

# 第一课-科学上网

---

科学上网是为了探索更加优质的学习以及工作资源，与国际接轨。掌握了科学上网你可以：

1. 使用Github
2. 使用Google，避免使用Baidu
3. 浏览Youtube
4. 大语言模型以及AI Agent工具（OPENAI、CLAUDE等）

首先我们需要确保能登录Github，这个网站是全球所有程序员都需要使用的网站，所有的优质开源项目都会在其中发布。  
请查看视频以及视频评论区：[手把手教你流畅访问Github](#)

如果你可以成功登陆Github，恭喜你，你还需要确保自己能够完全的科学上网，请在github的搜索栏搜索 VPN 或 科学上网 等关键字查找资料完成科学上网。

## 第二课-认清大学、认清自己

---

这部分我推荐阅读[上海交通大学](#)生存手册，这本开源书籍由多名上海交通大学的学生编写而成，揭露出当今本科教育的落后，早点脱离高中的学生思维，为自己的人生负责。

每一篇我都希望刚入大学的人仔细研读，如果可以的话分享给自己的家长。  
链接：[上海交通大学学生生存手册](#)

## 第三课- CS基础知识

---

对于刚接触计算机的同学，并不知道git怎么用，也不知道vim、shell、命令行、ssh这些是什么，这些知识我们日常都会用到，但是太过于零锁，国内没有大学专门开课讲这些，但是我们必须知道。MIT开了相关的课程并有相关讲解，课程名叫 `The Missing Semester`，在Bilibili 或者 Youtube等视频网站搜索都能查看到。  
这里是后人对于课程内容进行的笔记以及资源整合：[The Missing Semester-zh](#)

## 第四课- 定向学习优质的CS课程

---

如果你不安于仅仅完成学校枯燥无味老掉牙的课程，你的追求不只是为了不挂科，可以在[CS 自学指南](#)中浏览查阅每个课程相关最优质的课程，每个课程都是真金白银！趁着自己还年轻有时间，坚持下来，学习有用的东西。

## 第五课-语言

---

问题：程序员最应该学习的语言是什么？Java？Golang？  
答案：No, it should be English.

📖 详细论证：[英语才是最好的计算机语言【让编程再次伟大#3】](#)

## 作者观点总结

---

### 1. 获取全面的资源和解决方案

- **寻找答案效率高**：英语社区提供海量可靠解决方案，避免中文搜索引擎的低质量内容

- **直接沟通**：通过 GitHub Issues 或 Discord 联系技术作者

◦效率考量：如果不懂英语而过度依赖中文社区，寻找正确答案的过程会变得“无比漫长、无比艰难，有时候还可能是无果而终”，因此学习英语能更高效地解决问题。

## 2. 紧跟技术发展的步伐

- **技术源头**：绝大部分技术创新来自英语社区

- **及时准确**：避免滞后和错误的翻译信息

◦深度理解的缺失：对一项技术要有深入的认识，需要足够覆盖度的答疑、面向细分领域的技术拓展和知识，以及丰富的场景应用经验，而中文社区在这些方面普遍缺乏。这导致中文开发者总是“落后别人几个拍”甚至“被远远地抛在后面”。例如，GraphQL发布八年后，中文社区仍处于“启蒙阶段”，而英语社区已形成成熟生态。

◦技术的演进：即使是成熟的编程语言和技术（如 Python 2到3的迁移、C++从8到11的演变），也在不断进化。程序员必须跟上这些变化，才能保持竞争力，而这些变化主要通过英语文档和讨论来体现。

## 3. 促进职业发展

- **打破工作循环**：避免陷入低级代码编写的死循环

◦吸取他人经验：打破这一循环的最佳方法是“吸取别人的经验”。领导更看重你解决问题的能力，而非方法来源，会因此赋予你更大的责任。

◦高质量的内容：英语社区的开发者大会演讲和资深技术博主提供了大量有价值的“干货”经验分享。

◦中文社区的差距：国内的技术大会在数量、题材、范围、内容质量和演讲水平上与英语社区存在巨大差距，许多打着开发者大会旗号的活动实际上是在宣传产品和服务，而非讨论技术问题。中文技术博主（如B站上的计算机分区）也存在很大的进步空间，多集中于教程、热点或吸引眼球的内容，缺乏深度技术分析。

# 第六课-刷题

---

力扣 (LeetCode) 是全球著名的刷题网站，所有人在找工作前都会刷题。

labuladong算法笔记是针对题目各种算法进行的总结，可以搭配算法笔记进行刷题