

# IT 行业都有哪些职位，初学者（0 基础，新人）该如何选择，才能够快速进入这个行业？

作者: [ptteng](#)

原文链接: <https://ld246.com/article/1498187318531>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>互联网行业的薪资水准相对较高，刚入行一个月，半年，或者一年超过其他行业薪资很正常。<b>

</b>那么，互联网行业究竟有哪些职位呢，又分别适合哪些传统行业转型？</p>

<p>1.产品<br>

2.UI<br>

3.CSS<br>

4.JS<br>

5.后端 (Java/php/python) <br>

6.DBA(mysql/oracle)<br>

7.运维 (OP) <br>

8.测试 (QA) <br>

9.算法 (分类/聚类/关系抽取/实体识别) <br>

10.搜索 (Lucene/Solr/elasticSearch) <br>

11.大数据工程师 (Hadoop) <br>

12.Android<br>

13.IOS<br>

14.运营</p>

<p>然后等下我会分别介绍每个职业的工作内容，发展前景，入门门槛，适合哪些职业转行，成长周期，薪水范围等。</p>

<ul>

<li>薪水都以北京为例</li>

<li>运营和市场都有更细致的划分，然而我并不太了解了</li>

<li>其他的偏门的</li>

</ul>

<p>一.产品<br>

=====我是求某个姑娘快原谅我的的分割线=====</p>

<p>1 工作内容：了解用户需求，做竞品调研，画产品原型，写产品文档，讲解产品需求，测试产品，收集用户反馈，苦练金刚罩以防止程序员拿刀砍。</p>

<p>2 需要技能：PPT, Word, Axure,XP,MVP, 行业知识，沟通</p>

<p>PPT 是最容易被忽视的技能，很多时候很多人都把产品经理=画原型，如果你这么想，再做五年产品经理也是一个画原型的。我在修真院的时候，给修真院的弟子讲过很多次，产品经理更多的是理一个行业，熟悉这个行业的用户，知道这个行业有哪些没有满足的需求，寻找解决方案，对比行业，到自己的切入点，最终落地成：哪些功能要做，哪些功能不要做。</p>

<p>这些都是要在 PPT 里写清楚的。所以，PPT 并不是简简单单的一个工具，而且是用这个工具把对这个行业的了解和分析和解决方案展示出来。</p>

<p>XP 是敏捷开发，不会敏捷开发的产品经理很难说会是一个好的产品经理（我很偏激，所以接受了可以直接无视）。敏捷开发对于产品经理来讲，是逼着产品经理去用 Story 的方式去思考，去考虑己每一个 Story 的价值是什么，所有的 Story 优先级是什么。</p>

<p>MVP 是最小化验证产品的缩写，这在精益创业里是很重要的概念。也就是说，你臆想了一个产品，他有了什么样的功能，但是！请注意，我们认为所有没有经过用户验证的功能都是在扯淡。所以，其实是设计了一个扯淡的功能然后交给一堆人去开发，再交给人去运营，直到一个月或者是三个月以才能拿到反馈结果。我们等不了，所以你必须得在程序员开始启动之前，先想办法验证自己要解决的问题是存在的，要解决的需要是用户确实需要的---你有无数的办法可以这么做，真的。好比是，我做 IT 修真院，并没有先直接做线上的网站，而是直接去贴吧发帖，用 QQ 群，在论坛直接做，去观察这些弟的行为，然后去修正自己的产品设计，我发现了他们需要日报，需要点评，需要有班级的概念，需要学号，需要有首席大弟子等等等等这些。然后我会把这些需要分析整理之后加入到网站的设计里去，就是 MVP 的价值--当年穷疯了每个决策都慎重的时候，就会特别喜欢这种方式，土豪除外，只会浪费时间而已。</p>

<p>Word 是产品需求文档，我自己不太喜欢 Word 文档，更喜欢 XP 的 Story，不过会一点也没有系。</p>

<p>行业知识是产品经理最重要最重要的技能，没有之一，你可以不懂 PPT，不会 Axure，不会 Word，也不会什么 XP 和 MVP，但是绝对不能不懂一个行业。某种程度上来讲，产品经理拥有的其实是速了解一个行业的能力。现在的互联网已经和以前的不一样了。以前的产品经理懂点儿交互，知道什么是表单，知道点 IT 术语就能混日子了，现在必须要懂一个行业，地产，金融，汽车，教育，餐饮，

疗等等各种行业，你都需要了解这个行业的模式，如果你不懂，你就是一个画原型的人。

沟通也是产品经理的必备技能，出门不带沟通技能，一定会被程序员砍。一方面是要跟用户沟通一方面是要跟程序员沟通（这是世界上最难沟通的物种，没有之一），一方面要跟领导沟通（其实就想办法拍领导马屁然后哄着他们听自己的话）。

3 发展前景

不用说，产品经理绝对是一个炙手可热的职业，而且，TNND 的几乎是零门槛，这表示，是个都可以说，我想成为一个产品经理。

然而，好的产品经理很多么？并不多，所以，好像围棋一样的，入门容易，学精难。

而且，似乎并没有一个系统的教程（这真是一个很难让人相信的事情啊，这么一种职业，没有大的知识培养，甚至都很少有标准化的职业化培训，全靠业界手把手培养和自发的交流，就能够拿着无数 DS 程序员根本就想像不到的薪水--产品经理真的跟 DS 有点儿不搭边的，当然，我大修真还是在努力做产品人员的培养的）。

产品经理的发展前景是很赞的，只是越来越体系化了，而且很容易出人头地。

一般来说：

0<del>6 个月：4K</del>8K

6<del>12 个月：6K</del>12K

1 年 <del>3 年：10K</del>24K

3 年 <del>5 年：20K</del>40K

5 年~：30K~

三个月到半年左右就可以算是迈入这个行业啦，如果遇到 P2P 这种对于专业属性要求比较高的随便在一家公司混点经验，跳到另外一家新的公司就是一个产品总监：因为并没有其他人又了解金融了解互联网。

成长路径：

产品助理-产品经理-高级产品经理-产品总监-CEO

大致这个样子。

4.入门门槛

无（甚至你可以不认识字）

5.哪些行业适合做产品经理

IT 界：程序员，UI，QA，运营，市场

其他界：无限制

PS：想了想，还真是没见过身边有做运维的转产品经理的。

6.职业限制

很多产品经理都很怕跟程序员沟通。而且经常会苦恼，觉得自己设计的功能，程序员做不了，做出来，做的太慢，做的太差。惟独不会觉得自己设计的太烂。

所以，很多产品经理都会认为不同技术是自己的一个弱项，我曾经整理过一个产品经理需要了解技术术语，有需要直接看楼下的广告。

7.去哪里学，怎么成长

还用问，自然是跟着大修真院最好。

如果 IT 修真院你看不上，可以去一些有产品经理培训的地方去，虽然我肯定觉得自家的好。

如果这些产品经理培训的差不多了，有了一个基本概念了，找个产品助理或者是实习的职位去吧。

会画点儿原型图，有个线上的作品，就能找工作（虽然我很不喜欢这样的产品经理）。

然后有机会就去大公司了解一下各个部门是如何扯皮的，镀镀金，Over 了。

二 UI

====我是累死了的分割线=====

马丹，首先写软文，又要写成干货，又要带广告真心不是一件容易的事儿。我在之前[https://ld246.com/forward?goto=http%3A%2F%2Fwww.zhihu.com%2Fquestion%2F36823157%2Fanwer%2F69156567%3Fgroup\\_id%3D643447526423064576WEB](https://ld246.com/forward?goto=http%3A%2F%2Fwww.zhihu.com%2Fquestion%2F36823157%2Fanwer%2F69156567%3Fgroup_id%3D643447526423064576WEB) 的贴子中，写整整一个星期才写出来。我写小说都是分分钟好么。

可是没办法。一个是确实需要跟大家介绍一下每个职业的比较和特点，一个是也确实需要告诉更的人，写代码真的不是看视频看会的。

好了我接着说。

1 工作内容：收到产品原型，给原型上色，偶尔会自作主张调整下原型的位置，出不同的风格给

板和客户选，然后听他们的意见给出一个自己极不喜欢的风格，最好给 Android, IOS 或者是 CSS 好标注，还有的需要直接帮他们切好图，最后要练出来像素眼，看看这些不靠谱的程序员们有没有上色或者是有偏差。</p>

<p>2 需要技能: PS, Illustrator, Sketch, 耐性, 找素材</p>

<p>PS 是最基础的技能。各种图层通道蒙版神马的细节的东西就不用说了，很多时候很多公司都愿出所谓的高保真图。什么叫高保真图，简单说就是不折磨产品折磨 UI 的破玩意儿。就是把所有的图画出来然后对着 UI 图来改，偶尔还会要求做出来网页能够点，对于这点我不想再吐槽了，总之，PS 少不了的技能，基本上很多时候其他人都会以为你只需要用 PS 就能搞定一切。</p>

<p>然而 PS 始终只是一个工具而已，所以在这里，PS 代表的更多的是一种方案的选择。比如什么业用什么配色，怎么样对齐，怎么突出重点，PC 下应该是多宽，手机应该是多宽，等等等等。</p>

<p>ILLustrator 最近似乎是用的少了，我不太确定，但是做矢量图以前是就是靠他的。什么是位图什么是矢量图，自行百度就好。以及还有 CoreDraw，不知道现在是否常用。然而，慢慢的似乎 Sketch 流行起来了。</p>

<p>SKetch 和 Illustrator 和 CoreDraw 之间的差别我说不上来，我记得我的 UI 小姑娘跟我讲这个具很好用。</p>

<p>耐性必须是 UI 必备的技能，因为有的时候，老板跟你较真起来，一个星期不停的反复修改一个页是很正常的事儿。可能颜色要从红换成黑再换成绿。所以你必须要有耐性，除非你做的设计没人管你说了算。而且，UI 这东西，是个人都有自己的审美观。更不用说之前朋友圈的测试，人和人看到的色真心不一样。<br>

所以你必须要有耐性，这是很重要的。</p>

<p>找素材的能力也是必不可少的，你手上必须要备几个常用的素材网站，大部分的人都是在别人的材基础上改来改去。</p>

<p>说到这里，其实忍不住想加上一句。UI 设计师最重要的能力并非以上几种，而在于理解产品经理的需求--前提条件是产品经理很明确的说出来了而且讲清楚了需求。</p>

<p>比如说，一个页面的重要是什么？最想引人注目的内容是什么？应该配什么样的图？加上什么样元素能让网站的风格更醒目一些呢？</p>

<p>这里大部分其实都是原型设计师要做的事儿，我一直有一个观点，就是 UI 其实最应该做原型，有这样，才能让产品经理着重去考虑功能点，让设计完美传达产品的功能点。<br>

然而原型设计师这个角色慢慢的被消融了，已经变成了产品经理了。所以大部分产品经理都以为自己画原型的，然后大部分 UI 都会自作主张改掉原型，因为没有美感。所以通常你会发现美感说有了，有那么一点儿，可是总觉得会有一些不对。</p>

<p>我对他们的要求就是，先别管美不美，你先告诉我这张页面每个模块想要重点表达的内容是什么能不能让人分清主次。做到了这一点，再去考虑美感。</p>

<p>3 发展前景</p>

<p>UI 的发展前景是很赞的，在两到三年前，甚至都有人推动让 UI 来代替产品经理来驱动。某种程度上来说，UI 其实最好的发展方向就是 UE，也就是交互，然后就是所谓的用户体验优化师--坦白的说并未觉得有多少公司会真正的设置这个职位，因为太泛了。但是对于现在火热的 O2O 来说，无论是上还是线下，都希望提供一个极致的用户体验，所以我说不好。</p>

<p>但是大部分的 UI 都应该去朝着交互的方向发展，而不是去切页面。。。</p>

<p>UI 薪水其实跟产品经理在前期差不太多，中期差点。后期相差比较远，一般来说：</p>

<p>0<del>6 个月：4K</del>8K<br>

6<del>12 个月：6K</del>12K<br>

1 年 <del>3 年：10K</del>20K<br>

3 年 <del>5 年：16K</del>30K<br>

5 年~：24K~</p>

<p>UI 的成长时间大概是在两年左右，基本上 PC, IOS, Android, H5 全部都做过一些项目，设计几个 Logo, 做过一些专题和 Banner 图，才算是比较完美报。</p>

<p>然而想要做的更好一些就有点困难了，除非你的遇到了一家公司，对用户体验要求极致，然而那时候，并不仅仅是 UI 和 UE 相关的内容了。</p>

<p>成长路径：</p>

<p>实习 UI-初级 U-中级 UI-高级 UI (UE)</p>

<p>大致这个样子。</p>

<p>4.入门门槛</p>

<p>UI 对入门门槛要求还算是比较高的。首先你得有一台 Mac。。大部分的情况下你都需要 Mac。

br>

然后你要懂之前提到的常用软件，UI 的交互规范，配色等等等。</p>

<p>5.哪些行业适合做 UI</p>

<p>IT 界：无<br>

其他界：传统做 VI 的，就是平面设计的会转一些。跟着基本上就是小鲜肉了，刚毕业就从事这个行的会比较多一些。还有一些学美术的，会手绘的。</p>

<p>PS：想了想，从没怎么见过程序员转行做 UI 的。只见过 UI 人员会写代码的。</p>

<p>6.职业限制</p>

<p>UI 这个职业的限制在某种程度上就是不太清楚自己应该朝着 UE 发展。所以时间长了工作就会变单调，两年或者是三年之后原地踏步走没怎么有进步的人很多。像 Metro（扁平化）或者是 material esign 这样的风格一不熟悉二不了解三不想尝试的人还是挺多的。</p>

<p>很多人做了很久其实还是分不清产品经理想要在一个页面上表达什么--这有时候真的并不是 UI 错。</p>

<p>7.去哪里学，怎么成长</p>

<p>对于初学者来说，找个培训学校还真是比较好的选择。<br>

我大修真院暂时并未打算去培养 UI 人员，主要是没有时间也没有精力去做一个系统的教程，而且我自己自己在 UI 上也在不断的摸索。<br>

一般来说，最好的成长路线就是不断的去参考别人的例子，然后再多动动脑子，多看一些 UI 规范。<p>

<p>先想清楚自己要表达什么，然后想想用什么表达，再去查查怎么样能做到自己期望的结果。</p>

<p>三 CSS<br>

=====我是再也不要写软文的分割线=====

<p>没办法。自己拉的屎不能拉个开头就提裤子走。。。说起来这也是知乎挺好玩的地方，如果不是乎允许这种自问自答的方式，只要你说的是真正有价值的内容，并不拒绝你有做广告的嫌疑，恐怕我不会写这些东西，你们也看不到这个帖子。</p>

<p>以下内容是我从 UI 那边粘过来的，还没改呢，等我去沙发上喘口气歇会再写。中午吃的麻辣锅有点咸。</p>

<p>1 工作内容：产品设计好原型，UI 做出来了效果图，剩下的就是 CSS 工程师用代码把静态文件出来的。</p>

<p>所以 CSS 工程师和 OP 一样，都是可以在不用了解业务需求的情况下就能完成工作的唯一两个殊职业。<br>

一般来说，CSS 工程师并不存在，写 CSS 的人多少要写一些 JS 代码，然而确实也会有一些妹子真的是学不会 JS，有一些汉子真的就是写不出来 JS，所以我还是把他们两个拆开了。</p>

