关于本节中要掌握的内容

- 比较运算符与bool运算符
- 条件控制与代码块
- 循环语句与控制
- import与随机数
- 倍投法则

1比较运算符与bool运算符

我们之前已经接触过整数、浮点数、字符串，这几种数据类型都有无数种可能性，但bool只有两种值，True和Flase。

- 需要注意的是，True和Flase是大小写敏感，并且不能带引号的
- bool类型也可以被用在表达式和右值中

> 右值，一种只能放在等号右边的值类型，例如，常量整数（1、2、7、100），常量浮点数（2.0），常量字符串（"abc","lalallal"）
> 左值，与右值不同的是，它既可以放在等号左边，也可以放在等号右边

bool类型使用示例

```
a = True      # right
b = False     # right
c = a         # right
print(f"a = {a},b = {b},c = {c}")
True = a      # error
```

输出

```
  File <unknown>:12
    True = a      # error
    ^
SyntaxError: cannot assign to True
```

- 可以发现，对右值进行赋值，程序会直接出错，就连前面的print都不会打印

比较运算符

比较运算符就是比较两个数，得到一个bool类型结果的运算符。下面是

运算符	含义
>	大于
>=	大于等于
<	小于
<=	小于等于
	等于

- 需要注意的是>、>=、<、<=只能用来比较整数和浮点数
- "=="是比较运算符，"="是赋值运算符

猜猜下面的结果

```
print(18==18)
print(14==13)
a = 10
b = 1
print(a!=b)
print(a!=10)
print(10.0==10.000000000000000001)
print('abc'=='abc')
print('abc'=='ABC')
print('32'==32)
print(True=='True')
print(10==10.0)
print('abc'>10)
```

输出

```
True
False
True
False
True
True
False
False
False
True
Traceback (most recent call last):
  File ~\anaconda3\Lib\site-packages\spyder_kernels\py3compat.py:356 in compat_exec
    exec(code, globals, locals)
  File c:\users\peng\.spyder-py3\untitled0.py:20
    print('abc'>10)
TypeError: '>' not supported between instances of 'str' and 'int'
```

- 浮点数是不可以直接使用"=="进行比较的，会出错
- 字符串不可以和其他类型进行比较

bool运算符

bool运算符包括：and、or、not，它们通过比较值或表达式，得到一个bool值。

- and，只有当左边和右边的值或表达式同时为真才为真，记为**同真为真**
- or，左边和右边的值或表达式中的一个为真则为真，记为**一真为真**
- not，只有当右边的值为假时结果才为真。

bool运算符示例

```
print(True and True)
print(True or False)
print(not True)
a = 1
b = 2
c = 3
d = 4
print((a < b) and (c < d))
print((d > a) and (c > b))
print((c > a) and (b != d))
print((c > a) and (b == d))
```

输出

```
True
True
False
True
True
True
False
```

- 涉及到复杂的运算要用括号括起来

2条件控制与代码块

学习完前面的bool运算，接下来只需要关键字"if"，就可以掌握条件控制了。

if是控制语句，当它后面的条件满足时，则执行子句，例如：

```
a = 1
b = 2
if a != b:
    b = a
a = 2
```

这里的代码可以理解为，当a不等于b时，执行b=a这条语句，然后退出子句，执行a=2
我们可以看到，if语句包含这4个元素：

- if 关键字
- 结果为bool的值或表达式
- 冒号
- 子句缩进

else子句

if的含义是当条件为True时执行，而else是当条件为False时执行，当然，else后面是不跟条件的，以下是一个else的使用示例：

```
a = 2
if a == 1:
  print('True')
else:
  print('False')
```

这里的代码可以理解为，当a等于1时，打印'True'；当a不等于1时，打印'False'，我们可以看到，else包含这三个元素：

- else关键字
- 冒号
- 子句缩进

练练手

1.定义两个变量，如果它们相等，输出1；如果它们不相等，输出0
2.定义四个变量，如果其中有两个变量相等，输出1；否则，输出0
3.定义四个变量，如果其中有三个变量相加等于10，输出1；否则，输出0

3循环语句与控制

之前，我们都是对条件进行一次判断就结束。如果我们想要一遍又一遍的执行代码块，应该怎么做呢？这时就需要利用到for循环了：

```
for i in range(5):
  print(f"I'm {i}")
```

上面代码的含义是循环5次，i每次的值分别为：0,1,2,3,4。
这里还有一个需要注意的点，range可用的参数有三个：
range(start, stop[, step])

- start: 计数从 start 开始。默认是从 0 开始。例如range（5）等价于range（0， 5）；
- stop: 计数到 stop 结束，但不包括 stop。例如：range（0， 5） 是[0, 1, 2, 3, 4]没有5
- step：步长，默认为1。例如：range（0， 5） 等价于 range(0, 5, 1)

计算从1加到100的和

```
# 方法1:
sum = 0
for i in range(1,101,1):
  sum += i
print(f"sum = {sum}")
# 方法2:
sum = 0
for i in range(101):
  sum += i
print(f"sum = {sum}")
```

- 尝试一下1 * 3 * 5 * 9 * ... * 19的结果

可能的实现

```
sum = 1
for i in range(1,20,2):
  sum *= i
print(f"sum = {sum}")
```

循环的控制

循环中包含两种控制，分别是continue和break。

- continue：直接进入下一次循环
- break：退出循环

```
sum = 0
for i in range(10):
  if i == 5:
    continue
  if i == 8:
    break
  sum += i
print(f"sum = {sum}")
```

上面代码的含义是：循环10次，从0开始，如果i等于5，直接进入下一次循环；如果i等于8，直接退出循环。最后的sum的结果为：1+2+3+4+6+7=23

import与随机数

python包含了很多已实现的基本函数，包含之前用过的print、range。当然，它们属于内置函数，python中也包含着其他的模块，我们称为标准库，需要我们导入后才可以使用。

```
# 导入random模块
import random
for i in range(5):
    print(random.random())
```

上面的程序将输出5次0-1之间的随机数，如果想得到1-100之间的随机数，可以这样做：

```
import random
for i in range(5):
    print(random.randint(1, 100))
```

倍投法则

这里我们将会运用到本节课学到的所有内容来完成这个题目。

- 问题
  每次投掷硬币，我们需要进行押注，赢了获取双倍投入。
- 倍投法则
  为了获胜，我们每次从1开始押注，赢了继续从1押注，输了则压双倍。

这样的法则到底有没有用呢？我们使用python来编写一段代码来实现。

可能的实现

```
import random
# 初始资金
balance = 1000000
# 初始押注，输了加倍，赢了继续从100开始
bets = 100
balance -= bets
# 遍历1000000轮
for i in range(1000000):
  if random.random()>=0.5:
        balance += (bets*2)
        bets = 100
  else:
      bets *= 2
      balance -= bets
      if balance<0:
          print(f'The game is over, you survived {i+1} rounds')
          break
```

课后练习

- 程序随机生成一个数字，你有10次机会去猜中它，猜错了会输出大了还是小了，猜对了会输出正确。如果前5次就可以猜对，还会输出你真棒。

> 你需要自己查询input()函数的使用。

- 你和电脑猜石头、剪刀、布，如果你获胜了就输出获胜并结束程序；如果你输了就输出失败并继续

> 或许你需要自己查询elif

- 输出9*9乘法表
- 现在有三个人，初始都有100点能量，现在进行游戏，每轮每个人都可能会随机从某个人身上抢到1点能量；也可能抢不到。当有一个人能量为0时，结束这个游戏。