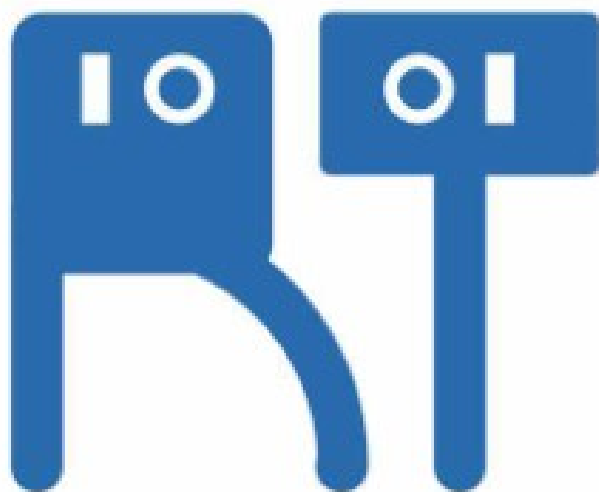


Python 语言速查手册



RealThink 上海睿兮教育科技有限公司

二零一九年三月

目 录

1 变量	3
2 运算与赋值	3
2.1 运算符	3
2.2 赋值符号	3
2.3 布尔运算	3
3 开始写程序	4
3.1 基本规则:	4
3.2 print()	4
3.3 input()	4
3.4 更多内置函数	4
4 if...elif...else...条件判断	5
5 for 循环	5
6 while 循环	5
7 list 列表	6
7.1 列表的概念	6
8.2 列表的方法:	6
8.3 多维列表:	6
9 function 函数	7
10 类与对象	8
11 异常处理	8
12 导入模块	9
12 Python 与 Pygame 的下载与安装	10
13 Python 初级班代码示例	15
13.1 蒙特卡洛算法求 π 值	15
13.2 RGB 绘点	15
13.3 验证哥德巴赫猜想	16
13.4 斐波那契数列	16
13.5 二分法迭代猜数字	16

13.6 冒泡排序	17
13.7 字符串与像素图	17
13.8 星座演示程序	18
13.9 计算 24 点	20
13.10 随机微信红包 ‘1 分钱’ 算法	21
13.11 随机微信红包 ‘官方’ 算法	21
13.12 太空大逃亡基本版代码	22

1 变量

语法：

变量名必须以下划线或字母开头，而后面接任意数目的字母、数字或下划线。

变量名大小写敏感（Abc 和 abc 和 aBC 都是不同的）。

保留字不能作为变量名（什么是保留字？）。

例如：a1, A1, ab_gogogo_3, a2_kiss, _ab5, _3, _3a3 都是合法的变量名，1a, a#, \$a 等都是不合法的变量。

变量类型：

整数 int、浮点数 float、布尔值 bool、字符串 str、空值 None。

布尔值：True or False, 等同于 1 or 0

字符串：如 'abc' , "oiewhfouwdwod" , 双引号与单引号等效

空值：None, 空值

type() 函数可以获得数据类型

2 运算与赋值

2.1 运算符

+ - * / 加减乘除：与数学完全一样
// 整除：3/2=1.5, 3//2=1
% 求余数：3%2=1, 10%4=2
** 乘方：2**3=8, 3**2=9, 9**0.5=3

2.2 赋值符号

= 赋值符号，左侧必须为一个变量名，右侧必须是已知的，可以为变量，常量，函数或表达式(其实前三者也属于表达式)。含义为将右侧表达式的值赋给左侧的变量。

2.3 布尔运算

and 与，运算符两侧逻辑值同时为真

True and True -> True, True and False -> False, False and False -> False

or 或，运算符两侧逻辑值至少有一个为真

True or True -> True, True or False -> True, False or False -> False

not 非，运算符后的逻辑值取反

not True -> False, not False -> True

== 判断两侧的值是否相等

>= 大于等于

`<=` 小于等于

`!=` 不等于

3 开始写程序

3.1 基本规则：

程序代码顺序执行

保留字和函数、变量，均大小写敏感

Python 是强制缩进语言，每条语句前面的空格缩进是语法的一部分

3.2 `print()`

将括号中的内容打印输出。

为什么 `print` 后有 `()`？因为它实际是一个函数。

输入需要在屏幕打印的内容和格式参数，输出就是在屏幕显示。

`print` 支持转义输出，转义符 `'\'`

`\n` 回车

`\t` 下一个制表符位置

`\'` 单引号

`\"` 双引号

3.3 `input()`

获得用户键盘输入语句。（实际也是一个函数）

输入从键盘获得，输出是一个字符串

语法：

`input(字符串)`

获取输入前会先在屏幕打印括号内的字符串。

3.4 更多内置函数

从 `print()`，`input()`，`eval()` 开始初步体会函数。

`eval()`：输入一个字符串，输出它表示的数字

`str()`：输入一个数字，输出对应的字符串

`int()`：输入一个数字，输出它的取整值

`len()`：输入一个列表，输出列表的元素数量

4 if...elif...else...条件判断

语法:

if 表达式: (表达式为真, 运行:后面的语句体, 为假则跳过)

...