<p>CSS 工程师要考虑更多的就是兼容性，简单说，IE 内核和非 IE 内核（WebKit），IE10 以下和他。我是强烈建议新人不要去学 IE10 以下的 Hack 的写法的，让 IE10 以下的，不支持 Html5 的都死吧，表再来跟我说用户还在用 IE678 了。</p>

<p>当页面切完以后，另一个工作就是去帮助菜鸟后端和某些 JSER 把页面又重新套错了，什么是套面，可以看这个 <a href="https://ld246.com/forward?goto=http%3A%2F%2Fwww.zhihu.com%2Fpeople%2Fptteng%2Fanswers" target="\_blank" rel="nofollow ugc">http://www.zhihu.com/question/22689579/answer/69899644</a> 。</p>

<p>2 需要技能：<br>

环境【IDE（WEBStorm，Sublime，EditPlus），源码管理（SVN/Git），WEB 服务器(nginx)】<br>

基础【PS，域名，Html，Html5，CSS，CSS3】<br>

扩展【自适应，响应式，Bootstrap，Less，Flex】</p>

<p>无论是哪一种程序员，学会搭好的自己的环境都是第一步。选好 IDE，选好源码管理工具，学会天把自己做好的 Html 文件发布到服务器上可以直接访问。这都是最基础的技能，记着这些东西并不，你只是需要不断的去练习去熟悉它而已。所以从第一开始就去适应它。</p>

<p>会用 PS 是因为一是要切图，二是要制作雪碧图。这儿简单说一下，很多人都会说，切图这个活底分给 UI 还是分给前端。我一直都觉得这是 CSS 必须要做的，因为只有 CSSER 才会知道怎么切合。然而大部分 Android 和 IOS 人员都不具备切图的能力，这也是我一直很无语的工作方式，反正，要求我的 Team，无论是 Android 还是 IOS 都必须要学会自己切图。</p>

<p>另外再解释一下所谓切图和切页面的概念。切图指的是把需要用到的小图标了神马的切出来。切

面一般指的是就是把整个效果图做成 Html 网页。两种叫法我都觉得 Low 的不行，然而很多人都喜欢这么叫，而且会混到一起，时间长了，也就习惯了。

自适应和响应式是前端人员必须要懂的概念，也是必须要能做到的。毕竟现在是各种跨屏时代，及各种分辨率都层出不穷，H5 又是眼下最火红的职业和技能。

曾经有过工作半年或者是一年都不太懂什么是自适应和响应式的，这并不好。

Bootstrap, Less 和 Flex 也是 CSS 工程师必须要会的。学会 Bootstrap 更重要的还是要去学习的思想，这是很重要的一点，它帮助你开阔视野，你才会明白，靠，原来可以这么写。

Less 也是一样，在推荐 less 和 Sass 之前我犹豫了很久，然而看到 Sass 要 Ruby 环境就有点不爽了，虽然 Bootstrap 已经转向 Sass，但是在一般的项目中，Less 也够用了。所以还是果断的推荐 Less 了。

Flex 是一个好东西，我了解的还不多，在真实的项目中还未有使用过，不过挺看好的。  
简单说呢，前端工程师需要的大多数就是技术相关的技能，倒不需要太多的沟通啊和耐性啊神马的技。

哦哦，想起来了，CSS 的知识比较零散，要死记的东西其实是比较多的。

3 发展前景

CSS 的发展前景最近几年好的不行。在 Android 和 IOS 大行其道的时候，一大部分人都转去做这两个行业，然而突然之间，因为微信的缘故加上 JS 的魅力，H5 突然又火爆起来，所以呢，CSS 整来说还是一个挺靠谱的职业。最关键的是，CSS 入门最快的职业啊，几乎零门槛，只是转 JS 还是挺容易的。

CSS 薪水如果纯粹只是做 CSS 的话，12K 到 15K 基本上就是封顶了。。。所以：

0<del>3 个月：4K</del>6K<br>

3<del>12 个月：5K</del>8K<br>

12 个月以上~：8K~15K

CSS 的成长是非常快的。修真院的线下弟子都是一个半月左右就可以独立做项目，可以进入 IT 修界闯荡江湖了。正常情况下，3 个月左右是没问题的。

我的要求比较高，所以所谓的独立做项目就一定是独立做项目，不用怀疑这一点。包括自适应，应式和 Bootstrap。

所以。想要拿高薪，CSS 都必须要学 JS，没有其他的途径。

成长路径：

CSS 工程师-初级 JS 工程师-JS 工程师-前端架构师-前端 Leader (IOS, Android 和 JS 通吃)

不要相信 NodeJS，那只是玩玩而已。不要想从前端转后端，太难了，后端和前端是两个不同的念，特别是对基础知识要求的特别高。

CSS 和 JS 最好的方式就是 IOS 和 Android 通吃，精通一种，然后其他的能做项目，并不需要考虑其他的选择，推荐再学点其他脚本语言，首推 Python，永远不要去学 PHP，表问我为啥，个人好。

你一定要问，等我心情不好的时候再说。

4.入门门槛

零门槛！你相信咩，一个程序员的职业居然可以零门槛进入。。。<br>

好吧，大多数人其实并不把 CSSER 称为程序员，不过，你至少学会了 CSS，就能在 IT 修真界这个神的元世界里生存，就有了进军其他职业的可能性。

5.哪些行业适合做 CSS

IT 界：All<br>

其他界：All

PS：正常人都可以学的会啦（马丹我就不是正常人）

6.职业限制

CSS 的职业限制有三个，一个就是不去学自适应和响应式，一个是不去学框架，另一个就是学不 JS。

这三点都是很重要的突破。然后这个职业呢，怎么讲，在一家公司里可替代的程度比较高，并不算特重要，做不了核心岗，也不存在什么代码交接。你就算代码写的再烂，也最多就是花一个月时间重写。。

所以如果你不想做 JS，你就真的真的发展受限制了，一定要记着，CSS 只是你的一个跳板。

另外，因为零门槛，所以你其实还是有很多计算机相关的知识要补的，这是一定的，没什么可商量的

</p>

<p>只是做前端还好，如果想做后端，只是不想花那么长的时间才进入这个行业，就必须要从 CSS 起，跟着用自己的其他时间恶补计算机基础了。</p>

<p>7.去哪里学，怎么成长</p>

<p>之前说我的贴子是广告的，你们错了。<br>

看到这里，这才是真正的广告。</p>

<p>要说去哪里学 CSS，绝对是 IT 修真院，请先看这里 <a href="https://ld246.com/forward?got=http%3A%2F%2Fwww.zhihu.com%2Fquestion%2F36823157%2Fanswer%2F69156567%3Fgrup\_id%3D643479018993934336" target="\_blank" rel="nofollow ugc">如何在一个月之内从零基础成为初级 CSS 前端开发工程师？ - xdy1 的回答</a>。<br>

IT 修真院跟其他任何一家培训机构都不一样，完完全全不一样的教学模式。看完这篇帖子，其实你能大概明白这是什么样的教学方式。</p>

<p>而且，线上学习完全是免费，不仅如此，大修真院还给你入院就提供了良好的人脉关系。嗯。<b>

然后，如果你真的不想来大修真院，或者我没看上你，那么你就也不要看视频或者是交钱学什么的，看 W3C 吧，自己偷偷按照修真院的 Task 做项目，跟着就找家公司实习去吧。</p>

<p>虽然慢了点，好坏也入门了。</p>

<p>四 JS<br>

=====我是第一次过 100 赞的分割线=====

<p>JS 说起来必须是一个神器，这个当年 10 天内被开发出来的神器，以一种谁也想象不到的速度快发展，它击败了 Java Applet，逼死 Flash，当 Android 和 IOS 看似一统全球的时候，JS 慢条斯理和 Html5 一起蚕食 App 市场，甚至还出来了 NodeJS 这种连后端都不放过的残暴东西。</p>

<p>原生和 WEB 的两种方式一直在争论不休，而我自己在三年前就是绝对的 WEB 支持者，windows 对于 JS 支持的更彻底，只是 Winphone 一直不给力。</p>

<p>好吧，对于拿 NODEJS 做后端工程项目我还是有点不情愿，然而提供各种辅助工具什么的还是不错的。JQuery, Ajax 和 RequireJS 和 Angularjs 和 PhoneGap 都是一些标志性的转变，还包括 ReactJS。JS 简直了，简直了，简直什么都不放过。</p>

<p>所以 JS 简单么？我带着你们了解一下 JS 工程师是什么样子的。</p>

<p>1 工作内容：JS 工程师其实分成两类，在之前讲 CSS 的时候已经提到过，一个是套页面的，一是前后端分离的。对这两个概念还是分不太清的，可以回过头去看 CSS 的部分。</p>

<p>对于套页面来说，JS 工程师的职责非常简单，一个是做一些简单的动画和交互和验证，轮播图弹框，验证用户名是否为空等等。另一个就是通过 Ajax 取一些后台数据，然后在页面上展示出来。<p>

<p>那个时候的 JS 工程师并不被太多人重视，很多时候一些后端的人把前端的工作也做了--除了写 CS，大部分的后端工程师写点 JS 还是不成问题的。</p>

<p>可是前后端分离的出现，彻彻底底的改变了前端的世界。前端居然有架构了，前端居然也有“编”的概念了（你能想象么，JS 的项目居然不能直接打开源码直接使用了，必须要经过编译才可以）。端工程师和后端工程师只通过 Json 数据交互（感谢 Nginx 提供了一个完美的跨域解决方案，再也不什么 Jsonp 了），然后服务器端的同学发现：好简单啊，我对 Android，对 IOS，对 Html 都只需提供一套 API 就够了~~</p>

<p>前端同学更开心，再也不用去搭建各种悲剧的开发环境了！前端变的很轻，而且控制欲很强（AngularJS 好赞），页面跳转神马的 SoEasy，而且也有了 Bower 这样的包管理工具，还有类似于 JSP 定义 Tag 的指令，MVC，还有了 Service 的概念，马丹，前端到底发生了什么！</p>

<p>所以我们现在说到的，基本上就是前后端分离的工程师，他们的工作职责就是和 IOS&Android 的工程师一样，跟后端定接口，确定框架和架构，分模块，然后跟后端联调互相吐槽对方是渣渣。

/p>

<p>项目结束之后修复线上 Bug，不停的跟用户说：在我这里是正常的，你换个浏览器试试？不行就一下 Cookie，再不然就换台电脑。</p>

<p>2 需要技能：<br>

环境【IDE (WEBStorm, Sublime, EditPlus) , 源码管理 (SVN/Git) , WEB 服务器(nginx)】<b>

基础【Http, REST, 跨域, 语法, 组件, F12, Json, Websocket】<br>

框架【jQuery, AngularJS, Bower, RequireJS, GruntJS, ReactJS, PhoneGap】<br>

业务【金融，教育，医疗，汽车，房产等等等等各种行业】<br>

第三方【微信，QQ 等各种第三方登录，支付，IM，地图，语音，视频，图片】</p>

<p>环境不说了，参考 CSS 的。<br>

基础跟 CSS 差别就好大了，要有一些网络协议的基本概念，要了解什么是 Http，什么是 Https，什么是 WebSocket，什么是 Rest，各种 JS 的语法（太复杂的完全不用管，别去管什么闭包什么 Prototype 什么 This 的指向），Json 是必不可少的东西。</p>

<p>框架就多了去了。直接拿 JQuery 来学习就好，表去太在意什么原生语法，大部分场景你都不用引入一个 JQuery 的库导致系统变慢，AngularJS 真是一个好东西，表太纠结一些细节，Bower，RequireJS 和 GruntJS 这些会有一些前端大牛帮你搭好框架你直接用就好了。<br>

想做点 App 就用 PhoneGap，想尝试一下 ReactJS 就去试吧（我很想试一下然而看到有 Render 就不下去了）。</p>

<p>必须提醒一次，如果你是从 CSS 转到 JS 的，那么一定要看一下这篇贴子。<a href="https://ld46.com/forward?goto=https%3A%2F%2Flink.zhihu.com%2F%3Ftarget%3Dhttp%253A%2F%2Fbbs.ptteng.com%2Fforum.php%253Fmod%253Dviewthread%2526tid%253D7473%2526extr%253Dpage%25253D1%252526filter%25253Dtypeid%252526typeid%25253D2" target="\_blank" rel="nofollow ugc">【IT 修真院】怎么样从 CSS 转 JS-IT 修真院</a>。这里强调的就是，你必须懂业务逻辑，这跟 CSS 完全不需要懂业务逻辑简直是天壤之别，很多时候你并不是因为技术而无学会 JS，而是缺少对于业务逻辑的理解。</p>

<p>当然啊，这并没有完。还有无数的第三方库等着你，做微信呢你必须要用 WX 的 SDK，要用百度地图呢就必须要用百度的 SDK，还有各种支付啊神马的。如果你要做 IM 就有环信啊神马的。</p>

<p>所以。JS 就是一个真正的程序员，一个真正的工程师，不但要能把代码实现，还需要理解需求需要懂得编码规范，需要知道如何控制项目流程和进度风险，需要修复线下 Bug，需要调优，需要做本管理，需要制订接口，需要编写技术文档，还需要做新技术的调研或者是带新人。</p>

<p>现在你了解了咩?(如果不了解，加入 IT 修真院嘛，老大带你从 IT 界从菜鸟修成大神)</p>

<p>3 发展前景</p>

<p>我说了那么多，你们对 JS 的期望也很高了吧。坦白的说，有点高的离谱！<br>

JS 是唯一一个可以在两年之内薪水到 20K 的职业。<br>

虽然工作五年或者是七年的 JS 在薪水上差别并不大，但是也已经让很多人难以置信了好么。</p>

<p>好多人在别的行业，工作十年都拿不到 10K 啊亲。</p>

<p>0<del>6 个月：6K</del>10K<br>

6<del>12 个月：8K</del>14K<br>

一年<del>两年：12K</del>20K<br>

两年<del>五年：18K</del>30K<br>

五年以上：24K~</p>

<p>JS 的入门相对来说比 CSS 难的多（表怕，有我大 IT 大修真院），但是他的好处是学会一点是一，并不像 Java 一样，绝对是前期 Gank 型英雄，到了 6 级就能四处杀人。Java 是后期，要到 16 级后才能出山才能够震撼全场。</p>

<p>所以 JS 相对来说比较平滑，而且现在学习的曲线也比较清晰，慢慢在业界都形成了相对完善的架。JS 可以选择前进的路有很多种，只要你不是太瞎（跟着修真院就一定不会瞎），就没什么大的问题。</p>

<p>成长路径：</p>

<p>JS 菜鸟-JS 初级工程师-JS 中级工程师-JS 架构师-UED-前端 Leader-前端大牛</p>

<p>大概这个样子啦，JS 工程师在后面有一点后续无力，一般来说，我还是强烈推荐 JS，IOS，Android 三者通吃，在我的带的 Team 中也一直要求他们这么做（虽然并未能完全实施成功）。</p>

<p>这三者有太多相似之处了，到最后你会发现，语言只是一种工具而已。要做的事情，其实是差不多的，并不需要把所有的语言都特别熟悉，有一种打底的，其他的都可以随意去玩了。</p>

