

计算机科学学什么

- 学理论（计算的本质，自动化重复而枯燥的操作）
- 学系统（让现代的计算机跑起来需要些什么）
- 学应用（在现代的计算机上可以做什么）

程序 = 算法 + 数据结构 软件 = 程序 + 软件工程 软件公司 = 软件 + 商业模式

在中国，“计算机科学”或“计算机科学与技术”是工科（一级门类）下的二级专业。一般可细分为三级专业：

- [计算机软件](#)：围绕着软件开发、[软件工程](#)、[编程语言](#)等
- [计算机应用](#)：计算机的各种应用[算法](#)、技术。如：[模式识别](#)、[计算视觉](#)、[计算机图形学](#)等等。
- [计算机体系结构](#)：关于计算机或计算机信息系统的“硬件”。

C语言是

古老而常青的语言

1972年 Dennis Ritchie from Bell Lab

C语言开发的东西有：UNIX, Linux Kernel, Git, Nginx, Vim, GNU Project, OBS Studio ...

就连C语言的编译器都是C写的

现在还有很多人在使用C进行开发

C的历史

摘自维基百科的历史

C语言最早由[丹尼斯·里奇](#)（Dennis Ritchie）为了在[PDP-11](#)计算机上运行的[Unix](#)系统所设计出来的编程语言，第一次发展在1969年到1973年之间。

C源于[BCPL语言](#)，后者由[马丁·理查德](#)（Martin Richards）于1967年左右设计实现。BCPL是一门“无类型”的编程语言：它仅能操作一种数据类型，即[机器字](#)（machine word）。1970年，肯·汤普逊为运行在[PDP-7](#)上的首个Unix系统设计了一个精简版的BCPL，这个语言被称为[B语言](#)，它也是无类型的。

Unix最早运行在PDP-7上，是以[汇编语言](#)写成。在PDP-11出现后，丹尼斯·里奇与[肯·汤普逊](#)着手将[Unix](#)移植到PDP-11上，无类型的语言在PDP-11上愈发显得合适。PDP-11提供了多种不同规格大小的基本对象：一字节长的字符，两字节长的整型数以及四字节长的浮点数。B语言无法处理这些不同规格大小的对象，也没有提供单独的操作符去操作它们。

程序设计语言多且杂，难于以一言而蔽之，C语言可谓是大半程序设计语言的前辈，大多数程序设计语言都有着C的影子，C++ C# Java JavaScript PHP Python Ruby 等等

为什么还要学C

虽然现在都是8102年了，但是C语言仍然稳居TIOBE前三

C语言被发明出来就是用来开发系统的，C设计出来就是对拼汇编语言的，虽然还是没有汇编快，但是赶不上它更方便（这个也印证着大名鼎鼎的安迪比尔定律，因为硬件性能越来越强，汇编语言对C语言的那一点点优势就微不足道了，C又比汇编好写很多，所以大家纷纷转向C。现在也是如此，C#，Python这些语言比C肯定是慢不少，但是挡不住它们更好写，也可以说更抽象，所以现在用Python的人也不少，而且更多的人愿意使用这些更方便的编程语言来进行开发）

C语言是贴近硬件的语言，可以直接在C语言里插入汇编语言，因为贴近硬件，所以在后面的计算机组成等课程的学习中，能够更好的理解其中的一些思想。

C语言也是在不断发展的，从最早的K&R C，到ANSI C，到新的C99 C11，C语言也在与时俱进。

Hello World

测试开发环境是否配置正常，测试基本输入输出是否正常

```
1  #include <stdio.h>
2  /* C语言的Hello World */
3  int main() {
4      printf("Hello World!");
5      return 0;
6  }
```

```
1  # Python的Hello World
2  print("Hello World!")
```

Extra Part

每个人都应该去装个 GNU/Linux 来用一用，不管是装虚拟机还是实体机，我都推荐大家去试试。现代操作系统不少，MS Windows, macOS, Linux Distros, BSD, Solaris ...，并不仅仅只有Windows一种，优秀的CLI(Command Line Interface)交互方式或许更胜于GUI (Graphics User Interface)。Debian, Red Hat 等著名的 GNU/Linux 发行版占据着现代服务器的大半边天。我相信，仅仅是掌握一些基本的CLI界面的知识，也可以让你在接下来的计算机科学学习中少走不少弯路。