第二章 Python 语言程序设计

1、Python 语言的数据类型与表达式

数据类型		描述	示例
数字类型	整形（int）	表示数学中的整数，包括正整数、零和负整数，没有取值范围的限制。	-2,0,100 等
	浮点型(float)	浮点型与数学中实数的概念一致，表示带小数点的数字，没有取值范围的限制。	如-2.50,3.14 等
字符串型（str）		字符串是由一对单引号、一对双引号或一对三引号括起来的字符序列。 字符可以是字母、数字以及能在键盘上输入的符号，可以用来存放姓名、电话号码、地址等。	如：“姓名”、“联系电话”、“School”等
布尔型（bool）		表示逻辑的真和假，布尔型只有两种值：True 和 False，注意首字符要大写，可以用来表示条件成立与否。	如“10＞5”的值为 True，*"9 < 8"*的值为 False。

2、强制类型转换函数

数据类型转换函数	功能说明
float(*x*)	将*x*转换为一个浮点型数据
int(*x*)	将*x*转换为一个整型数据
str(*x*)	将*x*转换为字符串型数据

type( )返回被检测的数据类型

len() 返回序列长度

3、Python 的运算符

（1）Python 的算术运算符

运算符	表达式	描述	示例(x=5, y=2)
+	$x + y$	加，x 与 y 的和	*5 + 2* 结果是 7
-	$x - y$	减，x 减 y 的差，也可表示负数	5-2 结果是 3
*	x*y	乘，x 与 y 的积	*5 * 2* 结果是 10
/	x/y	除，x 除以 y 的商，结果为浮点数	5 / 2 结果是 2.5
//	$x//y$	整除，x 除以 y 的整数商	5 / / 2 结果是 2
**	x**y	幂运算，x 的 y 次幂	5＊＊2 结果是 25
%	x%y	取余，x 除以 y 得到的余数	5%2 结果是 1

（2）Python 的关系运算符

运算符	表达式	描述	示例
>	$x > y$	x 大于 y	*5 > 2* 结果是 True
<	$x < y$	x 小于 y	*5 < 2* 结果是 False
>=	$x >= y$	x 大于等于 y	*5 >= 2* 结果是 True
<=	$x <= y$	x 小于等于 y	*5 <= 2* 结果是 False
==	$x == y$	x 等于 y	*5 == 2* 结果是 False
! =	$x! = y$	x 不等于 y	*5! = 2* 结果是 True
in	$x\ in\ y$	x 是 y 的成员	“5”in“2”结果是 False

（3）Python 的逻辑运算符

运算符	表达式	描述	示例
and	x and y	布尔“与”	True and False 结果是 False
or	x or y	布尔“或”	True or False 结果是 True
not	x not y	布尔“非”	not False 结果是 True

4、运算优先级

括号 > 指数 > 算术运算符 > 关系运算符 > 赋值 > 逻辑运算符

5、常量、变量和赋值语句

（1）常量：在程序运行过程中其值始终不发生变化。

（2）变量：在程序运行过程中其值发生变化的量。

（3）变量命名规则:

(1)变量名变量的标识符可以由字母（(A-Z, a-z)、数字（0-9）和下划线_组成。

(2)变量名不能以数字开头。

(3)变量名严格区分大小写。

(4)不能使用系统保留字作为变量名

and	as	assert	break	class	continue
def	del	elif	else	except	finally
for	from	False	global	if	import
in	is	lambda	nonlocal	not	None
or	pass	raise	return	try	True
while	with	yield			

- 由于 Python 是动态类型语言，因此在使用前不需要预先声明变量的数据类型。

6、输入，输出语句

Python 语言主要用函数 input( ) 实现数据输入，用函数 print( )实现数据输出。

■ 输入函数 input( )

主要用来接收键盘的输入，返回值为字符串型数据。通常，在输入时可以给出提示信息，例如： x = input(″请输入一个正整数：″)。

■ 输出函数 print( )

主要用于在屏幕上输出一个或多个输出项的值，多个输出项中间用逗号隔开

例如：print(x, ″是奇数″)。

7、注释

注释语句是对程序代码的解释和说明

- 单行注释用#
- 多行注释用英文的三对单引号或三对双引号

```
#这是单行注释
"""
这是多行注释
这是多行注释
这是多行注释
"""
'''
这也是多行注释
这也是多行注释
这也是多行注释
'''
```

8.Python 编程语言中的四种复合数据类型:

- 列表(List)
- 元组(Tuple)
- 集合(Set)
- 字典(Dictionary)

例如:

