

第 5 章 函数和模块

【学习目标】

通过本章的学习,应达到如下学习目标:

1. 阐释函数的概念,包括函数的定义与调用、函数的返回值、参数传递、变量作用域、递归函数。
2. 掌握函数的定义与调用、返回值、参数传递和变量作用域。重点掌握如何按照需求定义函数,如何构建函数的参数,如何调用函数并返回值。
3. 根据要求编写有关函数的案例,能说明递归函数的含义。
4. 掌握模块的定义与导入。
5. 了解模块的内建属性和内建函数,模块的搜索路径。

【单元导学】

第 5 章思维导图如图 5-1 所示。

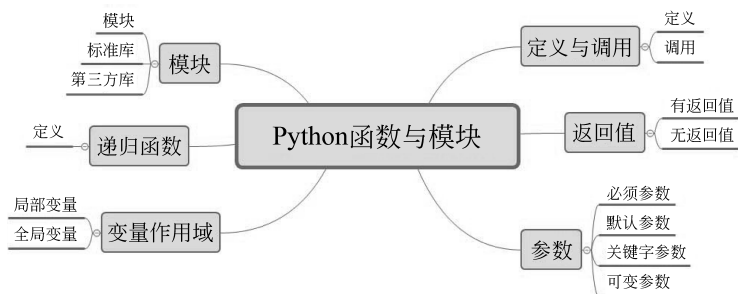


图 5-1 第 5 章思维导图

本章重点主要包括以下部分。

函数定义部分: 函数定义的概念、函数定义的格式、函数的返回值、函数的参数。

函数调用部分: 函数调用的格式。

模块定义部分: 模块定义的概念、模块定义的格式。

模块导入部分: 模块导入的格式。

本章难点主要包括: 递归函数、模块导入。

【知识回顾】

第 4 章重点介绍了 Python 的数据结构。复习第 4 章的内容,编写代码,从键盘输入任意多个数字,按 Enter 键退出,结束输入,输出存放在列表 ls 中的数,并统计输出最大的数和最小的数,以及最大的数和最小的数的平均值。

代码：

```
ls=[]
iNumStr=input("请输入数字(直接输入回车退出): ")
while iNumStr!="":
    ls.append(eval(iNumStr))
    iNumStr=input("请输入数字(直接输入回车退出): ")
max1=min1=ls[0]
for num in ls:
    print(num,end=" ")
    if num>max1:
        max1=num
    elif num<min1:
        min1=num
aver=(max1+min1)/2
print()
print("最大的数: {:d}, 最小的数: {:d}, 最大的数和最小的数的平均值: {:.2f}".format(
    max1,min1,aver))
```

【学前准备】

为了更好地完成本章的学习,请做到以下两点:

1. 查阅函数的具体定义,并能举例说明。
2. 查阅函数调用与参数传递的过程,并能举例说明。

实验 5-1 内建函数 ceil() 的使用

【实验目的】

1. 熟悉内建函数 ceil() 的含义。
2. 掌握内建函数 ceil() 的使用方法。

【实验内容】

利用数学函数实现从键盘输入一个数,求不小于该数的最小整数,运行结果如图 5-2 所示。

输入一个数:15.7 不小于15.7的最小整数是16

图 5-2 ceil() 函数的运行结果

【实验步骤】

根据要求编写函数如下:

```
import math
n=float(input("输入一个数: "))
```

```
m=math.ceil(n)
print("不小于",n,"的最小整数是",m,sep='')
```

程序使用了 math 模块中的数学函数,math 模块中的 ceil()函数返回参数的上入整数。
ceil()函数的语法:

```
import math
math.ceil(x)
```

注意: ceil()是不能直接访问的,需要导入 math 模块,通过静态对象调用该方法。

参数: x—数值表达式。

返回值: 函数返回数字的上入整数。

实例:

```
import math #导入 math 模块
print("math.ceil(-12.34): ", math.ceil(-12.34))
print("math.ceil(100.78): ", math.ceil(100.78))
print("math.ceil(1000.25): ", math.ceil(1000.25))
print("math.ceil(math.pi): ", math.ceil(math.pi))
```

输出结果为:

```
math.ceil(-12.34): -12
math.ceil(100.78): 101
math.ceil(1000.25): 1001
math.ceil(math.pi): 4
```

实验 5-2 内建函数 hypot()的使用

【实验目的】

1. 掌握内建函数 hypot()的含义。
2. 掌握内建函数 hypot()的使用方法。

【实验内容】

输入三角形的两个直角边,输出对应斜边的长度,运行结果如图 5-3 所示。

输入第一条直角边的长度: 3 输入第二条直角边的长度: 4 三角形直角边 3.0 和直角边 4.0 对应的斜边长度是: 5.0

图 5-3 hypot()函数的运行结果

【实验步骤】

```
import math
a=float(input("输入第一条直角边的长度: "))
b=float(input("输入第二条直角边的长度: "))
```

```
c=math.hypot (a,b)
print("三角形直角边",a,"和直角边",b,"对应的斜边长度是:",c)
```