<p>4.入门门槛</p>

<p>必须有 CSS 基础，必须有 CSS 基础，必须有 CSS 基础。<br>

或者是有后端 Java 基础。</p>

<p>总之你要么是会了 CSS，要么是精通了一种后端语言，千万不要 CSS 没学好，后端语言也不会直接学 JS，心里会没底，不踏实。</p>

<p>其他的都没什么了，列出来的基础知识，自己慢慢补就行了。</p>

<p>5.哪些行业适合做 JS</p>

<p>IT 界：CSS，然后是所有的都可以<br>

其他界：所有的都可以

好蛋疼的结论然而事实上就是如此，这波 JS 的红利受益于微信，以及移动端的普及。我不确定三年或者是五年之后会不会还是这种状态，然而三年和五年已经足够可以改变你的人生了，所以，你等毛线？

大修真院的弟子中有搬砖的有退学的有高中毕业的有做电气和通信的，你想想为什么你不可以？一不合适你的大概就是：你真的没有编程思想哈哈哈，然而这确实是一件很难解释和预测的事儿。我确实见过反应比较慢的，进度比较慢的，但是无论如何，混口饭吃，不成问题。

6.职业限制

JS 的职业限制也挺多的，主要就在于学习的方向上很容易误入歧途（至少在我眼里是这样的，信我，我不是什么好人，我并不会我说过的每一句话负责）。比如说一直坚持喜欢用原生，各种喜欢一些奇奇怪怪语法（我把我不懂的语法都归结为奇怪的语法），各种回调等等等等。

学到一些好的框架其实很赞的，对于大多数人来讲，就够了。只是缺少一个清晰的指引。有的时候你会发现，你对 Http 了解的比较少，会是一个瓶颈。

你对数组啊，循环啊不了解，是一个瓶颈。

你对 MVC 不了解，也是另一个瓶颈。

对 Json 不懂，不知道怎么定接口，也是个问题。

这些都学会了，也会遇到一些问题。就是 JS 的工程师三年，五年，七年的差别，大多数人都不是特别的大。

所以总会有人想学后端--但是我绝对不赞成学 Java，NodeJS 也可以不去学，不如去学习 Python，不要提 PHP 啦（我真心不喜欢 PHP，不知道为嘛）

或者说，后端并不是语法，而是牵涉到算法，架构，性能，业务，选型等等跟前端几乎没有任何系的知识，所以学习一门脚本语言，能做点简单的东西就好了。

还是需要把注意力转到 IOS，Android 上去。

7.去哪里学，怎么成长

不用说。首推的当然还是我大 IT 修真院。IT 修真院是先总结归纳一个能够独立完成项目的 JS 工程师需要掌握的技术体系，然后设计 15 个 Task 来让你逐步完成对于知识的掌握。修真院一直在传播码是写出来，而不是看视频看出来的概念。更看重你们在真正去做 Task 的时候独立解决问题的能力除此之外，还有跟你一起学习和成长的师兄。这就是 IT 修真院去培养 JS 工程师的方式，而且，线上完全不收费的。

如果你不想用这种方式去学习，还是我之前说的那些话，W3C，然后做点东西，跟着去一家公实习。

五 后端 (Java/python/go)

====我是想玩 Dota 的分割线=====

说到后端，其实我主要想说的是 Java。C 和 PHP 这两种语言我都不喜欢，之前也说过了，我绝对不是一个说话公平公正的人，什么观点都带着自己偏激和极端的调调。

我之前在贴吧跟 Java 吧的吧主一直在撕逼（原因很简单，我说教大家学 Java，一个月收 400 钱，他说我是骗子封我贴。然后我说好吧，我不说教大家学 Java 了，我来给大家解决在学习过程中到的困惑，他说贴吧不能发问答贴。我说行，那么我就写一些新人学 Java 必须要学数据库，数据结构和计算机网络，于是一群吧主过来喷我说，我没学过这些我也照样学会 Java 了啊，什么多线程什么承等等，我无语了解释说 Java 语法不重要，重要的是要学会后端的架构，要懂算法，要懂业务，要系统的扩展性，要会调试程序，于是吧主们就把我封了，我很不爽，就新开帖子跟他们撕逼对骂--我来不是一个重身份的人，我骂人会很脏，所以如果看到这个帖子觉得我很厉害那么你瞎眼了，我就是个不喜欢就说，谁喷我一脸我喷谁一身的性格，现在的结果就是 Java 吧的吧主每隔 10 天就来封我次-哈哈哈哈哈比闹钟都要准，所以三个月过去了，我带出来很多 CSS 和 JS 的学员，然而 Java 的学并没有多少。所以如果有人去 Java 吧替我骂一下那些 XX 吧主并且截图给我看，我会很开心很开心，说不定就会给你们开小灶哈哈哈哈哈。）

之所以说这些，一方面是 400 多的赞让我觉得有点羞愧，另一方面也是想强调一个概念，学后，学会语法只是开始而已，最后一个就是我是一个小人，谁欺负我我就想欺负回去。

我想想该怎么描述后端的工作。后端跟前端是截然不同的，之前讲过。前端是 Gank，后端是大期，要等到 16 级以后才能 V5 起来，而且我非常不推荐前端去学后端（所谓的全栈工程师完全是扯我有时间会写一下，为什么不要去做一个全栈工程师）。后端要积累到足够多的项目经验，才能够成一个靠谱的后端工程师。我觉得。我来举一个跟着我线下半年的小培宇的例子就能简单说明一下后端工作。

<p>小培宇是第一个来到我大修院面试（嗯，最初我是给他们发工资然后带他们学习的）的人，跟我他是考研失败，差了几分，然后也做过点项目，我随便问了几句就知道了他的状态：人挺聪明的，但在学校肯定玩的疯，所以问点排序算法还是能够答的出来，数据结构也懂一点儿，LinkedList 和 Arra List 删除数据谁更快也能答的挺靠谱的，但是绝对绝对没写过一行工程代码。</p>

<p>他打动我的那句话就是：不在乎工资多少就想多学点东西。很好，我默默的点个赞，因为我本身想把自己这几年积累的经验和知识和走过的坑整理出来，告诉互联网的新人，所以也不抵触带新人，坦说，愿意像我这样带新人的公司，几乎没有，带新人真不是一般的累。幸好我之前在各种公司中都带各种新人，好的坏的都带过，所以还算是熟悉。</p>

<p>于是我给培宇精心设计了的学习曲线，这也是我大 IT 修真院的核心观点：<br>

1.先搭建基础环境 (Maven, SVN, Eclipse, jetty,SecureCRT,Linux, Mysql) <br>

2.做简单的 CRUD (DAO-自己封装的数据层, junit,Log4j,Rest,Spring,Spring MVC,Json,JsonTagli ) <br>

3.做一个相对复杂的系统 DB 设计，接口设计，项目部署，错误提示，Bug 查找，怎么打系统日志。<r>

4.做了一个微信相关的项目，了解微信的 API，交互方式，Cookie，拦截器，AOP，登录系统的设计命名规范等。<br>

5.拆分 Service，将 Home 和 Service 分开，使用 RMI 调用，实现各个层次之间都可以完成分布式部署，使用 Tuscany (真心喜欢 Tuscany) 完成 SCA。<br>

6.使用 MongoDB 完成地理位置的搜索，短信，图片上传，云存储，使用 Tiles 来配置页面模板。</>

<p>这些内容他花了将近三个月的时间。对他来说已经是学会了很多东西了，这三个月是几乎没日没的学出来的，要知道他之前压根就不知道什么是 Spring，生成 Json 和套 JSP 的区别我骂了他好几次才弄明白，经常会遇到一些 Maven 或者是 Tuscany 的报错不知道该怎么解决，数据库字段的规范和口规范常常被我黑的体无完肤。很多东西都只是知道个皮毛而已，你们自己说说，学会 Java 语法算么？</p>

<p>这还是有我来带，有我来教，有我给他定制合适的项目教给他去做，如果没有这些，你们自己学学会我说的这些东西要多久？<br>

别的不说，学会怎么打日志，怎么根据线上的报错去找错就不是一个月两个月能解决的。做为一个工师，应该明白，很多时候要学会正确的路怎么走，还必须要知道错误的路是走不通的。正确的路大概那么几条，错误的路呢？</p>

<p>为什么很多时候我看到错误日志就会明白是什么地方报错了，是因为我之前花了无数的心血和心在查找这些错误上，所谓的经验就是这样，看的多了，一眼就知道大概什么地方报错了，然后随便百下，就能找到解决方案，跟着去尝试到底行或者是不行。这些是看书，或者是看视频能教会你的么？<p>

<p>到现在为止培宇已经跟了我快半年了，还是被我骂的狗血喷头，别的不说，就是接口的 Wiki 文和代码保持一致，他都会经常犯错---这跟 Java 语法有什么关系，然而不经过一个好的训练，想做的好，很不容易。</p>

<p>其实他接下来要学的东西更多。Memcache 或者是 Redis, ActiveMQ 或者是 RabbitMQ 或者是 QPid, Mybatis 或者是 SpringJDBC, Struts 或者是 SpringMVC，我告诉他的只是一个我们在项目经过实践的，认为最合适的架构体系，然而他并不知道是怎么选择的。他必须要把这些相关的选择都所了解，然后才能成为一个架构师。这个时间，如果一直跟着我，我觉得应该是在一年到两年左右。<p>

<p>这是一个横向扩展的内容，在这个时候我还没有要求他去看一些深层的东西。只是需要他停留在用的状态就可以。在会用这些技术之后，再去了解一些自己喜欢的技术的细节，不成为一个只会使用具的码农，所以培宇问我还需要多久才能达到我的水平的时候，我其实并不想打击他。我也是很刻苦力人啊，曾经无数个日夜也是默默的去一行一行代码去用最笨的方法调错，并没有人告诉我怎么样正确的只有靠一个又一个的项目总结出来的经验。</p>

<p>而且我还会一些 Drools, CRM, Lucene 等等一些和架构师关系不大的事儿偏算法一些的东西毕竟当年也学过点数据挖掘机器学习之类的内容。</p>

<p>就算是学会这些了，对于一个后端工程师来说就够了么？不不不，还需要学习 JVM 优化，监控部署流程，发布流程，项目进度管理，代码重构等等等等。</p>

<p>所以，你们自己算算，这些东西如果都学会，一个 Java 工程师要多久才能成为架构师？</p>

<p>然而我还是对带培宇很有信心，首先他相信我，他愿意学，跟我当年一样，不怕苦不怕累，人也聪明，做事也有责任心，其次我知道他应该怎么走这条路，先做什么，再做什么，哪些该花时间和精力

, 哪些不该花。 </p>

<p>我希望他能够在一年之内就成为一个架构师。就如他在三个月和六个月之间独立做项目已经不成问题了一样（记着，我说的是独立做项目，自己设计 DB 设计接口设计架构完成需要的功能，从设计到现完全自己来）。 </p>

<p>我也希望我能够帮助很多和培宇一样，有实力有能力只是没有遇到我的那些人，这也是我为什么知乎发帖的原因，IT 技术的培训，哪些培训机构能做到这一点？ </p>

<p>这个真实的小故事，就是想跟大家提前说清楚，我对后端的要求有多高，这也是后端特别好玩的方。你必须会很多种框架，有足够宽广的视野，还需要有足够多的项目经验（做金融和做地产是完全不同的概念），还需要懂项目开发流程以及快速定位线上问题的能力。 </p>

<p>这些，就是我说的后端的主要工作内容了，这也是为嘛我说到后端的时候，大部分就是在指 Java，而我说 Java 的时候，基本上是指后端，根本就不是指 Java 的语法。更不是说是 Android。 </p>

<p>现在明白为什么后端是大后期了么，为什么不建议前端学后端了么。后端要懂的东西，太多了。 <br>

言归正传，我来讲一下后端工程师的相关内容。 </p>

<p>1 工作内容： </p>

<p>大部分的后端工程师都停留在功能实现的层面上。这是现在国内二流或者是三流的公司的现状，至是在某些一流的公司。很多时候都是架构师出了架构设计，更多的外包公司根本就是有 DBA 来做设计，然后后端程序员从 JS 到 CSS 到 Java 全写，完全就是一个通道，所有的复杂逻辑全部交给 DB 做，这也是几年前 DBA 很受重视的原因。 </p>

<p>所以你能看到成千上万行的存储过程（存储过程，视图，事务，外键 这些东西我真心希望永远要在 Mysql 里出现），这就是外包公司中最常见的架构体系。来个 SSH, Over。 </p>

<p>好一点的会个 Webservice，用过 ActiveMQ，也用过 Redis，甚至还会用过 Dubbo。然而大多数情况也根本不了解为什么这么用。 </p>

<p>很多人写了两年或者三年代码都没做过独立的 DB 设计，不知道什么是 REST，不懂怎么做接口设计，也不知道怎么去定位问题。 </p>

<p>所以对于他们来说，拿到产品经理的需要，会有一个项目经理或者是 Leader 分配任务，跟着按就班的把代码写完，跟前端调试完，QA 测试不通过，加班改回来重新改，改完 QA 又没通过，再加再改，QA 终于通过了然后上线了突然发现另一个好的功能不能用了，跟着再接着改，在线上发布一次又一次。。眼睛熬的通红最终真的受不了了，休息几天换另一家公司涨个 40% 左右的薪水继续这样日子。 </p>

<p>不不不。我带出来的后端程序员并不要这么做。所以，我带的后端程序员的工作方式是这样的。 <p>

<p>拿到产品需求->后端程序员做接口设计，架构设计，DB 设计->拿出方案来做技术方案评审->审通过，开始预估时间->每日更新自己的 Task->接口完成自测一百遍，每日部署到开发环境，随时成->CodeReview->重构代码->性能测试->Demo 通过->发布到测试环境->修正 Bug->重新发->发布到线上环境。 </p>

<p>这中间需要理解需求，需要拿出多个方案，需要跟前端配合，需要跟 QA 配合，需要跟运维配合需要跟产品沟通，有时候还需要找 UI。后端几乎是一个核心节点，而这个核心节点接起来了所有的人 </p>

<p>我不知道我讲清楚没，很多时候我都发现我可能太久没做一个 IT 新人了，都忘记了新人们关心问题或者是困惑是什么。 <br>

这就是我知道的，两种后端程序员的工作内容。你选哪一种？ </p>

<p>2 需要技能： </p>

<p>环境【IDE (Idea/Eclipse, Maven, Jenkins, Nexus, Jetty, Shell, Host) , 源码管理 (SVN/Git , WEB 服务器(nginx,tomcat,Resin)】 <br>

基础【Http, REST, 跨域, 语法, Websocket, 数据库, 计算机网络, 操作系统, 算法, 数据结构】 <br>

框架【Spring, AOP, Quartz, Json TagLib, tiles, activeMQ, memcache, redis, mybatis, log4j, junit 等等等等】 <br>

业务【金融, 教育, 医疗, 汽车, 房产等等等等各种行业】 <br>

第三方【微信, QQ 等各种第三方登录, 支付, IM, 地图, 语音, 视频, 图片】 </p>

<p>环境不说了，搭环境永远是后端人员比较头疼的事儿，所以才会有很多人想用简单方便的的语言解决这些问题，比如说 Python 之类的。我还是喜欢 Java，大概很多人觉得重，然而我喜欢，我觉得是“重”，而是“正”。好像剑一样，王者之剑，路子很正。 </p>

