1.学习python基本语法。

2.上w3cschool这个网站，学习HTML/CSS下的html、xml、webservice三个教程。

3. 然后下一个python的requests库学习写最简单的网络爬虫。知乎上爬虫教程一大堆。

3是第一个里程碑，学写简单爬虫一方面有一定的成就感，一方面又知道了接口到底是怎么回事。同时还学到了怎么解析一个页面，

4.学习python的测试框架unittest，知道怎样用unittest和python的mock模块写一个小单元测试。

5.把3和4结合起来，你掌握http自动化接口测试。

6.学selenium的库和页面对象模式

7.把2、4、5、6结合起来，你应该能写既支持web测试又支持接口测试的自动化测试脚本了。

8.学robotframework，你可以把自动化测试变成关键字驱动和数据驱动的了。

9.学python的高级一点的语法。如装饰器、线程进程协程。你可以让测试并行执行，并自动记录测试步骤到log文件里了。

10.学jenkins，测试不再需要你手工去启动了。测试也可以分步式运行到多个环境上了。

11.学docker、git、gitlab等的简单使用，从此测试脚本不用再人肉更新、测试环境也不用人工搭建。这里也要学linux的简单使用，

12.回到w3cshool，学习javascript、ajax、jquery、bootstrap。至此你可以写50%以上简单网站的前端了。

13.学习python的flask库，学mysql或mongodb是怎样和flask一起用的。至此你可以写简单网站的后台了。你还可以快速开发webservice接口了。

14.综合12和13，你可以开发一些测试管理工具了。比如写一个管理很多jenkins master的ci调度平台。也可以写一个提供统一样式的测试报告的web展示平台。写一个监控所有测试情况的看板。写帮你生成测试文档的脚本。写测试环境的管理工具。

并且14是一个里程碑，你可以理解开发人员的一些思路了，比如为什么开发人员老是不愿意好好写单元测试、老是说这不是bug、老是说几点干完但却拖到半夜还做不完，特别是你做的东西交给别人测的时候。你也可以理解web测试的接口测试、单元测试怎样做比较好，因为如果不好，你是不愿意用到自己写的网站工具上的，你会觉得浪费时间。老实说我从来没用selenium给自己写的网站做过测试，完全不想做，但接口测试和单元测试我还是很愿意做的。主要因为基于图形界面的自动化测试效率低。我当时组里有其他测试人员执意要给我这个网站做selenium，我劝阻不了，但最后界面需求不断修改，他也没坚持做下来，从这个工具网站的第二个大版本开始就放弃了。

15.学压测工具locust、jmeter等，7里写的框架可以支持压力测试了。同一套接口测试脚本，既做自动化测试又做压测了。注：现在主流是jmeter，而未来属于像locust这样的工具，它让测试人员写代码来描述性能测试。jmeter的控件和图形界面适合新手入门，但基于图形界面做的话效率低一点。其他gatling、grinder都在自身的设计理念上比jmeter先进。

16.想做app测试的话，学一个appium之类的框架。p s：我没有学这个，主要看项目方向是否需要。

17.补课时间，好好把计算机网络补起来，这个是真有用，我原单位用openstack搞云计算的时候，网络这块是最经常出问题的。网络我学得不好，就得不断求实验室的人帮忙。

算法题刷起来，平时没多少用但是面试经常考，数据库我说实话我毕业后就没怎么系统学过，好像我用得不多。

linux操作系统再补补，鸟哥私房菜看看完。很多人用了n年linux连系统权限咋回事都不懂，这个有点无语的。

18.java补起来。没办法，用java的单位多。要找工作机会也多。现在流行java+python都要会。其实都差不多，举一反三。但能用python的地方我还是喜欢python。

19.不知道后面再怎么学了。我想再提高提高web开发能力。前端框架学一个，以后开发一些更漂亮更炫的前端页面。再学下什么消息中间件之类的，去面试曾经被这个的问题虐过。

我相信再怎么基础差，到了这里做做大部分单位的自动化测试是随便做做了。但是，恐怕你到这个时候也不愿意回去写那种常见的selenim脚本了吧。按我们单位的情况，原来写的脚本是robot脚本加一些自定义库，和别的单位的selenium、appium本质上都类似，都很简单不用动脑筋。这时候我原单位里有的人就转去做devops，搞ci、cd。有的人去做自动化测试教练（其实就是写测试基础库和搞培训）、测试工具开发（偏web开发）。这几类我都试过后感觉对devops更有兴趣，当然我做的具体工作还是从自动化测试开始的。自动化测试的学习路线和devops可以很好地衔接起来。因为自动化测试方面的专家较少，而devops讲究的是自动化整个流程，不仅是测试，但自动化别的阶段和自动化测试阶段，目前还没有太大本质区别。

至于自动化测试架构师的职位，其实就看你单位怎么设置职位了。做技术的人总想着挂个架构师头衔，但其实这个架构师和开发的架构师差距还很大啊。到这里也不要沾沾自喜…可能只是你在这个公司资历熬的久。

