

# 专访篇@前端早读课

---

这22篇专访，是前1，2年刚开始做早读课的时候本着好奇的，向各位大神、技术专家发出邀约分享他们的前端学习旅程。其中有的是我入门的前辈，有的是年纪轻轻但很有上进的小伙子，有的还处于一线开发的前端开发们。这种文章很适合还在学校的萌新，或者工作两三年感到迷茫的前端er，希望他们的分享能给我们带来一个方向。

## 专访 | 淘宝虚拟互动实验室@渚薰大神

---

### 前言

今天男一号，早读君只看过他几次分享的宣传照跟分享的内容，主要都是跟动画有关。本着好奇的跟 @ 渚薰大神约聊聊这些年他在前端方面的成长经验跟选择。

正文从这开始~

### 自我介绍

大家好，我叫渚薰，来自淘宝虚拟互动实验室，是一名工作了10年的老前端。非常高兴能在早读课和大家见面，并借此机会给大家聊聊我的经历和我对前端职业的思考。

### 入行的经历

我跟编程打交道，是从小学五年级开始。那会在Windows3.1上，开始接触Pascal，觉得编程这个东西，能给我来非常大的愉悦感，所以就入这个坑了。小时候，最喜欢在小霸王上用QBasic编曲子。这些很古董的东西，一直伴随着我渡过了童年的时光，也让激励我在长大后同时实现了写代码和成立乐队（阿里996乐队）的梦想。

真正开始写网页，是在大学时期，之前初中和高中经历了几届ACM，成绩比较一般（也证明了自己不适合搞算法）。那会加入了学校的网络联盟，其实也就是专门做网站的。最早用的还是ASP，微软大法那套，IDE也还用的FrontPage，确实那个年代的IDE流行所见即所得，手写代码的比例大概20%。后来才慢慢改用网页三剑客，也真正意义上让自己蜕变成了一介码农。然后开始混当年最火的前端论坛51js，也许我的ID（渚薰大神是有了微博后取的）不够醒目，所以也就没混出像寒老师那样的人缘来（我现在也比较少混某乎）。

大学毕业后，第一家公司是金山软件，而且阴差阳错的成为了一名服务端工程师。因为有ASP的功底在，对服务端的开发也能得心应手，于是先后又入坑了Java和Python。那会我对一些后辈的教导是编程语言并不可怕，只要你懂得怎么使用它们，做任何事情都可以。现在嘛老了，早就忘记了

当年的Java、Python和PHP是怎么写的，而且反正有JS一统天下，也懒的去学其它语言了（小孩子不要学我:-)）。在金山的最后一年，开始真正做前端的一些事情。不过金山算一个传统软件公司，所以并没有特别认知前端这个职业。当时就感觉到，自己还是希望捣鼓网页，看起来一些酷酷的东西，也感觉自己在前端路上越走越远，所以就从金山离职了。题外话，当年在金山还挺崇尚个人英雄的，比如求伯君、雷军，因此我到现在还一直是雷军和小米的粉丝。

从金山离职到加入阿里中间有2年多，和几个哥们折腾了下开奶茶店和建外卖网站（很可惜没做起来，不然绝对是中国最早的外卖O2O~\_~）。没折腾起来的原因很多，最主要的还是资源问题，比如硬件资源和人脉资源，这些都是硬伤。所以奉劝有创业冲动的年轻人，在没有积累够一定资本和人脉之前，千万不要尝试这条路，创业的甜头真的很难尝到，如果你把它当做一次涉世的机会到是不错的，至少我在开奶茶店时，纯粹觉得实业挺好玩的，做奶茶也挺好玩的，以至于后来用程序员的角度看待送外卖这件事，才有了后面正儿八经的创业。虽然缺了不少火候，但经历告诉我，机会真的都在身边，抓不抓得住并不是自己能决定的。之后在12年加入了阿里当年的一淘UX，成为一名前端工程师。

在加入阿里后，正好赶上移动互联网的浪潮，所以在手机淘宝一呆就是五年。作为一名正经的前端工程师，手淘确实给了我不少证明自己的机会，也正因此让我能一直长情的留在手机淘宝的大家庭里。到了工作后的六七年，每个人心里需要更加清楚职业道路应该怎么走，从不同的角度看待自己所在的公司和所在的部门，必须要有足够的判断力现在所处的方方面面适不适合自己，而不只是抱怨着离开。要知道，就算是公司内转岗，到了一个新的团队很可能面临着一切的重新开始，比如技术栈，日常工作流，甚至人脉和社交。其中，重新建立人脉和社交的成本是最高的。到了这个阶段，我会考虑很多人脉关系的因素，当然换个新环境也是一种挑战，但一定要想清楚自己追求的到底是一种新鲜和刺激感，还是用合适的方式证明自己。当原本的环境和团队给你了很多机会的时候，是不是在困难的时候也应该回报它们。

## 我在阿里从事的方向

在阿里的头三年，做了手淘前端架构的工作，整理整理规范，建立建立工程体系和资源库，推广推广热门的技术，打造打造像样的轮子。这样的工作，很多做架构的同学都会经历过，个人亲力亲为的时候很多，也或多或少有些感触。当团队内大多数同学都扑倒在产品业务中时，自己一个人在折腾各种东西，并伺机让其它同学来用这些东西。所以当年的KPI，就是各种落地，有时业务不给力，还得自己来寻求能落地的项目。所以，架构工作是看起来光鲜亮丽，实则做起来忍辱吞声的一件事情。因为短期是看不到多少成果的，而且兄弟团队之间都做着很多类似架构的工作，并都发誓要推广到全集团，这种竞争可想而知，更何况在整个前端界了。但实际上，真正的架构都是为某个团队量身定做的，要复制一份到其它团队阻力并不小，特别从业务角度产出的架构更是如此。所以，如果仍在担任这个工作的同学，首先服务好自己团队，服务好当前的业务，再者是不断的提升方法学上的能力，而不是某个具体架构的技术。

在手淘第四年，因为组织架构的调整，我顺其自然的加入了Weex团队。在从事Weex工作中，我获得了不少在运作一个大型开源项目时的宝贵经验。从工程化质量，到项目推进模式，从开源社区工作，到业界宣传发声，每一个都是我以前没有经历过或者深入过的。Weex团队带给我的并不是开发一个热门解决方案的能力，而是努力做到完美的态度。因此在手淘第五年，我选择带着这种热情和态度，去追求最初的一个梦想：动画和游戏。

说我自己是个重度游戏控，一点都不为过，接触FC的日子（小学2年级）比我开始码代码的日子还早，可以几乎清楚的说出从雅利达时代到PS4时代的总总变迁。所以，小时候的梦想是去日本的大厂开发游戏。现实让我把梦想变成在阿里这个大厂做动画和游戏。就像某职业眼中的其它职业的梗，H5动画/游戏行业的开发者并不愿在他们自己头上扣个前端的帽子，因为开发动画和游戏看起来要高深的多，前端很多时候是低门槛入行的代名词。而且前端在做的那点动效，确实没法跟动画挂上钩。选择这个方向，真的是为了兴趣和梦想。

现在，我在动画和游戏这个十分专业领域摸索一些东西。提到动画，可能很少有话题专门来讲述前端如何开发动画，它一般和性能挂钩在一起。但在做了很多次双11的大型红包『游戏』之后，觉得以往前端做动画的方式实在是太原始了。引用我们团队经历了无数次营销活动的小伙伴话说，做一个动画项目80%的精力是在不断的调参数，这并不是理想的工作方式。所以我们成立了专项，开始研究业界（动画公司，广告公司，游戏公司）是如何开发一个动画项目的。说实话，隔行如隔山。互联网公司的开发模式和这些业界公司真的完全不一样，单从职业分工来说，这些业界公司会有动画工程师的角色。而在我们这，界线很模糊，有时像是设计师的工作，有时又像前端的工作，还都不够专业。如果，我们能像业界这些公司一样工作，是否能给现在的团队和手淘带来一些变化呢？这是我们团队正在探索的。

## 如何在前端娱乐圈摸爬滚打

在一个足够细分的领域，可以挖掘许多意想不到的内容，前端圈已经不像10年前、5年前只是掌握HTML/CSS/JS这么简单了。我们不谈JS的蓬勃发展，但就前端的业务形态来细分，现在动画/3D/VR也是一门技术活。它不但冲击着前端工程师的知识领域，而且也被证明着HTML/CSS/JS也就只是工具而已。在这个领域，JS/CSS的什么高级特性看起来已经不那么重要。比如Threejs用最基础的原型链继承一直走过了80多个版本，还一直活跃在WebGL的圈子里，可见各种前端模式对于它们来说，又有何用。诚然像Vue/React，甚至Rxjs和Redux这些新奇的思想给前端圈带来了不少骚动，但在动画和3D的专业领域，更多讨论的是普通数组和类型数组哪个更快，如何解决单线程CPU运算慢的问题，如何模拟一种材质的表面纹理，如何实现自然光的反射等等更接近学术的东西。

在前端圈，有了跨栈和跨端之后，我也想来卖个概念叫做跨界。从前端圈跨到游戏圈，前端知识看起来已经变得微不足道了，任何一个前端大牛，在跨界后势必还是一名小白。但，每个人应该感谢前端知识的积累对于其它行业的正向作用。如果让寒老师来写个Threejs能比它更好吗？不能！但一定更好用！很多同学，在刚开始入门前端时，发现前端也就是HTML/CSS/JS，走着走着发现HTML/CSS/JS是个啥都不知道。这句话有有贬义也有褒义，入了这行，有很多坑让你跳，只怕你小心翼翼的沿着边缘蹒跚前行。

最后，文中有一句话：各种落地，有时业务不给力，还得自己来寻求能落地的项目。做过新技术落地的童鞋，对这个应该有更强的认同感。

## 专访 | @黯羽轻扬，CSS2.1规范中文版译者

### 前言

前段时间的某日依旧打开新浪微博，发现被@了好多次，本着好奇去看看到底是什么东西，一看不得了是一位今年刚要毕业的小朋友花了两个月的时间翻译了css2.1规范。这可是大工程，所以早读君就好奇跟他聊聊了。那么今天的文章就来自于他@黯羽轻扬，一位低调的前端新人。

正文从这开始～

## 零.我

接触前端差不多2年，今年6月份出厂（毕业），一个[用心写博客](#)的前端新人。

最近完成了CSS 2.1 spec的翻译，然后腆着脸用微博@了好多和我不熟的前辈，再后来的事情就完全出乎我的意料了：前辈们热情帮吼，支持鼓励，前端同行的问候.....

当时只是碰碰运气，想通过前辈们的影响力，告诉大家“有中文课本了”。因为个人功力不够，有些点理解得不准确不透彻（比如当时很难理解的clearance），只有更多人仔细阅读这份文档，迭代修缮，它才可能变得更好。

如果文档完备，我们这些新人入行就更容易，也会更合格。

## 一.原因

想看，但没有中文版，英文的一遍看不懂，所以先翻译，一遍一遍看。

有了目标，一切都水到渠成：

平均日翻500行HTML源码，需要8小时，每周翻3-4个500行，剩余的2-3天挖旧东西，尝试新东西，写博客。这样，2个月完成。

投入大量精力完成了一个可复用的snippet，相信您也会希望让所有人都知道它，以节省大家的时间。

## 二.前端窘境

学习CSS应该从熟读规范开始，而大多数像我这样的新人都不是从规范开始接触前端的。以致于：

- 面试的时候问个css的position属性能刷掉一半的人这是啥情况.....
- 其实这问题我本来打算的是可以顺着一路扯到normalflow、containing block、bfc、margin collapse, base line, writing mode, bidi, 这样一路问下去的，奈何第一个问题（亲我真的只问了position有哪些取值和行为啊）就悲剧了.....

（以上引自winter前辈的[谈谈面试与面试题](#)）

1年前，第一次看到这段话时，觉得自己很不合格，当时我只知道position: static|absolute|relative|fixed（其实还有inherit），对其用法知道一点，但似乎已经以入门级前端自居了，还妄想着加入大牛团队学习进步。

如果在我入门时，能有前辈告诉我先熟读规范，稳扎稳打，那该有多好？至少不会现在马上出厂了，要去做所谓的“前端开发工程师”，仔细一想还很心虚。

### 三.基本功

BFC、margin collapse等等这些理论性的东西有用吗？没有用吗？最初我也觉得“虚的”没用，知道BFC的人就比我强么？不知道这些就搞不定页面布局吗？

慢慢发现功力深厚的前辈“总拿规范说事”（当然，只是玩笑话），并不是说不懂BFC就会被鄙视，但想走得更远的话，那些欠下的东西早晚要还，何况作为前端，连课本都没读过，一提基础就心虚，怎么说得过去？

熟读规范的好处是能对布局知根知底，知道一些奇怪的表现背后的原因，而不是记得或者能找到解法。有一位擅长运用CSS规范搞定各种问题的前辈，我不说您也知道他是谁，对于这样的人，我们多半只是敬佩，却从未认真思考过他为什么这么厉害。其实如果您仔细读过规范，就能发现有些锦囊一样的神奇解法，在规范里写得清清楚楚，他只是读得熟想得多而已。

术业有专攻，作为前端，熟读规范绝对是最基本要求。

### 四.我们所缺少的

先看看W3C日文文档：

<https://www.w3.org/2005/11/Translations/Lists/ListLang-ja.html>

其完整程度简直令人惊叹，甚至CSS3的一些模块老早就已经翻译好了。以下公开邮件：

You may know this already, there is already a Japanese translation (CR) exists:

<https://triple-underscore.github.io/css-writing-modes-ja.html>

Yes, I know.

...It leads that the translation refers recent spec, but sentences are a bit hard to read. The latter tries to translate in fluent sentences. The maintenance cost becomes high, but it would be easy to read.

...So I decide to translate.

（以上引自2016-5-2 [Re: Japanese intention of translation](#)）

大意就是“我知道已经有日文版了，但翻得不好，所以还是决定重搞”。对比我们那寥寥31篇中文文档，实在汗颜。

完备的文档是行业发展的基础，我们正是缺少这些东西的支撑，才导致现在的前端窘境。翻译文档是所有前端开发者的责任，而不是中文小组的义务，W3C所有文档翻译者都是Volunteer，参与翻译没有任何门槛，与W3C工作人员的交流也并没有您想象的那么难：

Dear translator,

Thanks for your translation...

...And would you also please let me know your Full name both in Chinese and English to allow us

add your info into the Database?

Please feel free to contact me if you have any questions.

对方很亲切的，一点点都不可怕，真的。

如果您有前端工作积累的经验，恰好还会点英文，Please feel free to volunteer:

<https://www.w3.org/Consortium/Translation/#volunteer>

如果您确实想做点什么，建议先不要乱来，请注意：

- 确定文档状态。是Working Draft, Proposed, Recommendation, Candidate Recommendation还是Recommendation? 建议只翻译Recommendation（推荐标准）
- 确认不存在中文版。在W3C数据库以各种相关关键字搜索目标文档，并搜索公开信，确认没有Intention（有人要翻）或者Completed（有人翻完了）
- 发送翻译意向（Intention）。告诉大家，我说了要翻这个，别重复了
- Just do it.