<p>基础知识太多了，正是我一直强调的，做后端，这些基础知识了解多少，其实就是决定了你以后走多远。这些科班出身的计算机 ER，会了这些，才有了一个平台，才可以站在这个平台之上去搭建高层的建筑，如果根基不稳，你觉得你会对上层的知识理解透彻么？ </p>

<p>框架是 Java 最有资格说自己是架构师的原因。无数的开源框架，选型，筛选，对比，填坑，优，维护，寻找最适合的业务场景，很多时候很多公司的架构简直了（我不吐了，很多技术都在用然而一种用法几乎都是错误用例的典范）。所以你想想，你大概要有多少框架要学要用？很多时候，你必要想清楚，哪些是需要认真了解的，哪些是需要一笔带过的。 </p>

<p>业务对于后端人员来讲无比重要，不懂业务，就没有架构。这是我经常说的一句话，这个世界上存在不懂业务的架构师（我不怕被打脸），一个架构师必须要深入了解业务体系，知道哪些是会变的哪些是不会变的，哪些是重要的，哪些是不重要的，然后才能做出来适合某个应用场景的架构来。比如说，同样的表，几千万的量和几亿的量差别非常大，频繁读和频繁写的设计也完全不同。会有一些通的架构思想和理念在里面，但是都是需要跟业务结合落地的。 </p>

<p>PS：很多金融证券行业的程序员，就是靠业务知识混饭吃的。对他们来说，对业务体系的了解要在技术上的追求重要的多。 </p>

<p>第三方的东西和 JS 的内容相似，我不想多说了，而且 JAVA 的第三方的东西更是多的离谱，坦的说 Drools 这东西我就没彻底研究明白，虽然很喜欢。而像这种类似的东西，太多太多了。 </p>

<p>3 发展前景 </p>

<p>对于后端人员的发展前景，我有两点想说的。 </p>

<p>A.无论是 B/S 还是 C/S,无论是 WEB 还是原生，或者是智能硬件，后端都会屹立不倒。 <br> B.随着后端架构体系的稳定和成熟，后端人员在性能上需要担心的问题不多（再加上大部分应用场景实并不需要那么多的性能），所以更多的应该会关注于一个稳定的扩展性好的架构，以及快速实现能复用的业务逻辑模块实现上。 </p>

<p>最近后端人员在价格上，其实有点偏低于前端人员的，就向我之前所说。两年的 JS 可能拿到 20。两年的 Java 想拿到这个，非常难。然而，五年的 Java 或者是七年的 Java，拿到 30~40K，不难。 </p>

<p>更高的，也不稀奇。 </p>

<p>0<del>12 个月：4K</del>10K<br>

一年<del>三年：8K</del>20K<br>

三年<del>五年：18K</del>30K<br>

五年以上：30K~ </p>

<p>成长路径： </p>

<p>Java 初级工程师-Java 中级工程师-架构师-技术经理-技术总监-CTO-CEO </p>

<p>后端的爆发力并不差，只要你给他时间，只要你愿意前进，后端的路线很深，深到你有时候会觉得自己还没来得及全部了解，就已经有无数的新人涌进来要替换你的位置了。 </p>

<p>4.入门门槛 </p>

<p>计算机网络，数据结构，数据库，操作系统，Java 基础语法。 <br>

Java 是入门门槛最高的一个，没有之一。（好吧，我虽然说的是后端，然而一直把 Java 等同于后端 </p>

<p>当然，如果你的志向并不是一个架构师，只是像 NodeJS 和 Python 或者是 PHP 一样随便做点项目，那么也可以说的得上是没有门槛，但是我说过我有偏见，所以可以直接把我无视掉。如果你觉我说的哪点不对，你过来揍我啊。 </p>

<p>要跟我学 Java，就必须把这些基础知识学好，我只带想成为架构师的人。 </p>

<p>5.哪些行业适合做后端工程师 </p>

<p>IT 界：无 <br>

其他界：无 <br>

科班生：计算机专业的中等水平能力以上 </p>

<p>所以如果有各种培训学校告诉你零基础 4 个月 20000 块钱把你教出来做 Java 后端，然后你月上万，你就直接一锅盖盖他脸上吧。 </p>

<p>那么零基础的人想做后端，怎么样才能入门呢，我比较推荐的是先做前端，然后把自己的基础知各种补，补回来，再去转后端--但是实际上，你做了前端，再想转后端，就太难了，因为你到时候就太想放弃自己一年或两年就可以轻松拿到的高薪，去做一个苦逼的后端了。 </p>

<p>6.职业限制 </p>

<p>后端的职业限制有很多，第一个职业限制就是不去做独立的项目，不做 DB 设计，不做接口设计 <br>

第二个职业限制就是视野不开阔，不知道有什么样的开源软件可以用。 <br>

第三个职业限制就是不重视线上环境，不知道如何写日报，也不知道如何快速定位。我不得不说我带的兄弟，有一次解决线上问题的时候快把我气疯了，他们在那里猜测问题出现的原因，跟玩福尔摩斯样，不打日志不看日志，根据现象倒推结果，直接盲改代码再扔到线上看看有没有解决问题---那是最逼不得已的办法好么，在此之前能不能安静的把日志打出来，确认一下到底是哪里出错了？ <br>

第四个职业限制就是不懂版本管理，不懂 Bug 修复流程，不懂开发流程。这些其实都是一整套的流体系（等我心情好了，有人把 Java 贴吧吧主骂的狗血喷头了，我大概也会写出来） </p>

<p>大部分后端的人员都会抱怨自己不会写前端代码，不会写 Android 或者是 IOS，不能自己独立成项目，所以他们才倾向于自己做一个全栈工程师，做一个自己喜欢做的东西。 <br>

这也是后端人员会经常觉得不爽的地方，自己写的东西完全感受不到，而且一旦出问题很多时候都是问题，解决起来很麻烦，经常不敢改代码，因为看不懂前人的东西。 </p>

<p>有时候后端人员会比较木，虽然很各种职业都交流，但是多数都会觉得自己很 NB 其他人都是很 S 。 <br>

这也是后端人员比较大的问题，往上走的话也容易遇到各种瓶颈，做技术的，做到 CTO，再去做 CE ，其实很难的。 </p>

<p>而且，等你走到足够高的高度，你会发现，一个七年工作经验的正常发展的后端工程师，一定会一个七年工作经验的产品或者是运营，在薪水和职业上秒杀他。这也是做技术的最大的悲剧。 </p>

<p>不过大部分的后端工程师都比大部分的产品和运营人员薪水高，这也是这个行业的特征之一，所高不成低不就，小富即安，就是这样的。 </p>

<p>如果你是一个有理想的后端工程师，我建议你多关注一些敏捷开发，多关注一些项目管理，学会着自己的兄弟们一起做事儿。再不然，就是在技术这条路上一直走到黑。 </p>

<p>7.去哪里学，怎么成长</p>

<p>基础知识不要来找我，我不教。随便去其他的学校补基础也可以。想要提高或者是想要成为架构，就来找我。 </p>

<p>我在说培宇的时候其实已经讲清楚了我是怎么带人的，但是在线上做这些，确实有难度，而我又想教一些虚头巴脑的东西害人害己，所以我还在想办法解决怎么样在线上把后端工程师带好的问题。 <p>

<p>如果看到这里觉得我说的有道理，一定要称赞我，催促我，还有帮我宣扬大修真院的思想 and 做事情，让我觉得这个事情有意义，值得做。 </p>

<p>想做好，真的挺不容易的。所以，能转就转，能收藏就收藏，能推荐就推荐。我想我应该能想到个好的办法，来帮助后端工程师，把成长的时间从五年到七年，缩短到一年到三年，至少在线下，我绝对有信心的，这种信心来自于我之前这么做过。以及我正在做。 </p>

<p>你们可以随时找培宇聊聊，看看他这些日子倒底是怎么被我骂的狗血喷头然后开心快乐的活着的 </p>

<p>六 DBA<br>

=====我是被这几天加群的人气死的分界=====

===== </p>

<p>首先说，我对 DBA 的了解并不专业，也不够多，而且对这个职业也有偏见。所以，我只能把我受到的，我会的讲出来，然后如果说你们觉得我说的不对，要么自己开贴回答来打我的脸，我虚心学，要么就直接笑笑走开，表在评论里说三道四，最烦这个。 </p>

<p>七年或者八年或者很早之前，DBA 是非常吃香的职业。讲这个，大概要从系统的性能瓶颈说起。 <br>

很早之前，互联网刚开始的时候，算是蛮荒时代。那时候大家写代码还没有规范，能把功能做出来就错了，大家拼的是什么呢，Sql 的性能。基本上就是没有中间层，也不会分什么服务层和 Web 层，多时候 SQL 都写到页面上。 </p>

<p>然后 Sql 呢，又属于那种外键，视图，存储过程的天下。这就导致了出现一个问题。大部分的功都是通过 DB 来实现的，也就是说，什么计算啊，分组啊，排序啊，筛选啊，全是靠 DB 来做。 <br>

小功能还没问题，功能一多，问题就出来了，一个 Sql 语句执行了半个小时没做完，然后整个系统崩掉了。 </p>

<p>那么，怎么解决呢，解决的方案就是。。。我其实很难理解这种思考方式。。。就是找一人，对 DB 特别熟悉，他的职责就是审核所有程序员的 Sql 语句，去找出来这些 Sql 哪些用到索引了哪些没用，能不能执行，怎么优化，以及监控线上的慢 Sql。一个公司能养得起 DBA 的，很 NB 了很贵的！ </p>

<p>所以这是那个时候的 DBA，但是，很快大家发现有不同的方式了，这种方式就是，我靠，原来

可以用分库分表，我可以做读写分离，我能做主从。于是对于 DBA 的依赖又重了一些，再加上数据安全和备份，所以 DBA 的作用已经有点偏移，然而最关键的还是系统架构的发展变化了。分布式的念慢慢的起来了，大家明白了一件事儿：机器不够，并不是说把服务器升级成小型机就能搞定了，而应该用更多的机器来做，因为便宜，而且更简单。所以后台的系统架构慢慢的演化出来很多不同的层 WEB 层，服务层，缓存层，DB 层。对于缓存的使用越来越重要，由此而变化的观点就是数据分成了存和持久两种结果，DB 慢慢的变成了持久层-也就是说，只是要把数据持久化，并不希望它去承载用的压力，缓存主要用来扛并发，不需要做持久。这是一个很关键的点，也是决定 DBA 命运的转折点

当然现在还看不出来（像 MongoDB, Cassandra, 这些，又是另外一种不同的技术走向，包括 Mysql 也在不断的想要提升自己的性能），所以这些东西我们先抛到一边不谈。只说这中间发生了一变化，对于后端人员来说，对数据库访问的变的严格起来了。尽量单表操作，不允许复杂查询，设计结构的时候必须考虑缓存，甚至我们在白社会的时候还设计了一套通用的 DB 访问机制--虽然是七年前设计然而现在一直觉得很赞，只是再也没有如我在搜狐的时候那群人做这些事了-反正我见识少，直在小公司混，也不怕你们嘲笑我见识少。

这样就导致 DBA 的一个很重要的工作职责，失去意义：就是查找慢 Sql，因为我们在系统架构已经决定了不再这么使用 DB。这样使得 Oracle 什么的也慢慢的失去了价值-我知道我说的每一句话有可能会引起争论，所以我不得不再次强调一次，纯属个人的脑残观点，不喜欢的话，要么认真的回来打我的脸教我做人我认真学习，要么就是滚远点表理我。包括建表，去除外键，去除事务，去掉视等等等，一瞬间，DB 的使用简单多了。

那么，DBA 还能做什么呢？

对于我现在的理解来说，DBA 的职责慢慢变成了数据备份和安全策略--然而这部分又跟运维的工作有冲突，所以在某种程度上来讲，我都会在五十人左右的公司把 DBA 安排到运维部分，跟运维的兄弟做基友。可是现在云服务器也变的越来越好用了，这里也推荐一下好友的金山云和 Ucloud。阿里云我并没有神马认识的人，所以不推。

DBA 除了之前提到的主从，读写，数据备份，权限控制，分库等等，还应该再扩展视野，把 MongoDB, Redis, memcache, elasticsearch, hadoop 等等这些数据全部管起来。我觉得，更像是个运维的分支了。

这就是我目前认可的 DBA 的价值和意义，已经从之前的性能优化部分转移到了数据备份和安全

毕竟，性能，架构，和优化这些东西，是离不开业务系统的。

那么，接下来，和之前一样，继续介绍一下 DBA 的工作内容。

1 工作内容：

如果你做了一个 DBA，基本上会遇到两种情况。一种是你的后端工程师懂架构，知道怎么合使用 DB，知道如何防止穿透 DB，那么恭喜你，你只是需要当一个 DB 技术兜底的顾问就好，基本上没什么活可以做，做个监控，写个统计就好了。你可以花时间在 MongoDB 了，Hadoop 了这些，随便玩儿。再按照我之前说的，做好数据备份。如果需求变动比较大，往往会牵涉到一些线上数据的更改那么就在发布的时候安静的等着，等着他们出问题。。。如果不出问题就可以回家睡觉了。

另一种情况就是我刚刚提到的，大部分程序还是靠 SQL，然后有时候 DBA 还需要写几万行的存过程，那么你的主要职责还是优化 Sql，优化 Sql，永远不停的优化 SQL。

嗯。就这样。

还有就是多花点时间把 MongoDB 和 hadoop 这些都维护起来，或者简单说，只要跟数据安全备份相关的东西，都维护起来。

2 需要技能：

环境【Linux, Mysql, Oracle, MongoDB, Hadoop】

工具【各种 DB 的版本，工具，备份，日志等】

这个说是环境已经有点勉强了，毕竟是一些吃饭的家伙。就是各种 DB，各种维护什么的。工具也是相关的内容，再强调一下对版本的熟悉程度。

马丹我感觉我没什么可说的了。因为刚刚又被 <https://ld246.com/forward?goto=https%3A%2F%2Flink.zhihu.com%2F%3Ftarget%3Dhttp%253A%2F%2Ftieba.baidu.com%2Fp%2F136798230> target="\_blank" rel="nofollow ugc">我就静静的发图 ~ java 吧 这个帖子恶心了。我被 Java 吧封了，然后他们还 @ 我。我要赶紧写完然后想办法去跟他们对骂去。

3 发展前景

DBA 的发展前景我说不好。一些简单的工作。运维也是慢慢学会了。包括薪水，这个是我比较把握的。之前的薪水都是有迹可寻的，DBA 的薪水我接触的比较少，实在是没什么底气。

<p>1 年 <del>5 年: 8K</del>25K<br>

5 年以上: 20K~40K<br>

(我好心虚。。我只给一个 DBA 开过工资) </p>

<p>成长路径: </p>

<p>也不知道有啥成长路径, 感觉这个职业的物种越来越稀少了。</p>

<p>4.入门门槛</p>

<p>DBA 的入门门槛也是比较高的, 而且, 很少于有刚工始就是做 DBA 的, 大部分都是工程师转的所以呢, 至少要两到三年左右的时间才有可能做 DBA, 才能负责一些相对负责 DB 的事情。</p>

