第2章 数字类型与字符串

《Python程序开发案例教程（第2版）》

学习目标/Target

- 掌握数字类型，能够在程序中正确表示不同数字类型的数据
- 掌握运算符的用法，能够使用运算符进行数值运算
- 掌握运算符优先级，能够在数值运算中正确使用运算符
- 掌握字符串的创建方式，能够准确创建字符串类型的变量

学习目标/Target

- 掌握格式化字符串的方式，能够使用%、format()和f-string这3种方式格式化字符串
- 掌握字符串的常见操作，能够使用方法实现字符串的常见操作
- 掌握字符串的索引和切片，能够使用索引和切片访问字符串的字符或子串
- 掌握类型转换函数的使用，能够使用类型转换函数对不同类型的数据进行转换

章节概述/ Summary

在计算机编程中，数据是程序处理的核心。Python中提供了丰富多样的数据类型，简单的数据类型有数字类型和字符串，其中数字类型用于表示数值形式的数据，它可以联合运算符进行算术或逻辑等操作；字符串用于表示文本形式的数据，它能够对文本进行一些诸如分割、拼接、查找和替换操作，在程序中应用十分广泛。本章将对数字类型、字符串和运算符的相关内容进行讲解，并通过实例帮助大家熟练运用它们。

目录/Contents

01 数字类型

02 运算符

03 经典实例

04 字符串

05 类型转换

2.1 数字类型

2.1 数字类型

- 掌握数字类型，能够在程序中正确表示不同数字类型的数据

2.1 数字类型

表示数值的数据类型称为数字类型。Python内置的数字类型有整型（int）、浮点型（float）、复数类型（complex），它们分别对应数学中的整数、小数和复数，此外，还有一种比较特殊的类型——布尔类型（bool）。

整型 | 浮点型 | 复数类型 | 布尔类型

类似-2、-1、0、1、2这样的数据被称为整型数据。在Python中可以使用四种进制方式表示整型数据，分别为二进制、八进制、十进制和十六进制，其中二进制整型数据以“0b”或“0B”开头、八进制整型数据以“0o”或“0O”开头、十六进制整型数据以“0x”或“0X”开头，默认采用十进制方式表示。

2.1 数字类型

表示数值的数据类型称为数字类型。Python内置的数字类型有整型（int）、浮点型（float）、复数类型（complex），它们分别对应数学中的整数、小数和复数，此外，还有一种比较特殊的类型——布尔类型（bool）。

整型 | 浮点型 | 复数类型 | 布尔类型

类似1.1、0.5、-1.4、3.12e2这样的数据被称为浮点型数据，浮点型数据是带有小数点的数值。Python的浮点数可以采用两种方式表示，分别是十进制和科学计数法。

2.1 数字类型

表示数值的数据类型称为数字类型。Python内置的数字类型有整型（int）、浮点型（float）、复数类型（complex），它们分别对应数学中的整数、小数和复数，此外，还有一种比较特殊的类型——布尔类型（bool）。

整型 | 浮点型 | 复数类型 | 布尔类型

类似3+2j、3.1+4.9j、-2.3-1.9j这样的数据被称为复数，Python中的复数有以下3个特点：

（1）复数由实部real和虚部imag构成，其一般形式为real+imagj。

（2）实部real和虚部的imag都是实数。

（3）虚部imag必须有后缀j或J。

2.1 数字类型

表示数值的数据类型称为数字类型。Python内置的数字类型有整型（int）、浮点型（float）、复数类型（complex），它们分别对应数学中的整数、小数和复数，此外，还有一种比较特殊的类型——布尔类型（bool）。

整型 | 浮点型 | 复数类型 | 布尔类型

布尔类型用于表示逻辑值，它只有两个可能的取值：True和False，分别对应数学命题的真与假。Python中任何类型的数据都具有布尔属性，它们可以根据一定的规则转换为布尔类型的值，在进行转换时，符合以下条件的数据都会被转换为False。