----------------

回一下评论里的问题：

1.最快多久入门：

很快，

不一定要按顺序从1学下来，以最快加入项目组干活来看，像我现在单位，要求真的很低，掌握1、8、10、11就可以干活了。我原来组里带一个黑盒测试出身的人（三到五年经验）带到他能写出一个case，完成一个合格的脚本大概一两周吧。

但是，我不得不说，我也没办法把这些人带成高手。真正的入门我认为是要具备快速学习的能力开始。

就我在我原公司待过两个部门，我见到的具备快速学习能力的测试人员屈指可数，真的非常非常少，而且很多都是不再写代码的测试架构师。（敢情大部分人都是等别人学好了来教你？）

我的亲身经历就是，我学一个东西花两三天，用学的东西做一个东西花一个礼拜，优化到比较好用可能花个把月。

但我教别的测试人员接手这个东西要教一两个月？！要教好多遍？！给了官方教程的链接告诉我看不懂或看不完？！这就尴尬了。我想去做新的东西，而不是被自己做的东西绑住..变成唯一的维护人员...

所以我写的这么多点的学习路线，真正目的是希望读者在学习过程中培养快速学习的能力。然后摆脱学习路线自行学习。而并不是说让你按部就班一条一条学下来。我自己也才学到14和15。

1.从开发转自动化测试

2.学习各种测试工具：

不一一列举了，前面已经有了。

一般主流工具都会就可以了。记住这些工具仅仅只是一些api调用，调用这些工具和你asp拖拽一个控件并没有本质区别。

3.加强对测试和相关理论与的理解，以下不分先后：

单元测试原理（包括mock）

接口测试原理（不就是调接口么，顶多加个和单元测试差不多的mock）

性能测试原理（微软有个web性能测试的电子书很好，性能测试原理是唯一一个需要看书学习的，其他几个都太简单不需要看书）

自动化测试原理（归根结底就是一些简单抽象简单封装和一些第三方库的调用）

分布式测试原理（最简单就是master+slave，你可以看看jenkins、selenium grid、jmeter都实现了不同的分布式）；

持续集成原理（都不好意思说这能算原理，掌握jenkins和他的一些常用插件就好了）；

配置管理原理（为啥要用git，为啥要分branch等等，怎么做一个项目的配置管理）；

数据分析原理（我学的ELK，可以用来做测试结果分析和可视化————》进一步你可以转型做专业的数据分析，告别有点无聊的自动化测试，不过我原单位选择了从外面招一个人来而我也不想转这个方向）

4.写点自己的工具

原理都懂了，写起工具来简直势不可挡。

特别是你如果搞的python，很多东西都可以自己写一写啊，都很容易实现啊。有那么多第三方库对吧。实现起来还特别快。

比如，找一个可视化方面的库，你可以轻松自定义测试结果报告，做出你想要的任何高大上的图。这些图表在性能测试的时候很有用，那么尝试给jmeter或者给locust实现一个更漂亮的图表吧。

然后你可能觉得比如jmeter的图表不能实时展示很不好用（jneter会建议你用命令行模式执行较高负载的测试，那么报告要测试跑完你才看得到），自己写一个，让他能实时展示出来。其实网上已经有了，你都不用写直接看一下都行。

但应该要有能力写。

特别注意很多人把3和4颠倒了，写出来的工具根本就很不好用。遇到过这些人遗留的历史工具n多次了。神烦。他反正自己写出来自己不用的，工具质量低也没有领导会关注。

5.告别自动化测试

还是要和前面直接搞自动化测试的人一样补各种基础知识。以及掌握至少两个主流的语言（比如python和java）。以及具备从前端到后台的基本开发能力。至此你可以做大部分公司的自动化测试、测试开发相关工作。

别看前面12345好像很费时间，其实大部分人只是早年没有方向，没有好平台，所以才费时间。我大部分时间都在摸索这条路上。真的学得好的话也就三四年时间你就到这里了吧。到这步之后的区别就是开发能力了。

测试门槛低，天花板也低，这时你就快到天花板了吧，按我个人浅见，就是到了。我也很难想象，哪里还能找到一个月基本工资三万四万的测试工作（通过延长每天工作时间倒是可能拿到这个数）。但对开发来说，还是有这种基本工资的。（股票另算）

因为，自动化测试领域无大牛。

自动化测试的技术栈不够深。你看阿里有运维到p10研究员，却没有测试到p10。从侧面反映出来测试人员目前技术较低。招聘网站上你看不到3w4w以上月薪的测试岗位。但却有开发、技术专家、软件架构师岗位能给这种月薪。

或者你可以靠股票之类增加收入（一线大厂工资加股票一年几十万甚至上百万）。想做管理的可以赶紧转了，想继续技术路线的，是时候找一个领域来钻研，做领域大牛了。实际上对大多数人来说都已经有点晚了。但我们现在还是可以搭上devops的热潮，做devops，学云计算，在技术上更进一步做大牛的。

----我目前学习devops中...开发能力还是不上不下半吊子...上述建议不保证全对。话说回来，做自动化测试并不需要你的开发能力真的有多厉害。能写出简单清晰易维护的代码就合格啦。