<p>5.哪些行业适合做 DBA</p>

<p>IT 界: 后端工程师, 运维工程师<br>

其他界: 无</p>

<p>其他行业的想转 DBA, 刚刚也说过了, 不合适, 只能先写代码, 再慢慢的转。</p>

<p>6.职业限制</p>

<p>这个职业最大的限制大概就是。。很容易无事可做, 前面有后台架构师蚕食, 后面有运维工程师入, 中小公司都不太会设置这个岗位, 所以有的时候会比较尴尬, 大概还有一些外包公司, 或者是传的 IT 企业, 会是由 DBA 去设计表, 去理清业务还有一些岗位, 其他的都不太好。</p>

<p>所以对于其他的各种持久化数据的备份和优化, 特别是对一些正在使用的框架, 又不够成熟的东, 更容易找到自己的位置。如果你要做 DBA 的话, 就记着, 跟持久层相关的优化, 数据安全, 备份要去了解--顺便再学点运维的东西</p>

<p>7.去哪里学, 怎么成长</p>

<p>基本上, DBA 可以由运维工程师和后台架构师去转, 也会有一些数据库方面的认证之类的的过。这个岗位啊, 跟其他的岗位真心有点不一样, 很少有刚毕业或者刚入行的人就要去做 DBA 的--你做不了亲, 跟数据相关的东西, 没个三年五年的经验, 真的很难做, 而且有些技术细节, 是必须要了的比较多的。</p>

<p>PS: 刚刚想到了, 云服务器里是 DBA 的一个好去处, 其实这里跟大数据也有一些相关的。这么的话, DBA 的职业需求度跟云服务器公司的发展是有比较密切的关系。但是总之, 不太可能变成一标配了。</p>

<p>七 运维<br>

=====我是被知乎删除了答案的分界线=====

====</p>

<p>先说点的别的, 早上发现知乎登录不了, 提示说我的账户异常, 然后发了封邮件, 账户能登了。<br>

然后发现答案被删, 找不到了。</p>

<p>伤心难过了七秒钟, 然后我一直都没有备份的习惯, 在修真院的群里喊了半天, 终于有一位小兄出来了, 给了我一份保存的文档。</p>

<p>不然还真的是不想再写下去了。以及, 其他的答案也让我很无语, 特别是居然并不知道 CSS 和 JS 可以是两个职位, 实在忍不下去了, 我会在后续讲敏捷开发的时候再重新提一下软件开发的全过程, 告诉你们为什么 CSS 和 JS 要分开, 那个才是我执行敏捷开发落地七年的干货, 当然, 时间不定, 只赞越多, 分享越多, 我觉得自己的答案越有价值, 心情就会好, 然后知道修真院的人也会越多。</p>

<p>我可能会出一个系列, 这个只是第一个, 来讲一下互联网的各个职业发展, 第二个, 讲敏捷开发第三个, 讲应该怎么样对 IT 技术人员的团队管理和绩效考核, 第四个, 讲如何做自己的职业规划以及如何找到一份更好的工作或者是怎么判断一个公司靠不靠谱, 第五个, 大概会聊一下创业公司怎么做以及如何用 MVP 的方式去创业。</p>

<p>这些如果你们都看到了, 我讲出来了, 也差不多是我这八年中经历和感受到的所有了, 至少会对联网的各个方面都有了一个全面的认识。</p>

<p>这里面不可避免的会有对修真院的介绍, 但是我能确保的是, 每一篇都是干货, 只不过不保证客, 也不保证有用。所以, 让赞和分享更多一些好么。</p>

<p>言归正传, 讲一下运维, 这是一个很神奇的行业, 一个在渐渐蚕食其他各种职业的职业, 一个本不是编程但是现在却编程越来越多的职业, 你要问为什么?</p>

<p>一个字: 云</p>

<p>说到运维, 就不得不提起来服务器, 以前呢, 服务器很贵, 带宽也很贵, 机房也贵, 电源也贵, 正啥都贵, 而且还不稳定, 那个时候运维工程师的大部分技能还是在硬件上, 或者是说硬件是必不可少的技能, 所以守着机房也是运维工程师比较常见的事儿。</p>

<p>而且有的时候还会兼着网管, 因此, 大部分的运维工作就是装个系统, 搭些内部服务软件, 如邮

, Wiki, Jenkins, VPN 等等。很多的时候还要会布线, 会路由器, 会交换机, 会做资产盘查, 把管的工作也做了。

当然线上的正常发布也少不了, 规范的点的公司, 会分成三套或者是四套环境 (我是不太能接受套环境), 就是 Dev, Test, Online (三个环境的叫法会有不同, 我现在越来越喜欢这种叫法), 开环境就是给开发人员自己瞎折腾用, 运维只需要把机器安装好, IP 分配好就行了, 开发过程中你们自瞎折腾, 每天部署多少次都可以--细节请等我讲敏捷开发的时候再说。测试环境就会严谨一些, 正常情况下每天只发布一次, 太频繁 QA 会杀人的。线上环境更严格, 一般来说不允许频繁上线, 所以只能周二或者是周四发布, 一周两次, 对于 WEB 来说, 其实是足够的 (但是很搞乱的是, 很多公司是周定需求, 周二开发, 周三测试, 周四上线, 周五改 Bug, 这些都会在讲敏捷开发的时候提到)。

所以测试环境和线上的发布都是运维来做, 因为有一个最简单的真理: 只要你把权限扔给工程师, 他就会偷偷重启服务以及偷偷改数据以及一不小心删掉某个文件, 或者是清空所有数据。

而且一定会被你抓到, 不要问我为什么, 我要是有线上服务器的器权限, 我也经常干这种事儿。。这比偷情的感觉要好多了, 一个人偷偷摸的瞒着所有人把 Bug 改掉。。

这个时候运维还是运维工程师, 直到虚拟化技术疯起来, 直接生成了阿里云, 金山云 (修真院的要合作伙伴), Ucloud 等这些改变了运维工程师命运走向的公司。

云时代来临了。。。视频云, IM 云, 图片云, APP 云等等各种跟业务相关的云服务也做起来了确切的说短信, 邮件也是各种云服务。

那么运维工程师的变成两极分化。一个是, 云服务商的运维工程师越来越重要了, 一些大的公司也在建很 NB 的运维自动化系统。另一个是, 非云服务商或者是非大公司的运维工作变简单了, 再也不用机房换硬盘了。。反而是要熟悉和了解各种云服务器的特性。

所以, 当我们现在说到运维工程师的时候, 其实是分成两个阶段的, 一个是传统的运维工程师做东西, 一个是现在云时代运维工程师做的内容, 如果你现在想做运维, 你恰恰是处在这个变革中间, 说不好未来的走向, 但是这是一脉相传的, 所以并不担心运维中技术体系的中断。

这就是对运维的一个简单的介绍, 哦对了。运维是一个公司里非常重要的岗位, 刚刚还忘了说到数据备份和灾备和监控。如果你们公司有值班手机, 一定是运维同学手里拿着。24 小时不准关机。

无论你洗澡, 上厕所, 还是吃饭, 或者是逛街, 还是跟妹子恩爱 (天地良心, 我没见过一个妹子运维, 听都没听过), 只要你收到报警, 你就必须第一时间远程连接到服务器, 然后通知相应的工程, 立刻解决问题, 或者是直接滚蛋。

通宵熬夜发版本更是正常的事儿。你要问为什么会通宵, 呵呵。

发了一次版本, 上线, 发现不正常, 于是回滚, 或者是直接在线上 Bug。于是运维同学就等着, 老婆也没用。然后过了两个小时, 大后期后端同学说好了, 可以发布了, 于是执行脚本, 然后又发现出了题, 接着回滚或者是在线上改。。。

你要问不是有测试环境么, 不是有 QA 么, 我就呵呵一笑。我不知道我说的关于运维的介绍是多还是少了。反正, 这个答案只是一个初稿, 如果点赞过 5000 (感觉是不太现实的), 我就再整理一。

1 工作内容:

运维的工作大概分成几个部分, 我对于修真院学习运维的少年们都这么说, 大概是:

A. 基础环境的搭建和常用软件的安装和配置 (兼网管的还有各种程控机), 常用软件指的是 SV, Git, 邮箱这种, 更细节的内容请参考修真院对于运维职业的介绍。

B. 日常的发布和维护, 如刚刚讲到的一样, 测试环境和线上环境的发布和记录, 原则上, 对线所有的变更都应该有记录。

C. 数据的备份和服务的监控&安全配置。各种数据, 都要做好备份和回滚的手段, 提前准备好各种紧急预案, 服务的监制要做好。安全始终都是不怎么被重点考虑的问题, 因为这个东西无底洞你永远不知道做到什么程度算是比较安全了, 所以大多数都是看着情况来。

D. 运维工具的编写。这一点在大的云服务器商里格外常见, 大公司也是一样的。

E. Hadoop 相关的大数据体系架构的运维, 确实有公司在用几百台机器做 Hadoop, 所以虽然常见, 我还是列出来吧。

这些就是运维的基本工作内容了, 然后如我之前所说, 运维一般来说, 大点的有两个, 小点的就个, 再小的就是工程师兼着。

而且大部分都在使用云服务器了。所以普通的运维工程师的工作都可以标准化和简单了。这里并没有到性能优化相关的部分, 实际上这也是运维工作中比较常见的内容。特别是在出现问题的时候, 运维远是一个在不出问题可能就被忽视, 一旦出问题就会被痛骂的。。。职业。

之前有句话说的好, 线上不出故障, 是不现实的。现实的做法是, 出了故障之后, 多久能快速恢

。 <br>

我对这个很认可，原话应该是出自一家云服务器公司还是。。我不记得了。 </p>

<p>但是这里不得不推荐一下【猎聘网的技术总监 <a href="https://ld246.com/forward?goto=htp%3A%2F%2Fwww.zhihu.com%2Fpeople%2Fliu-zhong-bing" target="\_blank" rel="nofollow gc">刘中兵</a>】，我的老同事和好基友，前段时间刚刚在一个神马神马会（我靠，我只知道是很厉害的会）上分享了猎聘在自动化运维上技术积累，如果 PPT 能够公开的话，回头我会把它也放运维的体系里。 </p>

<p>当然最好的结识这些技术大牛的方式就是加入修真院，显示自己的实力之后，让修真院来帮你牵引线。 </p>

<p>2 需要技能： </p>

<p>环境【Linux, Mysql, Oracle, MongoDB, Hadoop, nginx,apache,F5,lvs,vpn,iptables, svn git,memcache,redis】 <br>

工具【linux 常用工具, Mysql 常用工具, Jenkins, zabbix,nagios】 <br>

自动化运维【openstack,docker,ansible】 <br>

语言【shell,python】 </p>

<p>运维的技术体系也很多，基本上所有的工具都想要你会用，我会慢慢的整理出来一套运维体系技术架构，当然不只包括运维，包括我说的这些所有偏技术的职业。 <br>

所以，如果觉得我写的还不错，还算靠谱，就关注我，关注群，关注微信公众号。 <br>

如果觉得我是在扯淡，直接评论里骂我就好了。 </p>

<p>这里说到的其实并不算全，确切的说，做为一个运维工程师，严谨的性格是绝对少不了的，因为上的操作全部由你来执行，你必须细心，要严谨，如果做不到，就别考虑了。 <br>

真出了事，压力又把你压死。想想最近几年出现的运维事故吧。。。 </p>

<p>还有就是必须对 Linux 系统的底层了解的深入一些。如果你想做一些高级运维的话。Python 和 shell 脚本都是必不可少的，最常见的运维必备语言。 </p>

<p>3 发展前景 </p>

<p>写的有点不耐烦了，马丹如果不是为了宣传大修真院，我才不要做这种丹藤的事儿。我去勾搭小姑娘多好。 <br>

运维和前端其实有点想像，都是属于柳暗花明又一村的。 </p>

<p>本来运维的工作要被后端工程师蚕食了（后端工程师真是一个可怕的职业，可以这么说，没有后端工程师做不了的。。。包括产品，市场和运营） </p>

<p>然而随着云的发展，运维工程师简直是有了新生。 </p>

<p>1 年 <del>3 年：6K</del>20K <br>

3 年以上：28K~40K </p>

<p>成长路径： </p>

<p>网管-初级运维工程师-中级运维工程师-高级运维工程师 </p>

<p>运维做到这个级别，已经到头了。很少有运维的同事能做到技术叫监的级别的。所以，对于资质般的你来说，选择了做运维，就代表着天花板很近，但是对于大多数人来说，已经可以了。 </p>

<p>4.入门门槛 </p>

<p>OP 的门槛其实并没有太高，经常玩玩 Linux 会好一些，但是要学的东西，确实挺多的。处理线故障，也是一个经验积累的过程，所以说呢，基本上算是 0 门槛，谁都可以考虑做，只是要看你喜欢或者是不喜欢，进度快和慢的问题而已。 </p>

<p>5.哪些行业适合做运维 </p>

<p>IT 界：后端工程师，网管 <br>

其他界：All </p>

<p>刚刚介绍过了，门槛不太高，所以比较容易一些，而且，可以考虑做一段时间网管。半年或一年后转运维，网管的薪水比较低。3000 或者是 5000 已经到顶了。 </p>

<p>但是转到运维，就不一样了，简直是新生。 </p>

<p>6.职业限制 </p>

<p>运维的大部分事情其实都是跟业务无关的。这在之前介绍 CSS 的时候说到的，两个不需要懂业务逻辑的职业。 <br>

曾经有过运维工程师觉得工程师写的 Sql 不对帮他改了。。。然后被我狠狠的骂了一顿。这表示着：运维工程师必须按工程师所写的步骤去执行，不允许有任何的差错，如果出错了，如果没在测试环境跑，或者发现有问可以跟后端工程师沟通，但是，不可以改他们的东西。 </p>

<p>这样责任和权利分开。所以运维工程师能够相对有更多的时间去研究技术，然而坏处就是对一个

业了解的会比较少。

所以这个职业比较大的问题就是，想要再转行成其他的职业，已经很难了，只能一条路走到黑。比较难。

天花板和 CSS 一样，能看得到，但是！已经很适合了，薪水毕竟不算低呢。

7.去哪里学，怎么成长

当然是修真院了，第一个。

然后还有一些其他的运维培训，我不太了解，运维这个其实是几种职业培训里相对来说比较简单的。为大部分都是跟业务关系不大，是可以培养出来的。

自己找资料的话，只能根据技术的关键字去找了，还好大部分都不算特别难，等遇到难题的时候你也一定是工作了不，然后会遇到一些技术上大牛的运维。

八 QA

====我是被小郭惹烦了的分界线=====

“QA 最好的出路就是产品经理”。这是我对 QA 最好的认知。

我知道这里有无数反对的声音，我说的每一句话大概都会有认同和不认同的人，所以如果看到这句话喜欢，就表再继续看了。

QA 的这个职位，大致分成两个流派，一个是功能测试，一个是性能测试。功能测试就是指的是种点点点。然后看看功能和数据有没有问题。什么浏览器，什么版本，什么情况，能否复现。

性能测试就是指：用 Jemter, LoadRunner 等做压测，测跨后端人员的信心。

基本上就是这两种，很少有专门针对安全的层次去做测试的。那么问题就来了，工程师和 QA 之，到底谁该做功能测试，谁该做性能测试？

我姐告诉我说：工程师才应该做性能测试，因为只有他们才最应该知道自己开发的系统性能瓶颈什么地方。

然后也是因为这句话，我举一反三的瞬间懂了什么才是一个工程师，并且在不断的践行者这个理念：

如果你是一个后端工程师，你必须对服务器的线上数据了如指掌，有多少张表，在哪台机器上，分了个库，一个表里有多少条数据，数据的分布是什么样的，起了多少台 Service，占用多大的内存，高期的 TPS 有多少，CPU 的负载怎么样，页面总响应速度有多少，一个请求调用了几个方法，每个方执行时间是多少，是否加载了缓存，从缓存里加载的数据是多少个，大概多少 MS，访问一个数据库要多久。