程序使用了 math 模块中的数学函数,math 模块中的 hypot() 返回欧几里得范数 $\sqrt{x^2 + y^2}$ 。

hypot() 函数的语法:

```
import math
math.hypot (x,y)
```

注意: hypot () 是不能直接访问的,需要导入 math 模块,通过静态对象调用该方法。

参数: x--数值表达式;y-- 数值表达式。

返回值: 函数返回欧几里得范数 $\sqrt{x^2 + y^2}$ 。

实例:

```
import math
print ("hypot(5, 8): ", math.hypot(5, 8))
print ("hypot(-6, 6): ", math.hypot(-6, 6))
print ("hypot(0, 5): ", math.hypot(0, 5))
```

输出结果为:

```
hypot(5, 8): 9.433981132056603
hypot(-6, 6): 8.485281374238571
hypot(0, 5): 5.0
```

实验 5-3 内建时间函数 time() 的使用

【实验目的】

1. 掌握内建时间函数 time() 的含义。
2. 掌握内建时间函数 time() 的使用方法。

【实验内容】

编程实现将当前时间转换为格林尼治时间,并同时输出当前时间和格林尼治时间,运行结果如图 5-4 所示。

```
Sat Apr 24 12:43:58 2021
time.struct_time(tm_year=2021, tm_mon=4, tm_mday=24, tm_hour=4, tm_min=43,
tm_sec=58, tm_wday=5, tm_yday=114, tm_isdst=0)
```

图 5-4 time() 函数的运行结果

【实验步骤】

```
import time
t=time.asctime(time.localtime(time.time()))
```

```
print(t)
gt=time.gmtime(time.time())
print(gt)
```

程序使用了 time 模块中的时间函数,Python 提供了一个 time 和 calendar 模块,可用于格式化日期和时间。

Python 的 time 模块下有很多函数可以转换常见的日期格式。

函数 time.time()用于获取当前时间,但是 1970 年之前的日期无法以此表示。

函数 time.asctime()接收时间元组并返回一个可读形式为"Tue Mar 31 21:19:29 2021"(2021 年 3 月 31 日 周三 21 时 19 分 29 秒)的 24 个字符的字符串。

asctime() 函数的语法:

```
time.asctime([t])
```

参数:元组或者通过函数 gmtime()或 localtime()返回的时间值。

返回值:返回一个可读形式为"Tue Mar 31 21:19:29 2021"(2021 年 3 月 31 日 周三 21 时 19 分 29 秒)的 24 个字符的字符串。

函数 time.gmtime()将一个时间转换为 UTC 时区(0 时区)的 struct_time,参数 sec 的默认值为 time.time(),函数返回 time.struct_time 类型的对象。

实例:

```
import time
t=time.localtime()
print(time.asctime(t))
```

输出结果为:

```
Tue Mar 31 21:19:29 2021
```

实验 5-4 通过自定义函数判断素数

【实验目的】

1. 熟悉自定义函数。
2. 掌握通过自定义函数判断素数的方法。

【实验内容】

编写一个判断素数的函数,实现输入一个整数,使用判断素数的函数进行判断,然后输出是否为素数,运行结果如图 5-5 所示。

输入一个数:11 11 是素数	输入一个数:58 58 不是素数
--------------------	---------------------

图 5-5 判断素数的运行结果

【实验步骤】

根据要求编写函数：

```
def isprime(n):
    i=2
    while i<=n-1:
        if n%i==0:
            break
        i=i+1
    return i
x=int(input("输入一个数:"))
y=isprime(x)
if y>x-1:
    print(x,"是素数")
else:
    print(x,"不是素数")
```

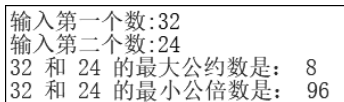
实验 5-5 通过自定义函数计算最大公约数和最小公倍数

【实验目的】

1. 掌握自定义最大公约数函数和最小公倍数函数。
2. 理解自定义函数调用过程中参数的传递。

【实验内容】

编写两个函数，分别计算两个整数的最大公约数和最小公倍数，运行结果如图 5-6 所示。



```
输入第一个数:32
输入第二个数:24
32 和 24 的最大公约数是: 8
32 和 24 的最小公倍数是: 96
```

图 5-6 计算两个整数的最大公约数和最小公倍数

【实验步骤】

根据要求编写函数：

```
def GY(m,n):
    r=m%n
    while r!=0:
        m=n
        n=r
        r=m%n
```

```

        return n
def GB(m,n):
    P=m*n/GY(m,n)
    return P
x=int(input("输入第一个数:"))
y=int(input("输入第二个数:"))
if y>x:
    t=x
    x=y
    y=t
z=GY(x,y)
U=GB(x,y)
print(x,"和",y,"的最大公约数是:",z)
print(x,"和",y,"的最小公倍数是:",U)