...

elif 表达式: (如果上面的 if 为假, 可以继续判断其他条件, 可选)

...

...

else: (如果上面的条件都不成立, 执行后面的)

...

...

表达式运算符: 'and', 'or', 'not', '>', '<', '==' (是否等于), '!=' (是否不等于)

注意:

' : ' 冒号后面的一组代码处理一流程, 称为一个代码块, 一定要统一缩进, 建议用 4 个空格, Python 自带编辑器的 'tab' 键就是 4 个空格。

5 for 循环

for 循环就是格式循环: 从 n1 到 n2 循环, 步长为 step, 都必须是整数:

语法:

for 变量名 in range(n1,n2,step) n1 和 step 可没有, 缺省从 0 开始, 步长

1

for 变量名 in list 遍历 list 中的元素

break 跳出本循环

continue 放弃本次循环, 执行循环变量下一个值

说明: for i in range(0,5): i 会是 0,1,2,3,4(没有 5)注意从 i1 到 i2 时, 不会达到 '==n2'

for i in range(5): 效果同上, 步长缺省是 1, 起始值缺省是 0

for i in range(0,5,2): n 会是 0,2,4

for i in range(1,2,4): n 会是 1

for i in range(1,2,4): n 会是 1

for i in range(5,0,-1): n 会是 5,4,3,2,1

6 while 循环

While 循环就是条件循环: 满足表达式条件的情况下反复执行语句体。

语法:

while 表达式:

...

```
...
```

如果表达式成立，就一直循环，否则退出。

同样支持 break 与 continue 操作

实际程序中经常使用

```
while True:
    if 条件满足:
        break
...
...
```

这样的无限循环，当条件满足时循环结束。

7 list 列表

7.1 列表的概念

list 列表是一个有序的数据集合，可以按照下标（从 0 开始）来访问、删除、增加和修改指定数据。使用 ' [' , '] ' 来表示元素及下标。

列表必须显式的申请内存，即确定列表包含的元素个数，比如：a=[1,3,5,7]，这样的语句，在赋值之前自动申请内存。运行完毕后：a[0]=1,a[1]=3,a[2]=5,a[3]=7；

这时如果执行 a[1]=5, 那么列表中的元素就变为 [1,5,5,7]，即 a[1]=5；如果企图给没有申请内存的位置赋值，会报错，比如：a[5]=10，会报错。

申请列表内存：

```
>>> a=[0]*5
[0, 0, 0, 0, 0]
```

8.2 列表的方法：

list.append(x)	末尾申请内存并增加元素 x
list.insert(i,x)	在第 i 的位置插入元素 x，并申请内存，后续元素后移
list.pop(i)	删除第 i 位置的元素，后续元素前移并回收内存，如果没有参数 i，就删除最后一个
list.remove(x)	根据 x 的值删除元素
list.index(x)	返回第一个与 x 值相等的元素的下标
del list[i]	删除第 i 个元素，等效于 list.pop(i)，只是 del 属于语句不属于 list 方法

‘引用’概念：列表变量实际是一个‘引用’，指向存放数组数据的开始的地方。因此列表之间赋值，必须对其中元素直接赋值，否则只是将数组指针赋给另外一个数组变量。

8.3 多维列表：

二维列表就是面（一页表格），三维列表就是体（多页表格），多维数组请继续理解。遍历多维数组的元素可以用多重循环，因此这两个概念有对应关系。

多维列表申请空间：

```
>>> a=[[0]*5 for i in range(10)]    #[0]*5,构成了1个有5个元素的1维数组,后面的for循环,继续
```

>>> a 将这个 1 维数组以 0 为初值申请了连续 10 个，构成 2 维数组

$$[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, \\ 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]$$

>>> a[3][4]=5 #访问2维数组时,第1个[]里的索引值是外重循环([3]表示第4个),

>>> a 第2个[]里的索引值是内重循环 ([4]是第5个)。

$$[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 5], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]$$

这里可以清楚的看到 2 维列表中哪个元素被修改

或者可以写成：

```
>>> a=[[0,0,0,0,0] for i in range(10)] #与上面完全等效
```

但是，下面这种写法会有问题：

```
>>> a=[0,0,0,0,0]*5
```

 $\ggg a$ [illegible]

或者，这样写：

$$a = [[0] * 5] * 10$$
 $\ggg a$
$$[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, \\ 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]$$

>>> a[3][4]=5 #如果1个列表被'*n'，那么其实是重复了多个该列表的引用！这样的2维
中，在其他维度的同一位置，都指向了同一个元素，如果修改，全局都被改！

