第五章：数字

本章讲的是数字方面的数据类型，以及数字方面的内建函数。

目录

[1·数字类型 1](#_Toc350442560)

[2、数字类型介绍 1](#_Toc350442561)

[3·运算符 2](#_Toc350442562)

[3、计算机中算数表达式的解析 4](#_Toc350442563)

[4、内建函数和工厂函数 5](#_Toc350442564)

[5·其他数字类型 6](#_Toc350442565)

[6·数字科学计算的包 6](#_Toc350442566)

[7、数字的练习题： 6](#_Toc350442567)

## 1·数字类型

Python 支持多种数字类型：整型、长整型、布尔型、双精度浮点型、十进制浮点型和复数。

1)创建

即赋值

2)更新（假更新，真替换）

通过给数字对象（重新）赋值，您可以“更新”一个数值对象。我们之所以给更新这两个

字加上引号， 是因为实际上你并没有更新该对象的原始数值。这是因为数值对象是不可改变对象。Python 的对象模型与常规对象模型有些不同。你所认为的更新实际上是生成了一个新的数值对象，并得到它的引用。

在学习编程的过程中， 我们一直接受这样的教育， 变量就像一个盒子， 里面装着变量的

值。在 Python 中， 变量更像一个指针指向装变量值的盒子。 对不可改变类型来说， 你无法改变盒子的内容， 但你可以将指针指向一个新盒子。每次将另外的数字赋给变量的时候，实际上创建了一个新的对象并把它赋给变量.(不仅仅是数字，对于所有的不可变类型，都是这么回事)

3)删除

按照 Python 的法则， 你无法真正删除一个数值对象， 你仅仅是不再使用它而已。如果你

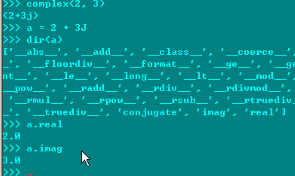
实际上想删除一个数值对象的引用， 使用 del 语句（参见 3.5.6 小节）。 删除对象的引用之后， 你就不能再使用这个引用(变量名)， 除非你给它赋一个新值。如果试图使用一个已经被删除的对象引用， 会引发 NameError 异常。

## 2、数字类型介绍

1)整型

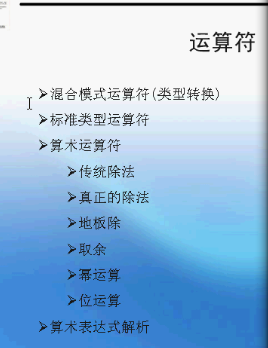
* 1. 布尔型 包含True和False两个值的整型
  2. 标准整数类型 0x表示十六进制, 无前缀表示十进制, 0表示八进制, 0b表示二进制
  3. 长整型 整数后加L, 表示范围与可用内存大小有关..

1. )双精度浮点数: 一个符号位, 52个底位, 11个指数位. 使用后缀e表示指数
2. )复数: 实数 + 序数J构成一个复数



* 1. python中的复数概念
     1. 虚数不能单独存在, 总是和一个值为0.0的实数部分一起构成一个复数
     2. 复数由实数部分和虚数部分组成
     3. 表示虚数的语法: real + imagJ
     4. 实数部分和虚数部分都是浮点数
     5. 虚数部分必须有后缀j或J
  2. 复数的内建属性
     1. real 复数的实部
     2. imag 复数的虚部
     3. conjugate() 返回该复数的共轭复数

## 3·运算符



1)强制类型转换规则



2)幂运算符\*\* 比左侧单目运算符优先级高, 比右侧单目运算符优先级高.

注意：

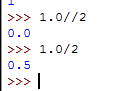


如果是and，看的是后面的，如果是or，看的是前面的不符再看后面。

3)位运算符: ~, &, |, ^, <<, >>

* 1. 负数会被当成正数的二进制补码处理
  2. 左移和右移N位等同与无溢出检查的2的N次幂运算
  3. 长整数的位运算使用一种经过修改的二进制补码形式, 使的符号位可以无限左移
  4. ~是单目运算符