说出来的话就得尽全力做到，只有Intention没有下文可不好。所以，翻译请慎重，建议从篇幅短点的开始，但无论怎样，多一篇极短的文档对行业来说都是极好的。

## 五.致FEers

### 1.致前辈

您可能也觉得行业现在的新人功底差、素质低、眼睛高，可能也认真思考过原因，但最后或许也觉得这些东西是nonsense（知道症结所在，却有心无力）。

当然，您没有大量的时间，也没有那些精力来做年轻人的体力活儿。但换个角度，年轻人翻译过程中遇到了各种困难，无法理解某些概念，非常希望能有一位经验丰富的前辈作出解释。而对您来说这些只是基本常识，一两句话就能说清楚。

但问题是他不知道您，更不知道您愿意帮助他。解决这个问题其实不难，各进一步，只要您对文档工作表示关注和支持（中文小组？或者其它更靠谱的方式），同时只要我们能通过一些方式找到您，这个结就解开了。

另一方面，面试时发现对方功底差到无法忍受，不妨直说，这样他对行业都好。（如果早有前辈这样痛骂我的话，我现在得多感激他呀，可惜开悟的晚了）。

这些话本不应该由稚嫩的前端新人来说，但老老大说过“光脚的不怕穿鞋的”，谢谢@情封前辈给我勇气

### 2.致像我一样的新人

如果接触前端不久，还处于学习积累阶段的话，强烈建议从规范开始学习。如果没有翻译得不错



的中文文档，自己翻一遍又何妨，这可是原汁原味的第一手资料啊，绝对不吃亏。

翻译过程中如果遇到有困难，就大胆去问前辈，大不了不回复而已，不行就多问几个。问不问是我的事，帮不帮是他的事，先做好自己的事。（P.S.求助时应该注意措辞突出重点，反正我丢给前辈们的求助邮件回复率是100%）

另一方面，前端东西确实非常多，而我们精力有限：

- 想必很多小伙伴都没见过。不知大家有没有跟我一样的感觉，去厕所蹲了个大号，再回到办公室就会遇到之前没见过的前端新特性。
- 像我这种只学HTML和CSS都有些应接不暇，我想，那些CSS3,HTML5,ES6/ES7,React,Angular,bootstrap,postcss,node都会的小伙伴，应该每天都只睡3-4个小时吧，好厉害！

前辈调侃时可能也觉得这是nonsense（他不说穿，让用心的人自己去悟），@余果前辈把它说穿了：“先精后广，一专多长”是成为全栈工程师的锦囊。

去年实习离职时向部门各前辈索要了一堆技能书，从bash到PHP|node到backbone到ionic到数学到产品（胃口略大），目前只吃透了一两本，因为在开始点技能之前看到了@余果前辈的书，暂时放弃了成为三代目（忍术之神）的幻想，很庆幸当时的选择。

最后一点，如果遭遇了困境，请大胆地向前辈求助，网络中我们之间都是0距离的，前端圈的前辈们非常亲切，都很乐意伸出援手（注意方式方法的话，没有搞不定的人）。亲身经历：

- @张鑫旭看过他很多篇博文
- @寒冬winter 他的博文告诉我应该稳扎稳打
- @前端开发博客看过很多篇博文
- @情封朋友告诉我有一篇关于点击穿透问题的文章被他收下了
- @愚人码头看过很多篇博文
- @Barret李靖从博客园追到小胡子哥主页
- @朴灵最近才基本吃透node那本书
- @CSS魔法看过一些博文
- @司徒正美想先学飞（打算点框架设计技能），后来失败了

如开头所述，我@了这些和我不熟的前辈，我知道他，他不认得我。万万没想到前辈们能非常热心地帮我，所以，大胆去做，前辈们都很愿意帮助我们。

## 后语

CSS2.1规范中文版，已经全部翻译完了，有兴趣的大家可以去看看，记得fork

进度：已完结（2016.5.23）

官方文档: <http://www.w3.org/TR/2011/REC-CSS2-20110607/>

W3C索引: <https://www.w3.org/2005/11/Translations/Query?titleMatch=css+2.1&lang=zh-hans&search1=Submit>

中文翻译: <http://www.ayqy.net/doc/css2-1/cover.html>

GitHub: <https://github.com/ayqy/CSS2-1>

## 专访 | 《编写高质量代码--web前端开发修炼之道》作者@阿当

### 前言

前端早读课第十三期专访来了。曾回忆2010年时候看过他写的书，近段时间微博上的各种“撕逼”，以及广开收徒事情，让他站在了风口浪尖，有人支持，有人反对。那么今天我们就来看看阿当前辈在自己的前端生涯的经历。

正文从这开始～

### 自我介绍，所服务的公司

我是曹刘阳，圈里的朋友们叫我阿当。现在在岗岭集团工作，互联网医疗领域。

**你是基于什么情景上选择了前端工程师，以及在所服务的几个公司来看，你觉得哪一段经历对你影响最大，为什么？**

说来话长，大学时我的专业是计科系，学校里教些c啊、java啊、软件工程啊什么的，死活提不起兴趣，总觉得这玩意儿离看得见摸得着的“作品”有太大的距离。直到开了网页设计和photoshop这两门选修课，才眼睛一亮，觉得能做些什么了。另外，我读大学那会儿流行flash，闪客。我觉得这东西很好玩，就自学了flash。虽然很多人觉得c语言是最好的入门语言，但于我来说，真正为我打开编程之门的，却是as2，没错，flash的脚本语言。flash方便快捷，加上as2写脚本，不用太多的知识储备就能做出些好玩的东西了。如果不是因为flash在今天已经快退出历史舞台了，我仍然觉得它是编程入门最好的选择。

大学时我非常喜欢读书，主要是ps、flash和网页制作，当时网页制作方面的书low到不像话，经常都只告诉你如何操作dreamweaver，偶尔有一点讲js的也是xxx特效300例之类的，只有代码片断，没有基础讲解。这种非常虐心的局面一直持续到06年下半年，人邮出版社引进了一堆国外的优秀前端书籍进来，比如《dom javascript编程艺术》《javascript高级程序设计》《ajax基础教程》《css基础教程》，同时国内也有了非常好的原创书籍《css网站布局实录》。这些书都在讲“基础”，比之前我读到的那些代码片断类的书好太多太多了。我开始大量吸收这些书的知识。我有个习惯，就是喜欢把未知的东西全部补齐后，在脑子里有个清晰的系统化的脉络后，再开始动手练习，书没读完我就像欠了债一样不舒服，拼命读，快的话一周一本，慢的话一个月一本。我觉得这些书上的知识比学校里老师讲的东西好太多，课都懒得去上了，除了去吃饭，就趴在床上看书。我有另一个习惯，就是喜欢自己按照自己的理解和记忆去练习，不喜欢照着书敲代码，哪怕



自己写的效果比书上的效果差也不要紧，重要的是通过练习去验证我的理解是否到位。

虽然我的兴趣在前端，但当时市面上并没有前端工程师这个岗位。当时要么是网页设计师，要么是工程师。我想做工程师。大学毕业前一年，我花了大量时间在后台方面，主要学j2ee，但出去找工作时发现当时市场上对j2ee的需求量非常少，不得已换了定位，干脆找网页设计师的工作好了。然后大学时自学的flash、选修的网页制作和photoshop反而帮我找到了第一份工作。然后第一份工作的title就成了网页设计师，从ps、ai、flash到网页制作全包。我大量的时间花在了设计上，配色啊、布局啊、点线面、冷暖色什么的。离后端越来越远，离前端也是若即若离。当时前端做网页，主流的做法其实还是dreamweaver里拉表格，还流行什么1\*1透明png的黑魔法。公司的两位老前辈用的是asp + table布局，我跟公司推荐说改用css布局吧，说这是未来趋势，老板问我说，为啥要用css布局呢？有啥好处呢？我balabala说了一堆，但老板是生意人，不懂技术，没有听到什么他感兴趣的点，不置可否，然后这事就不了了之了。曾经一度我想过干脆转设计师得了，当时流行两种设计风格，1是简洁的欧美风，2是漂亮花哨的韩国风。我买了一堆的美术设计方面的书，一心想做个一流的设计大师，走韩国风！

但生活总是爱和人开玩笑，我阴差阳错还是回到了前端圈。我的第一份工作是在深圳，华强北，我很喜欢两站地之外的深圳购书中心，经常下了班走两站地去深圳购书中心看书，当时薪水低，没钱买，我就坐在地上看，翻看过大量前端相关的书。工作快一年之际，通过大量阅读书籍，我的前端和flash编程能力都进步飞快，当时的年轻人和正在的年轻人也差不多，总喜欢加入些同行的群吹牛逼，吹的内容无非3个：公司、薪水、作品。我的作品总能在群里引起很多人的赞叹，猜测我的薪水应该很高，但事实上我的薪水却比很多能力一般的人少，这引起了我极大的心理不平衡。终于在一次心理极度不平衡下，我产生的跳槽的念头，我要找识货的东家给个对得起我能力的薪水。07年下半年，我找遍了51job、智联招聘，蓝色理想的招聘版，在深圳这个大城市却找不到前端工程师这个岗位需求。然后我将目光转到了北京，发现北京这时有大量的前端工程师需求，很多公司也开始有了UED团队。然后我就到了北京。

一开始去了家小创业公司，两个老板和我一个打工的，一共三个人。两个老板，一个是写php的，另一个是设计师，然后我就成了前端。北京的技术圈还是识货的，他们能理解到前端工程师的作用，很爽快地double了我的薪水，然后我就终于心理平衡了，成为班上同学里薪水上涨最快的一位。这份工作是我真正前端之路的开始，这家公司做的是web2.0的网站，对富媒体应用要求很高，各种效果，mp3音乐播放器、flash相册、多文件上传、自定义皮肤、各种js效果。然后这家公司为了生存也接外包需求。然后我就从前端开发、flash开发、flex开发到php开发，除了设计不需要，全部锻炼了个遍。平时加班、周末加班、多的时候三个项目并行开发，其中有的项目是我一个人全栈完成。而当时我虽然看过不少书，但其实刚从一个拉table的网页设计师转型过来，压力和挑战可想而知。也许确实是我对编程有天赋吧，也许是很会挑书和读书，我抗了过来。这段时间对我的编程能力和抗压能力有了质的提高。不逼自己一把，永远完不成质变的，就是在这段时间我认识到这个道理。之前看过再多书，也不如在这段时间我成长得快。值得一提的是，正是在这段时间jquery才开始红起来的，所以从深圳时开始我前后写了一年的原生js，这让我对js底层有了更深的理解。

之后有机会去了雅虎。没去过大公司的人，会特别向往大公司，特别是QQ群里一群人整天拿大公司说事，一个人自报家门“淘宝”“新浪”“腾讯”，然后一群人各种跪舔。我很向往去大公司，管它哪

家大公司，我也要一个招牌，降薪去都行。然后很幸运去了雅虎，hr问我你薪资要求多少啊，我说你们看着给吧，没有要求。去雅虎对我来说，有非常重要的意义，不是公司光环，而是我遇到了YUI。当时YUI还是YUI2，雅虎内部强制要求所有前端使用YUI，不能使用jquery之类的其他库。而在遇到YUI之前，我以为jquery已经非常强大了，比原生js强太多了，就像魔法一样，我以为不会有比它更好的东西了。然后遇到了YUI2，纯英文的一个官方网站，花了近一个月的时间去啃它的各种文档，然后惊为天人，和jquery的差异实在太大了，完全不同的思路。做内部分享的时候，我说jquery就像古龙的小说，江湖是二维的线，兵器谱排行，而YUI像金庸的小说，江湖是个立体的面，东邪西毒南帝北丐中神通，没有一个jquery对象那样的绝对主角。YUI的设计思路让我彻底知道了什么是框架，什么是库。直到后来YUI3的升级，我发现YUI就像apple一样，是个极致的封闭的完美的宝贝。我一直以为它会火，会大火。

在雅虎刚转正不久，遇到了雅虎裁员2/3，我这个新人理所当然被裁了。然后我内部面试成功转岗到了淘宝，我是淘宝北京的第一位前端员工。然后挑战了我这辈子的第一个spa，一个非常有挑战的项目。项目的核心部分我现在还保留着：<http://t.cn/RbrJFM8> 对于当时的我来说，这是个很考验前端编程能力的任务，而且时间要求非常非常紧，这个核心demo我花了不到3天时间完成。各种拖拽啊、换肤啊很复杂，而且没有任何人可以咨询可以帮手。我通宵着加班，完成了这个挑战。在这段期间我看了设计模式相关的书，理解了面向对象编程的思路，然后这个SPA项目正好逼着我用上了OO编程。这个阶段我完成了第二次质变，也变得非常讨厌加班。另一件重要的事是，我开始写书了。感谢我的朋友周裕波帮我介绍了华章优秀的编辑杨福川。

书写到一半，我离职去了新浪。在新浪，进入到的项目组也是个特殊的项目组：即时通讯部，新浪聊天室和web uc。新浪聊天室和web uc全是spa，面向对象组织代码，多人合作维护同一份长期维护的项目。这段时间其实蛮闲的，我学了ruby、rails、python、linux、django、wxpython、敏捷开发等知识。然后html5来了，移动开发也来了。因为2010年被称为移动开发元年，是一个崭新的，光明的平台，机遇很好，很多小伙伴在开始往ios或android转了。我也买了android的书，准备转android了。但当时我已经有点船大难掉头了，一是因为我在web上积累了很多经验，二是我的书出版了，也去参加了很多大会的演讲，圈内很多人认识了我。最终让我下定决心留在前端圈的原因是，想起了asp、delphi的没落，和iphone4对flash不支持，flash的窘境，以及移动平台崛起，windows平台的衰落，我想还是web这种公开的非私有的技术才会是安全的吧，再说html5比html4强大很多了呀，未来大有可为。

在新浪的时间我恶补了大量知识，随着能力越来越丰富，我开始反思两个问题：为什么我要听产品经理的，我能自己做自己想做的吗？以及我大学时选择it这个专业的初心，我想做游戏开发，想开发出让人喜欢的游戏，让人可以津津乐道于上上下下左右左右BA的秘笈。我觉得自己足够强大了，能做前端能做后端能做项管，我想做点自己喜欢的事了，我要做游戏去。然后很巧的是，我的老朋友周裕波当时在盛大创新院工作，他跟我推荐这家在国内已经名气不小的工程师乐园，说这里想做中国的google，院长是陈大年，盛大九千岁，工程师出身而且出非常文艺的一个人。我非常向往这种传说中的工程师文化乌托邦，更何况盛大是以游戏出名的，这么理想的环境简直就是在我量身打造的呀。然后我毅然离开了生活4年的北京，到了上海，加入了盛大创新院。

创新院当年是真的不错，自底向上自主立项，基本上项目负责人全是工程师，没什么产品经理。或者干脆说这家公司就基本没有产品经理也没有运营，全是工程师。大城小胖、flashlizi、

winter、hax、月影、正美当时都在创新院。记得我和小胖、flashlizi老吵架，吵不完的架，这俩又想做游戏又不想和我一起做，却偏偏都在我的项目组里。也是这段时间我发现传说中的微软大神，51js版主winter知识结构真奇葩。然后发现在线上咄咄逼人的hax，其实线下即视感居然有点憨憨的。真想不到是这样的人拍了我两次砖（一次是为朱印宏的《征途》连带被拍，另一次是拍我的书）。月影大侠说话结结巴巴的，还特别绕 @\_@ 。正美一直到离开创新院一直到今天也没见过真人，不喜欢走动的一个技术痴。米粽从那时就已经在讨厌我了，我也不知道为什么，可能是因为技术理念不一样？（耸肩）创新院的回忆其实大部分是不太开心的，最美好的部分，大概是和winter聊洪荒流的神话吧，比如佛啊、太上老君啊、原始天尊啊，谁比谁狠之类的。在创新院期间我立的项目是一个跨pc和手机、pad的实时联机游戏平台，我们基于YUI3 + websocket + twisted + mongodb + 自主研发的html5游戏引擎开发了疯狂坦克、台球、斗地主、五十K共4款游戏。然后盛大创新院就倒闭了。。。这期间我的收获是学习了mongodb、redis、twisted、html5各种api，canvas2d和webgl。另外，也完成了技术人到产品负责人的转型。

小伙伴们都四散去了不同的公司，除了小胖很倔强地自己开始了屌丝创业之路，捆着裤腰带过日子，其他人都找到了不错的归宿。我没有离开盛大，调岗去了盛大游戏创新院。再然后和江湖大佬出去联合创业，做html5游戏平台去了。再然后创业因为一系列的原因，我退出了。但这期间的收获是真真实实地看到一个公司在经营的过程中会遇到的问题，将我彻底带出了工程师思维，开始用更接地气地方式看待很多事情，包括人、公司运作、向上管理向下管理的艺术等等等等。发现工程师真的是群太单纯太单纯的动物了。说单纯一方面是工程师简单善良，另一方面是工程师真的愚蠢。这个世界并不是围绕工程师在转的，工程师的世界虽然别有洞天深不见底广阔无限但终究只不过是哆拉A梦的四次元口袋，它并不是真实世界，它的游戏规则出了那个圈子什么也不是。我本以为我能做个剑客的，却发现我不过是剑客手里的剑，哪怕我是倚天剑，我也不是剑客。创新院那样的乌托邦再也不会有了。这个很打击我，非常打击。。。

我的编程之路到这里差不多就结束了。再后面的故事，我就已经彻底转型管理路线了。带着技术驱动产品创新的梦想和出师未捷身先死的遗憾，我离一线编码越来越远了。不是叛逃程序员圈，也不是我编程编不下去了，而是我的初心就并不是想当一辈子程序员，编程于我来说，是个工具，实现我的游戏梦想的工具，学前端学后端学软件工程全是为了有能力开发游戏。我从一开始就把程序当做工具，只不过一不小心发现我居然还有点编程天赋。

我发现一个很严重的问题就是，很多人在进入职场后忘记了初心，或者干脆就没有初心，什么挣钱学什么，往前怎么走完全是被圈子的思潮所带动的，比如圈子里鼓吹geek文化，鼓吹传说中的大神，鼓吹白胡子白头发写一辈子程序，然后很多人就被这种文化给洗脑了，误以为这是他自己的追求，还被这个文化绑架了，鄙视甚至攻击不认同这种文化的人。真的完全没必要。

我的每一段经历都有不同的收获，我也不知道哪一段经历对我影响最大。回首来路，我想我的路线其实从一开始就注定了吧？性格决定命运，无论路线怎么变，到了某个阶段，我的位置和方向可能都差不多。不谦虚地说，如果当年坚持走设计师路线，没准今天也是个很棒的设计师，也转型做管理做产品了，哈哈。

