

[www.cc98.org /topic/4958232/1](http://www.cc98.org/topic/4958232/1)

## 自学CS路线、课程推荐 - CC98论坛

2-2 minutes

半年前我曾经写过一篇长文，主要是关于自己的面试、学习CS的经历和心得，[在这里](#)。这篇短文给出一个我认为比较合理的自学CS的路线，以供参考。其实也算是自己写的一些文章的汇总啦>\_<

个人推荐首先学好英语，同时跟美国名校有视频的课程，并完成作业。这套方法是偏学院风的，比较适合时间比较多的学生。

如何学好英语？可以参考[这篇文章](#)。

需要说明的是，我的学习方法是比较偏学院风的，不适合时间不多的同学；而且这些资料也有一些门槛，所以这套学习路径可能只适合一部分人吧。注意以下推荐的课程都是我认为很好的课程，每一块内容挑选1-2门对自己胃口的课程就足够足够了，然后看视频、看reading材料并完成作业。这些课程有些我写了点简单的介绍文章，一并附上；还有些课程的介绍文章会在未来有空的时候写。不过我还是强烈推荐自己找到课程主页，看一下课程大纲，且课程资源都在课程主页上。至于课程视频，在B站或者Youtube上都有。

下面是路线推荐：

1. 首先建议从计算机导论课程开始。推荐MIT的6.001或者Harvard的CS50或者Berkeley的CS61A都是可以的。关于CS61A，可以参考[这篇文章](#)
2. 之后建议学数据结构与算法。纯粹数据结构的话，Stanford的CS106系列是不错的，再包括算法的话MIT的6.006和6.046（进阶）都很好。学了之后也可以定期刷点leetcode。Berkeley的CS61A和CS61B当然也非常好，还有Coursera上的Princeton的课程，这些可以参考[这篇文章](#)。Stanford和UIUC也都有很好的进阶算法课程。
3. 操作系统：推荐CMU的15-213和Berkeley的CS162，都是有视频有lab的好课。进阶的话推荐6.828（小伙伴都说很好，我还没做）。
4. 组成原理、体系结构：MIT的6.004, CMU的15-213有一些，Berkeley的CS61C也非常好。有一个很有趣的游戏MHRD也不错，可以参考[这篇文章](#)。
5. MIT的6.null，实用工具介绍，可以参考[这篇文章](#)。
6. 计算机网络：Stanford的CS144，lab很棒，可以参考[这篇文章](#)。

基本上学了第一门导论课程之后，就会知道自己接下来想学/需要学什么了，也可以参考[MIT的CS本科的课表](#)。

我列出的这些课程的资源在网上都有，也基本都有视频。想要学好的话，作业一定要做。当然也有很多很好的课程这里没给出，大家可以自行发现，或者参考[github项目awesome-courses](#)（这个项目近期更新得不是很多了）

之后看走开发岗还是算法岗路线。走开发的话，可能还要学数据库(CMU 15-445)、分布式(MIT 6.824)、软件工程(MIT 6.031)等等；走算法的话，就从吴恩达的Coursera课程开始，然后Stanford的CS229或者CMU的10-701，然后Stanford的CS231n.....想走工程道路的话早日实习，有机会参加高质量的github项目开发；想走科研的话找靠谱老师跟着做，或者去发论文导向的靠谱的组实习；想去美国读phd的话找机会去参加暑研.....

关于编程部分，也推荐看看Norvig的这篇文章：[Teach Yourself Programming in Ten Years](#)

关于这么多课真的能自学下来吗，以及要多久，可以参考[这篇文章](#)。太长不读版：按我个人经验，视课程难度与个人基础，一门CS课一般需要150小时-300小时的有效学习时间，因此请自己计算。

学下来能拿到什么样的offer？我只是偷懒地完成了以上部分课程的一大部分内容，大概拿到了10多个offer，且基本都是硕士能拿到的很top的offer了。所以全部学下来的话应该至少能拿到比我多比我好的offer吧~~

另外发表一点小小的不成熟的看法。根据个人经验，这些课程每扎实掌握一门，以后从事程序员工作的话基本上就可以年薪提高x万。这样的话，哪怕职业生涯只有十年（35岁就被裁了？），学好一门课的收益也在十万以上了。所以这些不是课，都是钱啊~~想到这些，是不是上课会更有动力了？

祝大家自学CS之路顺利、开心~ Happy Learning, Happy Coding!