你做不到这一点，你就没办法做好性能优化。所以，QA 很难做到这些，也完全没有必要做到这-实际上我们的团队是没有 QA 的。工程师负责所有的问题。

前端也是一样的，做了一个 App，耗电量，带宽，内存，兼容性，都是前端应该关心的问题。

所以我在这里给出的一个明确答案就是：性能测试，开发人员做，功能测试，QA 做。

但是这里的性能测试，并不是指的是普通的功能测试。实际上，开发人员应该自测一百遍再提交测试环境（我在讲到敏捷开发的话，会再次提到整个开发流程是怎么样的，以我在几家公司实施的经来看，这套敏捷开发流程，对于很多公司梳理内部结构都会有帮助。）

QA 做的应该是自动化的回归测试，边界条件测试，极限条件测试等种种很难测试得到的问题。

对了，再解释一下为什么要有 QA，以及为什么 Bug 总是测不完。

因为在我根本记不清的一篇文章里说到，哪怕是最简单的几个功能组合，想无穷穷举测完都很困--马丹，原话我记不住了。

所以这就是给无数擅长性能测试的 QA 判了死刑，你之所以能做性能测试，是因为你们的开发模和流程不对--别看我，我就是喜欢这么说，一点都不委婉。爱听听，不听拉倒。

但是 QA 真的不是一个很简单的活儿，可以这么说吧。QA 几乎是所有职业中，最熟悉系统的人甚至包括设计它出来，开发它出来的产品经理和各种工程师！

很多时候 QA 做的事情都是非常单调的，但是又必须有责任心的，拿我之前的话说，QA 就是最一道关口。

所以，如果你想成为一个 PM，你可以考虑先做 QA，当然，前提是你必须遇到一个愿意给你机做产品的好老大-比如说我。

So，接下来就开始来闲扯一下 QA 了。

1 工作内容：

QA 需要了解需求，很多公司会要求 QA 写测试用例，我觉得是扯淡。完全是在浪费时间。

通常开发三周，QA 测试的时间只有一周到一周半。还有关于提前写测试用例的，都不靠谱。

但是总之，了解需求，就算是不写也要自己知道自己测，这是 QA 的必备职责。

跟着就是等开发人员开发，这个时候也会有一些奇葩公司，要求 QA 提前介入的，理由是加快上周期。之前还有说过完成一个 Story 就要测试完整的，我也不想吐槽了。总之，如果你真的遇到了这的公司，你的工作就是测试，然后跟开发人员说不对。然后开发人员说改好了，然后再测。然后你大一声，好毛线啊。然后开发人员说，稍等，我先梳个头。然后再告诉你好了。然后你再说，好毛线啊然后开发人说。。。这次真的好了，于是你测了一下，真的好了，开发人员很 Nice 的说：完美！你很开心，但是没过几秒钟，你就会喊：什么鬼，之前的功能是好的，为什么又改回去了？

如果你呢。遇到好点的流程。会在上 QA 之前，打版本（我始终无法理解不打版本是什么流程）会 Demo，那么你可以有一个安心的测试环境了。

这个时候稍微正常点的工作，就是测试，然后把 Bug 录入到 Jira，禅道，或者 Bugzilla 等各种用的 Bug 管理工具。

跟着就是追踪 bug(如果你们公司有 Bug 处理流程的话)，如果有晨会就在晨会上说明，如果有会，就在周会上统计。

大部分的 QA 都比较羞涩，并不太敢提 Bug，并不太敢说哪个程序员的代码写的有问题。毕竟，这是一个必须要当面揭短的职业，哈哈哈哈哈。

基本上到这里 QA 的任务就没了。等着发布上线就好了。实际情况上发布上线的时候 QA 要等着程序员改代码，验证。。。

2 需要技能：

流程【Bug 修复流程，版本发布流程】

工具【禅道，BugZilla，Jira,Excel 表格来统计 Bug 数，自动化测试】

性格【严谨，耐心】

QA 里经常会嘲笑自己是技术 Team 里最没技术的一个。如果说你是一个标准的 QA 的话，真的用去特地研究一些压测工具的。

不过也确实可以了解一下，Jmeter 怎么用，TPS 是什么概念，90% 线是什么意思，PostGet 什么的。

自动化测试工具是我一直都强烈推荐的，无论是神马办法，只要你能做到哪怕只有一部分自动化测试，你做回归测试就很容易了。相信我，回归测试在每一个版本发布都是需要的。只不过有的时候只是需要跑一下脚本就好了。有问题再详细测一下。

严谨和耐性是非常非常难得的，也是 QA 最重要的能力。

当然也包括需求的理解能力啦。

3 发展前景

QA 的发展前景。。真的不算好，跟网管一样，如果说并不能转产品的话，20K 基本上就封顶了就算是有公司，愿意出 30K 让你们做性能测试，也是到天花板了。

1 年 <del>2 年：4K</del>15K

2 年以上：12K~20K

成长路径：

测试工程师-测试组 Leader-PM

如果能转到 PM，真的就很赞了，我推荐的时间应该是在半年到一年左右就开始转。

4.入门门槛

QA 并没有什么门槛，一般来说，是妹子比较多。汉子比较少，是汉子的，也多数是做点性能测相关的。但是请相信我，这绝对不是一个好的天赋加点方式，就算是性能测试做的再 6 又怎么样？

所以稍微懂一点 Bug 修复流程就好啦。勇敢的去做 QA 吧。。。

5.哪些行业适合做运维

IT 界：all

其他界：all

虽然是 IT 界的 All 都可以转 QA，但是说实话，我没有看到过一个从其他职业转到 QA 的。

也许除了运维。。

6.职业限制

职业限制也说过啦。总有重复的使用技能的厌倦感，毕竟这些技术水平都不需要太多。理解能力一些，严谨一些就能做。

说的直接点，从事五年的 QA 和从事一年的 QA，本质上并没有什么差别啊。

但是好处就是，如果你有心，你会对系统特别特别的了解，这对于你转行做 PM，是一个非常非常大的优势。再强调一遍，并不推荐 QA 转走技术路线。 </p>  
<p>7.去哪里学，怎么成长</p>  
<p>修真院并不提供 QA 的培训，暂时并没有，以后大概会讲一下 Bug 处理流程和敏捷开发流程。 </p>  
<p>所以，我也不知道哪里有 QA 的成长机会，一般都是在公司。 </p>  
<p>九 算法工程师<br>  
=====我是伤透了心的分界线===== </p>  
<p>“真正 NB 的程序员都是写算法的” ----我自己瞎编的。 <br>  
算法工程师其实是一个很虚的概念，但是在大数据忽悠了人这么多年以后，慢慢的也开始吃香起来了。很多经典的算法都是十年前，二十年前的，在算法本质上，诸多数学家或者是 IT 学者没有做过太多改进，大家的统一的结论就是：先拿到最应用，然后改进一点是一点。 </p>  
<p>可是即便是算法很多时候不完善，也已经暴发出了强大的战斗力。 </p>  
<p>那么什么是算法呢，什么是算法工程师呢？简单来说，算法就是瞎扯淡想的，从各种自然现象映到计算机上去的。什么爬山算法，模拟退火。。种种种种，反正就是解决问题啦。算法工程师就是吃了没事干，调个参数，训练个模型玩玩的闲人。 </p>  
<p>一般来说。说到底，还是在机器学习和数据挖掘中最常见的几种东西，首先就是分类。分类算法最常见的。分类一般会用来匹配，比如说，上周刚刚有朋友来问我怎么做穿衣搭配种的一套方案可以速推给其他同类用户的设计。那么首先就要收集用户的信息，设定特征值，然后训练分类模型等等。常见的就是新闻的分类，如我之前做过的财经新闻分类，把 2000 多家上市公司相关的新闻自动分到属行业。 </p>  
<p>分类完之后就是聚类，聚类用的其实不多啦，算是用的比较少的。我了解到的使用聚类算法的并不多。 </p>  
<p>实体识别是一个核心的技术。在新闻的领域用的是比较多的，确切的说，在文本领域用的都比较多。人名，地名，股票，时间 等等都是实体识别的一部分。 </p>  
<p>关系抽取是一个相对来讲比较靠上的应用。事件识也是一样，我们曾经做过上市公司高管变动的东西。也看不出有太大的价值。 </p>  
<p>协同推荐是在三四年前，还是五六年前的时候比较靠谱的，那个时候个性化推荐刚出来，各种商品，书籍，还有个性化阅读，音乐等等等等。然而也并没有爆发出什么好的应用来。 </p>  
<p>豆瓣和亚马逊据说推荐的还不错，我没仔细对比过。 </p>  
<p>简来说，常用的算法，指的就是这些。而算法工程师呢？并不是去研究算法。真的。 <br>  
算法工程师只是拿别人的算法：有的是自己实现，有的是直接拿开源软件来用，跑语料，找特征，调数，优化修改而已。 </p>  
<p>在我还年少无知的时候，对算法这个很东西痴迷，后来就渐渐明白了，算法是那些博士要做的。博士只负责来做算法。在公司呢。大部分都是要接近业务的，而且说实话，算法之间的差别，并不大。多时候都是减轻人工负担，提供决策支持，更简单的更有效的可能就是关键词过滤。 </p>  
<p>嗯。还有很多算法工程师，做到最后，只能跑个 Demo，连个像样的工程代码都写不出来。经常一个月调出一个模型，然后再优化一两个月，然后再优化一两个月，准确率可能都提高不了一个点。 </p>  
<p>所以这其实是一个跟业务结合特别密切的岗位，很多时候的先难知识都比较有用。而且也是个很见到效果的岗位，需要太多语料的配合了。 </p>  
<p>好的吧。大概就是这么多。 </p>  
<p>1 工作内容： </p>  
<p>算法工程师的工作内容，大部分时间都是在调优。就是调各种参数和语料，寻找特征，验证结果排除噪音。 <br>  
也会和 Hadoop 神马的打一些交道，mahout 神马的，我那个时候还在用 JavaML。现在并不知道有什么更好用的工具了。 </p>  
<p>有的时候还要自己去标注语料---当然大部分人不爱做这个事儿，会找漂亮的小编去做。 </p>  
<p>2 需要技能： </p>  
<p>基础【机器学习，数据挖掘】 <br>  
工具【Mahout, JavaML 等其他的算法工具集】 </p>  
<p>这是一个技术含量很高很高的职业。。。特别是对不懂这个行业的人来说。 <br>  
虽然有时候大部分做的东西也就那么回事儿，但是对于最基本的 SVM 算法，隐马模型，最大熵等还

有比较高的要求的。毕竟，懂还是不懂，差别挺大的。

<p>3 发展前景</p>

<p>我对算法工程师的薪资并不太了解，只是知道不会比较架构师的薪水低。这个职业的人数比较少最少也是需要硕士以上（硕士需要学的东西和写的论文真不是盖的，并不排除有很多都是抄的），所很难给出一个精准的数据。

<p>1 年 <del>3 年：10K</del>25K<br>

3 年以上：20K~40K</p>

<p>4.入门门槛</p>

<p>嗯。门槛最高的一个。最低也是要硕士，基本上都是博士在做。<br>

大部分都是有一个博士震楼，然后一帮硕士帮着写具体的代码。

<p>所以入门的门槛极高。

<p>5.哪些行业适合做算法工程师</p>

<p>IT 界：计算机算法相关的硕士或博士<br>

其他界：无</p>

<p>6.职业限制</p>

<p>怎么说呢。算法工程师很少会写工程代码，所以写代码的能力基本上等于零。这个是一个瓶颈。外，很少有机会能让你研究算法，只能去不断的跟业务数据去死磕。

<p>7.去哪里学，怎么成长</p>

<p>说不上来怎么成长，学这东西只能是跟着学校走。自学可能性太低了。成长的路径也就是去一些数据的公司--放心，只要是大数据的公司，基本上都是会考虑算法的。

<p>十 搜索工程师<br>

====人啊，就是这样一波又一波分界线====

<p>这儿说的搜索工程师是偏 Java 的。并不是百度，Google 这种大型的搜索，而是指一般的网站用的搜索工程师。

<p>搜索工程师的主要工作流程就是抓取，解析，去重，处理，索引，查询。每个环节都有比较强的法要求。我自己比较喜欢不区分搜索和应用组，一直觉得这两个组的人技能是相通的，能做架构的，法也不问题。只能写模块代码的，也可以写搜索的项目。

<p>但是实际上搜索还是值得拿出一个单独的领域的。每个领域都有比较深的知识点，因此单独拿出个搜索工程师的概念也并不算错。特别是 NoSQL DB 兴起之后，我还是比较起 Search+NoSqlDB 架构。

<p>搜索工程师算是属于偏高级别的了，一般来说也是要求硕士以上。最早的时候能做搜索的都挺厉害的，后来有一段时间之后，开源框架越来越成熟，搜索工程师也变成了一个门槛低的行业，直到 NoSql DB 的兴起，搜索并不是局限于是来做前端模糊搜索了，反而成了系统架构的一种方式，同样是可以做分布式，做分层，分负载均衡，性能比单纯用 DB 好太多，也解析了 Cache 的问题。特别是 Solr 和 ElasticSearch 的出现。

<p>或者这么说吧。以前的搜索有很大一部分都是为了前端的索引查询，只是满足于搜索一个功能的要。后来慢慢演变出来了很多不同的需求，其中一个比较关键的词就叫做 Facet。

<p>某种程度上来讲, Facet 才给搜索带来了新生，慢慢的才发现，原来并不是我在搜索框中才叫搜索<br>

原来电商的分类搜索也是搜索，原来很多列表也可以用搜索来做。

<p>这个又刚刚解决了 Mysql 的问题，各种 NoSqlDB 做这种索引的东西实在又有点有心无力。所搜索才能慢慢演变成一种架构方式。

<p>嗯。至少在我这里一直是要不断的去尝试一下。

<p>1 工作内容：</p>

<p>所以搜索现在其实分成两种。

<p>一种是传统的搜索。包括：<br>

A. 抓取<br>

B. 解析<br>

C. 去重<br>

D. 处理<br>

E. 索引<br>

F. 查询</p>

<p>另一种是做为架构的搜索。并不包括之前的抓取解析去重，只有索引和查询。<br>

A. 索引<br>

B. 查询</p>

<p>从传统的搜索来讲。先说 A 抓取。</p>

<p>抓取最需要解决的问题就是抓取的优先级，理论上讲，对一个网站上的访问，不能超过超过 2 秒次的频率，这个并不会给别人网站带来压力，被称为是友好的访问--表问我怎么知道这些的，我也不得了，反正并不严谨。</p>

<p>因此你需要有一个很好的机制来给 Url 打分，判断哪些抓过了。哪些没抓。这就是所谓的蜘蛛，叫爬虫。但是实际上我被这个名字误导好久，真心是丹藤。只是发一个网络请求而已，根本没有到对网站去，马丹刚开始想出这个名字的人抽象的能力简直弱爆了。我一开始就在幻像各种程序写成的小子会在对方网络上跑。打野的最后才明白原来这些虫子都不出门，只是在家吐口水而已。</p>