**在当前前端技术更新这么快的情景上，你觉得前端开发工程师应该注重哪方面的学习？在学习过程中遇到瓶颈了应该如何突破呢？**

前面说了这么多我的成长经历，就是为了现身说法，从我的经历中给出一些建议：

- 不要迷信大公司出品的技术，google和facebook等大牌公司出品的东西多了，别见着是名厂出品就觉得一定是正确的。google出品的GWT死了、微软出品的silverlight死了、IBM出品的dojo也和死了没区别吧。
- 别看今天热闹，经得起考验的才是王道。曾经flash、flex、silverlight、as3、extjs、YUI、mootool、prototype、air、phonegap、jquery mobile、coffeescript、javascriptMVC、backbone、标准语义化、seo、残疾人可用性、flash全站、RIA等等等等也都热闹过一阵啊，今天这些还有几个人提？把时间拉长了看，只有活下来的，才是王道。而什么样的技术能活下来呢？以我的经验看是：极致的性价比。所以我一点不看好angular和react。哪些东西具有极致的性价比，哪些东西才可能存活下来，历史总是惊人的相似。我说这话的意思，并不是让大家不要学习，多学习总是好的，学习下框架里的思想。就像YUI一直没火起来，但我非常感谢它对我的帮助。我说这话的意思是，要小心这些框架，要辩证地看问题，而且要做好充分的心理准备，它们很可能被新的东西替换掉，不要陷入“不停追赶，不停放弃”的陷阱，精力会顾不过来。
- 别盲从意见领袖。对于很多技术人来说，他们追求的是“大神”之名，或“技而优则仕”之实，技术不过是工具，满足这些不甘寂寞的阿甘的陪跑教徒而已，他们才不管性价比高不高，才不管你们跟不跟得上，才不管你们基础扎不扎实学习效率不高，也不管技术会不会淘汰，反正他们江湖名声起来了再互相卖个面子，不愁找不到好雇主不到好薪水。还一群人大神大神地叫，可high了。
- 喜欢叫别人大神的，其实不少人心里也有个大神梦，也希望自己可以早点出点名气。所以呢，对别人总是巴不得别人和盘托出，然后直接拿来主义。有点成绩之后呢，端起碗吃饭放下碗骂娘。觉得看你教程、关注你、给你点赞是看得起你，是公平买卖。学到了知识之后呢，还会骂骂咧咧的，说你哪儿哪儿不对，说你误人子弟。然后问他们看不看书，答说，看，只看国外的书，叫扔个书单出来全是国外的书，显得有逼格。其实国产书有很多不错的，丝毫不出国外的书差，比如说《大话设计模式》《html5和css3权威指南》等等等等，非常棒，却很少看见他们出现在国人的推荐书单里，尽是些国外的书，有不少书其实跟实战离得老远，比如犀牛书，我觉得一点不实用，实战经验足够以后，再拿来当字典查查还行，也就那样。早些年的《网站重构》《css禅意花园》都什么玩意儿，看了和没看没什么区别。我对国人追求逼格的做法非常讨厌，不接地气，该成长不了就成长不了，活该。想学习的同学，别被这帮人绑架，什么东西对自己有用，相信自己的真实感觉。
- 看书非常重要。别整天东学一点西学一点，遇到问题查查百度（别跟我提你不用百度用google，这不是啥逼格，我不会翻墙，你爱用啥用啥）。看书是最系统地学习知识，将零散的知识点串成面的方法，没有之一。看书，看好书。
- 算法、数据结构啥玩意儿离前端实战很远，别听人瞎BB。我写了这么多前端代码，遇到过的挑战非常多，越是基础的css、dom api、bom api对我的帮助越大。别的都是扯。让那些整天提算法、数据结构重要的人举几个例子出来看看，是什么场景下这些重要，一准举不出几个常见场景。还不如好好学学css去实用。

- css是前端的基础，比js更基础。没学好css前，不瞎折腾js，更别折腾框架、工具、编译啥的。更别折腾node，妈的离前端越来越远。搞成了伪后端也解决不了你的实际项目问题。先学css、再学原生js、再学js框架、再学工具、最后学node后端。真想学后端的话，也不推荐用node，玩什么不好玩这个坑货。
- 别迷信大公司光环，幻想大公司的大牛如何如何牛逼，大公司的工作流程如何如何正规、大公司的技术选型如何如何正确。以我这么多家互联网大公司的亲身经历，和对其他大公司的了解，大公司其实也乱得一塌糊涂。都是混口饭吃，某些人喜欢装神弄鬼的，显得与众不同，公司光环既是他们的靠山，他们也为公司的光环添砖加瓦，相得益彰，或者说狼狈为奸吧。还是那句话，有用还是没用，相信你自己的感觉。别一边跟在人后面跪舔，一边发现这些玩意儿落不了地，该加班你还得加班。
- 别随便听任何人的意见，包括我，别搞偶像崇拜啥的，相信自己的判断。尽信书不如无书，尽信人言不如别听。

### **对你的了解，你做过工程师，创过业，那这些经历对你现在的感受是什么？**

这些经历告诉我，工程师30岁转型危机是真实存在的。每个人都需要做未来5年计划，你希望5年后在做什么？很多人能回答出来未来2年计划，但回答不出来未来5年计划。这点值得引起足够重视。在职场时间真的过得很快，特别是IT圈，5年工作经验就敢自称老人，10年工作经验不管愿意不愿意都会被当做老人。你真的以为你到30岁时不会遇到30岁危机？你真的以为你想做白胡子白头发编程老头？你真的以为你做得了这样的老头？今天你在做什么，取决于5年前你的计划，而5年后你在做什么，取决于今天你的计划。如果今天的你没有5年后的计划，那么等着5年后的危机吧。

### **市面上前端相关的书很多，在这块上你的书单上有哪些呢？**

html5和html4是个升级关系，不是个替换关系。html5相关的书，或者说当下流行的书，其实讲的是html5于html4的增量部分，而html4的书可能已经不再流行甚至不再出版了。这是很危险的，书断层有可能引起技术断层，特别是连互联网上的知识也是这种倾向的时候，非常可怕。不知道5年内工作经验的同学，基础扎实的有多少。我说的基础扎实是从html4开始算的。我个人的建议是，找找老书看，补补基础知识，从html4开始补，补到html5。也许这其中会遇到某些知识已经被淘汰了，但不重要，知道被什么淘汰了，以及为什么被淘汰了很重要。

希望直接就有现成的书，把最佳实践直接告诉你，这是不可能的。一是有能力写这种书的人很少，二是有能力写的人不见得愿意写。我建议是从html4时代的书看起，一路看到html5，移动前端开发。

那除了自己补，还有什么别的办法吗？也许你可以参加我的收徒计划。准备好这6个问题：

- 怎么看所谓高手、大牛？
- 如果钱不是问题，自己想做什么事？
- 如何看待工程师30岁转型问题？



- 自己和大多数人观点不一样的一件事是什么？不限于技术
- 如何看待拜师一事？
- 觉得我能教你什么？然后将答案投到 [caoliuyang@111.com.cn](mailto:caoliuyang@111.com.cn)
- 另外，入我们下还有拜师礼节和拜师费，详情看这里吧：<http://t.cn/RGQ0g5i>

为什么会想到收徒呢？自己想吧。

## 在前端技术的深广度上，要如何去权衡呢？

09年我在写自己的书的时候说过，欲精一行，必先通十行。深度当然是最核心的html、css和js上。广度上可以学学服务端编程、面向对象、代码重构技巧、软件工程、native开发、webgl等等。权衡嘛，注意你的岗位是前端，所以什么是珍珠什么是盒子要分清楚。盒子可以换可以扔，珍珠不能扔。

## 从技术岗转到管理岗，你切身的体会会有哪些呢？

team leader和geeker是不一样的，team leader的定位是保证团队合作的流畅与效率，团队成员的能力成长和稳定性。geeker并不适合当管理，甚至会成为团队合作的障碍。技术岗是第一人称视角，是打cs，而管理岗是上帝视角，是玩魔兽争霸。

但有意思的一点是，编程中的很多智慧放到真正世界来运行也是非常适用的，比如敏捷迭代的思想，先完成后完美，比如面向对象的思想，高内聚内耦合，放权与团队合作，比如抽象不足与抽象错误，团队分工协作，比如api封装需要考虑易用性，文档编写要尽可能对用户友好。总之，编程的思想在做团队管理时也是帮了我很多大忙。

只是这样仍是不够的，需要掌握的知识还有很多，比如时间管理、心理学、性格研究等等。另外，我在负责产品，所以最近一年在恶补大量的经济心理学、社会心理学、产品营销、销售技巧等等等等。活到老学到老，做个技术出身的产品人，在互联网+时代，广阔天地，大有可为。

## 专访 | 前端布道师@张克军

### 前言

今天专访的这位前端，在早读君刚入行的时候就听过这个名字，而且也是看着他的博客学习成长的。当初约他做专访的时候还是会有点小期待的，那么我们就来看看关于他的前端之旅。

正文从这开始～

自我介绍，以及现在所服务的公司

我是张克军，一般用名字的拼写“Kejun”出没在社区里。工作在豆瓣。

详细介绍下自己从事前端的经历，这过程中你是如何学习的？有没有遇到瓶颈过，有的话，如何去突破。



我专业是学计算机网络的，架局域网什么的。大概98年左右，那段时间混图书馆，看电脑报上介绍用Dreamweaver做动画，引起我的兴趣。做一个个人网站玩玩吧，于是就这样上道了。

那时候主要精力是在设计上，那个时代叫“网页制作”是很准确的，是“制作”，不是“编程”。现在回顾，这种背景决定我最终是产品、设计向的前端工程师，不太可能成长为语言专家或极客。这个阶段大量做网站，自己的、学校的、外包的、参加比赛等等，生活一度小富裕（哈）。另一方面也培养了耐心，做一个东西常常没有捷径，忍受过程，享受结果。周围同学很多不懈学这种低技术含量和需要付出枯燥劳作的东西，而我乐在其中。

02年第一份正式工作是在亚商在线（不存在了）做网页制作，可能表现出很高的性价比，带动了学校的就业分配（哈）。直到现在工作上，我也是很看重和追求效率。干了两年多去了一家用.net的公司。兴趣是很奇怪的东西，总是潜移默化拉着你走，最终还是不喜欢干后端的工作，05年去了1拍（新浪和雅虎合资的）。

1拍的工程师来自雅虎中国，其中有一位工程师Anson（现在是宝贝树的CTO）对我影响极大，向我输入了很多很新的来自雅虎美国的技术资讯，其中有一个是革命性的—Ajax。

05年底雅虎中国被阿里并购，我正式转到雅虎中国，地理上是从王府井转到土的多多的光华路，但技术上使我与雅虎美国的前端技术更靠近了。雅虎的内部技术交流非常活跃，我在后台可以看到很多很系统的技术资料。那个时候还是Blog时代，国内的开发者要了解新技术，主要靠国内前端第一批Blogger的搬运，像陈贤安、嗷嗷等。雅虎时期，最直接影响我的人是Hedger，他后来也数次来国内分享，是国内前端界的老朋友。他是用代码说话的人，满脑子前端黑魔法，极富个人魅力，直到现在也是我的偶像（哈）。由于算是阿里系了，跟淘宝关系紧密。那时淘宝前端的负责人是承志（后来蘑菇街的创始人），这家伙才华横溢，我们经常一起交流。后来证明他其实是更优秀的产品经理（哈）。

回顾一下这个时期，好比从沟壑上到山峰看到更广阔的风光，有幸遇到很多极富才华又很nice的人直接影响了。我觉得对于新人的借鉴是，刚毕业先别急于进高大上的公司，先进一个限制小、能更全面实践技术的公司，然后，再找一个更大更专业的平台，跟比自己有才华的人共事。

09年去了豆瓣。豆瓣没有明确的角色边界和刻板的流程。前、后端在开发上配合非常紧密。前端工程师做为一种中间角色，即要有能力弥补设计上的不足，又要会写一些Python，尽可能独立解决问题。豆瓣发展到三、四百人时，豆瓣的前端团队最多也没超过15人，如果说优秀工程师的生产力是平庸工程师的10倍，没那么优秀5倍也够了（哈）。角色之间的协作效率才是影响开发效率的主因，通过加人可以解决一时的问题，但慢慢会变得臃肿难以逆转。我在豆瓣写的代码是在雅虎的n倍，参与了很多项目，在这个过程中，有机会与不同优秀工程师合作受益匪浅。学学其它语言有助于建立更多维的编程思想。比如要深刻理解函数式编程，最好学学Haskell。

何时出现瓶颈取决于视野，当看不到发展方向和趋势时，便会苦于没有突破。结合我的经历，有三个明显结点：

1. 2005年的Ajax流行。之前更多精力是在CSS重构上，学习标准，积累解决兼容性问题的经验。Ajax的出现，让我意识到下一个学习方向是JavaScript。学习OOP、学习如何架构代码、学习YUI等优秀的框架、追求网站性能等等。当这些方面都涉足后，又会感到碰到天花板了。08年

去雅虎美国参加全球前端工程师大会，首次听Zakas讲基于Ant的自动化构建，豁然，回来后很快实践和分享。

2. 2010年Node加速发展。Node使前端生态迅速繁荣，通过不断翻新的工具，弥补标准的滞后和语言设计上的不足，前端开发逐步进入“工程化”开发的时代。各种工具、理念，一下子有点目不暇接。这时候的瓶颈慢慢变成学习方向，应该学什么？这部分展开可以看[这里](#)。
3. 2013年PC时代到移动时代。原来的“前端”其实就是浏览器。移动时代的“前端”变成多元客户端。Web方式由必选方案，变成可选方案。工程师必然希望参与一家公司主产品的开发，主产品从PC端转换到移动端，我应该怎么办？曾经想过学Android或iOS开发，解决跨平台问题，首先要先了解平台。但如果追随传统的native开发，从头学起，我个人不太能接受。纯属个人偏好，不喜欢传统native开发方式，一点兴趣也没有。混合开发有局限性，只能坐等完善。我很惊喜React Native的出现，它不会替代原来的native开发，而是传统native开发聚焦在相对底层可复用的组件上，应用层用JavaScript写，看起来这是比较完美的方案。

在所服务的公司中，哪一个公司对你影响最大。为什么？

影响最大的是雅虎，提升最大的是豆瓣。现在雅虎没落了，十年前还是末代王者。技术和设计，特别讲究从理念到方法论的体系性。目的是只要招一个不傻、好学的人，通过这些东西就可以快速培育成符合企业特点的人才。国内的壕公司，这方面做的还不够。豆瓣是一家追求产品创新的公司，如果个人意愿强可以参与各类项目的开发，应用各种新技术也没什么限制。这种环境非常有利于工程师成长。从2014年初开始孵化一拍一这个项目，在产品设计、商业方面有了不同程度的经验的积累，是我想超越工程师局限获得的成长。

对刚入门的前端，你有什么好的建议呢？在技术的深广度上如何考虑呢？

从前面我的个人经历出发，建议是：要先耐得住两三年的菜鸟期(T1)，这个阶段“广”最重要，“广”有两方面，一是全面掌握CSS/HTML/JavaScript的基础知识，二是广泛接收各种技术、思想，尽可能多的实践，多写代码，多有体会。书只是陈述技术是什么，无法告诉你什么场景下应该用什么、怎么用。然后是成熟期(T2)，可以从容的驾驭足够复杂的项目，这个阶段多看代码比较重要，多做思考和总结。最后是创造期(T3)，能造“新轮子”，能攻坚难题。我个人更看重技术在产品上的创新应用。

对你的了解，你现在在豆瓣参与了一拍一的项目，按你的理解，前端工程师的职业规划可以有哪些可能呢？

前端工程师如何发展其实非常不明确。只能说目前有几种可能性，一是成为专家，我感觉Hax还没老过四十吧，有机会跟他聊聊四十岁后有何打算。专家估计会转型为团队Leader，发挥经验和判断力。二是成为极客，超越前面提到的T3，追求T3+或T4。同样到四十岁后，可以创业、可以带团队。三是成为自由开发者，写出个爆款App就真正自由了。四是转行，转产品经理、转设计、转后端皆有可能，取决个人能力。不过没准哪天公司上市、彩票中奖就不用烦恼职业规划的问题了。

作为国内前端的老司机，有哪些图书值得推荐阅读的呢？

Zakas的“红皮书”不错，不过现要学ES2015/ES.next了。书看的不多，大家可以去搜一搜。

最后你可能还需要看看这篇：[【第229期】写给初学前端工程师的一封信](#)

## 专访 | 腾讯UI工程师@张鑫旭

---

### 前言

前端早读课第十一期专访来了，今天要分享的是经常在微博晒钓鱼，博文高产的 @张鑫旭童鞋，今天的故事你一定喜欢的，赶紧搬张凳子看起来～

正文从这开始～

### 1) 自我介绍、现在所服务的团队、负责的项目

我叫张鑫旭，网名也是张鑫旭，因自认为待人真诚，没有必要遮遮掩掩，于是一直使用真实姓名。我其实有个曾用名，名叫张旭，后来算命的掐指一算，我五行缺金，得，父母果断加了个金最多的字-鑫，看你还缺不缺，故得张鑫旭。

现在在[腾讯科技ISUX-社交用户体验部](#)上海设计中心主要从事UI开发相关的工作，“UI开发”这个职位名称起得好啊，听上去即高大上，又把工作内容包含进去了，搞设计的，但是是通过开发（技术）来实现。其实呢，拿出程序员的自嘲精神来讲，就是个写页面的。但是呢，能成为一个设计师认可和喜欢的UI开发，还是需要很多积累以及一些编程以外的东西。如果正在阅读的你觉得你就是这样的人，欢迎简历我([zhangxinxu@zhangxinxu.com](mailto:zhangxinxu@zhangxinxu.com))，加入国内顶尖的设计团队。

真巧，正好说到团队，我们可以从闪亮亮的title说起——「上海设计中心」！为什么是上海，恩，因为办公的地方在上海；设计，表明团队的小伙伴都是设计师……才怪，实际上由国内非常顶尖优秀的视觉、交互、用研以及目前独苗UI开发，也就是我组成；中心，带有支援各方的意味在里面，因此，基本上，上海这边与社交相关的产品我们设计中心都会参与。

就我个人而言，负责的项目主要是QQ企业产品（欢迎大家说服自己老板购买）以及QQ公众号（应该不用多久就会全面开放，到时候欢迎大家体验，可以直接向我反馈任何问题或建议）。

### 2) 本身是生物信息技术专业，到现在做前端开发，这期间你是如何思考并作的选择？

说穿了，就两个字：“兴趣”！



一事物能够成为兴趣，自然是要自己能够做得来，有成就感，释放能满足自身愉悦感的激素，然后才能上瘾并持之以恒走下去。

大家赶快搬好小板凳，我脑中的画面开始`filter:blur(0px)→filter:blur(20px)`模糊，以前的画面开始浮现出来了，故事即将开始.....