 $\ggg a$
$$[[0, 0, 0, 0, 5], [0, 0, 0, 0, 5], [0, 0, 0, 0, 5], [0, 0, 0, 0, 5], [0, 0, 0, 0, 5], [0, \\ 0, 0, 0, 5], [0, 0, 0, 0, 5], [0, 0, 0, 0, 5], [0, 0, 0, 0, 5], [0, 0, 0, 0, 5]]$$

9 function 函数

函数是一个重要的抽象概念，是编程的基础。数学中的函数，是给定输入，通过表达式，有输出。编程中的函数，含义及应用更为广泛。

语法:

def 函数名(形参表):

函数内容代码块，包括返回值

下面这个函数可以判断一个输入数字是否质数：

'def' 是保留字，意思是开始定义函数，'is_prime_number' 是函数名，'n' 是形参，后面介绍何为实参

```
def is_prime_number(n):
```

```
if (n<2) or (n!=int(n)):    #合法性检查: 如果 n 比 2 小或 n 是小数, 直接返回 False
    return False
```

```
if n==2: # 2本身就是质数, 不用验证
```

```

return True

```

```
for i in range(2,n-1):    #循环判断从 2 到 n-1 是否有能被 n 整除的
```

```
if n%i==0: #如果有能被 n 整除的, 说明不是质数
```



```

        return False
    return True        #没有能被 n 整除的，说明是质数

```

语法：

global：声明变量为全局变量。

10 类与对象

Python 是一种面向对象语言，实际上其函数、变量、模块，本身均为对象。Python 提供自定义类的方法。

语法：

```

class classname(object):                类名声明，从 object 基类继承
    def __init__(self,p1,p2...):        __init__() 类生成器
        self.p1=p1
        self.p2=p2
    ...
    def func1(...):                    类方法说明
        ...

```

classname 类名

类可以继承，源头都是 object

```
def __init__(self,p1,p2...)
```

类生成器，注意 self 是必须的

```
def func1(self,p1,p2...)
```

类方法，注意 self 也是必须的

使用示例：

```

class MyClass(object):
    def __init__(a):
        self.a=a
        self.b=0
    def do_this()
        print(self.a+self.b)

```

```
my_class= MyClass(5)
```

调用类生成器，获得实例 my_class

```
my_class.b=3
```

访问类属性 b

```
my_class.do_this()
```

调用类方法 do_this()

11 异常处理

语法：

```
try:
```

可能发生异常的代码块

```
except 异常名 (可省略，表示所有异常都执行这里):
```

代码块

12 导入模块

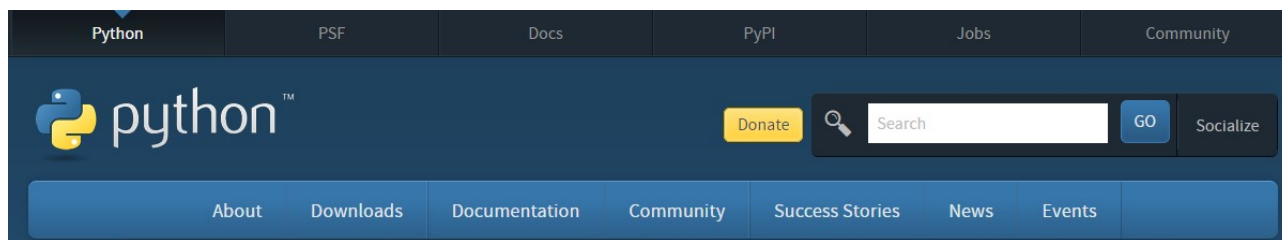
Python 有许多内置模块和几乎无穷多第三方模块，可以为我们带来很多新的功能和支撑。

<code>import pygame</code>	#引入 <code>pygame</code> 模块，使用其资源要以 <code>'pygame.'</code> 开头
<code>import random</code>	#引入 <code>random</code> 模块，使用其资源要以 <code>'random.'</code> 开头
<code>random.random()</code>	#返回一个随机数，是 0 到 1 之间的小数
<code>random.randint(n1,n2)</code>	#返回一个 <code>n1</code> 到 <code>n2</code> （含 <code>n1</code> 和 <code>n2</code> ）之间的整数随机数
<code>import sys</code>	#引入系统模块
<code>sys.exit()</code>	#程序结束退出
<code>import math</code>	#数学模块
<code>math.pi</code>	#返回常数 π 值
<code>math.sqrt()</code>	#开平方
<code>import time</code>	#时间管理模块
<code>time.sleep(n)</code>	#系统等待 <code>n</code> 秒
<code>time.clock()</code>	#第一次是初始化，第二次调用开始，返回的是距离第一次的时

间

12 Python 与 Pygame 的下载与安装

首先去 www.python.org 官网下载 python 安装包。



选择“Downloads”，进入下载页面，就下载最新版本，比如现在我看到的是 V3.7.2:

