

- 全部开发者教程
- 深度剖析C语言知识体系
- 第1章 导学
- 1-1 引论
- 第2章 C语言必备基础
- 2-1 环境搭建
- 2-2 第一行C语言代码
- 2-3 程序调试方法
- 2-4 C语言的代码风格
- 第3章 C语言快速入门
- 【加餐】背景知识
- 3-1 数据类型之int类型
- 3-2 数据类型之char类型
- 3-3 数据类型之浮点类型
- 3-4 变量
- 3-5 常量
- 3-6 运算符-1
- 3-7 运算符-2
- 3-8 条件分支语句
- 3-9 循环语句
- 第4章 函数与程序结构
- 4-1 函数基础

数据类型之 char 类型

窄字符类型 char

字符类型 char，就是用来存储字符的，例如字母，空格，换行符这些都是字符。

char 类型本质上也是整型，C 语言当中的 char 类型占用 1 个字节，因此其存储的值的范围是 0 ~ 255，或者 -128 ~ 127，取决于 C 语言的具体实现，C 标准允许使用 unsigned 和 signed 来限定 char，即无符号 char 类型和有符号 char 类型，如果明确符号类型，那么对应的类型的取值范围的定义与其他整型的行为一致。

不过，作为字符使用时，我们无需关心 char 到底是否有符号，例如一个字符 `a`，我不管你在 char 类型的内存空间当中怎么存储，它都是 `a`，我需要判断它是不是这个字符以及要打印这个字符的时候都不需要理会它的整型表示。

那么问题来了，一个字符究竟是如何转换成整数并且存储于 char 类型的内存空间当中的呢？实际上，由于字符是人类认知当中的符号，计算机无法直接表示它们，因此我们人为的制定了一个映射表，把字符与整数一一映射起来，这个映射表就是字符集。

C 语言的 char 类型由于表示的范围有限，因此只能表示计算机最基本的字符，选用的字符集是鼎鼎有名的 ASCII 码，全称是 American Standard Code for Information Interchange，美国信息交换标准代码，它定义了常见的英文字符、标点符号以及一些空白字符等等，总共 127 个字符，因此只需要 7个bit就可以表示。

在 ASCII 码字符集中，`a` 的值是 `97`，空格的值是 `32`，字符 `0` 的值是 `48`，等等。千万注意，这个字符集就是一个映射关系，记住即可，一边是用于人类认知的字符，一边是用于计算机存储的整数。

这里再稍微提一下前面我们见过的换行符 `\n`，换行符我们无法直接打出来，类似的字符还有不少，例如水平制表符 `\t`，退格符 `\b` 等等，他们的写法非常规律，都是用 `\`（反斜杠）加一个字符组成，这叫转义字符。这么规定其实也是为了我们使用方便，记住即可。

转义字符	含义
<code>\a</code>	警报
<code>\b</code>	退格
<code>\f</code>	翻页
<code>\n</code>	换行
<code>\r</code>	回车
<code>\t</code>	水平制表符
<code>\v</code>	垂直制表符
<code>\</code>	反斜杠 (<code>\</code>)
<code>'</code>	单引号
<code>"</code>	双引号

这里稍微提示一下，`\r` 是回车，其实是把光标跳到所在行的行首的意思，与我们键盘的回车键还是有点儿区别的。

`\`可以理解，因为反斜杠一旦出现，就表示与后面的字符连起来表示转义字符，因此如果我们真的只是想要一个反斜杠这样的字符，那么就要敲两个反斜杠。

单引号和双引号又是怎么回事呢？原因其实也简单，源自于字符和字符串的字面量。前面我们打印 Hello, World! 的例子当中，用到了字符串：

```
<> 代码块
1 printf("Hello, World!\n");
```

为了告诉 C 语言的编译器我们想要打印的是 `Hello, World!\n`，我们用双引号引了起来，表示这是一个字符串，这种直接用双引号写出来的字符串叫做字符串字面量。而单个字符，就用单引号，例如：

```
<> 代码块
1 'a'
```

这表示是字符 `a`。既然如此，引号必须转义，不然 C 语言编译器还以为你要写字符和字符串的字面量呢。

除了直接写出来的字符和转移字符，字符既然是整数，那么我们也可以直接用整数来表示字符，怎么写呢？

例如字符 `a`，我们说它的整型值对应于 `97`，那么用八进制表示就是 `\141`，用十六进制表示就是 `\x61`。

宽字符类型 wchar_t

大家很快就能发现 char 类型的问题，它能存储的字符实在有限。ASCII 字符集仅仅包含了英语字母以及一些符号，别说中日韩三国的文字，就连希腊字母、法语字母的表示都是个问题。因此，涵盖了几乎所有文字符号以及符号表情的字符集 Unicode 就应运而生了。Unicode 总共包含 111万个字符，显然 char 是无法存下这么多字符的，因此 C 语言当中提供了 wchar_t 这个类型，规定大小不得小于 1B。宽字符在涉及到非英文字符的文本处理时非常有用，目前来看，Windows 上占用 2B，Linux 和 macOS 上占用 4B。

wchar_t 与 char 不同，char 当中直接存储的就是字节的原始内容，因此字符串又经常被称为 byte string（二进制字符串），而 wchar_t 当中存储的字符一般来说都是字符的 Unicode 码点，例如：

```
<> 代码块
1 L'中'
```

那么这个中字在内存当中以宽字符存在时，就是对应的 Unicode 码点：4e2d，也可以写作 `\u4e2d`，下面的写法与上面的等价：

```
<> 代码块
1 L'\u4e2d'
```

其中 `\u` 表示这是一个 Unicode 字符的码点。

注意，我们在宽字符的单引号前加了一个 L 表示这个字符比一般的字符要 Large。如果用普通的字符，我们需要使用字符串来容纳中这个字，例如：

```
<> 代码块
1 "中"
```

有关字符串的内容我们后面会逐步讲解。

索引目录

数据类型之 char 类型

窄字符类型 char

宽字符类型 wchar_t