我这个人很随性的，很多很重要的决定都是凭感觉走的。比方说大学填志愿，我们是分数出来再填志愿。结果知道分数后，花功夫去研究学校、研究专业了？屁来，连一个动画帧的时候都没用过，干嘛去了呢？那还用说吗——钓鱼，恨不得睡在河边上。得，要填志愿了，屁颠屁颠跑到学校，完全没准备怎么办？啊，我机灵劲来了，点兵点将，点到谁谁就上，哦，不是不是，点到谁就上谁。老夫看你骨骼惊奇，就你了，华中科技大学；至于专业，翻开本子，从上往下看，有生物字样的（我本来想成为一名生物学者，野外生物探险专家那种），从上往下一个一个填。然后第一志愿第一专业进去了。当时有思考吗，没有，反正就是，先进大学再说，至于以后要干嘛，看天数。

然后，从小孤家寡人长大，没人指引，大学要干嘛，以后要干嘛，完全没概念完全没想法，用一个词形容就是“迷茫”，以至于做什么都没有激情，没有参加任何社团，篮球场上的眼神防守专家，同学提起我，别人会“哦~~我知道，就是最后一排一直睡觉的那个”，成绩也是高不成低不就，稳定的中间游击手，当然，毋庸置疑，是单身狗。

后来自己想了想，这样蹉跎下去不行啊，反正时间都是一样的过，那就做点自己想做的事情。于是，我去干嘛了呢？哈，没错，有人猜到了，去钓鱼，翻过学校可能会断子绝孙的铁栅栏，去后面的东湖钓鱼；还有串巷，随便搭一辆公交去陌生的地方走街串巷；以及一些没想到会对自己职业生涯有影响的事情，比方绘画，图书馆所有绘画相关的书都看过，素描，彩铅，水彩都自学过，多次通宵作画。大二可以有电脑了，然后，对着80年代电视机模样的1024分辨率显示器自学photoshop，暑假同学都回家避暑了，我一个人练PS技术；下学期开始捣腾flash，搞些酷酷的效果自己得瑟下。

现在想了想，自己是个典型的内向的人，上面那些想做的事情都是一个人完成的，不需要交流损耗精力的，不需要逻辑思维损耗能量的，两年下来看似无所事事，实际上一点一点积攒了很多精神能量，厚积薄发一触即发。

故事即将进入高潮.....

大二暑假，在河边上，钓了一个暑假的鱼，顺便思考了一个暑假的人生，突然就想通了，男人有实力才是王道。大三回到学校，整个人精神暴走，凭借自己纯粹兴趣使然学的点PS和Flash技术分别进入了校义工部做平面设计，校网络应用研发中心做网站(flash方向)，没错，应该是别人大一干的事情，我起步比其他同学慢了2年。所以，基本上，每天早起晚归，拼命追赶，没有任何时间的浪费，每天时间都不够用，做设计做动画。然后还得了义工之星奖励，标识设计还拿奖，班级篮球队也得了冠军，似乎我的大学生活开始有了亮彩。

在网络应用研发中心，让我开始接触互联网，接触网页，当时找我进去值做flash特效的，当时这种酷站挺火的。然而，进入后，flash没什么活，到是有很多外包的写CSS的活，人手不够，我就主动请缨，帮忙写页面，花擦擦，全是通宵搞，不要太过激情。那个时候，我还不知道，这玩意是可以帮你找工作的。

转眼大四了，我擦，校招来了，对于从没踏出校门的我来说，第一次有个认识外面世界的机会。那个时候，我才第一次知道，原来世界上有一种职位叫做前端开发，需要懂PS懂CSS懂JS，虽然哥JS很挫，但AS写得不错，我当时就high了，这不是给我量身定做的职位吗，我大学瞎折腾的不就是这些鬼吗！当时那个激动啊，要是兴趣能当饭吃，我就可以上天了。

此时我看了一眼我的专业“生物信息技术”，爱过。然后，我就决定走前端这条路了。

告诉大家一个秘密，我其实还犹豫过要不要去做网页设计，职位要求要会PS,最好懂CSS，好像挺符合的嘛。后来我为啥放弃了呢，某天兴致勃勃来到社团办公室，然后大家在围观团队里工设专业第一设计师妹子的网站改版作品，我也跑去凑热闹，当我看到作品的那一瞬间，我一下子懵逼了，突然意识到，原来人和人的差距可以这么大，一下子明白有些东西不是有兴趣有干劲就能超越的，遂放弃。

当然，毕竟比别人晚两年，且一直没出去看看，眼界什么的狭隘，加上自己的表达能力是罗永浩的补集，校招惨败。举个例子：兴致勃勃，西装笔挺跑去面试阿里，老激动了，年轻人嘛，为了想展现自己能力很强，说话嘛，有时候分寸拿捏不好。

“我一天就学会了CSS?”

“你确定你学会了CSS~”

我愣了1秒，吧嗒吧嗒我的大眼睛，补充道：“恩，我是1天就学会了！”

然后一男一女两个面试官就在面面相觑咯咯咯咯的笑。

当时我就嘀咕了：“笑毛线啊，靠，尼玛以后请我去你那儿上班我都不去，太不礼貌了”！所以，很多年后，当阿里和腾讯offer同时摆在我面前的时候，大家都懂的。

因为起步慢，校招不顺利，才有了后来大四我去实习，毕业时候，辞去实习工作，去闭关学习php和JavaScript，第二年也就是10年才开始正式参加工作的经历。也就是，虽然我09年毕业，但10年才正是从事前端这个岗位。

期间遇到的挫折和迷茫还是相当大的，后来，是凭借俞敏洪，乔布斯以及罗永浩的励志演讲，以及对前端的真爱才走下来的。不过那是后话了。

总结下就是：当年我是个很普通的屌丝，一事无成，同学都很优秀，各种奖学金拿到天上去。后来阴差阳错遇到前端，发现这玩意门槛低，好入门，进入大学这么多年，终于发现一件可以做得来的事情，于是有了兴趣，然后在“follow your heart”这句话的激励下，披荆斩棘在前端这天路上走了出来。至于专业，不念到博士难有大成，想都没想过。

### 3) 跨专业做前端开发这几年中，你觉得你最大的收获是什么？以及跨专业对你有什么影响？

\*\*

\*\*

最大的收获嘛，往高大上方向讲，就是通过不断地分享，让很多前端同行的得到了学习和成长，也对整个前端行业的发展做出了一点自己力所能及的贡献；如果往世俗方向讲，有了自己的车子，虽然是小车，有个自己的房子，虽然在乡下，有个娇妻，没有虽然，然后还有儿子一枚，哈哈，没错，我就是传说中的人生赢家。低调低调。

跨专业嘛，只能说跨半个专业，做前端，计算机专业对我还是帮助很大的。虽然说，当年在大一学C语言，根本就不知道，在黑底框框上敲一些白字都有些什么用啊，没兴趣，但是，毕竟是专业课，计算机的理论什么的，程序基础都还是知道的。于是，当我学习flash发现as可以实时呈现一些动态效果的时候，我的天窗被打开了，原来程序是这么个东西，兴趣也起来了。所以，我一直认为，大学应该先学JavaScript这种所见即所得的程序语言，先培养兴趣，然后在C/C++走起。

至于跨专业的生物，我本来想说没怎么用心学，不影响，后来一琢磨，不对，影响非常大！正是因为我选择了生物信息，我才能遇到了我的室友宏达，于是才有了下面的对话，“鑫旭，你最近都在干嘛？”“在弄网站。”“是吗？那很好啊，互联网很有前途的！”就是这么不经意的一句话，如果没有这么一句话，前端界说不定就看不到我的身影，从一点上讲，跨专业影响巨大，哈哈！

不过，我突然一想，大学里有“前端开发”这个专业吗？貌似没有吧，大家应该都是跨专业的吧~~

### 4) 博客这么高产并且研究的问题还很细致，在这点上，你是有什么特别的学习方式吗？

\*\*

\*\*

这个问题，之前有人邮件或其他方式问过我类似的：“哥，我写技术文章比女人生孩子还难受，半天都憋不出一句话，你有什么诀窍吗？”

我的回答都是“这个嘛，多写写就好了！”其实内心真实的想法是：“这个是需要点天赋的”。

为什么当年我看到工业设计专业第一妹子作品的一瞬间就放弃了呢？有些东西是很难逾越的，一个是很多年的积累，二是内在的天赋。有些事情要想做得好，勤奋就可以；但是要想顶尖，天赋是必不可少的。

我想说的是什么呢？博客为什么这么高产，戳穿了，因为写作本身也有天赋同时有很多的积



累，写篇文章就跟吃顿饭一样稀松平常。小学时候就开始在国内期刊发表作品，拿奖，一直笔耕不辍。以我现在的目光来看，与自己之前内向敏感细腻的性格有关，艺术气息浓郁的人，往往内向孤独。所以，小孩子性格内向，并不是一件坏事情。

同样的，存在这样的人，说话很有天赋，任何场合都能即兴演讲，像说相声一样，语言表达很有天赋。你问他有什么特别的学习方式吗？估计他的回答也就是“多练练”，实际上，我是很清楚的，人家语言区逻辑思维很强，汪涵可不是人人都能做的。

第二个小点，“研究问题细致”，还是与人特质有关，我能一个整天坐在河边，像个傻子一样，看着鱼漂一动不动，估计很多人是做不来的？能够长时间静下心来专注在某件事情上，这是做研究非常重要的特质。所以，性格活泼外向，非常能说会道的，可能管理有道，但是一般技术尔尔。

然后，还想补充一点，兴趣很重要，写文章做研究是件很辛苦，并且没有明显收益的事情，如果只想把前端开发当作吃饭的工具，而不是真正发自内心热爱它的话，是很难坚持下来的，这个时候，才能体现“follow your heart”这句话的价值。

最后，分享下我做研究的些心得吧。其实呢，我们做前端技术研究，跟大学里做科学研究是一样的。首先，明确研究对象，比方说我想研究[CSS中的层叠上下文](#)，OK，1.先去找资料，官方文档或者MDN等外文站点（科研是找文献）；2.验证资料中的观点，例如，inline元素覆盖float元素，此时可以设计实验，在网页中就是写demo，看看是不是这种覆盖关系，一定要亲自验证，其他地方的文章会有错误或者观点过失；3.提出资料中没有的自己的疑惑，例如，层叠上下文的层叠顺序是哪个级别？我觉得貌似是z-index:auto级别，好，为了验证这个观点，我们再次设计实验，让几个z-index:auto的元素与之重叠，且先后顺序不一样，看最后的表现，得到我们的结论。

其实现在想想，我其实是个做科研的好手，只是学校的教育没能及时吸引我的兴趣，如果走自己的专业“生物信息技术”，一直研究下去，应该对人类的贡献更大。不好意思，扯远了~

## 5) 在前端技术的深广度上，你觉得在这条路怎么走？

\*\*

\*\*

我就说说我自己在前端技术的道路上如何积累深度广度的吧。

关于广度的学习广度学习，在我职业生涯早期，一般都是关注很多知名前端团队的官方博客。后来，发现，这些博客一是大部分都坚持不了多久，二是内容已经不能满足我的胃口的，三是加上工作事务越来越多，后来，这些团队博客基本就放弃了，转而集中关注国外知名的前端学习站点。

包括：<http://www.sitepoint.com/>，<http://maxdesign.com.au/>，<http://www.smashingmagazine.com/> 以及<https://css-tricks.com/>

基本上，每月都会花几天时间，认真浏览每一篇文章，觉得有价值的，就是记住，或整理成文章，或者编辑成小tips微博备忘或传播。

由于自己有刷微博的习惯，因此，还有很多新的知识，是微博上一些积极传道的同行小伙伴传播

的，自己或转发或评论，等需要的时候，就可以找到。

关于深度的学习就我个人而言深度学习在于多想和多写。

而这里的深度，又包括两方面，一是上面广度内容的深度学习；二是基本知识的深度学习。

### ① 广度内容的深度学习

我们都会见到或听到一些新的属性、名词或者特性或者技术实现。一般的同学都是看一下，知道有个这么回事。

而我自己，一定会自己亲自实践，至少整个demo什么的，如果在实践的过程中发生一些有意思的事情，一定会整理篇文章，和大家分享下。一来思路提炼，二是知识备忘，三是日后用到的时候可以快速上手，四是帮助其他同行的学习。

### ② 基本知识的深度学习

这里基本知识就是我们平常经常使用的一些东西，看上去简单、朴实，实则深有学问。学习的关键就是多想，就是多提几个为什么？这需要一定的前端敏感度。比方说，一个普通的标签里面有个图片，结果图片下方会有几像素间距，这个间距是怎么来的？

一般的同学遇到问题都是去度娘或谷歌搜索，找到解决方法，然后就没有然后的。并没有去深究背后为什么会这样。

为什么呢？因为，越是简单的东西深究起来越麻烦。就好比问为什么 $1+1=2$ 这样的问题。但是呢，前端这东西门槛低，如果你没有这些深入的理解，如何跟其他茫茫多普通的前端做区分，从而迎娶白富美，走上人生巅峰呢？

项目这么紧，哪有闲工夫去研究这些看不到价值的事情，类似这样的心理导致学习总是遭遇瓶颈，若要在技术领域有所成，这样的心理一定要有所转变。

而我会牺牲几十个小时的钓鱼时间，去思考：会是什么原因呢？然后不断写demo，变换属性等方法验证自己的想法，看看能不能解释得通；网上会找资料，尤其外文，有没有相关解释；会去CSS官方文档寻找蛛丝马迹。直到全部梳理清楚。

然后，很重要一步，就是要把所想所得记录下来，一定要记录。因为我们都不是天才，过了一段时间，都会有所遗忘，唯有笔耕不辍方能弥补。

最后呢，很重要的一点就是：学习这东西一定是要肯花精力的，而且无法一蹴而就，需要日积月累，贵在坚持！

有些同学成长初期可能会疑问，人精力总是有限的，我是先专注于广度的成长呢还是深度的成长呢？从长远来看，都是深度和广度都是需要积累的，没什么差别，那么初期应该侧重哪一点呢，我个人觉得应该根据自己公司、自己从事的项目或者业务需要走。中小企业可能需要你是多面手，页面写的起来，开发也做得来，此时，广度应该先重视起来；职位划分非常细致的大企业，需要你某一方面是顶尖的，显然，深度的学习是优先关注的。

## 6) 现在前端技术层出不穷，你对前端新人有没什么建议？

建议谈不上，说说自己的看法吧。

从生物学的角度讲，一个物种要想在地球上存活的久，多样化很重要，无论是外形，性格以及食物。如果体型很大，需要能量就多，食物匮乏就惨了，优先饿死，此时矮人(侏儒)存活机率大；再说性格，如果物种性格都是一样的，很容易被利用和灭绝，比方说渡渡鸟，都不怕人，70年时间就被灭光了，如果只要有中华田园犬这种程度的机敏，保证可以活到今天；食物更好理解了，考拉只吃桉树叶子，一旦桉树遭遇大规模死亡，考拉日子也不好过了。

扯这么多是想说明什么呢？从宏观角度看，对待新技术的各种态度并没有谁对谁错，这些观点态度其实对于整个前端生态而言都是很重要的。我觉得应该用包容开放的心态看待不符合自己作风的态度，放下自尊，主动吸收对方之长，这对于自己的成长很重要。

就像投资一样，有人喜欢价值投资，稳固保守；有人喜欢追逐热点，快速收益，其实都是为繁荣经济做贡献。

我们做前端的在职场中打拼，虽然说是做技术的，但最后世俗定义的成功，技术的能力只是一部分元素，一个加分项而已。基本上，公司里管事的都不是技术最厉害的，而是当年技术做过贡献，其他领域如沟通，领导力也比较出色的人。可能几年前，我会建议新人：“你想成为技术大牛，在技术领域有所造诣，时髦的新技术可以先不急着重究，夯实自己的基础最重要，一定要系统学习，而不是靠有限的经验积累。”但是，现在的我并不会这么说了，因为每个人追求的不一樣。比方说我，我更看重个人技术成长，对团队的帮助以及对行业的贡献。但是，很大一部分同行，追求的是升职加薪，如果某个新出现的技术对其职业发展很有帮助，例如React.js帮助公司做移动开发。我们还会要求这个小伙伴，不要跟风新技术，好好扎实基础吗？错过了时机，会影响人家的职业发展，表现和升职的机会就会错失。

所以说，对于前端新人而言，重要的是自我认知清晰，知道自己想要的是什么，知道自己的特长在哪里，这个很重要，一定要认识清楚，如果只是盲目跟风，或许既不能收获技术，也不能收获职业发展度，那就比较不幸了。

## 7) 如果让你推荐你工作中所用到的工具，你会推荐什么？

这个问题还真烧了我不少脑细胞，我并不是一个热衷于工具的人。非要推荐一个，那就Faststone Capture吧，截图、标注以及量尺寸，是我使用频率相当高的一个工具。其他嘛，如果要录制屏幕为gif动图，可以试试使用ScreenToGif，我一般会再再Photoshop中再重新保存一篇，优化下大小。

“没了？”

“是啊，没了！”

“有点短，要不再说一点？”

“这个.....那我推荐下自己的博客吧，如果有想了解的前端知识，试试去我博客哪里搜索下，说不定会有不一样的发现！”