4)除法运算符



在未来的 Python 版本中，Python 开发小组已经决定改变 / 运算符的行为。/ 的行为将变

更为真正的除法， 会增加一种新的运算来表示地板除。 下面我们总结一下 Python 现在的除法规则， 以及未来的除法规则：

传统除法

如果是整数除法， 传统除法会舍去小数部分，返回一个整数（地板除）。如果操作数之一

是浮点数，则执行真正的除法。包括 Python 语言在内的很多语言都是这种行为。看下面的例子：

>>> 1 / 2 # perform integer result (floor) # 地板除

0

>>> 1.0 / 2.0 # returns actual quotient#真正除法

0.5

真正的除法

除法运算总是返回真实的商， 不管操作数是整数还是浮点数。在未来版本的 Python 中，

这将是除法运算的标准行为。现阶段通过执行 from \_\_future\_\_ import division 指令， 也

可以做到这一点。

>>> from \_\_future\_\_ import division

>>>

>>> 1 / 2 # returns real quotient

0.5

>>> 1.0 / 2.0 # returns real quotient

0.5

地板除

从 Python 2.2 开始， 一个新的运算符 // 已经被增加进来， 以执行地板除： // 除法不

管操作数何种数值类型，总是舍去小数部分，返回数字序列中比真正的商小的最接近的数字。

>>> 1 // 2 # floors result, returns integer # 地板除， 返回整数 0

>>> 1.0 // 2.0 # floors result, returns float # 地板除， 返回浮点数 0.0

>>> -1 // 2 # move left on number line# 返回比 –0.5 小的整数， 也就是 －1

-1

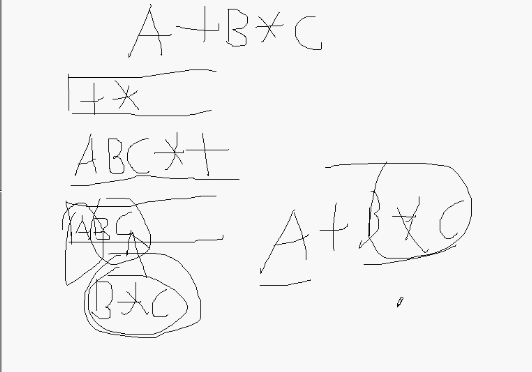
5)运算符的解析

## 3、计算机中算数表达式的解析

3+2中缀表达式，操作符的位置

32+后缀表达式

+32前缀表达式



计算机中怎么计算a+b\*c

通过栈

遇到后括号就出到前括号为止。

## 4、内建函数和工厂函数

* 1. 数字类型函数:
     1. 转换工厂函数: int(), long(), float(), complex(), 接受参数第一个是要转换的值, 第二个是进制..如果要转换的值是string才能使用第二个函数指定进制.
        1. python2.2开始, 加入了bool(), 用来将整数1和0转换成为标准布尔值(所有非0数都返回True).
        2. python2.3的标准数据类型添加了Boolean类型, true和false也有了常量值True和False, 而不再是1和0
        3. bool(obj) 返回obj对象的布尔值, 也就是obj.\_\_nonzero\_\_()
        4. complex(real, imag=0.0)
     2. 功能函数
        1. abs(number) 返回数字的绝对值, 如果是复数, 返回math.sqrt(num.real2 + num.imag2)
        2. coerce(number1, number2): 返回按照类型转换规则转换得到的两个数字组成的元组
        3. divmod(number1, number2) 返回一个包含商和余数的元组, 整数返回地板除和取余操作结果, 浮点数返回的商部分是math.floor(number1/number2), 复数的商部分是ath.floor((number1/number2).real)
        4. pow()和\*\*功能相同
        5. round(number[, base]) 对浮点数进行四舍五入运算, base参数是小数位参数, 如果不指定, 返回与第一个参数最接近的整数的浮点形式
        6. int() 直接去掉小数部分, floor()得到最接近但小于原数的整数, round()得到最接近原数的整数
     3. 整数的内建函数: hex(), oct(), bin()…ord(“A”) 接受一个ascii或unicode字符, 返回相应的unicode值 🡸🡺 char(65L)接受unicode码值, 返回对应的unicode字符.

## 5·其他数字类型

* 1. 布尔数:
     1. 布尔型是整型的子类, 但是不能再被继承而生成它的子类.
     2. 没有\_\_nonzero\_\_()方法的对象默认值是True
     3. 对于值为0的任何数字或空集(空的list|tuple|dict)在python中值都是False
     4. 数学运算中, True ==1, False == 0
  2. 十进制浮点数: from decimal import Decimal

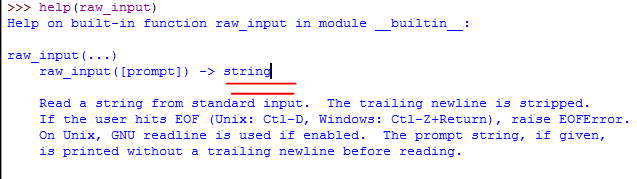
## 6·数字科学计算的包

* 1. 高级的Third package: Numeric(NumPy)
  2. python自带的数字类型相关模块
     1. decimal 十进制浮点运算类Decimal
     2. array 高效数值数组(字符, 整数, 浮点数)
     3. match/cmatch 标准c库数学运算函数. 常规数学运算在match, 复数运算在cmatch
     4. operator 数字运算符的函数实现
     5. random 伪随机数生成器
        1. randint(start, end)随机生成start, end之间的一个整数
        2. uniform(start, end)随机生成范围内的一个浮点数
        3. randrange([start ,]stop[, step])随机生成start, stop内按step步增数字范围的一个整数
        4. choice(sequence)随机返回给定的序列中的一个元素

## 7、数字的练习题：

Python接收到的是啥类型的：

Raw\_input



5–9. 数值形式 回答下面关于数值格式的问题：

(a) 为什么下面的例子里 17+32 等于 49， 而 017+32 等于 47， 017+032 等于 41？

>>> 17 + 32

因为017代表八进制

（b）为什么下面这个表达式我们得到的结果是 134L 而不是 1342 ？

>>> 56lL+ 78l

134L

不要把小写的l看成1，所以为了避免这种情况，最后将l大写L

Ranom这个模块很有用，随机数肯定是有很多地方要使用的。