<p>B 解析<br>

解析并不是一件容易的事儿。解析要做的就是将网页的文本内容和其他没用的信息去掉。知道什么是题，什么是作者，什么是时间，或者是其他特定的需要。一般来说，都是有两种方式。一种是配模板不同的网站我有不同的模板。对方网站改版我也改。这种方式最精准。但是很愁人。<br>

所以另一种方式，自动解析，用算法或者是规则等自动解析内容的东西层出不穷。我到现在为止都觉得一切都是扯淡。寻找最大正文本块再反向解析是最科学的。</p>

<p>之前还听到过说，模板的自动生成。抓下来一个网站的数据，然后从上千个网页中自动解析网页析。这个我挺感兴趣的，但是一直没试过。</p>

<p>总之，如果只是抓一个网站上的数据（比如说抓取百度的搜索结果哈哈哈哈哈）。用 Python 的 Scrapy 这种神器，几分钟搞定还是没问题的。</p>

<p>C. 去重。<br>

去重是一个很好玩儿的事儿。不去重不行。因为国内的风气就是这样。你抄我，我抄你。而且。为了怕大家发现是抄袭，还会手动的增加一段，删除一段。做为一个智能的去重系统，你就必须把这些问题解决掉。</p>

<p>而且，随着时间的积累，你甚至有可能去做全量的去重（虽然我一直理解不了，为什么要做全量。六七年前我知道的最好的算法就是 SpotSig。嗯。一个很机智的算法。</p>

<p>D. 处理。<br>

这里的处理多了去了。所有的跟 NLP 相关的东西都可以放到这里。包括分词，词性识别，实体识别关系抽取，事件识别，分类，聚类，热点等等等等。</p>

<p>而且对于实时性的要求比较高。一篇新闻从别的网站发布，到我们的搜索系统，最快的可能就 30 秒或者是 1 两分钟--全部都是 30 秒不可能，因为抓取受限，只能说，最快的能看到这种新的数据。<p>

<p>E. 索引。<br>

索引是比较简单的。主要牵涉到的就是怎么做实时索引。大概是在内存里做两个索引，然后文件有一索引，跟着来回倒腾。。具体算法我不记得了，也是一个很机智的策略，在 Bob 中使用的。<br>

如果有 Facet 的话，也要考虑一下 Facet 的设置。</p>

<p>F. 查询。<br>

查询里主要是分词，并不牵涉到句法分析。还有一个高亮算法，以及 Facet 的查询。相对简单一些。过如果是分布式的搜索就会麻烦一些，又会出现如何分片的问题。是垂直分，还是按时间分？不同的务有不同的选择。</p>

<p>2 需要技能：</p>

<p>环境【Linux】<br>

框架【Luence,Slor,ElasticSearch,Cassandra,MongoDB】<br>

算法【倒排索引，权重计算公式，去重算法，Facet 搜索的原理，高亮算法，实时索引】</p>

<p>嗯。算法偏多。跟 NLP 的算法又有很大的不一样。比那些分类聚类的难度要低一些，又比做项和工程应用要再高一些。有点儿不上不下的感觉。</p>

<p>3 发展前景</p>

<p>现在来看的话，搜索工程师的作用越来越大，某种程度上和后台工程师是重合了，之前说到过拿的 Facet 来做架构。</p>

<p>1 年 <del>3 年：10K</del>20K<br>

3 年以上：15K~35K</p>

<p>成长路径：</p>

<p>搜索工程师-搜索组 leader</p>

<p>在我印象中，很少有做搜索的能够做 CTO 或者是架构师的，这跟搜索工程师以前的工作职责是



<p>3 发展前景</p>

<p>不知道。看不清楚。从某种程度来说，并不觉得对这些东西的人很 NB，但是总会有一些公司人的把这些东西加上去。所以应该说现在还处于比较混乱的地位。一般人要学的话，也确实是有的一些门，所以，也确实是属于偏高薪的职业。</p>

<p>1 年 <del>3 年：10K</del>20K<br>

3 年以上：15K~35K</p>

<p>成长路径：</p>

<p>说不上来，大数据这个部门以后还会不会存在都不一定。</p>

<p>4.入门门槛</p>

<p>入门的门槛算是比较高的吧。比不上 NLP，跟搜索比较类似，主要就是五六个框架的原理问题。<br>

全熟悉下来，也就是两三个月或者是半年左右的样子。</p>

<p>5.哪些行业适合做大数据</p>

<p>IT 界：后端工程师中有两年经验,或者是硕士,或者是 NLP<br>

其他界：无</p>

<p>6.职业限制</p>

<p>想想该怎么说，熟悉了这些架构之后，也并没有什么东西可以做。<br>

跟 NLP 相比呢，在算法的深度上比不过。跟原来的搜索工程师比较相似，但是现在搜索和 NoSqlDB 也成为了现在系统架构的主要场景，所以在工程方面也会偏差很多。</p>

<p>所以有点不伦不类，高不成低不就的样子。要说以后会不会发展的很好，或者说会不会是这个职的空缺很大，我并不知道。</p>

<p>7.去哪里学，怎么成长</p>

<p>不知道。有一些培训在做这个，但是，也只能说讲讲算法的东西。这个还是需要使用场景的。</p>

<p>十二 Android 工程师<br>

=====我原谅某人了我还是爱她的=====

<p>Android 工程师在两年前是最疯狂的时候。一个五年的 Java 工程师的薪水，比不上一个三年的 Android.一方面是苹果这种高端的装逼工程师成了很多人不想去学 IOS 的门槛，一方面是 Android 的用使用人数实在是太多了。</p>

<p>在三年前和四年前，个人开发者还很多的时候，经常盛传一个人开发了几个 Android 小应用。年收成过三百万或者是一千万的事儿。这事是真的还是假的我不知道，我身边还没有遇到过这种人，少都是听别人讲的，也许是有人挣了钱并不会告诉我们。</p>

<p>反正，现在啊。这种可能性越来越少了。所以 Android 工程师整体来讲呢。属于高潮过后的喘，慢慢平稳。不算高也不算少。另外，也是受 H5 的影响，现在有两种应用会使用 H5，一种是做的别简陋。低制烂造的，一种是懒得频繁发布版本，经常需要动态替换内容的。</p>

<p>前者是特殊产物，慢慢会被终结掉。后者呢，是正常的需求，以后的发展空间会越来越大。所以 Android 也是会受到 JS 的侵蚀。记着我之前说的。前端工程师一定要 Android，IOS，JS 全通，然主精一种。</p>

<p>还有，Android 的兼容性是最烦的了。碎片化很严重，版本，机型都是有问题的。特别是哪家公司的老板用的什么机型，这款机型的适配就必须做的很好。否则。。。</p>

<p>以及 Android 是用 Java 开发，资料神马的都是比较全的，上手也比较容易。不过。Android 的事儿，和 Java 后端做的事儿，几乎没有任何关联。所以，当说 Android。就是 Android。当说 Java。就是指后台架构。</p>

<p>1 工作内容：</p>

<p>Android 工程师的日常就是听产品经理讲需求，跟后端定接口，听 QA 反馈哪款机器不兼容，着申请各种测试机，以及悲催的用 Android 做 IOS 的控件。</p>

<p>马丹我其实并没有什么东西可以了，一瞬间不知道该对这个职业说些神马。嗯。想起来了。WEB 几乎都是要学会自己切图的，Android 往往会要求 UI 切图，原因是啥我不知道，大概是 Android 程序员都认为自己是程序员？而 CSS 工程师往往不会被当成人看？</p>

<p>还有还有，想起来了。各种闪退的修复，以及各种性能的调优，这些都是 Android 工程师的日。<br>

崩溃啊，慢啊。开了会就没电啊，如果是要做到跟视频音频相关的东西更悲剧。Android 和 IOS 之还需要做一些转码。</p>

<p>别的似乎也没了。嗯。我又想起来了。<br>

Android 其实是对地图，支付，IM，第三方登录，短信，统计，音视频，文件，照片等等要求比较的。业务上其实一般并不会会有太多的内容。也很少会要求一些动画效果。再加一些手势密码。</p>

<p>还有各种奇葩的问题。比如说 Android 的 65536 的问题。曾经折磨了一家公司半年多。</p>

<p>2 需要技能：</p>

<p>环境【Android Studio, Maven,Gradle】<br>

基础【数据结构，Java，计算机网络】<br>

组件【IM，地图，支付，拍照，视频，音频，统计，分享，手势密码】</p>

<p>做的好的会对控件做一些自适应的改动。会把网络请求也封装一下。处理好本地缓存，处理好内的占用。<br>

另外，接口的制订是比较重要的，而且，果断的使用 Rest 吧。表再使用一些过时又无聊的接口风格。</p>

<p>3 发展前景</p>

<p>单纯纯 Android 的话，我觉得发展前景不算大。如我之前所说的一样，差不多两到三年就已经用了。五年的 Android 并不见得比三年的 Android 强太多，或者站在公司的角度上来说，两到三年的 Android 基本上就够用了。</p>

<p>0 年 <del>1 年：6K</del>10K<br>

1 年 <del>3 年：8K</del>30K<br>

3 年~：20K~</p>

<p>成长路径：</p>

<p>Android 工程师-&gt;Android Leader-》前端 Leader-》</p>

<p>很少有 Android 工程师能够做到技术总监的。所以还是早做准备，Android, IOS 和 JS 通吃吧</p>

<p>4.入门门槛</p>

<p>门槛比较低了。比 JS 稍稍高一点儿，但也差别不大。主要是因为 Android 是纯客户端，要做就须什么都懂。但是 JS 就不太一样了，有的时候要求不高也能做事儿。</p>

<p>这就是 Android 比 JS 的门槛高一点的原因。简单来讲的话，最好是懂 Java 语法，然后是懂 Http 请求的含义，然后懂点儿 MVC 的概念，就可以了。</p>

<p>5.哪些行业适合做 Android</p>

<p>IT 界：都可以。哪怕你是产品，运维或者是 QA<br>

其他界：都可以，只要你有点儿基础。</p>

<p>6.职业限制</p>

<p>偏客户端的职业怎么说呢，做的时间长了，公司对你的要求其实并没有那么高了。这个是 Android 工程师有点儿不太好的地方。靠。现在已经快两点了。我有点神智不清了。</p>

<p>7.去哪里学，怎么成长</p>

<p>做 Android 培训的挺多的，基本上是四个月 1 万 6 的价格。<br>

所以，为嘛不考虑修真院的线上免费教学呢？</p>

<p>我对于看视频一直都不看好啊。</p>

<p>十三 IOS 工程师<br>

=====我好困啊我好困啊我好困啊=====

<p>什么是 IOS 工程师？培训 4 个月，薪水 10K 的工程师，这就是 IOS 工程师。<br>

虽然有一点夸张，但是也确实存在，只是那个 666 的人不太可能会是你而已。</p>

<p>IOS 之前是用 ObjectC 开发，现在又推出来了 Swift。至少从目前来看，Swift 并未成为主流和发 IOS 语言。所以现在要学的 IOS 的话，还是会以 OC 为主。</p>

<p>从难度上来讲，IOS 比 Android 要简单一些，Android 要兼容 N 多机型，系统底层的 API 也没封装的很好，IOS 的编译器偶尔会出点问题，但是总体来讲还是比较简单的。</p>

<p>只是 IOS 对 UI 的要求会比较高一些。做出的效果也会炫很多。</p>

<p>1 工作内容：</p>

<p>IOS 工程师的工作内容真的挺简单的，听需求，定接口。做个适配，抛弃一下 iphone4。<br>还有啥。。</p>

<p>马丹，以我为数不多的 IOS 知识来讲，真的不知道还有啥了。<br>

我知道的比较复杂系统也是各种背景高斯模糊，各种渐变，各种图片滤镜处理，其他并没有什么。<p>

<p>支付，地图，统计这些东西。 嗯。 </p>

<p>2 需要技能： </p>

<p>环境【Xcode】 <br>

基础【数据结构，Object，计算机网络】 <br>

组件【IM，地图，支付，拍照，视频，音频，统计，分享，手势密码】 </p>

<p>有崩的地方注意一下就好。 </p>

<p>3 发展前景</p>

<p>IOS 是属于高端的项目必备的，因为 IOS 的使用者比 Android 的还要多一些。 很多人都在解释什么自己不用 IOS，最被认可的答案就是：穷。 </p>

<p>所以 IOS 的发展前景怎么样？ 我不知道。但是在五年之内，IOS 的发展不会有问题。 </p>

<p>0 年 <del>1 年：6K</del>10K<br>

1 年 <del>3 年：8K</del>30K<br>

3 年~：20K~</p>

<p>成长路径： </p>

<p>IOS 工程师-&gt;IOS Leader-》前端 Leader-》 </p>

<p>4.入门门槛</p>

<p>IOS 的门槛比 Android 稍稍低点儿。毕竟不用做兼容真的是一件很 Happy 的事儿。 </p>

<p>5.哪些行业适合做 Android</p>

<p>IT 界：都可以。哪怕你是产品，运维或者是 QA<br>

其他界：都可以，只要你有点儿基础。 </p>

<p>6.职业限制</p>

<p>偏客户端的职业怎么说呢，做的时间长了，公司对你的要求其实并没有那么高了。这个是 IOS<b>

> 工程师有点儿不太好的地方。靠。现在已经快两点了。我有点神智不清了。 </p>

<p>7.去哪里学，怎么成长</p>

<p>做 IOS 培训的挺多的，基本上是四个月 1 万 6 的价格。 <br>

所以，为嘛不考虑修真院的线上免费教学呢？ </p>

<p>我对于看视频一直都不看好啊。 </p>

<p>十四 运营<br>

=====岁月没有假期 | 前行不需周末 | 再远的星空 | 也比不过内心的宁静=====

<p>运营，产品，技术是创业 Team 的三大主心骨，这从侧面也显示出来了运营同学的重要性。而且在收入上，运营也是和技术远远无法比拟的。你想想。技术人员做了七年，积累的是神马？一堆代码，一身颈椎病。产品人员做了七年，积累的是神马？一堆破原型，一身铁布衫。而运营人员呢？了七年之后，产品再烂，技术再烂，身边的渠道啊，人脉啊，行业的内幕啊这才是最有价值的。毕竟什么事情能离开了人？ </p>

<p>不过运营也不好做，因为很多脑残的公司都有这样的逻辑，通过一个脑残的想法拉来一个活不过集的替死鬼产品经理，折腾了老实巴交的技术人员半年多弄出了一个不伦不类的破系统，一切就绪，差一个 NB 的运营了。 </p>

<p>如果运营做的不好，就是运营的错，如果运营做的很好，但是用户还是不喜欢，那就是产品的错如果用户喜欢了，但是用不了系统总是闪退和 504，那就是技术的错。总之，产品技术运营都有错，一没错的就是老板自己。呃。不对，老板的错就是太信任手下这些人了。 </p>

<p>运营这个概念呢。在互联网上怎么兴起来的，说不上来，很多传统互联网公司在转行的时候，经会不清这些概念，我遇到过好几家公司，都以为做个网站，找个 CTO 就行了。CTO 下面管着产品，着技术，管着运营---不对，你太天真了，他们会以为产品运营都是技术做的。 </p>