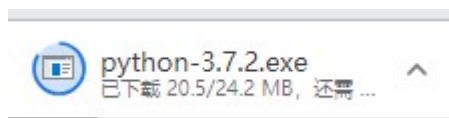


这里点击大的黄色按钮就可以直接下周 windows 版。

当然也可以自己选择下载版本：

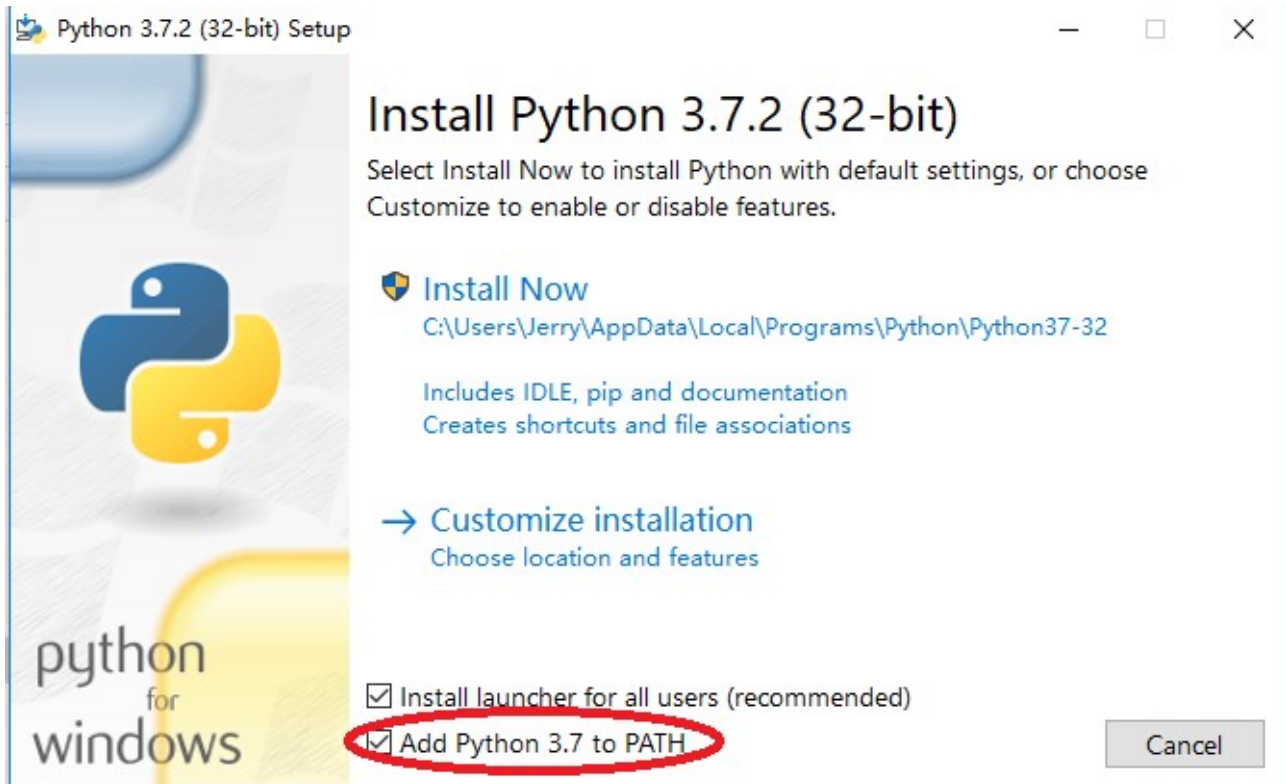
Files		
Version	Operating System	Description
Gzipped source tarball	Source release	
XZ compressed source tarball	Source release	
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later
Windows help file	Windows	
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64
Windows x86 embeddable zip file	Windows	
Windows x86 executable installer	Windows	
Windows x86 web-based installer	Windows	

(mac请下载macOS 64-bit/32-bit installer版本，Windows如确认是64位，下载Windows x86-64版本)
耐心等待下载：

