

Python 科学计算基础

第二章 内置数据类型及其运算

2025 年 9 月 5 日

目录

变量和类型

数值类型 (int、float 和 complex)

bool 类型

NoneType 类型

序列类型: list(列表)、tuple(元组)、str(字符串) 等

set 类型 (集合)

dict 类型 (字典)

实验 2: 内置数据类型及其运算

类型

Python 语言支持面向对象编程范式，Python 程序中的数据都是对象。每个对象包括的属性有：身份 (identity，例如内存地址)，类型 (type) 和值 (value)。

类型规定了数据的存储形式和可以进行的运算。例如整数类型可以进行四则运算和位运算，而浮点类型可以进行四则运算但不能进行位运算。

变量

变量是计算机内存中存储数据的标识符，根据变量名称可以获取内存中存储的数据。

变量的类型由其存储的数据决定。

变量名只能是字母、数字或下划线的任意组合。变量名的第一个字符不能是数字。以下 Python 关键字不能声明为变量名：

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

动态类型

和 C、Java 等静态类型语言不同，Python 是一种动态类型语言，变量的类型在使用前无需声明，而是在程序运行的过程中根据变量存储的数据自动推断。

动态类型的益处是程序的语法更简单，付出的代价是程序的运行效率不如静态类型语言（如 C 和 Java 等）程序。原因包括：

- ▶ 无法进行编译时的优化；
- ▶ 类型推断占用了程序运行的时间。

动态类型

Python 赋值语句的语法形式是“变量=表达式”，其中的等号是赋值运算符而非数学中的相等运算符。该语句将表达式的求值结果设置为变量的值。

例如赋值语句 `x=1` 运行完成以后，变量 `x` 的值为 1 并且其类型为整数类型 `int`，因为 `x` 存储的数据 1 是整数。赋值运算符可以和二元运算符组合在一起形成复合赋值运算符，如 `x+=1` 等同于 `x=x+1`。

数值类型

数值类型包括 `int`(整数)、`float`(浮点数) 和 `complex`(复数)。

int 类型

int 类型表示任意精度的整数，可以进行的运算包括相反 (-)、加 (+)、减 (-)、乘 (*)、除 (/)、整数商 (//)、取余 (%)、乘方 (**) 和各种位运算。两个整数进行的位运算包括按位取或 (|)、按位取与 (&) 和按位取异或 (^)。单个整数进行的位运算包括按位取反 (~)、左移 (<<) 和右移 (>>)。

int 类型

bin函数将十进制转化为二进制

表达式是对数据进行运算的语法形式。当一个表达式中出现多种运算时，这些运算的运行次序由运算符的**优先级** (precedence) 和**结合性** (associativity) 确定。优先级高的运算先运行。若多个运算的优先级相同，则根据结合性是从左到右还是**从右到左**确定次序。

程序 2.2

float 类型

float 类型根据 IEEE754 标准定义了十进制实数在计算机中如何表示为二进制浮点数。**64 位**二进制浮点数对应的十进制实数的**有效数字的位数为 15**。标准库的 **sys 模块的 float_info** 属性提供了 float 类型的取值范围 **(max, min)** 和有效数字位数 **(dig)** 等信息。绝对值超过 max 的数值表示为 **inf**(正无穷) 或 **-inf**(负无穷)。

float 类型

float 类型可以进行相反 (-)、加 (+)、减 (-)、乘 (*)、除 (/)、整数商 (//)、取余 (%) 和乘方 (**) 等运算。由于需要进行进制转换和表示位数的限制，实数在计算机中的表示可能存在误差，称为舍入误差 (rounding error)。例如十进制的 0.1 表示为二进制无限循环小数 $0.000\overline{1100}$ ，在截断后导致舍入误差。舍入误差在计算过程中可能不断积累，导致最终计算结果出现较大的误差。

程序 2.3 和程序 2.4

complex 类型

`complex` 类型表示实部和虚部为 `float` 类型的复数，可以进行相反 (-)、加 (+)、减 (-)、乘 (*)、除 (/)、和乘方 (**) 等运算。

程序 2.5

数值类型的内置函数

对于一个整数或实数，`abs` 函数获取其绝对值。对于一个复数，`abs` 函数获取其模。

`int` 和 `float` 函数可以进行这两种类型的相互转换。

`complex` 函数从两个 `float` 类型的数值生成一个 `complex` 类型的数值。`pow` 计算乘方，等同于`**`运算符。

程序 2.6 1.`pow(x,y)`表示`x**y`
 2.`round` (计算式, 小数点后保留几位有效数字) 四舍五入

math 模块和 cmath 模块

`math` 模块定义了圆周率 `math.pi`、自然常数 `math.e` 和以实数作为自变量和因变量的常用数学函数。常用的函数包括 `ceil`、`floor`、`factorial`、`sqrt`、`exp`、`log`、`log10`、`sin`、`cos`、`tan`、`asin`、`acos` 和 `atan` 等。