<p>不过他们对如何能保障公司正常运转还是有了解的。所以对运营还是有一些模糊的概念，但是，丹，就算是我也对运营只有一些不清楚的理解而已，他们能理解多深刻呢？简单说，产品决定做什么技术来把它做出来，市场来把它推出去，运营来把它用起来。 </p>

<p>嗯。我本来计划写一点市场相关的东西，后来想想，自己也实在分不清市场和运营究竟该怎么分什么活动，策划，执行，渠道，品牌，内容，数据分析。并不能向技术一样分的这么清楚。 </p>

<p>所以就把市场和运营混在一起了。对于你们这些新入行的人来说，基本上是老板让做什么，就做什么呗。只是不管怎么说，运营是一个非常非常有前景的，在之前的介绍里，我想你们都看到了，有很职业不适合做 CTO，而后端是做 CTO 的最佳人选。然而，也大多数要止步于此，很多时候，我们要奔着 CEO 的话，你会发现，排名第一的是产品，排名第二的是运营，排名第三的。。没有了。 </p>

<p>而且，很多时候产品和运营是分不开的，也是要经常勾搭在一起的。当然，产品和技术也分不开技术面向产品，产品和运营沟通，运营收集用户反馈，交由产品设计，然后技术开发。嗯。</p>

<p>1 工作内容：</p>

<p>我之前给我们家超乖简单的介绍过，运营可以从几个大方面来分一下。</p>

<p>1.推广运营：这里包括各种论坛，知乎，贴吧，自媒体，大号，微博，SEO，导流等等所有的手，简单说就是用很多种方式来扩大自己网站或者是 WEB 的知名度，也有可能是策划一场活动，去人大大会堂办个结业典礼，也有可能是找某个漂亮明星做阵，来做一出秀，也有可能是找很多行业资源一来打造个什么百家蛋糕店对修真院弟子免单之类的东西。</p>

<p>2.内容运营：推广的话，肯定要有东西才会推，刚刚提到的很多内容，有很多都是要靠内容。以前是软文，现在的方式比较多了。什么易企秀，秀米之类的 H5 制作，但是绝大多数都是以 H5 主，大部分都是线上传播，形成热点，靠成口碑效应，一般来说，三天五天之内是传播的最佳时间，了三天或者是五天，热度就会直线下降。</p>

<p>这也是内容团队需要不断的生产内容的原因。并没有一招鲜吃遍天的说法。这篇文章同样，在最好的一个星期之内，瞬间挤进来 1000 人，现在呢，每天稳定在 4~5 个人左右。</p>

<p>3.系统运营：系统运营并不仅仅是包括给出一个后台。开个账户，找回个密码，查一下用户的数是否异常等。有的时候还包括你的系统的核心设置。比如说，要不要有 VIP，VIP 应该有什么特权，长时间合适，能谈到多少合作伙伴，应该分几个级别。这些跟产品要做的事情比较接近了，只是这些是要根据用户的表现不断的反馈来调整的。别的不说。现在大修真院就有好几个 QQ 群。这些 QQ 怎么设计，怎么管理，本身也是运营工作中的一种。</p>

<p>4.数据分析：这个其实是运营中最有技术含量的一种，也是让运营这个岗位比产品接地气的原因运营人员是最接近用户的，也是最了解用户行为的，在大数据的时代，对用户行为的分析真的不是一半点的重要-虽然在某种程度上，你一点不分析也可以，反正都没有人用。</p>

<p>数据分析包括很多，用户的留存率，日活，月活，召回率，转化率，跳出率，PV,UV，男女，职，地域，年龄，浏览时长，页面热点，浏览路径，高峰低谷。这些数据运营人员要头脑清晰，把这些学全部整合在一起，结合自己对业务的了解，然后制订定出后续的计划，看看是需要提高召回率，还需要提高转化率，看看系统的主要用户属性，借此去寻找更多的用户。</p>

<p>嗯。很有意思。<br>

所以运营的工作，有的时候是需要你去申请一个百度认证，提交各种资料，有的时候是需要你去找一合作伙伴，商量换个广告位，有的时候是需要你去谈几个合作商家，签几个战略协议，有的时候是需要你去策划一场活动，挨个去跟活动参与者去谈，有的时候是要去刷脸，请朋友来帮你站台，有的时候需要你坐下来，写一篇色情幽默的文章，更多的时候要收集用户的反馈，帮助他们解答为什么你们的站注册不了，短信验证码收不到等一系列问题，后台加个 V，换个广告，最重要的是每周或者是每月个报表，来给大家汇报一下当前的用户情况，给领导层做决策使用--虽然我一直都觉得完全是在扯淡再好的数据都比不上真实的了解这个行业，数据只能是辅以支持，二手数据的价值并不是没有，只是容易让人跑偏，就好像我今天刚好看到的说 Facebook 几年没改版，是因为在小范围内改版发现用户活跃度下降--真心觉得蛋疼。FaceBook 最终的价值并不应该仅仅是在页面上，这些会有影响，但是对不应该是一个决策的关键性因素。</p>

<p>然而做运营最重要的一点就是，不要去询问用户，而是去观察用户。对于这点我是非常认可的。</p>

<p>2 需要技能：</p>

<p>行业知识【各种行业知识，必须要懂行业，像推广，渠道，活动，SEO 这些都需要了解一些】<b>

人脉【各种刷人脸，凑热闹，换资源】<br>

文笔【文笔太差就找个小妹妹，风趣幽默点，否则没人看】<br>

数据分析【看数据，写各种各样的 PPT，然后做决策】</p>

<p>还有就是，跟产品和技术搞好关系。运营的职位其实是天生比产品和技术高一层，也正是因为这，才会经常招产品和技术骂，哈哈哈哈哈。特别是当一部分需要从运营这边传递到产品里的时候</p>

<p>如果你绕过了产品，直接用所谓的运营驱动去做开发，那更完蛋了。</p>

<p>3 发展前景</p>

<p>绝对好。无论技术发展到什么程度，无论产品变换多少次风格，运营的经营理念都不会变，积累人脉不会变。运营，可以做 100 年。</p>

<p>0 年 <del>1 年：6K</del>10K<br>

1 年 <del>3 年: 8K</del>20K<br>

3 年~ : 20K~ (百万年薪级别的运营多的是。) </p>

<p>成长路径: </p>

<p>运营专员-》部门经理-》运营总监-》CEO</p>

<p>4.入门门槛</p>

<p>这个说要门槛,有点谈不上。毕竟有深有浅,情商要高是必须的,其他的只能说有了更好。<br>好的吧,就算是情商低也可以去做编辑,每天只是对着后台系统一阵改。</p>

<p>5.哪些行业适合做运营</p>

<p>IT 界: 都可以。哪怕你是产品,运维或者是 QA<br>

其他界: 都可以,只要你有点儿基础。</p>

<p>6.职业限制</p>

<p>我想想,运营人员常常会受限于资源,也常常会迷失在数据里,很多运营的人,其实是做不到那高的职位的,对于他们来说,自己的行业积累和人脉积累没什么用处,因为只有你本身有价值,你才跟别人互相交换。所以有的时候会高不成低不就,退回去做产品和技术,又缺少专业技能,也是一场人间悲剧。</p>

<p>这大概也是运营这个岗位的职业限制吧,缺少干货,心里容易没底气。</p>

<p>7.去哪里学,怎么成长</p>

<p>有没有公司或者是机构做运营?我其实并不知道。修真院还有一个项目叫做草船云,是帮助传公司和创业公司做技术孵化服务,能够提供产品和技术的支持,但是运营这部分不太敢碰,嗯。所以营的最快成长方式,还是从一个岗位切入,传帮带。</p>

<p>数据分析神马的还好,我会在接下来的修真院系列中写一些运营相关的基础名词(所以快关注我收藏夹),能有一些入门的术语。其他的呢,大概只能是见多识广才能做。<br>

嗯。我可以在修真院尝试一下,看看能否有一些运营简单的培训,只是太浅了,并不能像技术一样,上钉钉的保证,只要你听话,认真,努力,就能成为一个初级工程师。</p>

<p>备注 14: 收尾了<del>么有了</del>累死了~这些不是一天写的,所以有时候心情好写的多有时候心情差,写的少。我打算过几天再整理一下~~尽可能的风格,结构都一样。</p>

<p>备注 13: 马丹,实在不知道怎么写了。</p>

<p>备注 12: 好烦躁啊好烦躁。</p>

<p>备注 11: 没什么。赶紧写完</p>

<p>备注 10: 嗯.越来越不想写了.因为没多少人看了.</p>

<p>备注 9: 今天又不开心。算法工程师也好久没接触了,也写的磕磕绊绊的水份比较多,其实我能的更好的,只是被最近知乎上的一些人恶心到了,也提不起来精神去做这个事了。</p>

<p>备注 8: 不开心。我是一个容易抽风的人,到了一个临界点我就会毁掉我自己辛苦做好的一切。以,如果觉得有用但是又没备份,就赶紧下手吧。我写贴子都不备份的。</p>

<p>备注 7: 今天啊。写的运维有点儿神智不清了。写东西都不是件容易的事儿,而且写的时间长了很容易发现自己思路走偏,变成套模板了。所以这个答案我想,还是要找时间再整理一下的。把硬广稍软一下,把内容编排的更合理一些,把图做的更好一些。。就这样,大家晚安~</p>

<p>备注 6: 这个 DBA 写的我真的是比较纠结,有太多很容易引起争议的内容(其他的内容也会引争论啦然而我不在乎,因为我有了我确定的答案)。所以这个是我比较希望的有其他人能够补充的。<br>

简单说,就是三个阶段,最早的 DBA 是处理优化的 SQL,后来变成了数据的安全和备份,再往后就集中到专业的云服务器公司,提供更专业的底层服务--这个看起来才是 DBA 最佳的归宿。</p>

<p>下次更新时间大概会在两三天之后。毕竟我还要回贴吧骂街。</p>

<p>备注 5: 今天本来没打算上来看知乎的。因为瞬间从知乎来到了群有近 600 人,折腾了我一天。<br>

本来想回去睡觉了,然后呢,发现有几个人在评论里举报说我是广告。</p>

<p>我就想说一句话,是广告没错,但是谁能觉得我说的没问题,就请过来补充交流,我向你学习。而如果是没事找事,纯属蛋疼,不好意思。我不是一个文明的人,我骂人句句带脏字。</p>

<p>以及,本文的内容之所以加上广告,正是以实例来证明文中所举的例子是真实发生的事情。</p>

<p>还有,加了群的小伙伴们请认真看群公告。</p>

<p>有些是我懂的,有些是我不懂的,我懂的我会说,我不懂的就乱说。如果你们觉得你们的答案我写的好,比我了解的多,比我更专业。欢迎你们去再回答好么。<br>

本答案中的所有观点都带有偏见,等下我就编辑到开篇,我不负责,也一定不客观。<br>就这样,睡觉去了。</p>

<p>备注 5:正准备继续编辑帖子呢，然后发现了中兵大人也在回贴。中兵大人是猎聘网的技术总监，是修真院的客卿长老，在 Erlang 和自动化运维部署方面都很犀利，出过好几本书。所以也顺便给猎做个广告，猎聘是我知道的猎头聚集地最多的地方，职位也挺多的，如果说有个几年的工作经验，其就可以去猎聘上试试啦。</p>

<p>备注 4: 其实我不太清楚，写的这些东西你们能否看得懂。毕竟我和一个 IT 新人的差距还是挺的，我也经常说，最好的就是师兄带师弟。如果你们有觉得我没说清楚的，评论里多说几句，给我一反馈。我可以保证我一定不会百分之百的听，哈哈哈哈哈。但是我会听，然后找到我自己觉得有道理，改进一下。</p>

<p>备注 3: 我每写一段，就会加一段备注，我得很认真的告诉你们，我说的每一句话，都是我从业十年以来的心得和体会。而且绝对不会为我的态度负责，还有，加入 IT 修真院的小伙伴要记着，IT 真院跟其他的培训结构截然不同，不要进去喊老师，老师不够凶残。还有，我并不是一个好脾气的人而且也丝毫不讲风度，可以随时随地跟任何一个人对喷，也特别喜欢这样。</p>

<p>写这么多，一方面是希望大家知道修真院的正确学习方式，表再看视频然后感觉懂了结果一行代都写不出来了，另一方面，也确实想给愿意转行的人提供一个参考。</p>

<p>备注 2: 好了。让我休息，休息一下。一天能写两个对我来说已经很难了。所以，再多点赞再多点感谢，再多点人加群，再多点认同给我--我还是一天只能写两个，哈哈哈哈哈，不过我会感谢你大概因为你这么做，可以让更多的人知道。</p>

<p>备注 1: 未完待续，一直在开会。让我歇会儿。先把广告打上。<br>

PS: 求助 为毛我不能给字体加黑，看起来好麻烦。</p>

<p>另外也推荐大家关注一下我的收藏 <a href="https://ld246.com/forward?goto=http%3A%2F2Fwww.zhuhu.com%2Fcollection%2F77083735" target="\_blank" rel="nofollow ugc">IT 修真系列 - 收藏夹</a><br>

接下来的帖子，我要写的话，还是会收藏到这个系列里。</p>

<p>第一篇对 IT 职业做了一个相对深入的介绍，给了想从事互联网职业的人一个了解各个职业的机会，已经有 4000+ 赞了，我想是真的帮助到了很多人。 <a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.zhuhu.com%2Fquestion%2F36991788%2Fanswer%2F69902291%3Fgroup\_id%3D685282044187512832%23comment-120731263" target="\_blank" rel="nofollow ugc">IT 行业都有哪些职位，初学者 (0 基础，新人) 该如何选择，才能够快速进入这个行业? - xdy1 回答</a></p>

<p>第二篇是对敏捷开发和项目管理做了一介绍，这篇帖子还没写完，其实它的价值远远大于第一篇马观花的介绍。只是大家还没有到能够理解敏捷开发的时候，所以我想了很久，决定暂时不写了。<br>

<a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.zhuhu.com%2Fquestion%2F39757751%2Fanswer%2F82927612%3Fgroup\_id%3D685376196313116672" target="\_blank" rel="nofollow ugc">互联网公司的“敏捷开发”流程是怎么样的，每个职位的角色和分工是什么? - xyl 的回答</a></p>

<p>这是第三篇，写这个贴子的动机是因为，在修真院有不少人在问，我要学到什么程度才能找到工，我是零基础啊，有没有视频和教程可以教我。 <a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.zhuhu.com%2Fquestion%2F40662462%2Fanswer%2F87649925%3Fgroup\_id%3D688073935970893824%23comment-122073494" target="\_blank" rel="nofollow ugc">有些 IT 初学者 (新人) 成长为技术大牛的真实经历? - xdy1 的回答</a></p>

<blockquote>

<p><strong>技能树.IT 修真院 <a href="https://ld246.com/forward?goto=https%3A%2F%2Flik.zhuhu.com%2F%3Ftarget%3Dhttp%253A%2F%2Fwww.jnshu.com%2Flogin%2F1%2F1108145" target="\_blank" rel="nofollow ugc">http://www.jnshu.com</a></strong></p>

</blockquote>

<p><strong>“我们相信人人都可以成为一个工程师，现在开始，找个师兄，带你入门，学习的路不再迷茫。这里是技能树.IT 修真院，初学者转行到互联网行业的聚集地。”</strong></p>

<p>感兴趣请加群：修真院 | 无名弟子 | 七群 562814359</p>