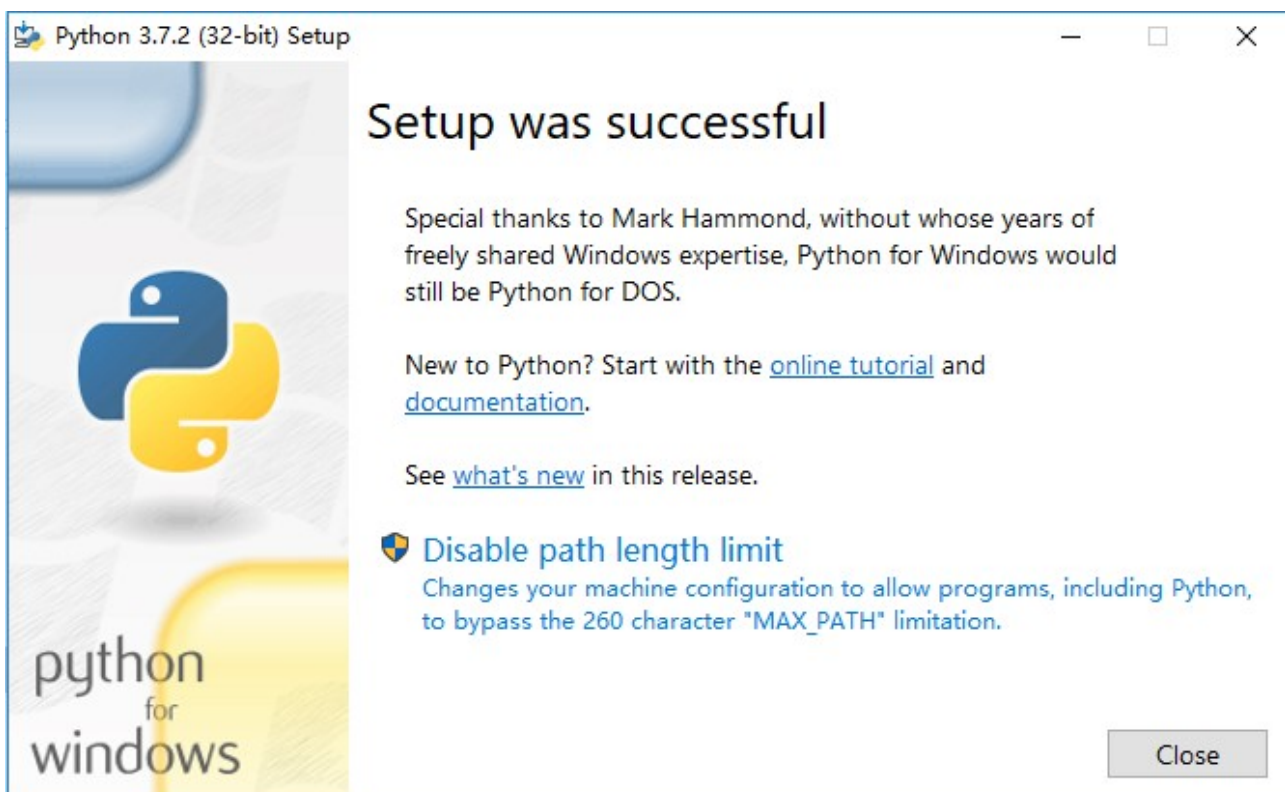


下载完毕，开始安装。

请注意第一步，一定要勾选“Add Python to PATH”，然后“Install Now”一路默认安装即可：



成功安装完毕，close 安装窗口：



下一步安装 pygame。

请打开 Windows Shell（屏幕左下角点击鼠标右键可以出现菜单）：



或者 cmd 命令提示符：



然后输入命令：pip install pygame，系统会自动下载 pygame 模块并安装，请耐心等待安装完毕。如果网络速度较慢，可能这一步会失败，那就再输入一遍命令直到成功：

```
PS C:\Users\Jerry> pip install pygame
```

这样，python+pygame 安装完毕。

如果是 mac 系统，可能稍有不同（比如最后一步的命令要在 terminal 中输入，具体请自行解决）。

13 Python 初级班代码示例

13.1 蒙特卡洛算法求 π 值

#蒙特卡洛算法求 π 值

```
import random

n = eval(input('Number: '))
s = 0
for i in range(n):
    x = random.random()
    y = random.random()
    if x*x + y*y < 1:
        s += 1
print('  $\pi$  =', s/n*4)
```

13.2 RGB 绘点

#Pygame 框架和 RGB 绘点

```
import pygame, sys

pygame.init()
screen = pygame.display.set_mode((256, 256))

#####
for y in range(256):
    for x in range(256):
        #只改下面这一句 set_at() 的第 2 个参数，通过循环变量改变 RGB
        screen.set_at((x,y), (x,y,x))
        pygame.display.update()
#####

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    pygame.display.update()
```


13.3 验证哥德巴赫猜想

#验证哥德巴赫猜想

#函数 is_prime_number(n), 判断是否质数

```
def is_prime_number(n):
    if n < 2 or n != int(n):
        return False
    for i in range(2, n):
        if n%i == 0:
            return False
    return True

n = eval(input('n='))
if n<4 or n!=int(n):
    print('Illegal input')
for i in range(2, n - 1):
    if is_prime_number(i) and is_prime_number(n - i):
        print(n, '=', i, '+', n - i, sep='')
```

13.4 斐波那契数列

#斐波那契数列

```
s = eval(input('s='))

print(0, '\t', 1, end = '\t')

n1, n2, n = 0, 1, 1
while n1 + n2 <= s:
    n = n1 + n2
    print(n, end='\t')
    n1, n2 = n2, n
```

