

## 3.2 基本数据类型

当谈及计算机中的数据时，我们会想到文本、图片、视频、语音、3D 模型等各种形式。尽管这些数据的组织形式各异，但它们都由各种基本数据类型构成。

**基本数据类型是 CPU 可以直接进行运算的类型**，在算法中直接被使用，主要包括以下几种。

- 整数类型 `byte`、`short`、`int`、`long`。
- 浮点数类型 `float`、`double`，用于表示小数。
- 字符类型 `char`，用于表示各种语言的字母、标点符号甚至表情符号等。
- 布尔类型 `bool`，用于表示“是”与“否”判断。

**基本数据类型以二进制的形式存储在计算机中**。一个二进制位即为 1 比特。在绝大多数现代操作系统中，1 字节 (byte) 由 8 比特 (bit) 组成。

基本数据类型的取值范围取决于其占用的空间大小。下面以 Java 为例。

- 整数类型 `byte` 占用 1 字节 = 8 比特，可以表示  $2^8$  个数字。
- 整数类型 `int` 占用 4 字节 = 32 比特，可以表示  $2^{32}$  个数字。

表 3-1 列举了 Java 中各种基本数据类型的占用空间、取值范围和默认值。此表格无须死记硬背，大致理解即可，需要时可以通过查表来回忆。

表 3-1 基本数据类型的占用空间和取值范围

类型	符号	占用空间	最小值	最大值	默认值
整数	<code>byte</code>	1 字节	$-2^7$ (-128)	$2^7 - 1$ (127)	0
	<code>short</code>	2 字节	$-2^{15}$	$2^{15} - 1$	0
	<code>int</code>	4 字节	$-2^{31}$	$2^{31} - 1$	0
	<code>long</code>	8 字节	$-2^{63}$	$2^{63} - 1$	0
浮点数	<code>float</code>	4 字节	$1.175 \times 10^{-38}$	$3.403 \times 10^{38}$	0.0f

类型	符号	占用空间	最小值	最大值	默认值
	double	8 字节	$2.225 \times 10^{-308}$	$1.798 \times 10^{308}$	0.0
字符	char	2 字节	0	$2^{16} - 1$	0
布尔	bool	1 字节	false	true	false

请注意，表 3-1 针对的是 Java 的基本数据类型的情况。每种编程语言都有各自的数据类型定义，它们的占用空间、取值范围和默认值可能会有所不同。

- 在 Python 中，整数类型 `int` 可以是任意大小，只受限于可用内存；浮点数 `float` 是双精度 64 位；没有 `char` 类型，单个字符实际上是长度为 1 的字符串 `str`。
- C 和 C++ 未明确规定基本数据类型的大小，而因实现和平台各异。表 3-1 遵循 LP64 数据模型，其用于包括 Linux 和 macOS 在内的 Unix 64 位操作系统。
- 字符 `char` 的大小在 C 和 C++ 中为 1 字节，在大多数编程语言中取决于特定的字符编码方法，详见“字符编码”章节。
- 即使表示布尔量仅需 1 位（0 或 1），它在内存中通常也存储为 1 字节。这是因为现代计算机 CPU 通常将 1 字节作为最小寻址内存单元。

那么，基本数据类型与数据结构之间有什么联系呢？我们知道，数据结构是在计算机中组织与存储数据的方式。这句话的主语是“结构”而非“数据”。

如果想表示“一排数字”，我们自然会想到使用数组。这是因为数组的线性结构可以表示数字的相邻关系和顺序关系，但至于存储的内容是整数 `int`、小数 `float` 还是字符 `char`，则与“数据结构”无关。

换句话说，基本数据类型提供了数据的“内容类型”，而数据结构提供了数据的“组织方式”。例如以下代码，我们用相同的数据结构（数组）来存储与表示不同的基本数据类型，包括 `int`、`float`、`char`、`bool` 等。

Python

```
# 使用多种基本数据类型来初始化数组
numbers: list[int] = [0] * 5
decimals: list[float] = [0.0] * 5
# Python 的字符实际上是长度为 1 的字符串
characters: list[str] = ['0'] * 5
bools: list[bool] = [False] * 5
# Python 的列表可以自由存储各种基本数据类型和对象引用
data = [0, 0.0, 'a', False, ListNode(0)]
```