```

实验 5-6 通过自定义函数计算阶乘

【实验目的】

1. 掌握通过自定义函数计算阶乘的方法。
2. 掌握调用函数的方法。

【实验内容】

编写一个函数,计算输出 n 的阶乘值。

【实验步骤】

根据要求编写函数:

```

def jiecheng(x):
    y=1
    for i in range(1,x+1):
        y=y*i
    return y
n=int(input("输入一个整数:"))
s=jiecheng(n)
print(n,"的阶乘是",s)

```

实验 5-7 函数的参数传递

【实验目的】

1. 掌握函数参数中形参和实参的含义。
2. 体会函数的参数传递过程。

3. 准确编写包含参数传递的程序。

【实验内容】

1. 定义一个有 3 个形参的函数,在函数体内实现 3 个形参相加。

【实验步骤】

根据要求编写函数:

```
def add(a,b,c):  
    d=a+b+c  
    return d  
x=int(input("input x:"))  
y=int(input("input y:"))  
z=int(input("input z:"))  
s=add(x,y,z)  
print(s)
```

【实验内容】

2. 定义一个有 3 个形参值为 100,200,300 的函数,在函数体内实现 3 个形参相加。

【实验步骤】

根据要求编写函数:

```
def add(a=100,b=200,c=300):  
    d=a+b+c  
    return d  
s=add()  
print(s)
```

上面两种方式定义的形参个数都是固定的,例如定义函数的时候如果定义了 n 个形参,那么在调用时最多只能传递 n 个实参。

实验 5-8 递归函数

【实验目的】

1. 了解递归函数的含义。
2. 掌握递归函数的使用方法。

【实验内容】

递归解决年龄问题:有 5 个人坐在一起,问第五个人多少岁?他说比第四个人大两岁。问第四个人多少岁?他说比第三个人大两岁。问第三个人多少岁?他说比第二个人大两岁。问第二个人多少岁?他说比第一个人大两岁。最后问第一个人多少岁? he 说是十五

岁。编写程序,输入第几个人时输出其对应的年龄。

【实验步骤】

根据要求编写函数:

```
def age(x):
    if x==1:
        y=15
    else:
        y=age(x-1)+2
    return y
n=int(input("输入人数: "))
s=age(n)
print("输入第",n,"个人对应的年龄是",s)
```

递归就是在调用一个函数的过程中又直接或间接地调用该函数本身。本例子中,age()函数被递归调用。递归过程分为两个阶段:第一阶段是“回推”,由题意可知,想知道第五个人的年龄,必须知道第四个人的年龄,想知道第四个人的年龄,必须知道第三个人的年龄……直到第一个人的年龄,这时 age(1) 的年龄已知,就不用再推;第二阶段是“递推”,从第二个人推出第三个人的年龄……一直推到第五个人的年龄为止。需要注意,必须有一个结束递归过程的条件,本例中,当 $x=1$ 时, $y=15$ (也就是 $\text{age}(1)=15$ 时) 结束递归过程,否则递归会无限进行下去。

实验 5-9 输出杨辉三角形

【实验目的】

1. 了解杨辉三角形的含义。
2. 掌握用自定义函数输出杨辉三角形的方法。

【实验内容】

编写函数,输入一个整数 t 作为参数,打印杨辉三角形的前 t 行,运行结果如图 5-7 所示。

```
输入打印杨辉三角形的行数: 10
[1]
[1, 1]
[1, 2, 1]
[1, 3, 3, 1]
[1, 4, 6, 4, 1]
[1, 5, 10, 10, 5, 1]
[1, 6, 15, 20, 15, 6, 1]
[1, 7, 21, 35, 35, 21, 7, 1]
[1, 8, 28, 56, 70, 56, 28, 8, 1]
[1, 9, 36, 84, 126, 126, 84, 36, 9, 1]
```

图 5-7 杨辉三角形

【实验步骤】

根据要求编写函数：

```
def yanghui(t):
    print([1])
    line=[1,1]
    print(line)
    #从第三行开始打印其他行
    for i in range(2,t):
        r=[]
        #按照杨辉三角形规律生成每行除两端之外的数字
        for j in range(0,len(line)-1):
            r.append(line[j]+line[j+1])
        #把两端的数字连在一起
        line=[1]+r+[1]
        print(line)
n=int(input("输入打印杨辉三角形的行数："))
yanghui(n)
```

杨辉三角形是二项式系数在三角形中的一种几何排列,特点是：最左侧的一列数字和右边的斜边都是 1,内部其他位置上的每个数字都是上一行同一列数字与前一列数字的和。

实验 5-10 通过模块计算列表中偶数的和

【实验目的】

1. 掌握模块的含义。
2. 掌握导入模块的使用方法。

【实验内容】

通过模块计算一个给定列表中偶数的和。

【实验步骤】

根据要求建立文件 osh.py,编写下列代码：

```
def func_sum(L1):
    s=0
    for i in range(0,len(L1)):
        if L1[i]%2==0:
            s=s+L1[i]
    return s
```

新建文件 exp10.py,编写下列代码：