`cmath` 模块定义了以复数作为自变量和因变量的常用数学函数。

程序 2.7

bool 类型

int 类型和 float 类型的数据可以使用以下这些关系运算符进行比较： $>$ (大于)、 $<$ (小于)、 $>=$ (大于等于)、 $<=$ (小于等于)、 $==$ (等于)、 $!=$ (不等于)。比较的结果属于 bool 类型，只有两种取值：True 和 False。

bool 类型的数据可以进行三种逻辑运算，按优先级从高到低依次为 not(非)、and(与) 和 or(或)。

程序 2.8

NoneType 类型

NoneType 类型只有一个值 `None`，表示空值。

序列类型

序列类型 (Sequence Types) 可以看成是一个存储数据元素的容器，这些元素是有序的，每个元素对应一个索引值。用 n 表示序列的长度，则索引值的取值范围是区间 $[-n, n-1]$ 内的所有整数。序列中的第 k 个元素 ($1 \leq k \leq n$) 对应的索引值有两个： $k-1$ 和 $k-n-1$ 。例如，索引值 0 和 $-n$ 都对应第 1 个元素，索引值 $n-1$ 和 -1 都对应第 n 个元素。

序列类型包括 list(列表)、tuple(元组)、range(范围)、str(字符串)、bytes、bytearray、和 memoryview 等。range 通常用于 for 语句 (第三章)。bytes、bytearray 和 memoryview 是存储二进制数据的序列类型。

list(列表) 和 tuple(元组)

列表和元组通常用来存储若干同一类型的元素。

- 语法 ▶ 列表的语法是用方括号括起的用逗号分隔的若干元素，例如 `[2,3,5,7,11]` 是一个存储了五个质数的列表。
- ▶ 元组的语法是用圆括号括起的用逗号分隔的若干元素，例如 `(2,3,5,7,11)` 是一个存储了五个质数的元组。

语义

列表是可变的，即可对其存储的元素进行插入、删除和修改。
元组是不可变的。

列表和元组的部分共有运算

- ▶ `x in s` `x not in s`
- ▶ `s + t` `s*n` 或 `n*s`
- ▶ `s[n]` `s[i:j]` `s[i:]` `s[:j]` `s[i:j:k]`
- ▶ `len(s)` `min(s)` `max(s)` `s.count(x)`
- ▶ `s.index(x)` `s.index(x, i)` `s.index(x, i, j)`

程序 2.9

1.`type(s)`表示输出s的类型, list or tuple

2.`id(s)`表示输出s的地址 (一串数字)

3.`r=[s,t]`为嵌套列表, 二维列表, 可理解为矩阵或二维数组

取r中的元素需要2个索引值, 如`r=[0][2]`, 改变r同时改变s/r中的数

列表的部分特有运算

赋值 ▶ `s[i] = x` `s[i:j] = t` `s[i:j:k] = t`

删除 ▶ `del s[i:j]` `del s[i:j:k]`

▶ `s.append(x)` `s.clear()` `s.copy()`

▶ `s.extend(t)` `s += t` `s *= n`

▶ `s.insert(i, x)` `s.pop(i)` `s.remove(x)`

▶ `s.sort()` `s.sort(reverse=True)`

程序 2.9

1. `s.reverse()` 表示将 `s` 中所有元素的次序反转, 返回 `s`

2. 删除列表中的所有元素 `x`:

```
my_list = [item for item in my_list if item != x]
```

str 类型 (字符串)

str 类型表示不可变的使用 Unicode 编码的字符构成的序列，即字符串。UTF-8 是一种广泛使用的 Unicode 实现方式，即用一个或多个字节存储二进制编码。

ord 函数可获取一个字符对应的 Unicode 编码值，例如 ord('A') 值为 65。chr 函数可获取一个 Unicode 编码值对应的字符，例如 chr(90) 值为 'Z'。

字符串可用单引号、双引号或三个相同的单引号或双引号括起。单引号和双引号可以相互嵌套。用三引号括起的字符串可以包含程序中的多行语句，常作为函数和模块的注释。

返回字符串的部分特有运算

- ▶ `s.lstrip(t)` `s.rstrip(t)` `s.strip(t)`
- ▶ `s.replace(t, u)` `s.replace(t, u, k)`
- ▶ `s.split(t)`
- ▶ `t.join(ss)`

逆{

程序 2.10

格式化输出

字符串的`%`运算符可以使用多种转换说明符为各种类型的数值生成格式化输出。运算符的左边是一个字符串，其中可包含一个或多个转换说明符。运算符的右边包含一个或多个数值，这些数值必须和转换说明符一一对应。如果有多个数值，这些数值必须置入一个元组中。