Python 三大控制结构

1.顺序结构：程序按照编写顺序依次被执行（自上而下依次执行）

```
a=2
b=4
c=a+b
print(c)
```

2.选择结构：根据条件语句的结果选择执行不同的语句

常用选择语句有 if、if……else、if…elif…else

```
age = 10
if age >= 18:
    print("你已经成年了")
    print("即将步入大学生活")

print("时间过的真快呀")
```

（1）单分支结构

```
#双分支结构
age=int(input("请输入你的年龄："))
if age>=18:
    print("欢迎进入网吧！")
else:
    print("未成年禁止进入网吧！")
```

（2）双分支结构

（3）多分支结构

```
    #成绩等级判断
    s=int(input("请输入你的分数:"))
age = 11
year = 1
level = 1
if age >= 18:
    print("你是成年人")
    if age < 30:
        print("你的年龄达标了")
        if year > 2:
            print("恭喜你，年龄和入职时间都达标，可以领取礼物")
        elif level > 3:
            print("恭喜你，年龄和级别达标，可以领取礼物")
        else:
            print("不好意思，尽管年龄达标，但是入职时间和级别都不达标。")

    else:
        print("不好意思，年龄太大了")
else:
    print("不好意思，小朋友不可以领取。")
```

if 语句的嵌套

循环结构:for、while

1.for 循环:

格式：for 循环变量 in range(start,end,step):

循环体(一组被重复执行的语句)

- start：用于指定计数的起始值，可以省略，如果省略则从 0 开始
- end:用于指定计数的结束值(但不包括该值,如 range(7)得到的值为 0~6 不包括 7),不能省略。当 range()函数中只有一个参数时，即表示指定计数的结束值。
- step:用于指定步长，即两个数之间的间隔，可以省略，如果省略则表示长为 1。例如，range(1,4)将得到 1、2、3

注意:

- 在使用 range()函数时，如果只有一个参数，那么表示指定的是 end;
- 如果是两个参数，则表示指定的是 stant 和 end;
- 只有三个参数都存在时，最后一个才表示步长。

例如：

```
#for循环输出1-10的数
for i in range(1,11):
    print(i)
```

2.while 循环

格式：while 条件表达式:

循环体(一组被重复执行的语句)

例如：

```
#while循环输出1-100的数
i=1                  #把i的值初始化为1
while i<=100:   #循环100次
    print(i)
    i=i+1
```

跳转语句 break:完全终止循环

跳转语句 continue:直接跳转到下一次循环

自定义函数

实现某一功能的代码，定义为一个函数，在需要使用时，随时调用即可，简单理解就是可以完成某项工作的代码块，类似于积木块，可以反复使用。即通过将一段有规律、重复的代码定义为函数，来达到一次编写多次调用的目的。

使用函数可以提高代码的重复利用率。

1、创建函数：

def 函数名(可选参数)：

    函数体

2、调用函数：

函数名(参数)，例如 input()

3、参数传递：

函数定义时参数列表中的参数是形参，而函数调用时传递进来的参数是实参

4、返回值：

return

注意:当函数中没有 return 语句时，或者省略了 retum 语句的参数时，将返回 None

5、变量的作用域：

局部变量、全局变量

例如：

例1，使用函数来输出“hello world”

```
def hello():
    print("hello world!")
hello()
```

例2，使用函数来比较两个数的大小，并返回最大的数

```
def max(a,b):
    if a>b:
        return a
    else:
        return b
a=4
b=5
print(max(a,b))
```

常用库的使用

1、导入库：

import 库 as 重命名

import 包 from 库

from 库 import 包

语句	语句作用
import numpy as np	引入 numpy 库模块，用 np 替代
import matplotlib. pyplot as plt	引入 matplotlib 库模块中的 pyplot 方法，用 plt 替代
from pylab import *	引入 matplotlib 库模块中的 pyplot 方法，用 plt 替代

2、常用库

语句	作用
numpy	构建科学计算最基础的软件库，为 n 维数组和矩阵的操作提供有用功能
matplotlib	2D 绘图库，可生成绘图、直方图、功率谱、条形图、散点图等
math	求数学公式或相关函数
pandas	用于快速简单的数据操作、聚合和可视化呈现。库中有两个主要的数据结构，一维数组（series）和二维数组（dataframe）结构。
image	主要的 PIL 库，可以完成对图像的常用操作，如获取图像尺寸和像素颜色、旋转图像或改变图像格式等。
random	产生随机数
OS	实现部分操作系统功能（文件、目录等操作）
xlrd	excel 数据读取
xlwt	excel 数据写人
time	时间处理