13.5 二分法迭代猜数字

#猜数游戏

```
n = eval(input('n='))
a1,a2 = 1, n + 1
while a1 != a2:
    a = (a1 + a2)//2
    print(a)
    s = input('> or < or = ?')
```

```
if s == '>':
    a2 = a
elif s == '<':
    a1 = a
else:
    print('a=', a)
    break
```

13.6 冒泡排序

#RT Lesson 冒泡排序

```
nList = list(eval(input('n[]='))) #元组强转为列表

for i in range(len(nList) - 1): #最多'数量-1'次冒泡
    print(nList)
    flag = 0 #循环前，假定已经排好
    for i in range(len(nList) - 1):
        if nList[i] < nList[i + 1]:
            nList[i], nList[i + 1] = nList[i + 1], nList[i]
            flag = 1 #有过交换位置，说明没有排好
    if flag == 0:
        break
print(nList)
```

13.7 字符串与像素图

#RT Lesson 字符串与像素图演示

```
import pygame, sys

#绘图函数，对指定像素集 pixel，在坐标(x0,y0)以 size 大小绘图
def RT_draw(screen, pixel, x0, y0, size):
    color = ( pygame.color.THECOLORS['black'], pygame.color.THECOLORS['gray32'],\
        pygame.color.THECOLORS['gray64'], pygame.color.THECOLORS['white'],\
        pygame.color.THECOLORS['red'], pygame.color.THECOLORS['green'],\
        pygame.color.THECOLORS['blue'], pygame.color.THECOLORS['orange'],\
        pygame.color.THECOLORS['brown'], pygame.color.THECOLORS['purple'],\
        pygame.color.THECOLORS['yellow'], pygame.color.THECOLORS['cyan'],\
        pygame.color.THECOLORS['sienna'], pygame.color.THECOLORS['chocolate'],\
        pygame.color.THECOLORS['coral'], pygame.color.THECOLORS['darkgreen'] )
    for y in range(len(pixel)): # 在高度方向循环
        s = pixel[y] # s 是第 y 行像素字符串
        for x in range(len(s)): # 在宽度方向循环
            if 'A' <= s[x] <= 'F':
```

```

        c = color[ord(s[x]) - 55]
    else:
        c = color[eval(s[x])] # 取得颜色编号
    pygame.draw.rect(screen, c, \
        (x * size + x0, y * size + y0, size, size), 0)

#----- pygame 初始化 -----
pygame.init() # pygame 初始化函数
screenSize = (800, 600)
screen = pygame.display.set_mode(screenSize) # 生成窗口, 获得窗口对象 screen
clock = pygame.time.Clock() # 帧率定时器初始化

#----- 程序初始化 -----
size = 8 #显示比例
pixel = [ ]
pixel.append('00077000')
pixel.append('00077000')
pixel.append('06666660')
pixel.append('60066006')
pixel.append('00666600')
pixel.append('06600660')
pixel.append('66000066')

while True: #主循环
    for event in pygame.event.get():
        if event.type == pygame.QUIT: # 鼠标关闭窗口
            pygame.quit()
            sys.exit()

    screen.fill((0,0,0)) # 背景重绘为黑色

    RT_draw(screen, pixel, 0, 0, size) # 绘制动画

    pygame.display.update() # 画面刷新
    clock.tick(50) # 帧率可调

```

13.8 星座演示程序

```

#RealThink Lesson
#星座演示程序

class CLS_Star(object): #Star 星座类定义
    def __init__(self, name, month1, day1, month2, day2, likeList):
        self.name = name