# 专访 | 美的电商高级前端工程师@姬小光

---

## 前言

hello，新一期的前端早读课又来了，这期应该是第十期了，那么今天要跟我们分享的是美的电商的@姬光。他2012年出版过《精彩绝伦的CSS》，写的文章在微博上被优秀网页设计转发，那么他还有什么故事呢，且看今天的专访～～

正文从这开始～～

大家好，我是姬光，在理想的情况下，我本是希望实名行走江湖的。只是当时注册微博的时候，我的名字被某位大神抢注了，所以我只能在名字中间加个小字，叫做 @姬小光，当然，其实并不小的。

很荣幸能够通过前端早读课这个平台与大家交流，早读君和我已经很熟悉了，起初邀我写专访，我是拒绝的，因为实在觉得谈不上老司机，怕乱写误人子弟。但事不过三，实在盛情难却，故厚颜聊聊我的一点从业经历罢。（早读君怎么可能会放过呢？）

我的本科其实是学电气工程及其自动化专业的，后来方向是楼宇自动化，也叫智能建筑。硕士才转学了软件工程，算是广大非科班出身的从业大军之一。我目前在美的电子商务有限公司（深圳），负责美的商城，会员活动等相关业务。

## 我的前端之路

虽然我也是众多非科班出身的前端工程师之一，但若说到跟前端的渊源，那要从我的小学开始说起了。

我在小学一年级的時候，父母感觉我画画貌似还可以，刚好假期没事还容易出去惹事，于是就把我送到了美术班。之后的每个假期我都是在美术班度过的，期间四五年也拿过省市县全国等各种奖项。

后来到了 2000 年，家里有了第一台电脑，那时我就开始接触 PhotoShop 6.0 以及其他一些乱七八糟的软件，比如友立系列的，Ulead PhotoImpact 这种在当时已经是切图神器了，还预置各种水晶按钮效果，所以，我的前端之路应该算是从设计切入的。

由于有了基本的切图技能，在加上对计算机的热爱，2002 年的时候，我已经可以做些超级简单的 table 布局网页了。虽然有的人说看起来像黄网，但好歹也算是入行了。从那时起，就算是走上了前端这条不归路了。

后面的流水账我就不多说了，对其他人也没什么价值，对我本人经历感兴趣的同学可以关注我的微信公众号 lasertime，那里有更真实的我。

## 快速成长阶段

如果说我的职业生涯中有哪个阶段感觉成长非常迅速，还真是不好想。我一直是处在一个相对平

稳的增长状态，没有爆发的时候，也没有低谷的时候。无论在淘宝 UED，还是腾讯京东美的电商，我的绩效始终保持在 A 或 S，目前还没有拿过 B 的考评。

当然说这段话肯定不只是吹牛逼，能保持稳定的增长也是一种能力，下面就简单说一下我的一点经验。

首先，如果你能明确你努力的目标，那就再简单不过了。比如硕士校招前期，必须是努力充实知识去准备面试，就算每天泡在图书馆都是泡妞睡觉，多少也是能看几页书的。

其次，到了一个环境中，要尽快弄清楚游戏规则，说白了大部分公司都是 KPI 导向的，你要明确你的 KPI 是什么，再去努力达成。比如腾讯比较看中的，业务能力，分享，专利，自研工具等等，既然有那么多的硬性指标摆在那里，那就一个一个去达成就好了。

当然，我只待过一个部门，各个部门之间肯定差异很大，不可一概而论。

## 关于一专多长

首先说说我自己的经历吧，前面说到，我算是从设计切入的，但是我本身也算不上个设计师，虽然我也为社团做过很多巨幅彩喷海报，不过现在看来，不过垃圾而已。

如果有哈尔滨工业大学的同学，去翻 201 讲坛办公室的电脑 2004-2006 年度的资料，应该基本都是我的“作品”。

网页的制作，我是完全自学的，从 FrontPage 2002 开始啃书学习，到后来的 DreamWeaver 等，各种虚拟主机，网站域名的也折腾过无数个了，从中也学到了不少技能。

Anyway，前端工程师也好，全端工程师也好，全栈工程师也好，叫什么并不重要。重要的是你是什么，你自己是怎样定位你自己的。

如果只是混口饭吃，那切切页面足矣，切到精巧细致也可丰衣足食。

我自己大部分时候其实是个懒人，我很少带着目的去专门学习一门技术。但是，当我遇到疑问或者问题的时候，如果我觉得这个需要解决，那么一定会想办法攻克。在这期间遇到任何的相关技术，或者难题，我都不排斥。

从产品思维营销技巧，到运维服务器配置优化，我都有兴趣去了解。甚至有多少个假期，我帮家里卖货，还积累了不少门店销售经验。

经历了所有这些，到了最后，我觉得能沉淀下来的，只有一样东西，那就是解决问题的能力。

解决问题的能力，虽然听上去很虚，实则可大可小。关键就在于，你能解决多大问题，基本就等同于你有多大的价值。

所以，关于一专多长，我的经验是，不刻意去排斥，保持对新事物的热忱，慢慢的你就会成为一个全能的人。

## 对新人的建议

其实这应该是我比较擅长的部分，我的公众号简介里自称是资深毕业生导师，不夸张的说，我手把手教出来的徒弟，一定是广受好评的。

那么作为一名前端新人，我想最重要的就是保持激情了。你们现在正赶上前端爆发增长的大潮，很多高中生都开始参加 D2 等前端会议，比我们当年不知道要幸福多少倍。

我刚开始的时候，没书没资料，就只能一点一点改代码看看有什么变化，根本不懂都是干嘛的。软件也是各种功能慢慢尝试过来的，甚至 windows 系统目录都逐个点开无数遍。同样，靠的也是一腔热血。

关于学习方式，有一点可能需要注意一下，那就是一定要转变在学校时啃书习惯。除了几本经典的大部头书可以啃一下，其他书都没必要从头到尾去啃，真正进入工作以后，可能你也没有那么多时间去啃。

所以读书的技巧就变得非常重要了，要把每本书都当成考试前一周的冲刺一样对待，直击要害，看重点关注的内容。看懂了 DEMO 就早点上手实践，这样才学的更快。

另外一个很重要的能力就是文献检索的能力，这里确切地说应该是各种资料的检索能力。搜索引擎谁都会用，那些特殊命令的奇技淫巧也不是重点，重点是你的归纳总结能力，快速提炼出能代表这个问题的若干中英文关键字，如果你的关键字足够准确，那就什么技巧都不需要了。

同样的一个问题，同样的搜索引擎，为什么总是有些人可以更快的找到答案，这其中一定有其不可告人的秘密。

所以，新人小鲜肉们，关于技术方面已经有各种大神的谆谆教诲在前，而我想强调的是，软实力其实更加重要。

软实力，包括解决问题，文献检索，写作能力，PPT，沟通能力，演讲能力等等，无论是在工作还是生活中，都将是你可以一生受用的宝贵财富。

## One More Thing

最后的最后，想跟大家分享关于个人口碑的问题。说实话，技术人员在某些方面跟古代的文人很像，是有些清高的。除了我代码的世界，其他事情我都可以不 care，我技术牛逼，我靠技术吃饭。

当然我说的有点极端，我真正想说的是，技术永远不是你的核心竞争力，你的核心竞争力首先应该是“靠谱”二字。靠谱这个词远远比技术牛逼有用的多，关于这个词我想我不用过多解释了。

那么，如何才能给人留下靠谱的印象呢？这个事就没那么简单了，信任与靠谱往往是日积月累，大事小情，一桩桩一件件积累起来的。如果想打破可是很容易，只需要一件事即可。

所以工作久了你就会发现，大家跳来跳去基本都是朋友介绍，圈子也很小，几年之后前同事遍布大小公司。如果工作了5年之后，你还要靠投简历来找工作，那么即使你简历上的技术写的再牛逼，我也觉得有问题。

当然，不是谁都有机会去认识那么多同行的，重点是，你一定要珍惜你的每一份工作，给你的每



一个同事留下靠谱的印象，这样，你的路就会越走越宽了。

如果大家还有什么想跟我交流的，可以关注我的微信公众号:姬光并留言给我，感谢大家阅读。

## 专访 | 360奇舞团团长@月影

---

### 前言

2016年第一期专访来了，也是前端早读课第九期专访了。早读君也是15年的时候在广州FEDay上认识的@月影，在那个时候有通过短暂的沟通，对奇舞团有了更多的了解。同时@月影在08年的时候出了一本《javascript王者归来》的图书。

正文从这开始～

我是月影，2004年毕业于杭州，先后在微软亚洲研究院、金蝶、百度、盛大工作过，现在是360奇舞团团长。

之前我做的东西比较杂，在MSRA做过图形图像的基础研究（C语言和C#），在金蝶做过售后、项目实施、平台开发（Java）、项目管理等，06年开始接触的前端，08年到百度之后主要负责百度有啊前端团队，11年到盛大创新院一年，主要做技术开发和团队管理，12年至今在奇虎360奇舞团，这是一只专业的前端团队，主要支持公司许多互联网产品的前端业务。

我一开始做的东西比较杂，所以做前端也算是偶然，但也不全是偶然，因为我从90年代起就接触互联网，当时我还在上中学，就自己学做网站，和高中的同学一起开发了班级的留言板、聊天室等应用，当时用的是perl写cgi服务。

在金蝶的时候我们开发公司内部使用的web系统，服务端用java，当时团队里面没人写前端部分，所以我就一边学js一边写前端，当时研究多了，对js有比较深入的了解，常在51js上交流，因此就这样开始专门往前端方向做。

前端行业如果从1994年w3c成立算起，到今天也不过是21年，而如果从所谓的web2.0概念（没记错的话大概是05年的样子提出的）算起，到今天才整整10年，而这10年里前端经历了巨大的变革。总体来说，前端目前是一个非常年轻、有活力、变化速度极快，充满挑战的行业，我觉得选择这样行业作为职业是非常幸运的，因为变化快意味着创造机会——前端一直是充满着机会的。

有些同学可能会觉得这几年前端变化太快了，多少有些压力。其实我倒是认为技术变化快并不是什么不好的事情，至少目前来说产品和互联网技术推动着前端这个行业往前发展，而前端技术的发展又反过来改变互联网技术和产品，所以这是一个非常良性的生态圈，我们作为从业者所应该做的，是对自己有更高的要求，顺应趋势，站在技术的前沿，去努力推动良性的变化更快更好地发展。整个行业变得更好了，作为从业者，我们每一个人都能从中获得更大的好处。

要站在技术的前沿，我认为最需要要有深厚的技术功底、开阔的视野和开放的心态。因为前端行业年轻，入门门槛低，造成从业者各自的经历有很大的差异，有的同学做前端可能是半路出家，并不是计算机相关专业出身，有些同学可能计算机理论基础不好，这些客观原因再加上主观上一些

同学可能并没有对这个行业有更加深入的思考和认知，本着做事情，完成项目的完全实用主义的心态，不知其所以然，蒙头做东西，以为做出东西来就是好的。别人今天说YUI好，就用上YUI，明天说jQuery棒，就用jQuery，再过几天，说AngularJS火了，就换AngularJS，没多久React来了，结果就又去追React。眼见过很多技术，能说出不少概念，但多少有深度的？可能没有。

有的人道听途说，别人说静态资源要合并，就一股脑都统统合并，别人说图片用webp好，就跟着用，别人再说要做什么优化，也就去做，最后再炫耀自己做的技术有多么多么牛，但其实技术背后的深层原理是不懂的，这样的“技术专家”好像自己也很牛，但也只是跟风而已。真正要做到极致，背后要下很大的研究功夫。每一种优化，背后的原理、问题细节都要掌握清楚，奇舞团之前的屈屈同学，现在是好搜前端团队的负责人，他的博客很有名，上面的文章质量很高，那是因为他不但能做优化，而且能够把其中的原理、细节、问题深度挖掘，真正做到知其然亦知其所以然，这是需要下苦功的，没有什么捷径，要说唯一的捷径，可能是对技术领域的兴趣和追求，有兴趣和追求，研究技术之“苦”于自身来说并不是苦，而是乐在其中的。

有另外一种“高手”，对别人的东西不屑于研究，喜欢自己闷头玩技术，也觉得自己做的东西很牛，大部分做技术的人都有这个毛病——包括我，喜欢自己写代码，不喜欢折腾别人的代码。但是，做技术，开放的心态太重要了，封闭的人觉得自己很牛逼，实际上有可能他是自己内部王国的“土皇帝”而已，没有见到外面世界更好的东西，这样的人带一个封闭的团队，往大了说是很难给行业有什么贡献，往小了说，自身的心态限制了自己和团队的技术发展，毕竟现在是集体智慧的时代，我们都是站在github巨人肩膀上的程序员，自己不利用好这样的平台，真是可惜！一定要用别人的东西，就算打算做自己的东西超越前人，那也是需要建立在对前人的东西彻底的了解的基础上的，如果不了解，谈何超越。

貌似前面说的太多快成了说教，不过说教就说教吧，我真心觉得目前前端大环境挺好的，每一位选择从事前端的小伙伴们请相信自己的选择是正确的，还是前面那句话，在这个时代做前端是很幸运的事情。至于说如何学习新技术，其实我一直有个观点——行业、团队、每个个体都是多元化的，没有绝对的学习方法，每个人有最适合自己的学习方法，不论是过去的百度有啊前端团队（我们当时叫WED团队），还是现在的奇舞团，我们都欢迎各种各样不同的小伙伴加入，只有多样化的团队才是精彩的。那有同学担心说自己成长遇到瓶颈，这个其实也没什么可怕的，没有人在技术成长的时候不遇到瓶颈的，我自己在学习成长的过程中也遇到过好几次瓶颈，遇到瓶颈不可怕，最重要的是不浮躁、不忘初心，只要遵循这两条原则，瓶颈总是会被克服的。初心是很重要的，我做技术的初心是把技术当做创作，因为我走上程序开发的道路的原因是因为我想要自己创作好玩的软件和游戏，因此每解决一个新的问题的过程中，我都会有创作的成就感，正是因为这个感觉推动我在技术路线上前进并战胜困难的。每一个人都会有自己的初心，不要忘记了，就一定能迈过一道又一道坎儿。技术问题是最好解决的问题，技术并没有什么特别困难的，最难战胜的，只是自己的心。

## 专访 | 大搜车前端负责人@芋头君

### 前言

2015年前端早读课专访最后一期了。今天的这位童鞋就是昨天早读君推荐把代码压缩成圣诞树的

作者@芋头君。那他又有怎样的故事呢？

正文从这开始～

## 1) 自我介绍，网名，你大学的专业，在公司所服务的团队，现在负责的项目

大家好，我是芋头，在杭州大搜车负责整个前端研发部，我们部门比较特殊，糅合了NodeJS服务端开发和前端开发两部分开发，因为Javascript这个语言，我们碰到了一起，不过NodeJS在我们公司还是纯后端的，只是因为大家写的都是JS，所以糅合在一个团队里方便交流，也是期望这种糅合能够碰撞出一些火花来，目前已经在团队内部孵化出不少服务出来。

我大学时候读的是电子科学，主要是理科，光学相关的，因为高中时候对理论物理非常向往，所以选了类似的专业，不过后来发现学什么讨厌什么，后来就各种厌学，后来机缘巧合接触了编程，于是入了这个坑。

后来大学没毕业就去阿里工作，后来也的确也一直没有毕业。不过有个有意思的事情，我们现在公司的ceo，cto大学都没有毕业。。。想想也算一件有趣的事情了。

大搜车的前端团队目前也有20人了，在这样一个团队中，我的角色已经不是简单的定位在某个项目里了，我觉得人在每个阶段都要给自己一个符合当前阶段的定位，所以项目相关的事情，会给小组leader或者个人更多空间来把握这些事情，而我做的事情主要是推进一些新技术和一些团队开发规范相关的事情，致力于让团队从草创阶段进入一个成熟的阶段，向一些大公司的成熟团队学习经验。然后也会负责推进一些技术驱动的创新项目，这也算是兴趣驱动了，希望能够带着团队做一些有意思的事情，最近在搞的例如炫酷的可视化，还有视频直播PPT同步的一些技术，还有react native在业务中的推进等。

## 2) 工作这么多年，有没哪一个时间段让你觉得成长进步很大？为什么？

不管是做什么工作的人，工作一段时间之后就会遇到瓶颈，而突破瓶颈的时候的成长其实就是进步最大的时候。

对我来说，其实有几次这样的感受。

第一个，在阿里负责交易线前端的时候，阿里的交易线前端逻辑算是非常复杂了，在接手以后，开启了重构计划，大概投入了3个月的全职时间来做这个优化，尝试了前后端分离，以及粗糙的mvc模式，模块化开发等开发模式，感觉对自身的成长非常有帮助。那是2012年吧，搞完了这个项目之后我就从阿里离职了。

第二次，我从上海回杭州之后，一个人在家里呆了一年，然后这期间我有了很多空闲时间，于是我对自己的技术栈进行了横向的扩充，深入学习了iOS开发，Nodejs，java开发。后来利用这些知识，我做出了一个用户突破60W的ios app，一个从0用nodejs构建的前端乱炖网站，好几个大搜车初创时候的java系统。每个技术不是简单的看看书，而是有机会给自己深入，我感觉这些机会非常来之不易。

第三次，就是今年吧，我感觉自己在两个方面的成长很大，第1个是接手了公司的NodeJS业务和团队，逼着我在实际的后端业务中深入进去，对服务端的开发模式有了深入的了解，虽然中间趟过