pygame	多媒体及游戏开发
tkinter	图形处理

（1）math 库 该库提供了基础数学函数的访问

math.fabs（x）返回 x 的绝对值

math.factorial（x）以一个整数返回 x 的阶乘

math.gcd（＊integers）返回给定的整数参数的最大公约数

math.lcm（＊integers）返回给定的整数参数的最小公倍数

math.pow(x,y)将返回 x 的 y 次幂

math.sqrt（x）返回 x 的平方根

math.pi 数学常数π=3.141592...，精确到可用精度

（2）random 库 该模块实现了各种分布的伪随机数生成器

random.randrange(stop)

random.randrange(start,stop[,step])从 range(start, stop, step）返回一个随机选择的元素

random.randint（a，b）返回随机整数 N 满足 a<=N<=b

random．random（）返回［0. 0, 1. 0）范围内的下一个随机浮点数

（3）turtle 库 绘图

forward（）｜fd（）前进

backward（） ｜ bk（） I back（）后退

right（）｜rt（）右转

left（）｜lt（）左转

goto（）｜ setpos（）｜setposition（）前往 / 定位

setx（）设置 x 坐标

sety（）设置 y 坐标

setheading（）丨seth（）设置朝向

home（）返回原点

circle（）画圆

dot（）画点

stamp（）印章

clearstamp（）清除印章

clearstamps（）清除多个印章

undo（）撤消

speed（）速度

pendown（）lpd（）丨down（）画笔落下

penup（）丨pu（）up（）画笔抬起

pensize（）丨width（）画笔粗细

color（）颜色

pencolor（）画笔颜色

fillcolor（）填充颜色

filling（）是否填充

begin__fill（）开始填充

endfill（）结束填充

reset（）重置

clear（）清空

write（）书写

3、常用第三方库：numpy、pandas、matplotlib

（1）第三方库的安装

pip install 库

（2）numpy 库 开源的数值计算扩展库，可用来存储和处理大型矩阵

（3）pandas 库 为解决数据分析任务而创建的库

（4）matplotlib 库 用于数据可视化的库

scatter（）绘制散点图

plot（）绘制拟合曲线图

show（）显示图

解析法与问题解决

基本思想：

用 Python 编程通过解析式求解数学函数

在解析算法的程序实现过程中，首先要确保数学表达式的正确性，然后在程序中正确描述该数学表达式。

任务：判断一个数是否是3和7的公倍数

（1）抽象与建模

◆找到核心要素

假设要判断的数为x
判断条件:3和7的公倍数

◆得出计算模型

3和7的公倍数
x%3==0 and x%7==0

x是公倍数 或
x不是公倍数

（2）设计算法

◆输入数据

输入X

◆处理数据

x%3==0 and x%7==0

◆输出数据

输出x是公倍数或x不是公倍数

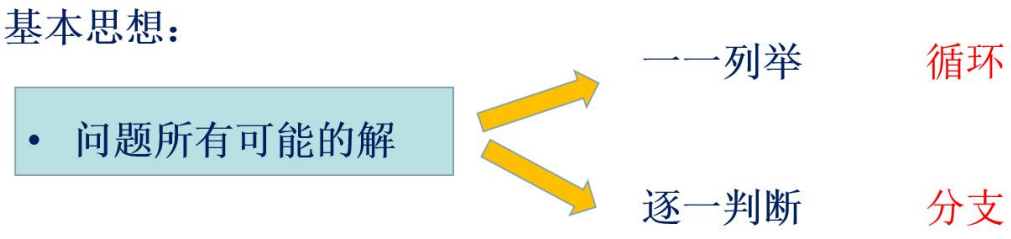
（3）编写程序

```
x=int(input())
if x%3==0 and x%7==0:
    print(x,"是公倍数")
else:
    print(x, "不是公倍数")
```

```
21
21 是公倍数
```

枚举法与问题解决

枚举算法思想是把所有可能解一一列举，然后判断每一个列举出的可能解是否为正确解。



在枚举算法中，逐一列举出每一个可能解，判断其是否为正确解的过程采用循环结构来实现；而在利用问题提供的约束条件判断正确解的过程中，则需要用到分支结构。

在设计枚举算法时，不能遗漏任何一个正确解，又要尽可能地缩小列举范围，以提高算法的工作效率。

用 Python 编程通过枚举法求解数学函数

File Edit Format Run Options Window Help

```
x=1
y=1
z=1
for x in range(1,11):
    for y in range(1,9):
        for z in range(1,7):
            if 4*x+5*y+6*z==50:
                print(x,y,z)
```

＃枚举法求解 3 元 1 次方程