CS Five

菜单 页面导航

2.1 引言

▶ INFO

在第一章中，我们主要讨论了计算过程和函数在程序设计中的作用。我们学习了怎么使用原始数据（数字）和原始操作（算术），怎么通过组合和控制形成复合函数，以及怎么通过给程序命名来创建函数抽象。我们还了解到：高阶函数使我们能够根据通用的计算方法进行操作和推理，从而增强了语言的功能。这就是编程的本质。

本章重点介绍数据：我们研究的技术使我们可以表示和操作许多不同领域的信息。由于互联网的爆炸式增长，所有人都可以在网上免费获取大量结构化信息，并且计算也可以应用于范围广泛的不同问题。有效使用内置数据类型和用户定义的数据类型是数据处理型应用（data processing applications）的基础。

2.1.1 原始数据类型

Python 中的每个值都有一个类（class）来确定它的类型。拥有相同类的值，行为也相同。例如，整数 1 和 2 都是 `int` 类的实例，我们就可以使用相似的方法进行处理。例如，它们都可以取反或与另一个整数相加。内置的 `type` 函数允许我们检查任何值的类。

py

```
>>> type(2)
<class 'int'>
```

到目前为止，我们使用的值都是 Python 语言中内置的少量的原始数据类型的实例。原始数据类型具有以下属性：

1. 有一些可以求解为原始数据类型的表达式，被称为字面量（literals）。
2. 有用于操作原始类型值的内置函数和操作符。

`int` 类是用于表示整数的原始数据类型。整数字面量（相邻的数字序列）会求解为 `int` 值，并且数学运算符可以操作这种值。

py

```
>>> 12 + 3000000000000000000000000
3000000000000000000000012
```

Python 包含三种原始数字类型：整数（ `int` ）、浮点数（ `float` ）和复数（ `complex` ）。

```py
>>> type(1.5)
<class 'float'>
>>> type(1+1j)
<class 'complex'>
```

浮点数：“Float”这个名字来源于 Python 和许多其他编程语言中对实数的表示方式：就是“具有浮动的小数点”的值。虽然关于数字如何表示的细节不是本文的主题，但了解 `int` 和 `float` 对象之间的一些区别是很重要的。特别是， `int` 对象可以精确地表示整数，没有任何近似处理或大小限制。另一方面， `float` 对象可以表示很大范围内的小数，但并不是所有的数字都能被精确表示，它有最小值和最大值之分。因此， `float` 值应被视为真实值的近似值，它们只能保证有限的精度，组合 `float` 值可能会导致近似误差；如果不进行近似处理，下面两个表达式的计算结果均为 7。

> 译者注：不同于其他的编程语言，Python3 中的 `int` 值是无界的，也就是说它可以存储任意大小的数，具体可以查看 PEP 237。

```py
>>> 7 / 3 * 3
7.0
>>> 1 / 3 * 7 * 3
6.999999999999999
```

尽管上式是 `int` 值的组合，但一个 `int` 值除以另一个 `int` 值，却会得到一个 `float` 值：一个截断的有限近似值，相当于两个整数相除的实际比值。

```py
>>> type(1/3)
<class 'float'>
>>> 1/3
0.3333333333333333
```

当我们使用等式测试时，这种近似就会出现问题。

```py
>>> 1/3 == 0.33333333333333312345  # 请注意浮点数近似
True
```

`int` 和 `float` 类之间的细微差别对编写程序有着广泛的影响，因此，这也是程序员必须熟记的细节。幸运的是，原始数据类型的数量很少，这无疑减少了精通一门编程语言所需的记忆量。此外，这些细节在许多编程语言中都是一致的，它们会由 IEEE 754 浮点标准 等社区指南强制要求执行。

非数值类型（Non-numeric types）：值可以表示许多其他类型的数据，比如声音、图像、位置、网址、网络连接等等。它们中间的少数可以用原始数据类型表示，例如用于值 `True` 和 `False` 的 `bool` 类，其他大多数值的类型必须由程序员使用我们将在本章中学习到的组合和抽象方法来定义。

下面几节将介绍更多 Python 的原始数据类型，并将重点介绍它们在创建数据抽象方面所起到的作用。那些对更多细节感兴趣的人，可以查看在线书籍 Dive Into Python 3 中的一个关于 原始数据类型 的章节，它给出了所有 Python 的原始数据类型以及如何操作它们的实用概述，包括大量的使用示例和实用技巧。