了很多坑，但是也让我们团队慢慢成熟了起来。第2个就是团队的组建，从0开始到组建，中间经历过许多人入职，也有不少人离职的状态，促成我不断思考如何管理一个团队，感觉这条路还很长。

### **3) 基于对你的了解，除了前端外，还会其他技能，那你是怎么理解一专多长的？一专多长对前端有什么帮助？**

对于开发者来说，其实就是要不断给自己创造机会去深入挖掘一门技术。

在刚毕业的时候，我给自己想了一个很大的前端框架，我每天上班下班都会花费很多时间来构建这个框架，其实框架本身挺烂的，但是会促进我去思考、去学习别人的写法、去解决问题。慢慢的，这些时间都转化成了我对前端知识的了解。

后来，我学ios开发，其实我看了很久的书，一直没有进步。后来我在我从来没动手写过app的情况下接了一个外包，时间一个月，做一个阅读类的app出来，为了做这个app，我几乎一个月没好好睡过觉，各种坑，然后各种翻阅资料。后来app凑合做出来了，我也基本开始入门ios开发了，后来我又独立在业余时间做了好几个app，耗费在这个app上的开发时间，慢慢转变成了我对ios开发的理解。

后来，我开始看nodejs，开始也没什么起色，后来我想要不用nodejs把我的博客重写一遍吧。后来的事情大家都知道了，我不小心把一个博客写成了一个社区，后来还玩转了起来，代码变得越来越复杂，当然我在这些上面投入了大量的时间，代码复杂了之后我对nodejs的理解自然而然就变得更深入了。后来在公司接手了一个40W行的纯服务端Nodejs项目，挑战更大了，每天各种报bug，报性能问题，报莫名crash，在此基础上，我和团队一起做了很多性能优化，监控，重构，服务分离，这些都是非常宝贵的经验。

对于java，其实我经验也不多啦，当初入职大搜车，因为对前端需求不多，老大让我去看java，于是我就去看了，然后照着葫芦画瓢做了几个早期系统出来（微信服务，官网h5，自动发车），后来每个系统都有人接手，然后他们就被坑了。。不过在java spring里的实践让我解除了spring风格的mvc，对于后来在nodejs中的开发模式的思考非常有帮助。

其实，平常我不太趋向于向别人表达我会多少技术，因为其实单纯的“会”一门技术是非常简单的，但是要真正的“会”则是需要不断实践和投入的，会耗费掉你很多精力。所以对于新人来说，我一般推荐可以看一下别的类型的技术，但是没必要投入太多，等你对于前端真正到了一个瓶颈之后再深入其他的技能。

掌握多种类型的技能，对于前端开发来说，一个是上面说的释放瓶颈的一种手段，另外就是对于思考改进开发模式方面会有所裨益，每个语言和框架都有它核心的一些开发理念，这些理念对于其他语言来说，虽然不可照搬，但是却可以借鉴，这方面我也算深有体会。还有就是没种语言都会对同一种模式有不同的表达方式，理解其中一种，对于理解其他也非常有帮助，例如javascript中的闭包和obj-c中的block，例如ios中的delegate和js中的事件机制，例如cocoa和react非常共性的一些组件化机制，还有很多很多。

### **4) 作为一名老司机，你对前端新人有没什么建议？包括学习方式等等**

虽然还不是什么老司机，但是建议还是要给一点点的：

- 1) 保持好奇心。一位优秀的工程师最重要的品质，持续的好奇心，不管是对于新技术，还是新事物，保持好奇，持续探索。
- 2) 持续专注。工程师要做好牺牲个人时间的准备，因为技术这门活需要大量连续时间的投入，通宵调bug这种事情只是小菜一碟。
- 3) 多看多读。不管是看源码，还是看前辈的总结，或者是关注技术最新动向，一定要不断吸收知识。
- 4) 多总结多思考。光吸收也是不够的，要善于甄别精华，自己在实践中思考，然后总结沉淀。写博客是个很好的方式，既能总结，还能让别人受益，何乐而不为。

今年其实我面试了不少人，能够进入我们团队的人大多都具备上面这些品质，即使现在你还是处于很初级的阶段，但是有了这些品质，你变成大牛是迟早的事情。

## 5) 有没其他问题值得让你分享给的！！

不想太误人子弟啦，说多了我总是忍不住要发招聘，最近脑子里全是这事。。

招聘链接：<http://f2e.souche.com/blog/jia-ru-wo-men-tuan-dui-qian-duan-nodejs/>

后语

大家看完之后有问题的，欢迎在下方评论交流，老惯例获赞数最多者除了可以获得@芋头君的回答问题外，还可以获得由@三川瓜子赞助的《Polymer:面向未来的Web组件开发》一本，时间截至到当日22点。获奖信息将在获奖者下方回复。如赞数相同的，取排序上方的评论。

欢迎各位童鞋推荐前端工程师来做专访，聊聊Ta的前端之旅。

## 专访 | 腾讯微信支付设计中心重构负责人@Ghostzhang

前言

前端早读课第七期专访来了，今天这位朋友的博客，早读君在刚学前端那会就认识了，那就是css森林以及他的群。博客的质量跟思考问题的深度都很值得一看。

正文从这开始～

其实蛮早之前就已经知道并加了前端早读课的官网（<http://www.zaoduke.net>），可惜一直没有怎么去看。几周前在朋友圈看了别人转发的前端早读课，觉得内容质量蛮高的，于是在群里找到了@情封，关注了前端早读课的公众号，现在基本上每天能看一篇。回想自己也曾经想为行业做点什么，到头来却渐渐找到各种理由，渐行渐远。能坚持下来真不容易，敬这份热情，推荐。

关于这次专访

在关注了前端早读课的公众号之后，@情封找我说要做一次专访，我一听心就虚了，专业上的内容现在跟我的距离还真有点远，自从开始带团队，已经很久没写代码了，这并不是好事，好在并未停止思考。以下是这次的几个问题：

Q：自我介绍，网名，在公司所服务的团队，现在负责的项目

A：每次做自我介绍都得经历一次内心的斗争，这是一次自我定义的过程，我是怎样的一个人？我希望别人如何认知我。虽然最后也就是陈述历史，但选择说哪些不说哪些呢？好吧，已经看得出我是个『纠结』的人，现在的网名是Ghostzhang，用了有10年了吧，这也是我在腾讯的ID，目前服务于微信支付设计中心，负责重构前端团队的日常管理。

Q：是在什么样的机缘下走进前端的？

A：我2005年毕业，经师兄介绍进入广州一家电脑代理商的设计部做美工，视觉和页面一起做，当前还在用DW画表格布局。虽然也学过美术，但个人的设计水平远没到商业的程度，所以美工的工作做得并不太好，自己也很痛苦。然后就是看到了关于Web标准的内容，直觉这是以后的方向，于是开始关注Web标准，学习CSS布局。然后是写博客，然后就被当时腾讯ISD的总监看到，约了面试，然后就进了腾讯，开始了重构的路。更多的内容在我的博客上有分享，有兴趣可以看[这里](#)。

Q：这么多年了，你是怎么看待前端？以及在前端深广度上，你是如何考虑的？

A：交互、视觉、重构、前端，这个过程基本上是不能减少的，不同的只是比例。早些年交互、重构不被重视，大家以为不需要，现在回过头去看，这部分的工作还是在做的，只不过没有专门的人独立去支持，更多的可能是产品经理把交互的工作做了，前端把重构的工作做了，甚至把视觉的工作也做了.....

经过了这么多年，交互、重构、前端已经比之前得到了前所未有的重视，都有独立的岗位和越来越专业的知识体系，这是好事，也是发展的规律。既然是规律，就会兴极而衰，这『衰』就意味着变革，在以为重构、前端已经找到位置，可以大展拳脚的时候移动端开始兴起，面对一个新的领域，又是一片『未知』和『机会』。这时候除了感叹之外，很多人开始转型，IOS开发、安卓开发等等，java、C#、C++之类的开发语言又重新进入大家的话题里，这也是规律之中的，关于程序语言的兴衰规律。每一周期重新产出新事物，淘汰不合适的旧事物，而随着人对同一领域理解的加深，这个过程只会越来越快。因此，所谓深广度，有时候只是用于『回顾』或者类似这种『经验分享』时的一个谈资。当然深度、广度还是很重要的，对于当前所处的环境中，更深更广意味着有更强的竞争力。说到竞争力.....又是一个大话题.....

Q：在专业技术上，是否遇到过瓶颈？如果有的话，你是如何突破的？

A：专业上的瓶颈，应该就是对JS的学习上吧，到现在都还是半桶水，也许这也跟我对设计感到困难一样，思维的方式决定了学习的效率，俗称『天赋』。但再怎么困难，都只是比较慢而已，试着找到一些自己比较感兴趣的切入点，会是不错的方式。比如我写的几个工具，大部分都是因为我要学JS而发起的，虽然对于JS的学习并没能真的提升多少，但后面写的比起前面写的在代码上已经有所提升，更高效些、更可维护，自己也没以前那么怕面对JS了。



Q: 有没学习前端的经验，可以分享给前端新人，包括实习生？

A: 兴趣爱好很重要，找你喜欢的、感兴趣的东西学，坚持下去。至于是不是前端，已经不是那么重要了。也许到时你要解决的问题是如何保持兴趣。然后，别只是低头做事，时不时停下来想想，多思考是很重要的，就像是打地基，是不能只是一味把土往坑里填，还需要压平，加上其它一些材料才能让它更坚固，才能支撑起更高的建筑物。

Q: 你现在带团队，在对于新人在团队中如何更快融入团队、怎么学习等有什么建议？

A: 这个话题其实也是很虚的，每个团队的环境不同，所需要用到的方法也就不尽相同，最好请教你团队的同学，会更靠谱。就职业化来说，让自己更专业，更高效是不会错的，没有哪个老板会不喜欢能赚钱的员工。这里又引到另一个话题，不只是新人，所有人都会面临的问题就是外部环境不断在变，需要不断去适应，去学习新东西，从这个角度来讲，每个人都是『新人』，《给不同阶段的自己》中有这部分的思考，保持『空杯』、『谦虚』的心态，保持『敬畏』之心，每个阶段都是一个新开始。

希望未来还能有机会拿出技术上的东西跟大家分享，而不是谈『经验』。

## 专访 | 百度高级前端开发工程师@叶小钗

### 前言

前面五期的专访早读君都觉得看的不过瘾，那么第六期的专访来了，早读君了解他也是通过他博客，每一篇都有一个共同点那就是写的很仔细，很“长”。今天来看看百度高级前端开发工程师@叶小钗的前端之旅。

正文从这开始～

### 1) 自我介绍，网名，在公司所服务的团队，现在负责的项目

大家好我是叶小钗，有以下途径可以更加了解我：

博客：<http://www.cnblogs.com/yexiaochai/>

微薄：<http://weibo.com/yiquinian>

网名来源于霹雳布袋戏刀狂剑痴叶小钗，是个威武的汉子，所以各位不要搞错咯

小钗最初在腾讯打了一段时间酱油，由于团队业务过于边缘被解散了；然后就到携程做移动端框架，在携程的日子是我人生最大的财富之一，那段的经历对我后续的工作产生了很大的帮助。现在在百度创新业务发展部团队负责移动端开发，团队的产品关注于衣食住行中的行，所以产品名称为：百度快行。

### 2) 是在什么样的机缘下走进前端的？

小钗进入前端是比较巧合的，我大学主要用.net做开发，当时比较复杂的项目单个存储过程都写了

1000多行，可以说我在server端的还是有一些项目经验的。但是当时找实习工作时候发现.net好像大公司不太招，于是就开始看数据库的底层设计想走深度（现在看来这个事情太二了），结果发现确实看不懂，当时觉得服务器的水实在太深了，就开始质疑自己的选择是否正确，于是选择了前端。很多年以后，看见前端的发展速度，小钗突然醒悟，原来前端的水似乎也很深啊！

### 3) 有了解到你的博客，对技术的分解很有深度，你是如何去发现这种问题？有没什么技巧或方法？

我所了解到的技术获取方式有两种，第一种是自我学习，一种是项目实践。

可能因为小钗记忆力不好、或者理解能力不行的原因，事实上我不太信任阅读这种事情，我之前尝试过阅读一个感兴趣的博文，然后由该博文去了解很多其它知识点，不断延伸，然而我发现这种方式似乎不太适合于我，因为没过多久我不是又忘了，就是与人讨论的时候完全拿不出自己的观点，所以我的信条是：

听过&看过 < demo过 < 实际项目用过 < 实际项目被坑过 < 实际项目被多次坑过 < 实际项目被多次坑过并且深入研究过

被坑过所以知道痛，真正痛过才会义无反顾的去解决这个问题。以模块化为例，我们是在实际项目中自己的代码经常被别人覆盖，每次都要合并代码时才深刻了解到多人协作的痛，才能知道requireJS的珍贵；我们是在不断的重复合并、压缩代码，才深刻理解到了构建工具（grunt、gulp、webpack）的好；我们是在使用模块化后不能解决独立模块的缓存问题，才会知道MD5带来的便宜。

之前很长一段时间，我都不能明白什么是表现与行为分离，数据与行为分离，我之前也与Arron等大神做过交流，他们只是说了一个MVC啊，于是我便去寻找MVC的实现，写MVC的demo，但很长一段时间我仍然不能了解什么是MVC，直到我碰到复杂业务发现DOM操作不可控，自己探索出解决方案时，再翻过去看网上关于MVC的探讨，便会突然吃了一惊，原来这就是MVC啊！！！！

就是因为这个原因，我相信实践出真知，而工作中总会不停的暴露各种问题，不停的去发现工作中的痛点，性能的瓶颈点，开发效率的提升点，然后尝试去解决他们、优化他们，最后将之整理成文，如此一来便形成了那些博客。

### 4) 你是怎么看待新技术在项目中的使用？

因为我之前从事的是框架开发工作，深刻了解到在框架被10多个团队使用后，你突然告诉他们框架升级了意味着什么，如果那次升级还不向下兼容的话，那么接下来的工作会很难，这样也会伤害业务团队。

所以，我对新技术在生产项目的使用是比较慎重的，要看是不是能做到透明，比如最初我们做webapp时使用的是hash做标识，大家都知道pushState也许更好，但是这个改变可能会导致外面的SEM投放，各频道的入口URL发生改变，如果对将会导致的各个变化没有明确的概念，如果对将要产生的问题没有解决方案的话就需要多考虑了。

从团队的角度出发，公司不是让你来实现你的技术理想的，不能单单因为技术喜好去折腾各个业

务团队，因为每次推翻每次颠覆都会有代价的，如果新技术确实能解决一些问题，当然应该使用，但是如果新技术对现有业务并没实质的提升就应该慎重。

以react为例，如果是要用他来替换我们现有H5框架，我是持否定态度的，因为每个框架都要经过生产的洗礼验证才能真正达到可用的程度，新的框架势必又会产生其它问题，你如何评估这些问题带来的影响，如何评估将给项目带来的风险，我们不能因为其它团队使用了某种技术，我们就必须使用，技术这种东西是为了推动产品进步的，而不是给我们炫耀的，使用新技术要确保自己对其可控，如果没有这种能力还是再看看吧。

当然，如果是新开了一个研究性课题，比如是ipad项目，那么我是非常赞同使用react试试水的。

## 5) 对待知识的积累，你是怎么做到的？平时通过什么方式获取新技术？在深广度上，你是怎么看的？

首先，前端技术是出了名的更新迅速，与此同时前端圈子也是出了名的喜欢分享，这个情况可能一些前端会感到忧虑，会去盲目追求广度，今天学angular，明天学react，但我们都知道如果不能深入理解框架解决的痛点，并且落地到实际项目中，那么这个技术对你来说是不值钱的，所以追求广度前需要在自己已有技术上有深度。

不积跬步无以至千里，不积小流无以成江河，程序员这种东西是做不得假的，很多优秀的前端在工作一段时间就不知道怎么进步了，好的程序员应该让心保持平静，去工作中寻找你需要突破的点，落地你的想法。比如你的项目会有各种统计需求，你如何设计你的程序满足订单来源的统计，就这一个小需求点却需要你了解整个前端乃至服务器端的交互。像这样不停的解决工作中实际的问题，慢慢的你就会发现，瓶颈已经突破啦。

其次，我们需要与优秀的人交流，多参加优秀团队的分享，但这种交流不是无意义的交流，比如你对工程化一知半解，甚至连为什么要以工程化的思想解决问题都不清楚，那么这次交流不会对你有帮助，交流需要明确自己需要什么。

所以，前端们多积累吧，想办法把自身工作做好，如果没有挑战了再换个更难的平台，渐渐的你的深度与广度就都有啦！

## 专访 | 手机百度前端负责人@三水清

前言：

来，新一期的专访又来了，早读君在很早很早刚学前端的时候就有看过@三水清的博客，今天专访的内容非常接地气，看起来～～

正文从这开始～

### 1) 先自我介绍，网名，你大学的专业，在公司所服务的团队，现在负责的项目

我是王永青，网名三水清，09年开始参加工作，算是一个前端老兵了，先后在新浪微博、腾讯工作过，现在在百度负责手机百度（以下简称手百）前端和运营团队，团队主要负责手百的垂类和

结果页的前端工作，另外负责手百快速搭建运营活动的Tiny运营组件化平台、运营API平台，还有手百广告运营位管理CMS和小流量实验平台

## 2) 开源轮子介绍，为什么想造轮子？从造轮子过程中学到了什么？推荐一些工作上常用的轮子

其实我写过不少开源的项目，其中比较出名的就是nodePPT了，另外还有node版本的spider和实时打点服务器等，没有开源的东西也不少，比如chrome的购物插件。

很多项目源自自己的需求，比如：我找电影麻烦，就做了电影网站；逛购物网站麻烦，就做了个chrome购物插件，找到想买的东西就提示我；想做个分享，觉得用PowerPoint做页面效果演示不方便，就写了nodePPT；甚至有时候觉得别人写的东西用起来不方便，就自己fork下，然后改动一下....

