right(90)
t.forward(100)
t.left(90)
t.forward(100)
t.backward(100)
t.right(90)
t.forward(100)
t.left(90)
t.forward(100)
t.ht()
turtle.done()

```

64. from bs4 import BeautifulSoup

```

html_doc="""
<html><head><title>睡鼠的故事</title></head>
<body>
<p class="title"><b>睡鼠的故事</b></p>

<p class="story">从前有三个小妹妹,她们的名字是
<a href="http://example.com/elsie" class="sister" id="link1">埃尔西</a>,
<a href="http://example.com/lacie" class="sister" id="link2">拉西</a>和
<a href="http://example.com/tillie" class="sister" id="link3">蒂莉</a>;
她们住在一口井的底部。</p>

<p class="story">...</p>
</body>
</html>

```

```

"""
soup=BeautifulSoup(html_doc, 'html.parser')
title=soup.find('title')
print("网页的标题:", title.text)
link_list=soup.find_all('a', class_='sister')
for link in link_list:
    print("锚文本:", link.text)
    print("  对应的链接:", link['href'])

```

上述代码的执行结果,如图 3-2 所示。

```

网页的标题: 睡鼠的故事
锚文本: 埃尔西
  对应的链接: http://example.com/elsie
锚文本: 拉西
  对应的链接: http://example.com/lacie
锚文本: 蒂莉
  对应的链接: http://example.com/tillie

```

图 3-2 网页信息提取

```

65. import string
    import random

    s=string.ascii_letters+string.digits
    print("".join(random.sample(s, 10)))

66. import random

    lst=[]
    for i in range(10):
        lst.append(random.randint(1, 20))
    print(lst)
    st=set(lst)
    for i in st:
        print(i, '=>', lst.count(i))
    或者
    import random

    lst=[random.randint(1, 20) for i in range(10)]
    print(lst)
    dt=dict()
    for i in lst:
        dt[i]=dt.get(i, 0)+1
    for k, v in dt.items():
        print(k, '=>', v)

67. def calc_y(x):
    if x<0:

```

```

        return 0
    elif x<5:
        return x
    elif x<10:
        return 3 * x-4
    else:
        return 0.5 * x-3

if __name__=='__main__':
    x=input('请输入一个实数:')
    x=eval(x)
    print(calc_y(x))

68. import turtle as t                #导入 turtle 库并起别名 t

    t.pensize(2)                      #设置画笔的粗细为 2 像素
    t.pu()                            #提起画笔
    t.goto(-50, -50)                  #将海龟移动到绝对坐标(-50, -50)
    t.pd()                            #放下画笔
    for i in range(3):
        t.fd(100)                    #将海龟向前移动 100 像素
        t.left(120)                  #将海龟向左旋转 120°
    t.ht()                            #使海龟不可见

69. import turtle as t                #导入 turtle 库并起名为 t

    t.pensize(2)                      #设置画笔的粗细为 2 像素
    t.setup(500, 400)                 #设置绘图窗口的长和宽分别为 500 和 400 像素
    t.pu()                            #提起画笔
    t.goto(-50, -50)                  #将海龟移动到绝对坐标(-50, -50)
    t.pd()                            #放下画笔
    for i in range(3):
        t.fd(100)                    #将海龟向前移动 100 像素
        t.lt(90)                     #将海龟向左旋转 90°
    t.fd(100)                         #将海龟向前移动 100 像素
    t.hideturtle()                   #使海龟不可见

70. f=eval(input("请输入一个华氏温度:"))
    c=5/9 * (f-32)
    print("摄氏温度为:%.2f"%c)

71. def score_transform(num):
    if num >= 90:
        return 'A'
    elif num >= 80:
        return 'B'

```