（1）None；

（2）任何为0的数字类型，如0、0.0、0j；

（3）空字符串,如''、""；

（4）任何为空的复杂类型，如空元组()、空列表[]、空字典{}；

2.2 运算符

2.2.1 算术运算符

- 掌握算术运算符的用法，能够使用运算符进行数值运算

2.2.1 算术运算符

Python中的算术运算符包括+、-、*、/、//、%和**，这些运算符都是双目运算符，一个运算符可以和两个操作数组成一个表达式。当解释器执行包含算术运算符的表达式时，会根据运算符的功能对两个操作数进行特定操作，并将操作后的结果进行返回。

2.2.1 算术运算符

以操作数a = 3，b = 5为例，Python中各个算术运算符的功能及示例如表所示。

运算符	说明	示例
+	加：使两个操作数相加，获取操作数的和	a + b，结果为8
-	减：使两个操作数相减，获取操作数的差	a –b ，结果为-2
*	乘：使两个操作数相乘，获取操作数的积	a * b，结果为15
/	除：使两个操作数相除，获取操作数的商	a / b，结果为0.6
//	整除：使两个操作数相除，获取商的整数部分	a // b，结果为0
%	取余：使两个操作数相除，获取余数	a % b，结果为3
**	幂：使两个操作数进行幂运算，获取a的b次幂	a ** b，结果为243

2.2.1 算术运算符

无论参加运算的操作数是什么类型，解释器都能给出运算后的正确结果，这是因为Python在对不同类型的对象进行运算时，会强制将数据的类型进行临时类型转换，这些转换遵循如下规律：

- 布尔类型在进行算术运算时，将False和True被视为数值0或1；
- 整型与浮点型进行运算时，将整型转化为浮点型；
- 其它类型与复数类型进行运算时，将其他类型转换为复数类型。

简单来说，类型相对简单与类型复杂的操作数进行混合运算时，会被转换为与复杂操作数相同的类型。

2.2.2 比较运算符

- 掌握比较运算符的用法，能够使用比较符进行数值比较

2.2.1 算术运算符

比较运算符用于比较两个操作数的大小，Python中的比较运算符有==、!=、>、<、>=、<=，比较运算符同样是双目运算符，它与两个操作数构成一个表达式。当解释器执行包含比较运算符的表达式时，会根据运算符的功能对两个操作数进行比较操作，并返回布尔类型的值。

2.2.2 比较运算符

以a = 3，b = 5为例，Python中各比较运算符的功能及示例如表所示。

运算符	说明	示例
==	比较左操作数和右操作数，若相等则结果为True，否则结果为False	a == b不成立，结果为False
!=	比较左操作数和右操作数，若不相等则结果为True，否则为False	a != b成立，结果为True
>	比较左操作数和右操作数，若左操作数大于右操作数则结果为True，否则为False	a > b 不成立，结果为False
<	比较左操作数和右操作数，若左操作数小于右操作数则结果为True，否则为False	a < b成立，结果为True
>=	比较左操作数和右操作数，若左操作数大于或等于右操作数则结果为True，否则为False	a >= b 不成立，结果为False
<=	比较左操作数和右操作数，若左操作数小于或等于右操作数则结果为True，否则为False	a <= b 成立，结果为True

2.2.2 比较运算符

需要注意的是，比较运算符只对操作数进行比较操作，不会对操作数自身造成影响，即经过比较运算符运算后的操作数不会被修改。比较运算符与操作数构成的表达式的结果只能是True或False，这种表达式通常用于布尔测试。

2.2.3 赋值运算符

- 掌握赋值运算符的用法，能够使用赋值运算符为操作数赋值

2.2.3 赋值运算符

赋值运算符是双目运算符，它的功能是将运算符右侧的表达式或值赋给左侧的操作数，其中左操作数必须是一个值可修改的变量。“=”是基本的赋值运算符，此外“=”可与算术运算符组合成复合赋值运算符，从而实现同时进行算术运算和赋值操作的便利。Python中的复合赋值运算符有+=、-=、*=、/=、//=、%=、**=，它们的功能与相应的算术运算符相似，例如“a+=b”等价于“a=a+b”，“a-=b”等价于“a=a-b”，诸如此类。

