零基础入门（一）人人都可自学

这个零基础，指的是只要你用过电脑，会开关机，英语有最基本的阅读能力。我曾用本文所讲的方法实战教学，让我侄子5个月整，从零基础到入职找到工作。总共只面试了2次。没报培训班，我每天只给他讲2个小时，白天自己学习，加上偶尔的答疑。

侄子资质普通，大学专科，之前连计算机文件，环境变量这些概念都不甚了了，更不要说什么数据结构，计算机原理等等。为什么可以做得到？以下是主要原因：

**主次清楚，内容适当**。相比于大学计算机专业4年的学习，想要速成，时间非常宝贵，必须精挑细选，掌握最核心的概念和技能。

**学习顺序方法安排得当，集中精力攻克重点**。内容有主次，学习方法更不能平铺直叙，该重点的，需要反复练习和反思，否则，一定要有目的地放松要求，节约时间和大脑的注意力。

**及时反馈，及时解惑**。自学并不比培训的学得效果差，但是关键在于疑惑能得到及时解决。这样就不会停滞不前，而是不断在主动思考的过程。

当然，侄子自身持之以恒的学习非常关键。这一点，无论什么形式的学习都一样。

具体来说，我给他选择的是后端开发，语言选Java。这样的选择对跨专业入门来说，其实挑战不小。**对入门来说**，后端开发所需的技能相对前端来说更繁杂，而且没有界面可见，更加抽象。但是他了解之后，决定挑战，觉得前端开发不过瘾，我就让他试试。所以这次先说，对这样的目标，怎么自学达成。其他的方向，以后再说。

自学所需要的全部重点：学习内容，学习计划，学习锦囊。

1. 学习内容  
   Java后端开发所需掌握的知识，相信你在网上可以搜到很多。但是如果你把它当作零基础自学的目标，可能就失之毫厘谬以千里了。为什么？因为零基础，你的目标是入门，然后找工作。网上的知识点一般都会往大了去，往全了去。但是你要利用好你宝贵的时间，必须取舍。

这是我取舍的结果，按这个范围去学，既不浪费时间，又可保证掌握最核心的部分。后续文章，会详细讲解其中内容，这里我总结出哪些需要重点掌握，哪些是可以熟悉一遍的，哪些又是可以舍弃的。

**计算机理论核心基础**：

这部分极难把握，又极其重要。我挑选的重点是2进制，计算机组成核心原理，操作系统多任务时间片，内存访问模式，方法调用栈，数据结构，计算机网络概念等。大学里这些已经构成了好多门课，显然不可能在几个月的时间速成。所以这里有大量的舍弃，还有精心设计的案例学习，才能掌握最核心概念而不至于发散太多。教学这部分最难的，在于取舍和教学设计。市面上找不到这样的教材。

**Java语法基础**：

这部分是整个过程的重点。学好了这部分，就相当于学会了游泳，可以到大海中遨游。不管是JDK也好，还是SSM套路也好，都基于非常扎实熟练的Java语法。所以也要有取舍，比如内部类，匿名类，静态块，Java8的新语法Lambda等，需要刻意弱化，避免花费太多精力。而基础语法部分，看似平淡，却是要掌握到无法感知其存在的熟练程度，所谓日用而不知。

**JDK常用类**：

学习初期是无法区分什么是Java语法，什么是JDK类的。但是这部分必须要精心挑选一些类来熟悉。其他类，暂时先略过吧。这部分以后会有详细列表说明，初学者应该重点掌握的类。

**Java高级话题：**

多线程，线程池，GC概念，栈内存堆内存，类加载，反射等。这些内容对初学者比较硬核，其中关键在于计算机理论基础。所以是需要结合理论基础一起学习，效果最佳。

**数据库基础：**

数据库理论是没办法学一遍了，但是要从问题出发，理解数据库系统的作用，然后才是具体的SQL语法。这部分是后端开发重点，是除Java基础外最重要的基础。

**后端开发预备知识：**

如Maven, HTTP协议，计算机网络，浏览器工作原理，rest API测试工具，git，linux常用命令，docker等工具。要点是掌握核心概念，工具要在平时学习过程中熟悉起来，不是学而不用。

**SSM套路：**

从SpringBoot搭建项目开始，学习常用的Spring和SpringMVC注解，同时用ApiFox、PostMan等工具测试。MyBatis的学习占用精力较多，这个过程也是加强SQL的机会。

**工程实战：**

程序调试，日志配置，错误分析诊断，甚至日常git代码提交的要求，都按照正规工作要求来，可以达到实际工作经验的效果。甚至还有软件开发流程，人员组织，CI、CD概念等等。

**Spring核心原理：**

不必很多，从Bean，Bean工厂的概念开始，理解Spring对Bean的处理，能学到哪里算哪里。这部分原本不在计划内，因为基础已有了，学习这部分可以检验，加强基础，对自信心，简历，面试都有很大作用。

**常用中间件入门：**

这部分初学者最容易沉迷，喜欢各种入门。但这部分确实需要挑选几个来感受一下实际工程中的功能，是怎么通过组合一些开源组件来完成。同时感受一下，Java基础在此过程中的重要性。

1. 学习计划

有了内容，并不是挨着顺序一个一个来。

计划的核心在于：按照知识的依赖关系安排学习顺序；安排好重点内容，不达标不前进；做好每日学习清单。

Java语法，JDK常用类，SSM套路，SQL，错误诊断，内存模型，git操作等。

另一个诀窍是，整个过程是有反复，有总结的。比如Java基础，第一遍下来后，不可能达到要求。那么就会在适当的时候，进行第二遍学习，换一本书，换一个学习的资料，这种反复的学习方法，对背单词也适用。以后会详细介绍。

限于篇幅，学习计划以后会专门详解，这里先说这些。

1. 学习锦囊

锦囊中有妙计，是到山穷水尽时，救急用的。

一般说来，容易有以下关口：

**虽然网上书籍、文章、视频众多，但不知道怎么挑选合适自己的。**

因为没有成系统地介绍自学入门的资料，要做好这样的资料本身就很难。要么专注于SSM套路，要么专注于多线程开发，或者其他专题。

**我的锦囊：**加入我的星球，得到指导和独家原创系统性学习的资料。一本书，我给你一个导读计划，就能节约你一大半的时间了。

**初学时，无人指导，工具的使用几十个菜单功能就足以让人产生放弃的念头。**

工具和知识一样，本应该有主次之分，也就是有些功能是每天要用几十遍，有些功能，你可能一辈子都没用过。

**我的锦囊：**我的星球会专门介绍常用工具的使用技巧，让你不至于迷失在众多菜单项之中。

**关键概念难以理解，导致后续知识无法继续学习。**

知识体系往往构建在几个核心概念之上，掌握了核心概念，就等于抓住了牛鼻子，就是从0到1的突破。剩下的事情是从1 到 100，只不过是不断重复，扩展记忆而已。

**我的锦囊：**我的专栏文章很大程度上，就是解决从0到1的问题。从1到100，是学习者自己修行要解决的。

**知识间依赖关系无人指导，很难自学。**

如果不知道学习顺序，很难顺利地保持学习进步的正反馈。

**我的锦囊：**我会有详细的学习计划，让你可以按部就班，每天都有进步的快乐。

如果你看了这篇文章，当然不足以完全开始自学。但是你对自学这件事，应该有了整体感觉，而不是无头苍蝇一般了。

后续我会持续更新整个自学过程所需的关键。