在造轮子的同时，你会发现有很多自己不会的知识点需要学习，于是查资料解决问题。所以造轮子是学习的过程，并且是非常好的学习过程。

## 3) 有什么机遇让你走上前端这条路的？这过程中你是如何学习的？

我接触计算机比较早，三年级拿过市里的LOGO画图比赛一等奖，那时候真的很傻很天真，自己连圆周长是 $360^\circ$ 、五角星的角是 $72^\circ$ 都不知道，竟然凭背公式就能画出一个一个的图形，从这个时候开始就喜欢上了计算机。

大学没有考上计算机系很沮丧，不过大二的时候买了台电脑，当时很流行的是校内网（人人网前身），涂鸦板可以填写css和注入js，发现很多好玩的效果，于是一开始是模仿，后来遇见不明白的内容就上网查，查到的都是别人写好的现成代码，看不懂就去图书馆借基本的语法书看。当时玩校内网学到了很多知识，后来一起网上玩的朋友也出了不少牛人。

在这个前端学习的过程。首先要有兴趣，其次要有目标。我学习一个新东西都是直接找项目来做，遇见不会的问题再找解决方案。当时我想把涂鸦板的代码分享给玩校内网的人一起用，就做了个涂鸦板代码网站。这个过程中，除了前端知识外，还学习了利用dedeCMS建站，也会了一些简单的php脚本。参加工作之后一直到现在，都是没事喜欢做网站玩，所以这些年，学会了简单的抓取、数据处理、检索服务，从前端到php知识都得到了很快的提升。

还有一点是要勇于放弃存量，寻找增量。拿抓取来说，一开始我用php来抓取，正则匹配内容。后来学习用node来写抓取，直接用DOM选择器来抓数据。最终我将抓取的选择器写到conf.json文件，扔给spider一个url，程序会根据配好json的选择器，自动将数据处理完返回一个只有想要数据的对象。

所以我认为最好的学习方法是：动手去做项目，而不是看代码，看代码不会发现问题，只有用起来，才会发现问题，然后想办法去解决问题。

## 4) 现在前端的更新速度很快，在广度跟深度上，你是如何看待的？以及如何突破自己的瓶颈？

互联网这个行业，不像医生、会计这些行业，随着工作经验加深，自己越来越得心应手。互联网行业要不断学习，只有不断充电才能跟上行业发展。

前端现在新技术越来越多，让人眼花缭乱，很多新技术还没来得及使用，可能就成了过去。

对于刚刚踏入前端行业的新人，一般的感受是觉得新技术就是前端技术，所以新人对新技术都有涉猎和研究，甚至可能在找工作的时候对于不使用react/angular的团队都打心里鄙视。对于前端新人，面对新技术，我给的建议是：打好基础，切忌眼高手低，当你真的参加工作，你会发现可能切页面会占用你80%的时间。

对于老司机，面对新技术会很彷徨，担心现在不学习，以后会不会被淘汰，但是要学习又不知道从哪里下手，最可怕的是学习完之后，还没在项目中实践呢，发现这个技术又成为过去时了。。。所以对于老司机，我建议：选择性学习，利用自己的经验，进行类比学习，搞明白其中的原理，知道新技术的优劣，如果有机会可以在自己的项目中尝试，也可以利用新技术的思想来解决项目中遇见的问题；但在没有完全搞明白的之前，不建议放到重要项目上使用，更不建议新技术没成熟、生态没有建设完善之前，就开始用新技术颠覆以前的架构。

## 5) 瓶颈问题

首先要知道自己是不是遇见了瓶颈，当你发现做的工作没有挑战性，经常做的是重复性的工作；还有的朋友会私下找我：说自己很迷茫，工作久了之后，觉得自己还有很多不足，但是又不知道从哪里下手学习，不知道如何改变现状，甚至连个有效的学习方法都没有。如果你也有这些问题，那你就要考虑下自己是不是遇见了瓶颈的问题了。

怎样去突破瓶颈，我结合自身和团队发展，说以下几点意见：

1. 很多前端可能不是科班出身（大学也没有对应课程），所以算法和抽象能力可能不如科班出身的强；前端应该更需要注意抽象能力，将遇见的重复性工作进行抽象。手百现在活动特别多，有日常活动、电视合作、日常生活服务运营、电影上映等各种运营活动，以前是one by one的做，后来我们将常用的功能组件化、常用的接口API化，满足PM需求的同时还形成了公司内部的两大平台。所以抽象是很重要的，如果没有这些事情做，可以考虑从小事情着手，比如：以前经常copy的代码想办法模块化，这时候可能会学习到AMD、CMD的库，还会找到一些grunt、webpack、fis的打包工具和模块化管理器；以前经常copy的文件想办法写个工具来解决，这时候可能就需要nodejs知识来写工具。
2. 如果做的工作和项目没有挑战，可以尝试找一些有挑战的项目来做，或者完全陌生的技术项目来做，不断挑战自己，强迫自己学习和运用新知识来解决问题。说到这里，举个例子，在团队分享的时候，我会要求分享的同学找自己不会的知识点给大家分享，不要讲基本的用法、语法，要讲网上不好找不到的知识点，或者直接放到项目中遇见的问题能否解决，保证自己先完全搞明白，然后分享时不被人问倒。
3. 要学会运用工程化的思想解决问题，这是个很重要的事情，前端早已经过去那种“刀耕火种”的时代，现在可谓是进入工业革命时代了，需要用工程化来武装自己。刚加入手百的时候，前端项目是很乱的，甚至连个打包工具都没有，更没有一套工程化的解决方案。我们团队在满足版本迭代的同时，将工作标准化、模块化、工程化，形成了自己的js框架和组件库，并且有一套针对移动端的解决方案。在这个过程中不仅仅自己得到提高，而且团队整体技术水平和视野也得到很大提升。

4. 多参加交流，跟优秀的人和团队一起做事，发现自己的不足，通过他人提高自己。

## 专访 | 腾讯高级前端工程师@余果

---

### 前言

新一期的专访又来了，还记得上一次@灵感的导航站吗？这次我们来看看《全栈工程师的自我修养》的作者@余果的前端之旅。

知道@余果，之前有看过他的博客，后来又在豆瓣上订阅了他写的电子书，最后在出版社的微信号中预览了该书的内容，为了书本的认同感，果断入了一本看了。

大家看完本篇后，有什么想向@余果提问的（内容不限），可以在本文中直接评论，获赞数最多者除了可以获得作者的回答外，还可以收到一本签名版的《全栈工程师的自我修养》，包邮噢～时间截止到当日 22 点。

正文从这开始～

### 1)自我介绍，现在在所在公司负责什么？

我是2010年从西安电子科技大学毕业，腾讯是我工作的第一家公司，我在腾讯一共有四年的工作经验。然后我在QQ空间做前端开发，期间得过几次优秀员工，也在技术上一步步升级，现在是高级设计师，负责管理QQ空间的前端团队。

### 2)如何走进前端这条路？坚持还是突破？

我在《web全栈工程师的自我修养》书中写了我走上前端这条路的契机。我本科学的是软件工程专业，跟操作系统、数据库、算法打交道，但我更喜欢跟用户界面打交道，所以就自学网站开发，走上了这条路。

其实随着前端工作经验逐渐丰富，也渐渐明白前端其实需要掌握的知识和技能比一般的软件工程师更加广泛，比如设计原理、用户心理、上下游沟通，甚至销售心理学等，大学里学的专业基础对我的思维方式也有很大的帮助。但是相对来说，我认为后台软件工程师的工作更加纯粹。

### 3)不管是广度还是深度，在职业上如何突破自己的瓶颈？

前端工程师其实是很辛苦的一个职业，其他职业比如金融行业，会随着经验的增加，筑起自然增长的护城河，会让你比年轻人有很大的优势。但是前端的技术发展太快，护城河没有什么用，你在SVG技术上挖得很深，也许过几年HTML标准就推行新的技术方案，就绕过了你的护城河。如果不持续学习，就被淘汰掉了。

反过来说，前端也是大有机遇的一个行业。因为没有护城河，所以有很强学习能力、问题解决能力和沟通能力的新人，在入行前几年就能获得几倍的收入增长，以及快速的职位提升。

### 4)如何学习？



学习技术最快的方法是直接开始用这个技术去做项目。也许自己在一开始摸索的时候，有一些不规范的用法，但是没关系，先让系统工作起来，然后再根据业界最佳实践、框架来优化你的代码。

不知道怎么开始怎么办？去Google搜一些简单的教程，边学边做。比如要学习使用angular来制作网站，就搜索“angular tutorial”，在第一页找到合适的教程就可以了。

对了，英语阅读能力至关重要，我们缺乏语言环境，所以英语阅读要持续使用，才不会忘记。

没有项目做的时候，看看大牛写的书，我在《web全栈工程师的自我修养》里的每一个章节后面都有推荐国内外的相关好书。

## 5)工作多年，有没给年轻人一点建议？

找自己的节奏，最高效的时候是早上，那就早上早起，晚上高效就晚点睡，习惯番茄工作法就番茄，午休的时候写代码也OK（前提键盘不要太吵）。

分清重要的事跟紧急的事，优先做重要的事情。

要有一个清晰的长远目标，但是不要眼高手低，抬头看星星指明方向，低头看路稳扎稳打。

养成习惯，不要急躁，很多美好的事情都需要时间锤炼，讲英语、跑步、减肥、写书.....都不是一个月就能看到成果的。如果习惯于看到反馈才有继续努力的动力，可以把行动的步骤拆分一下，把动作本身作为目标。比如把每天跑5km作为目标，而不是把减肥10kg作为目标。

不要说自己忙。说忙，要么是用借口把事情往外推，是无能的表现，或者一揽子把所有事情揽入怀中，是缺乏控制力的表现。要去排期、去分配、去沟通、去交接、去拒绝，但是不要说忙。

余果 2015年10月26日

## 后语

重要的事情，在说一遍：

大家看完本篇后，有什么想向@余果提问的（内容不限），可以在本文中直接评论，获赞数最多者除了可以获得作者的回答外，还可以收到一本签名版的《全栈工程师的自我修养》，包邮噢～时间截止到当日 22点。

在校招的时候，有问面试者，为什么走进前端？很多人都说是因为兴趣，兴趣～～

如果你身边有值得你推荐的前端朋友参加专访的话，可以给早读君留言～～

## 专访 | 腾讯UI工程师@灵感

---

### 前言

说到前端导航站，早读君在11年的时候就做了一个ux265.net的，但后面由于维护的力度原因，没

在继续做下去了。

今天我们要聊的这个人，是前一段时间在微博上被各大微博号推荐的一个前端导航站的负责人灵感\_idea，所以我们今天就来让他来分享这个导航站是如何出来的。

先来认识下这个人，昵称：灵感\_idea，现任职于腾讯理财通产品部前端开发组，职位：UI开发工程师，主要的工作是负责理财通日常活动页的UI开发和理财通官网的维护，现在有个开源的项目就是醉牛前端网址导航<http://f2er.club/>

## 由来

现在是互联网的时代，我相信不管是从业者还是普通的用户，都已经或者正在被它的威力所震撼并且受益。从当年初中的时候老师开玩笑说不用出门就可以购物，到现在成了家常便饭。每个人都曾有过一个“神奇”的想法——“如果能有一个东西能干嘛，就好了”。现在都在一步步的成为现实。为什么要扯这些呢，貌似跟话题无关，其实不然，那个理想中的“东西”，说专业点就是产品。我也曾做过做一款完美的产品然后暴富的美梦，显然目前还力不从心。那么作为前端从业者，我比那些入行时间更短，和平时不太关注行业动态的同行们可能接触的、了解的东西更多，我有什么可以为他们做的吗？于是就开始慢慢的思考。

## 思考用加法，做事用减法

学习前端一路走来，我跟很多人一样，逛了很多网站，找了很多资料，我想把它们分享出去，于是最简单的方法就是建立了一个QQ群，然后，自己写了个稍微有点煽动力的文案，到群里或者贴吧里发了几次，慢慢的人多了起来，我不是很追求人数，所以后来就没怎么推过。现在为止我已经在群里分享过很多好的资料和文章，很多会议和活动的资料也都是第一时间献上。慢慢的我发现，这样还不够，回答过很多次的问题总还有人在问，总会有人不知道某个东西的存在或者找不到某个资料，如果每次都重新说一遍，或者重新帮他找一遍，这成本就大了，而且觉得没必要，于是，我就开始想别的更好的方案，我最初想做的是一个学习网站，里面有文章，有教程，有很清晰的栏目分类，并且如果有问题可以直接跟相关的人进行实时沟通。总而言之，就是提供齐全的学习资料供大家查找，提供很便捷的渠道为大家解惑。使得一些被问了无数遍的问题，被找了无数遍的资料不需要再反复的去耗费大家的时间。我觉得它是一件很有意义的事情，当然，不会收费，我对一款产品的追求就是零门槛，体验好，内容充实。

可是问题来了，这是个五脏俱全完整的网站啊，我不是全能战士，自己可搞不定，于是我就到自己加的一些群里去发广告了...也是找来了一部分人，甚至分成了设计、前端、后台组，但是呢，终究还是遇到了两个拦路虎，沟通和执行，慢慢的，也就夭折了。可是我不甘心，就想做出个什么东西出来。于是开始新一轮的思考。

## 从雏形到成形，简单的事情做好，就不简单

复杂的网站做不了，不可以做简单的吗？想过做像html、css、js在线手册之类的东西，但已经有类似的东西了，似乎再造个轮子没有太大意义，也想过写电子书，但这是需要耗费大量时间的，周期长，形式受限，内容也受限。然后我就回想了一下平时自己或者同行所会遇到的问题，所需要的东西，从需求出发，而非最终要的东西。对的，一些常用的工具、组件、库等等，还有一些

社区平台，业界知名牛人的博客，虽然大家多多少少都会知道一些，也会自己有收藏，但我考虑的还是，够不够全，要用的时候找起来够不够方便，这两点就是核心价值所在，怎样做到全？不断积累加集思广益，怎样最方便？做成分类清晰的网址导航放到收藏栏。再加上自己已经有利用百度的首页去进行分类存放导航的习惯，就开始构思和行动起来。

在最初做的时候肯定不是现在这个样子，内容分类也没这么多，都是一点点的思考，寻找，再加进去的，UI参考了一点别的成熟UI，其他部分从文案到图标，到结构都是一针一线编织而成。有人问过我：就只是link？我说是的，虽然看起来简单，要想做好一点都不简单。其实我觉得链接还有一点天然优势，可以很容易的把新的好资源加进来，也就只是加一个链接搞定，聚合成本低，所以，最简单也是最强大。然后我还很多遍的问过自己：要不要再丰富点？只做链接吗？只做前端吗？然后又一次次说服自己：是的。为什么呢？因为我觉得，把看似简单的事情做好，就不简单，专注一个领域越做越好，就够了，而且我自己就是做前端的，这样才能把它做得“最专业、最懂你”。

## 分类和体验

### 分类方面

从最初上线到现在，有一些分类被新加入了进来，有些是因为最初没有想到要做，有些是因为资源很少，还不适合拿来做一个单独的分类。其实上线之前就在进行着这样一个过程，出于哪些方面考虑呢？

- 一、需要用到的、实用的东西和常见问题、解决方案
- 二、可以获取相关专业干货的渠道
- 三、可以参与其中的业界活动
- 四、可以学习的好教程
- 五、可以学习和借鉴的团队/个人博客、博文、demo等

以上分类基本涵盖了可以聚集的大部分网络资源入口

### 体验方面

- 一、色彩不要太扎眼
- 二、哪些最常用到，放在前面或者打开就是，哪些是空闲时间可以慢慢翻翻看
- 三、怎样让用户也能够比较方便的参与进来
- 四、点缀效果适当就好，过犹不及

上线之前有过调整，上线之后根据大家的反馈又有一些改进，以后还会不断的改善

## 人生在于折腾

在导航发出来之后，有朋友说我，你还挺喜欢捣腾网页的啊，我说那当然，就是干这个的嘛。其

实不只是网页，我觉得做技术本身就应该出于爱好，玩出乐趣，当然，人人都有惰性，都会有倦怠期，或者称之为瓶颈。怎么看待瓶颈呢？我觉得是人性的弱点在作怪，跟技术本身关系不大，你不敢去尝试新的陌生的东西，总在熟悉的范围内活动，永远不会迷路，永远发现不了新的东西，找不到新的乐趣。当然，也并不是鼓励一味的追求新的东西，其实我们人对每一个事物的认知基本都是片面的，知识就像是一张拼图，经常是没有确定的方向，我们找到一块就往上拼一块，都是逐渐的去完善，从广度、深度、细度、灵活度、技巧、量级、形式等，总有我们未曾触及的地方，只要我们想要去寻找和探索，就没有无法前进一说，敢于突破自己，敢于折腾~

