

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

前端开发

JavaScript

JavaScript 入门教程
本教程带你从初级到高级全...

TypeScript 入门教程
这是一个很好的简单课程，...

Vue 入门教程
本教程带您从零开始学习 V...

Ajax 入门教程
本教程涵盖Ajax的实现原理...

ES6-10 入门教程
对比 ES5 进行学习 ES6+...

Yarn 入门教程
Yarn将相关基础知识和高级...

ECharts 入门教程
从零开始学习 ECharts，...

HTML & CSS

CSS3 入门教程
本课程从盒模型、文字、颜...

雪碧图入门教程
本文详细介绍了雪碧图的由...

移动端布局教程
由于移动互联网的兴起，移...

Html5 入门教程
最新一代的HTML标准，增...

Sass 入门教程
前端项目中 Sass 的快速入...

HTML 入门教程
从零讲解 HTML，掌握基础...

canvas 入门教程

uni-app 入门教程

是程序，就连操作系统本身也是一系列程序。我们常说的电脑软件其实就是程序。

本质上讲，程序就是人与计算机交流的工具，如果给你一个机房，你只要学会编写相应的控制程序，你就是这一支电脑军团的将军。

什么是 C 语言

世界上现存的编程语言可谓不计其数了。实际上现在自己创造一门编程语言的难度也已经被大大降低，因此编程爱好者有不少都喜欢自创一门编程语言来满足自己各方面的需求。

C 语言作为最早期的一批编程语言，由 Dennis Ritchie 为 UNIX 操作系统设计。这个操作系统大家可能并不像 Windows 那样熟悉，没关系，现在不少操作系统都与 UNIX 有着深厚的渊源，例如大家经常用到的 macOS 就是从 UNIX 衍生而来的，Linux 系统的设计思想也同样源自于 UNIX，所以大家如果是初次接触 UNIX 也没必要感觉害怕。

最早的 C 语言版本可以追溯到 1972 年。1978 年 Brian Kernighan 和 Dennis Ritchie 合著的《The C Programming Language》一书第一版则成为了 C 语言当时事实上的标准，这一版本的 C 语言也被称为经典 C（Classic C）或者 K&R C。后来在 1983 年美国国家标准协会（ANSI）组建了一个委员会开始制定一套 C 语言的标准，于 1989 年正式公布，是 C 语言的第一个正式的标准，称为 C89。

这个标准被 ISO 与 1990 年批准，因此又被称为 C90，实际上 C89 和 C90 是同一个标准，为避免混淆，我们在课程中始终以 ISO 标准来指代 C 语言的各个标准，即使用 C90 来指代 ANSI C 或者 C89。在往后，C 语言标准又有了 C99、C11、C17 等等新版本，改动都不大，加上 C++ 对于 C 语言的语法兼容大多停留在 C90 的标准上，因此 C 语言现如今最为流行的写法仍然以 C90 标准为主。

本课程会主要以 C90 为基础，对比 C99、C11 的新特性展开对 C 语言的讲解，这样做有以下几个好处：

- C90 是最基础的 C 标准，也是应用最为广泛的标准，涵盖了 C 语言当中最核心的功能和特性。学习 C90 足以应对绝大多数的 C 语言的应用场景。
- C++ 诞生于 C90 之前，对 C 语言的语法兼容实际上与 C90 更为接近，因此按照 C90 标准编写 C 程序对于 C++ 有着更好的兼容性。
- C90 标准下的 C 语言学会之后，再来学习 C99、C11 这些新标准中提出的一些改动会非常容易。

由于 C17 相较于 C11 没有引入新特性，同时目前生产环境中主流的编译器版本对 C17 也尚未提供支持，为降低课程学习的复杂度，我们在课程当中不会提及 C17 的改动，有兴趣的同学可以结合课程所讲的内容来自行查阅相关资料。

我们的课程当中主要使用 Visual Studio 2019 的编译器 MSVC。MSVC 在 C 语言标准的实现上比较任性，整体上讲基于 C++ 兼容 C90 的需求，MSVC 对 C90 做了支持，但它也对一些新标准做了部分支持。选择 MSVC 作为我们的主要的编译器，原因主要有：

- 我们的课程开发环境是 Windows。
- 大多数同学的开发环境是 Windows，广大计算机专业的在校生的学习环境也是 Visual C++ 6.0 或者 Visual Studio 2010、2012 这样的版本，使用 MSVC 更容易方便大家上手。
- Windows 环境下的 C 开发会面临很多 Linux、Unix 环境中不存在的问题，选择 MSVC + Windows 作为课程的开发环境也是为了方便在课程当中对此类问题进行说明，方便大家了解多平台的差异。
- Windows 上可以使用 MinGW 来获得 POSIX 标准的支持，我们也可以基于 MinGW 为同学们介绍应用最为广泛的 gcc 编译器的一些相关知识。
- Windows 上也可以安装 WSL 来获得一个真实的 Linux 子系统，我们可以很轻松的在 Windows 上对比多套环境多个平台之间的差异。