```

```
        self.month1 = month1
        self.day1 = day1
        self.month2 = month2
        self.day2 = day2
        self.likeList = likeList
    def is_in(self, month, day): #判断 month/day 是否在本星座内
        if month == self.month1:
            if day >= self.day1:
                return True
        elif month == self.month2:
            if day <= self.day2:
                return True
        return False

def get_star_index(month, day): #输入月日，输出星座编号
    global starList
    for i in range(len(starList)):
        if starList[i].is_in(month, day):
            return i

#----- 星座数据初始化 -----
starList = []
st00 = CLS_Star('白羊', 3, 21, 4, 19, [3, 4, 7])
starList.append(st00)
st01 = CLS_Star('金牛', 4, 20, 5, 20, [4, 9, 8])
starList.append(st01)
st02 = CLS_Star('双子', 5, 21, 6, 21, [0, 1, 10])
starList.append(st02)
st03 = CLS_Star('巨蟹', 6, 22, 7, 22, [11, 7, 5])
starList.append(st03)
st04 = CLS_Star('狮子', 7, 23, 8, 22, [2, 6, 8])
starList.append(st04)
st05 = CLS_Star('处女', 8, 23, 9, 22, [10, 3, 11])
starList.append(st05)
st06 = CLS_Star('天秤', 9, 23, 10, 23, [8, 4, 5])
starList.append(st06)
st07 = CLS_Star('天蝎', 10, 24, 11, 22, [10, 4, 6])
starList.append(st07)
st08 = CLS_Star('射手', 11, 23, 12, 21, [0, 1, 3])
starList.append(st08)
st09 = CLS_Star('摩羯', 12, 22, 1, 19, [3, 9, 8])
starList.append(st09)
st10 = CLS_Star('水瓶', 1, 20, 2, 18, [7, 8, 9])
starList.append(st10)
st11 = CLS_Star('双鱼', 2, 19, 3, 20, [1, 2, 6])
starList.append(st11)
```

```
wordsList = ['对方非常喜欢你', '对方觉得你还不错', '对方觉得你凑合']
```

```
#----- 主程序 -----
```

```
while True:
    month1, day1 = eval(input('请输入你的月日:'))
    star1Index = get_star_index(month1, day1)
    print('你的星座是' + starList[star1Index].name + '座')
    month2, day2 = eval(input('请输入他的月日:'))
    star2Index = get_star_index(month2, day2)
    print('你的星座是' + starList[star2Index].name + '座')
    flag = 0
    for i in range(3):#关联星座有 3 个
        if starList[star2Index].likeList[i] == star1Index:
            print(wordsList[i])
            flag = 1
    if flag == 0: #star1 不属于 star2 的关联星座
        print('你不要胡思乱想了!')
```

13.9 计算 24 点

```
#RT Lesson 计算 4 张牌 24 点
```

```
#处理除法误差
```

```
pList = eval(input('请依次输入 4 张牌的面值，以逗号隔开'))
cList = ['+', '-', '*', '/']
kList = []
```

```
kList.append(['', '', '', '', '', ''])#0
```

```
kList.append(['(', '', ')', '', '', ''])#1
```

```
kList.append(['', '(', '', '', ')', ''])#2
```

```
kList.append(['', '', '', '(', '', ')'])#3
```

```
kList.append(['(', '', '', '', ')', ''])#4
```

```
kList.append(['', '(', '', '', '', ''])#5
```

```
kList.append(['(', '', ')', '(', '', ')'])#6
```

```
for p1 in range(4):
    for c1 in range(4):
        for p2 in range(4):
```

```

    if p1 == p2:
        continue
    for c2 in range(4):
        for p3 in range(4):
            if p1 == p3 or p2 == p3:
                continue
            for c3 in range(4):
                p4 = 6 - p1 - p2 - p3
                for k in kList:
                    #拼合出算式
                    s = k[0] + \
                        str(pList[p1]) + cList[c1] + k[1] + \
                        str(pList[p2]) + k[2] + cList[c2] + k[3] + \
                        str(pList[p3]) + k[4] + cList[c3] + \
                        str(pList[p4]) + k[5]
                    try: #是否计算出错 (除 0)
                        if 23.999 < eval(s) < 24.001: #考虑除法误差
                            print(s)
                    except:
                        pass #pass 表示什么也不做

```

13.10 随机微信红包 ‘1 分钱’ 算法

```

#RealThink Lesson
#随机红包 “一分钱” 算法
import random

while True:
    s, n = eval(input('随机红包总金额, 人数: '))
    if s >= n*0.01: #金额够分, 则跳出循环
        break
    print('金额太少不够分, 请重新输入! ')
mList = [1] * n #初始化
cents = int(s * 100 - n) #需要做随机处理的总金额, 单位: 分

for i in range(cents):
    mList[random.randint(0, n - 1)] += 1

for i in range(n):
    print(mList[i] / 100) #打印时恢复单位: 元

```

13.11 随机微信红包 ‘官方’ 算法

```

#RealThink Lesson
#随机红包 “一分钱” 算法

```

```
import random

while True:
    s, n = eval(input('随机红包总金额, 人数: '))
    if s >= n*0.01: #金额够分, 则跳出循环
        break
    print('金额太少不够分, 请重新输入! ')
mList = [1] * n #初始化
cents = int(s * 100 - n) #需要做随机处理的总金额, 单位: 分

for i in range(n - 1):
    r = random.randint(0, int(cents / (n - i)) * 2)
    mList[i] += r
    cents -= r
mList[n - 1] += cents #最后一个人, 不需要再随机

for i in range(n):
    print(mList[i] / 100) #打印时恢复单位: 元
```