2.2.4 逻辑运算符

- 掌握逻辑运算符的用法，能够使用逻辑运算符为操作数进行逻辑运算

2.2.4 逻辑运算符

Python支持逻辑运算，但Python逻辑运算符的功能与其它语言有所不同。Python中分别使用“or”、“and”、“not”这三个关键字作为逻辑运算“或”、“与”、“非”的运算符，其中or与and为双目运算符，not为单目运算符。

2.2.4 逻辑运算符

逻辑运算符——or

当使用运算符or连接两个操作数进行逻辑或运算时，若左操作数的布尔值为True，则返回左操作数，否则返回右操作数。注意，如果操作数一个表达式，则不会直接返回操作数本身，而是会返回表达式的计算结果。

```
result_one = 0 or 3 + 5          # 左操作数布尔值为False
print(result_one)
result_two = 3 or 0              # 左操作数布尔值为True
print(result_two)
```

2.2.4 逻辑运算符

逻辑运算符——and

当使用运算符and连接两个操作数进行逻辑与运算时，若左操作数的布尔值为False，则返回左操作数，否则返回右操作数。注意，如果操作数一个表达式，则不会直接返回操作数本身，而是会返回表达式的计算结果。

```
result_one = 3 - 3 and 5          # 左操作数布尔值为False
print(result_one)
result_two = 3 - 4 and 5          # 左操作数布尔值为True
print(result_two)
```

2.2.4 逻辑运算符

逻辑运算符——not

当使用运算符not进行逻辑非运算时，若操作数的布尔值为False，则返回True，否则返回False。

```
result_one = not 3 - 5          # 操作数布尔值为True
print(result_one)
result_two = not False          # 操作数的值为False
print(result_two)
```

2.2.5 成员运算符

- 掌握成员运算符的用法，能够使用成员运算符为检测给定值是否存在

2.2.5 成员运算符

成员运算符用于检测给定值是否存在字符串、列表、元组、集合、字典中，并返回检测后的结果。Python中提供了两个成员运算符，分别是in和not in。

- in：如果给定值在字符串、列表、元组、集合、字典中，返回True，否则返回False。
- not in：如果给定值不在字符串、列表、元组、集合、字典中，返回True，否则返回False。

2.2.5 成员运算符

注意，列表、元组、集合、字典是比较复杂的数据类型。接下来以字符串为例，演示如何使用成员运算符检测一个字符串是否在另一个字符串中。

```
words = 'abcdefg'          # 定义一个变量，保存字符串类型的数据
print('f' in words)        # 检测'f'是否在字符串中
print('c' not in words)    # 检测'c'是否在字符串中
```

2.2.6 位运算符

- 掌握位运算符的用法，能够使用位运算符对数据进行计算

2.2.6 位运算符

位运算符是一组特殊的运算符，用于对整数在二进制位上进行操作。Python的位运算符有<<、>>、&、|、^、~，其中运算符~是单目运算符，其余全部是双目运算符，操作数必须是整数。

2.2.6 位运算符

以a = 3，b = 5为例，Python中各个位运算符的功能及示例如表所示。

运算符	说明	示例
<<	按位左移，作用是将二进制形式操作数的所有位全部左移指定位数，高位丢弃，低位补0	a<<b，结果为96
>>	按位右移，作用是将二进制形式操作数的所有位全部右移指定位数，低位丢弃，高位补0	a>>b，结果为0
&	按位与，作用是将两个二进制形式操作数进行逐位的与运算，当两个对应位均为1时，结果位为1，否则为0	a&b，结果为1
|	按位或，作用是将两个二进制形式操作数进行逐位的或运算，当两个对应位有一个为1时，结果位为1，否则为0	a|b，结果为7
^	按位异或，作用是将两个二进制形式操作数进行逐位的异或运算，当两个对应位有一个为1，另一个为0时，结果位为1，否则结果位为0	a^b，结果为6