- ▶ `%d`、`%i` `%o` `%x`、`%X`
- ▶ `%e`、`%E` `%f`、`%F` `%g`、`%G`
- ▶ `%s`

程序 2.11

不返回字符串的部分特有运算

- ▶ `s.startswith(t)` `s.startswith(t, start)` `s.startswith(t, start, end)`
- ▶ `s.endswith(t)` `s.endswith(t, start)` `s.endswith(t, start, end)`
- ▶ `s.find(t)` `s.find(t, start)` `s.find(t, start, end)`
- ▶ `s.rfind(t)` `s.rfind(t, start)` `s.rfind(t, start, end)`
- ▶ `str.isalpha(s)` `str.isdecimal(s)` `str.isalnum(s)`
- ▶ `str.islower(s)` `str.isupper(s)`

程序 2.10

set 类型 (集合)

集合可以看成是一个存储数据元素的容器，存储的元素是**无序**且**不可重复**的，类似于数学中定义的集合。

集合的语法是用大括号括起的用逗号分隔的若干数据。例如 $\{2,3,5,7,11\}$ 是一个存储了五个质数的集合。**frozenset** 类型是**不可变**的 set 类型。

集合的部分运算

集合的部分运算如下：

- ▶ `len(s)`
- ▶ `x in s` `x not in s`
- ▶ `s.isdisjoint(t)`
- ▶ `s.issubset(t)` $s \leq t$ $s < t$
- ▶ `s.issuperset(t)` $s \geq t$ $s > t$
- ▶ `s.union(t)` `s.intersection(t)` `s.difference(t)`
- ▶ `s.add(x)` `s.remove(x)`

通过`s=set(t)`,可将序列或元组`t`中的元素赋值给集合:

程序 2.12

dict 类型 (字典)

字典可以看成是一个存储数据元素的容器，存储的每个元素由两部分组成：键 (key) 和其映射到的值 (value)。键的类型必须是可求哈希值的 (包括但不限于 int、str、tuple、frozenset 和用户自定义的类等)，但不能是可变的 (包括但不限于 list、set 和 dict 等)。

字典的语法是用大括号括起的多个元素，每个元素内部用冒号分隔键和值，元素之间用逗号分隔。例如 {2:4, 3:9, 5:25, 7:49, 11:121} 是一个存储了五个元素的字典，每个元素的键是一个整数，其映射到的值是键的平方。

字典的部分运算

字典的部分运算如下：

- ▶ `len(d)`
- ▶ `key in d` `key not in d`
- ▶ `d[key] = value`
- ▶ `del d[key]`
- ▶ `d.clear()`
- ▶ `d.get(key[, default])`
- ▶ `d.pop(key[, default])`
- ▶ `d.items()` `d.keys()` `d.values()`

程序 2.13

实验 2： 内置数据类型及其运算

本实验的目的是掌握常用的类型 (float 和 list) 的运算。

在 Blackboard 系统提交一个文本文件 (txt 后缀)，文件中记录每道题的源程序和运行结果。

1. 计算等额本息还款金额

等额本息是一种分期偿还贷款的方式，即借款人每月按相等的金额偿还贷款本息，每月还款金额 P 可根据贷款总额 A 、年利率 r 和贷款月数 n 计算得到，公式为

$$P = \frac{\frac{r}{12}A}{1 - (1 + \frac{r}{12})^{-n}}$$

计算当贷款金额为 1000000，贷款时间为 30 年，年利率分别为 4%、5% 和 6% 时的每月还款金额和还款总额。输出结果在小数点以后保留两位有效数字。

答案：

(4774.15, 1718695.06)

(5368.22, 1932557.84)

(5995.51, 2158381.89)

2. 计算三角形的内角

定义三个变量 “ $a=3$; $b=6$; $c=7$ ” 表示一个三角形的三个边的长度，使用公式

$$a^2 = b^2 + c^2 - 2bc \cos \alpha$$

$$b^2 = a^2 + c^2 - 2ac \cos \beta$$

$$c^2 = a^2 + b^2 - 2ab \cos \gamma$$

分别计算三个内角 (α, β, γ) 的度数，然后检验等式 $\alpha + \beta + \gamma = 180$ 是否成立。

答案：

25.208765296758365 58.41186449479884 96.37937020844281

3. list 类型的运算

定义两个列表 “`s=[2,4,0,1,3,9,5,8,6,7]`; `t=[2,6,8,4]`”。在 lpython 中对于 `s` 和 `t` 运行程序 2.9 中 “`In[30]:`” 行开始的所有运算，并记录输出结果。若某个运算修改了 `s`，在运行下一个运算之前需要再设置 `s=[2,4,0,1,3,9,5,8,6,7]`。若某个运算需要一些参数 (如 `i`, `j` 和 `x` 等)，可自行设定。