13.12 太空大逃亡基本版代码

```

#RealThink Lesson
#太空大逃亡基本版

import pygame, sys, random
SCREEN_W, SCREEN_H = 800, 600 #屏幕尺寸

class CLS_box(object): #box类定义, 陨石和飞船都是Box
    def __init__(self, rect, speed, color = (255, 255, 255)):
        self.rect = pygame.Rect(rect)
        self.speed = speed
        self.color = color
    def draw(self, screen):
        pygame.draw.rect(screen, self.color, self.rect, 0)
    def move(self):
        self.rect.x += self.speed[0]
        self.rect.y += self.speed[1]
def create_box(rectList): #随机产生不重叠的Box, 并保证不在基地homeBox内
    while True: #随机产生新box, 并保证符合要求
        x = random.randint(0, SCREEN_W)
        y = random.randint(0, SCREEN_H)
        w = random.randint(10, 40)
        h = random.randint(10, 40)
        rect = pygame.Rect(x, y, w, h)
        if (not rect.colliderect(homeBox)) and (rect.collidelist(rectList) == -1):
            break #如果不在homeBox内, 又不与已有box重叠, 符合要求
    while True: #随机产生初始速度
        speed = [random.randint(0, 2) - 1, random.randint(0, 2) - 1]
        if speed != [0, 0]:
            break #初始速度不是静止, 符合要求
    color=(random.randint(0, 255), random.randint(0, 255), random.randint(0, 255))
    return CLS_box(rect, speed, color)

#----- Pygame初始化 -----
pygame.init() #pygame初始化函数
screen = pygame.display.set_mode( (SCREEN_W, SCREEN_H) ) #产生窗口对象
pygame.display.set_caption("RT Space") #窗口名称
clock = pygame.time.Clock() #帧率定时器初始化
font = pygame.font.Font(None, 32) #font对象初始化
#----- 程序初始化 -----
homeBox = [300, 200, 200, 200] #基地矩形初始化
myBox = CLS_box([380, 280, 10, 10], [0, 0], (0, 255, 0)) #飞船初始化
boxNumber = 100 #陨石数量
boxList, rectList = [], []
for i in range(boxNumber):
    b = create_box(rectList)
    boxList.append(b)
    rectList.append(b.rect)
timeStart = pygame.time.get_ticks() #记录开始时间

#----- 主程序 -----

```



```

#----- 主程序 -----
while True: #主循环
    for event in pygame.event.get(): #事件遍历
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        if event.type == pygame.KEYDOWN: #如果是按键KEYDOWN事件
            #----- 方向键处理, 改变速度 -----
            if event.key == pygame.K_LEFT:
                myBox.speed[0] = -1
            if event.key == pygame.K_RIGHT:
                myBox.speed[0] = 1
            if event.key == pygame.K_UP:
                myBox.speed[1] = -1
            if event.key == pygame.K_DOWN:
                myBox.speed[1] = 1

    screen.fill((0,0,0)) #背景重绘为黑色
    myBox.move() #飞船坐标计算
    myBox.draw(screen) #绘制飞船
    for b in boxList: #遍历每个陨石box
        if b.rect.x > SCREEN_W - b.rect.width or b.rect.x == 0:
            b.speed[0] = -b.speed[0] #如果这个box水平过界, 则x方向转向
        if b.rect.y > SCREEN_H - b.rect.height or b.rect.y == 0:
            b.speed[1] = -b.speed[1] #如果这个box竖直过界, 则y方向转向
        for b0 in boxList: #遍历所有其它box
            if b == b0: #如果要比较的box b0是b本身, 不需要判断碰撞
                continue
            #----- 4个方向的碰撞检测 -----
            if (abs(b0.rect.bottom - b.rect.top) <= 1) and \
                (b.rect.right >= b0.rect.left) and \
                (b0.rect.right >= b.rect.left):
                b.speed[1] = -b.speed[1]
            if (abs(b0.rect.top - b.rect.bottom) <= 1) and \
                (b.rect.right >= b0.rect.left) and \
                (b0.rect.right >= b.rect.left):
                b.speed[1] = -b.speed[1]
            if (abs(b0.rect.right - b.rect.left) <= 1) and \
                (b.rect.bottom >= b0.rect.top) and \
                (b0.rect.bottom >= b.rect.top):
                b.speed[0] = -b.speed[0]
            if (abs(b0.rect.left - b.rect.right) <= 1) and \
                (b.rect.bottom >= b0.rect.top) and \
                (b0.rect.bottom >= b.rect.top):
                b.speed[0] = -b.speed[0]
        b.move() #box坐标计算
        b.draw(screen) #绘制box
        if myBox.rect.colliderect(b.rect): #判断myBox是否和这个box相碰
            print("score:", (pygame.time.get_ticks() - timeStart) / 1000)
            pygame.quit()
            sys.exit()

    imgText = font.render(str((pygame.time.get_ticks() - timeStart) / 1000), \
        True, (0,0,255))
    screen.blit(imgText, (SCREEN_W - 100, 0)) #在屏幕右上角绘制当前时间
    pygame.display.update() #屏幕刷新
    clock.tick(50) #帧率可调

```