2.2.6 位运算符

```
num_one = 10
num_two = 11
result_one = num_one << 2          # num_one的二进制数按位左移两位
result_two = num_one >> 2          # num_one的二进制数按位右移两位
result_thr = num_one & 2           # num_one与2的二进制数进行按位与运算
result_fou = num_one | 2           # num_one与2的二进制数进行按位或运算
result_fiv = num_one ^ num_two     # num_one与num_two的二进制数进行按位异或运算
result_six = ~num_one              # num_one的二进制数进行按位取反运算
```

2.2.7 运算符优先级

- 掌握运算符优先级，能够在数值运算中正确使用运算符

2.2.7 运算符优先级

Python支持使用多个不同的运算符连接简单表达式，实现相对复杂的功能，为了避免含有多个运算符的表达式出现歧义，Python为每种运算符都设定了优先级。

2.2.7 运算符优先级

Python各种运算符的优先级由低到高依次如表所示。

运算符	说明
+=，-=，*=，/=，//=，%=，**=	算术并赋值
or	逻辑或
and	逻辑与
not	逻辑非
in，not in	成员测试
<，<=，>，>=，!=，==	比较运算符
|	按位或
^	按位异或
&	按位与
<<，>>	按位左移，按位右移
+，-	加法，减法
*，/，%	乘法，除法，取余
~	按位取反
**	指数

2.2.7 运算符优先级

默认情况下，运算符的优先级决定了复杂表达式中的哪个单一表达式先执行，但用户可使用圆括号改变表达式的执行顺序。通常圆括号中的表达式先执行。例如，表达式“3+4*5”，若想让加法先执行，可写为“(3+4)*5”。此外，若表达式嵌套了多层圆括号，则最内层圆括号中的表达式先执行。

2.3 经典实例

2.3.1 实例1：计算BMI指数

- 根据任务描述实现实例1：计算BMI指数

2.3.1 实例1：计算BMI指数

任务描述

BMI指数即身体质量指数，是目前国际常用的衡量人体胖瘦程度以及是否健康的一个标准。BMI指数的计算公式如下：

体质指数（BMI）= 体重（kg）÷ 身高（m）÷ 身高（m）

2.3.1 实例1：计算BMI指数

实现步骤

（1）接收用户输入的体重数据。

（2）接收用户输入的身高数据。

（3）按照BMI指数公式计算BMI值。

2.3.2 实例2：计算三角形面积

- 根据任务描述实现实例2：计算三角形面积

2.3.2 实例2：计算三角形面积

任务描述

已知三角形三条边的长度分别为x、y、z，此时若要计算这个三角形的面积，还不知道三角形的高。为了能计算三角形的面积，我们根据海伦公式先计算三角形的半周长为q，再计算三角形的面积。三角形半周长和三角形面积公式分别如下所示：

三角形半周长q = (x + y + z) / 2

三角形面积S = (q * (q - x) * (q - y) * (q - z)) ** 0.5

2.3.2 实例2：计算三角形面积

实现步骤

（1）接收用户输入的三角形三条边长。

（2）根据三条边长计算三角形半周长。

（3）根据半周长计算三角形的面积。

2.3.3 实例3：判断水仙花数

- 根据任务描述实现实例3：判断水仙花数

2.3.3 实例3：判断水仙花数

任务描述

水仙花数是一个3位数，它的每位数字的3次幂之和等于它本身，例如，13 + 53+ 33 = 153，153就是一个水仙花数。

2.3.3 实例3：判断水仙花数

实现步骤

（1）接收用户输入的三位数字。

（2）获取三位数的每个数字。

（3）根据水仙花数特点判断输入的三位数是否是水仙花数。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
https://d.book118.com/775232100114012012