MSVC 并未完全对 C99 和 C11 标准进行支持，对于已经支持的部分我们会在课程当中指出，而对于不支持的部分，我们也会结合 MingW 来提供相应的说明。必要时，我们也会使用 WSL（即 Windows 上的 Linux 子系统，视频中使用的是 Ubuntu 18.04）来对比讲解 C 语言特性在不同平台上实现的差异。

可以说，这一个选择并不容易，但同学们却可以学到更多广大课程当中并未涉及的知识。

为什么要学习 C 语言

在很长的一段时间里，大学计算机专业的编程入门语言都是 C 语言，这并不是学校跟不上社会的节奏，而是自然选择的过程。主要的原因如下：

- C 语言与底层接近，代码与汇编语言几乎可以直接对应，能够帮助开发者直接了解程序如何驱动机器的运行。学习 C 语言之后，我们无需额外了解许多复杂的概念就能对程序最核心的作用做到了如指掌，这一点是具备现代各种复杂的编程语言特性的语言所难以企及的。也正因为如此，操作系统、底层驱动的开发到现在仍然以 C 语言为主，其他语言在性能上始终难以望其项背。
- C 语言的编程风格对现代编程语言的影响不可估量。从 C++ 到 Java，再到 PHP、Python，绝大多数的语言的语法体系都与 C 语言的风格如出一辙，业内也基本上存在这一的共识：有扎实的 C 基础的程序员在学习其他语言时也会非常容易就能上手。
- C 语言几乎可以认为 C++ 的子集。绝大多数情况下符合 C90 标准的 C 语言代码都也同时是合法的 C++ 代码，极少情况下只需要做些许修改就能正常运行。任何一本讲 C++ 的书籍都无法抛开 C 语言，因为任何一个 C++ 开发者都无法直接略过 C 语言。尽管现在企业当中使用纯 C 语言开发的应用程序相对较少，但使用 C++ 开发的应用服务、桌面程序、系统应用却广泛的存在于整个行业。
- 操作系统提供的系统调用为程序的开发提供了基础，而绝大多数系统调用都是 C 语言接口。因此 C 语言可以方便的直接调用系统 API 来实现自己的功能，可以尽可能的编写与系统紧密相关的程序。而其他语言例如 Java 则需要通过 Java 的 Native 方法调用 C 语言函数来转调系统 API。
- C 语言自身特性相对较少，也相对基础，各类课程也基本默认其受众具备 C 语言基础，例如操作系统，不懂 C 语言的开发者学习时，在系统调用层面就无法下手；再例如编译原理，现代语言的特性通常非常复杂，引入了太多概念和系统，不方便作为案例讲解，因此通常也会用 C 语言作为案例来讲解其实现方式。

还有一个很现实的问题，作为一个程序员，如果你只是想要快速掌握现在热门的技术混碗饭吃，可以直接去学习 Python，去学习 JavaScript 等等，你可能会马上照着教程写出个大家分分钟都能学会的 Demo，但随着从业人员越来越多，岗位需求也越来越，行业门槛越来越高，只懂皮毛，不懂深挖，不懂底层的程序员，必然要承受行业的各种冲击。

简单来讲，C 语言是程序员从业的基础。对于 C 语言的学习不应该像对待一些热门的内容，像机器学习入门，跨平台开发入门诸如此类一样抱着功利的心态，想要速成一门技术，对于基础技术的投资它的汇报是持续的，长期的。可以说，这一段 C 语言的学习经历将会在你后续持续对任何编程语言、操作系统、编译原理、系统架构设计等等的过程中发挥作用，助你在编程领域披荆斩棘，乘风破浪。

如何使用课程配套的资料

课程配套提供了全部源码和电子书方便大家学习。其中：

- 视频当中源码按照小节进行拆分，每一节对应的源码同学们从课程的 git 上下载后一目了然。课程源码是学习课程的重要参考资料，我们非常建议大家自己动手编写课程当中涉及的案例，也鼓励大家对于课程源码进行修改尝试，深入了解 C 语言的语法细节。如果大家不能充分利用课程的源码，学习效果将大打折扣。
- 电子书作为课程内容的补充，在章节安排上与视频尽量保持一致，方便快速查阅和复习巩固。视频与电子书结合学习效果更佳。

下一节 > 2-1 环境搭建

我要提出意见反馈

索引目录

导学

- 课程导学
- 课程所需基础
- 什么是程序
- 什么是 C 语言
- 为什么要学习 C 语言
- 如何使用课程配套的资料