## 理想主义者方能成事

这句话是我最近刚感悟出来的一句话，具体什么意思呢？理想主义者给人的印象，简单说可以归为三个字“想得美”，呵呵。想得美=不现实吗？其实不然。想得美会带来什么呢？首先是思考，思考一件事情的意义，思考怎样能够把它做得更好，既然想了这么多，不去动手把它做出来又憋得难受，然后就会整理思路，着手规划，将想象中的东西联系实际，怎样能在现有条件下真正的把它实现出来，然后从头开始一点点的做，边做边补充，边修改，这当中必定会经历一些纠结的思想斗争，要不要这么做？这块是不是必需的？是不是重要的？甚至做到一半的时候，回过头来去怀疑自己的初衷，这么费力的做的这件事，何苦？有时候，把自己以前想到的一个看起来很好的idea后来要砍掉的时候，还是会痛的，但为了最终的大局，依然能坚定的走下去。“理想”就是坚持去把一件有意义的事情做成，做好的动力所在。

## 关于读书

我本人买的书不算少了，了解的也不少，导航上面列出来的那些书籍，不敢说全部，90%都是热销经典书。不过真正能读得进去，然后在实际运用中用得也不多，学习是个很微妙的东西，你吸收到了多少，能做出什么东西，很大程度上不取决于书上的铅字，而是你本身的投入程度和思考深度，还很重要很重要的是实践。这里可以引用一段我在知乎上的回答：

“学习技术的人问的最多的问题，推荐好网站，好的在线教程，好的书籍等等。我们总是会觉得课程或者书籍简单，介绍的都是概念性或理论性的东西，浅尝辄止，我们看到的一些线上的网站或者demo源码，代码结构复杂——知识一看都知道了，实际写代码却没有思路，更不知道怎样写是更好的，好像二者之间有着跨越不过的鸿沟。其实，就是有些浮躁，缺乏实践的表现，理论是必要的，它能保障你知识体系的系统和完整性，但不动手是不行的。”（有删减）

书有优劣之分吗？当然有，上面说不取决于它，但它也能够给你带来很多收获和启发，关键看书籍的章节内容安排、示例和语言描述，是否跟你现有水平相吻合，是否能把带入进去，而不是大家都说好，都说经典的书就适合你马上拿来读，另外，每本书都会有它的侧重和详略，也都会因为书本限制，内容有所保留，当然，每本书都是人写出来的，不可否认作者会尽自己最大的智慧和努力去解释和表达清楚书要讲的内容，但也多少会用自己更擅长的方式和加入一些自己的见解，读者读起来是否能对等，就当另说了。从积极层面来说，好书还是应该多读，但要书尽其用。

## 关于大公司

能进入大公司，应该是每个技术员或者将要毕业的学生最初的理想愿景，有些优秀的幸运儿比较

顺利的就获得了机会，大部分还是没有，但是又会去想。

或者自己尝试过，或者听人说，总会觉得，门槛高，工作压力大，这似乎是人们对于大公司，对于名企的印象。那么为什么都还想去，直接一点，冲着好的福利待遇和所谓光环。

不过最近两年，大家应该也都有感觉，移动互联网的热潮，互联网+的概念，互联网向传统行业疯狂渗透，IT人也随之疯狂，创业公司如雨后春笋般出现，动辄融资千万或亿级，不论是从办公环境、设备，还是待遇上来说，大公司似乎已经没了优势，仅存的优势，就是光环。

那么这个光环是否也是必须的，显然不是，有句话说得好——“你认识牛人不代表你牛，你在的公司牛也不代表你牛”。如果你只是在意去在意那个光环，而没有从自身有一个清晰的职业规划和目标，不断学习追求进步，那么也只是背了一个虚的壳而已，说不定某个时候，就轻易的破掉。真正能代表你的是什么呢？

个人比较欣赏最近在余果博文里看到的一段话：

“优越感应该来自输出，而不是输入。阅读一本书不应该有优越感，要自己写一本；使用某种新潮的编程语言不应该有优越感，要创建一个项目；买了摄影器材不应该有优越感，要拍出伟大的照片。”

这段话后面当然只是举例子，重点是第一句，我们有很多速食的东西，看过，了解过，大概知道，等，当我们能够输出什么的时候，才代表对其有一定的理解和掌握。

我所知道的不在大公司的，也有不少优秀的团队和个人，工作的很好，也有着不错的输出。大公司的业务量级更大，用户更多，赚钱更多，所以相应的能提供更多的东西，也乐于吸纳优秀的人才。但并不代表你需要通过它来证明自己。就像我QQ签名一样，“保持一颗纯粹的心”，做自己喜欢的事情，简单快乐就好。

上面谈的是不盲目追求进入大公司，但如果你有幸到了大公司，需要在意和可以为我所用的是什么呢？

一、工作流程

二、工具、规范

三、平台资源、优秀的人

是的，你会接触到很多之前没有接触过的东西，有机会和优秀的人一起讨论问题，看到更多好文章，听一些分享。仿佛进入了一片更加广阔的海洋，你自己也显得更加渺小，你需要更加的谦逊和更加积极的吸取营养，让自己得到成长。

好了，说了这么多，有些是跟f2er.club直接相关的，有些是受情封之邀扯扯的，为了让大家更清楚的知道背后的故事，也顺便谈了一下自己对于一些事情的看法，希望能给一些同行一点点的帮助，欢迎多交流。

**专访 | 互远信息实习生@钱育辉**

## 前言

还记得第一次专访腾讯实习生童鞋@chee吗？这次早读君又带来一位上海第二工业大学信息安全专业大三的童鞋@钱育辉，来看看他所走的前端之路。

认识@钱育辉，应该是前端早读课沙龙群的@CIN介绍进群的，平时会在群里聊聊，扯扯蛋。

正文从这开始！

事实上我不是一个很喜欢回忆很久以前事情的人，因为我对于这些记性比较糟糕，就像javascript垃圾回收机制。所幸之前有写日志备份资料的习惯，从中一点点回忆曾经摸索的经历。因为感觉自己技术比较渣,但一直坚持学习，今年刚入大三也算完成自己大一制定的职业规划（现于某创业小公司作为前端开发）。

## 大一

其实高三之前我有过编程经历,不过一直没想到自己会走上web开发这条路。如果早知道如此，初中便跟随大神S学php或许现在还能有点沉淀。:)

大一刚进来就准备当学霸,没错,目标就是这么”伟大”。当初一头扎入学习的海洋。装了半年学霸后除了这个头衔能伴随我的大学生涯，我收获了：

1.知乎

2.时间管理

3.大量阅读技术类书籍以及编程思想相关的书。

4.Mooc（完整的跟完史记和中国历史人物秦始皇。玩票的跟完andriod开发）

5.Acm经历,后来觉得自己不适合于是逃票了。

因为大量的阅读，web这块我好像名词懂的很多。

也是这个阶段,我有了一种职业应该是网站开发这块的想法。

### 第一次接触前端项目

当初不知道名声是怎么传出去的,反正后来有个朋友介绍一个外包,一位艺术系老师需要一个产品展示站。事实上我当初水平很浅，但是初生牛犊不怕虎接了下来。这位艺术系老师自己设计网站，设计稿有了,整站用php做,上面说过大神S是初中开始学php的，因此对于整站开发我也有大概的思路，去图书馆借了几本书.然后就直接开做了,网站基于PHP+mysql.后来这位老师在开发期间请我去咖啡馆调试细节(人生第一次咖啡馆)。结束后还请吃了顿大餐。并付了报酬。这对我来说鼓舞很大。而且因为这个项目的缘故我开始对前端深入学习。

### 大一暑期实习

因为我奶奶的对我的关心，于是我被一名亲戚托关系送到一家公司实习，打杂。当时很惶恐，因为感觉自己什么也不懂。于是每天早晨公司都没开门的时候就到了。那家公司Boss人很棒,他给我



指了学习的道路。那时候第一次听说有Bootstrap。第一次听说BackBone(但对于当时的我来说, 那很难)。在公司其实没做什么产品, 但是Boss给了我很大的触发, 他让我负责一个项目的设计。这对于我来说受益无穷。不过当时公司的前端因为产假并不在,因此我没有接触到真正的前端应该如何工作的。后来因为Boss让我学Node.js,没错, 大一暑假我就知道了Node.js,当时如同天书一般。因此我逃避了。干满一个月我提出了结束。

现在想想当时内心是非常恐惧的,因为当时确实什么都不懂,一下子接触到Node.js这么高大上的东西,感觉自己无法掌握。

后来我自己琢磨着, 悟出了一个道理,一家公司需要的是能干活的人,一旦你进入公司,你不可能从基础做起, 你需要的是能产出产品的能力。

于是我开始发奋学习。

## 大二

大二我正式确定我的职业为前端,大多数前端都是野路子,我也是,因此我大量阅读,幸好当初还有个更加困难的东西-ACM在摧残我的自信,因此我学习基础的并没有多大问题。

后来在ACM群一位学姐邀请我去实习,闭关几个月的我感觉应该做些东西, 因此在那家公司实习了几个月。后来因为一些原因,我感觉不到我的进步,于是我辞职了。这次不是逃避, 而是看清了自己努力的方向。接着我学习了Ruby,学习了Rails。第一次知道社区,从社区中我看到了很多,学会了很多。还线下参加过活动。后来还开发了一个阅读分享的Web应用。从后台到前端一手包办。

不过那个阶段心性没定, 还想着学ios, 学swift。后来学了一个月, 觉得非人哉,弃坑。

这个时候第二个转折点来了。

这个时候, 不知道名声怎么传出去的, 咦, 好熟悉的感觉。

有人找我一起做外包。当时没啥事, 就应下了。

期间各种学习各种磨合各种感想我都忘了。于是我们来说结果, 第一个外包飞单了, 第二个外包拿到了钱。

## 大二暑期

这是我人生的爆发期,我什么地方也没去就窝在家里学习。我定下了学习目标, 按部就班。

我学的是Node.js和React。

学Node.js的时候顺便学了Express4.x, 并写了几个项目。

学React的时候结合Node.js写了一个私人定制版咨询应用, 现在我每天看的咨询都是它爬的。

还在github开通了博客, 记录学习经验。这个期间我积累非常多的东西, 明白了前段自动化流程, 学了gulp。期间跟大神S聚餐, 看到他在给一个开源翻译项目作审核。于是我在博客开了一个国外优质文章翻译系列, 坚持了20多天, 把积累的大部分给输出了。这个过程让我眼界大开。

并且我养成了屯学习资源和每天学习的好习惯。

暑假两个月，脱胎换骨。

大三

开学初直接跟暑假谈好的公司,于是真正成为了一名前端工程师。

于是我完成了大一的职业规划之旅，并开始我大三的职业规划之旅。

我的学习经验

曾经为了回报知乎，我做了一个前端技能树的网站，结合自己的学习经验我谈几点：

初学者：

1.学会翻墙

2.学会自己搜索资源（更全的学习曲线知乎现在都有很多大牛总结了）

3.学会坚持

初学者Js路线：

第一步果然还是w3school。

<http://www.w3school.com.cn/js/index.asp>这个，刷的又快，半天能解决，然后有一点概念你就能写最基础的js。

第二步

根据自己的学习能力分为看书或者视频，但是！没有人能从视频中把js学的很好，顶多就是会用！

因此看视频能掌握很高深的内容只是一个自我心理欺骗而已。所有的知识除了书本就是网上大牛的文章或者回答。

这一步的书我推荐Head First系列的Javascript系列，我认为对于新手而言，没有什么比直接上手制作一个小项目更有成就感更加有效果了。在这一步中可以去看慕课网的一些基础视频。

第三步

学习一些常用框架和库，推荐Jquery上手，然后可以接触其他框架，并不需要强迫学习，不过还是建议用用看。比如BackBone。

第四步

学习一些Mvvm的框架或者库，比如React,Vue.js,Avalon。

基础

不懂就百度，百度不到就google，不会google就去学翻墙,不会翻墙就学会买买买。这么简单的事

情不需要说第三遍。

国外的内容超级多有质量的文章。还有多订阅一些国内的前端优质内容。这年头，咨询是王道。

还有开发工具的使用一个是sublime2/3，一个是国内开发的Hbuilder.至于其他Atom什么的我没怎么用就不说了。

善于自己去学习这是非常重要的。

初学者Css路线:

第一步果然还是w3school。

<http://www.w3school.com.cn/css3/index.asp> 这个，刷的又快，半天能解决，然后有一点概念你就能写最基础的css3。==>你会发现这边的说辞和javascript一样一样的，毕竟最简单的就是刷概念。

第二步

1) 根据自己的学习能力分为看书或者看文档。

2) 所有的知识除了书本就是网上大牛的文章或者回答。

这一步的书我推荐Head First系列的css3系列或者Html系列，我认为对于新手而言，没有什么比直接上手制作一个小项目更有成就感更加有效果了。

3) 在这一步中可以去看慕课网的一些基础视频。

然后就是去大漠的站学习<http://www.w3cplus.com/>

想提高自己，看不懂一些含义？

张鑫旭的博客你刷几遍就行了。 <http://www.zhangxinxu.com/wordpress/>

第三步

学习一些常用框架，推荐Bootstrap上手，然后可以接触其他框架，并不需要强迫学习，不过还是建议用用看。

第四步

去codepen.io把popular的你挑喜欢的看一遍，自己尝试一遍。但是，codepen可能需要翻墙，还有些效果需要js这类。还有些css代码是sass来写的。所以还需要自己多提高然后再去尝试这些难的。

关于Sass

请自己查资料学习，快的一天就能上手。主要是编译问题，可以看我博客前端环境的搭建。

关于书籍推荐,首先我个人认为书是死的，你能从什么书中学到一些东西，那么这本书就是好书。然后看书的时候你必须做笔记，必须做笔记，必须做笔记。嗯，重要的事情说三遍。

## **\*\*书单\*\***

javascript:

JavaScript DOM编程艺术

JavaScript Dom高级程序设计

JavaScript高级程序设计（第3版）

精通JavaScript

ppk谈JavaScript

锋利的jQuery

高性能JavaScript

JavaScript语言精粹

JavaScript权威指南

编写可维护的JavaScript

Effective Javascript

JavaScript设计(和下面的是两本书哦)

JavaScript设计模式

CSS:

CSS权威指南（第3版）

精通CSS

Css禅意花园

图解CSS3：核心技术与案例实践（大漠的书，必看。可以当成字典。）

总结

现在我每天依旧坚持学习刷咨询刷优秀博客，不管每天上班还是上课，我都会抽时间看书而且写博客。

我现在觉得写博客是提升水平的最好途径。因为你只有输出才会有积累。

初学者最好建立自己的技术博客，当你学到一定程度时你会发现自己走过的路有多美。

后语

早读君看到这个的时候觉得以前我的大学都白过了，整个下来好丰富的大学经历。

早读君为什么要做这么这件事，是因为觉得每个人走进前端这个行业都是非常有故事的，并且每个人在学习的过程中都是非常精彩的。每个人的故事都是可以值得参考。

## 专访 | 腾讯实习生@Chee的前端之旅

### 前言

为什么会有这类型的文章呢？是因为有不少的童鞋在公众号后台或直接在微信上跟我探讨如何学前端，在小公司里就我一个前端该如何学习？还有平时在面试的时候，早读君都会问的一个开放性的题目是你为什么想做前端？

带着这些疑问，早读君觉得这种类型的人应该还蛮多的，所以我就跟今天的要男一号@chee聊了想法，让他分享自己在前端的道路上如何走的？

正文从这开始～

### 1) 自我介绍，包含现在在哪里实习，所学专业，现在在公司负责什么项目或业务

Hi，大家好，我是Chee，来自广东工业大学，计算机科学与技术专业。目前在腾讯公司实习，主要负责一些移动端 H5 运营活动小页面的开发。现在手机QQ的手机版官网就是我参与制作的（可以用手机打开<http://im.qq.com>查看）。

### 2) 什么时候认识前端，如何入门的？平时怎么学习的

我接触前端比较早。在我初中的时候，WHATWG（其实我已经忘记这个组织的名字了，是google找回来的）发表过一系列的web标准化教程，从0开始解释HTML是什么、css是什么、js怎么用，有点像今天Head First系列教程的感觉。那时候Opera公司在官网上提供了这个教程的中文的翻译，我就是跟着那个教程初步了解了前端的。可惜后来中文版下线了，再后来这个系列的教程我也找不回来了。

真正开始做项目也是大学以后了，和大部分前端初学者一样，最初也是从w3school里面学习css和jQuery，后来就是根据项目的需要用到什么学什么。渐渐得在大脑里形成一定体系之后，又去买了《js高程》回来看。应该是属于比较中规中矩的学习路线吧。到这里的时候其实已经有点沾沾自喜了，因为项目里遇到的困难几乎可以很快解决、遇上不会的css样式也能通过搜索引擎找到答案，再加上一直使用别人封装好的Bootstrap，几乎不会遇到什么问题，自己开始有点飘飘然。现在回头想想，只能用“井底之蛙”形容当年的自己。

直到后来在慕课网看了D2峰会的演讲，才发现现在的前端和我当年学习的已经大不一样了，各个公司都在讲自己的“工程化”方案、大家在用的也是以前从没听过的框架和工具，于是赶快恶补。由于我手里的项目大都是中小型项目，一个人课余时间一两个月就可以开发完成了，所以我给自己定下目标，每个项目都要比上一个项目多学点东西。从最原始的jQuery+hash实现简单的路由、到自己写jQuery插件、再到Angular、gulp+sass、最近也在着手一个基于React.js的个人项目。

我是一个比较懒的人，累了就喜欢刷微博。后来觉得这样不行，就直接在微博上搜索“前端”，把那

些我能找到的大公司的前端微博全部关注了一遍，看看他们都在研究些什么。这才慢慢接触到了最近吵得很火的 BFC、ES6 这些概念。后来也找到了早读课，所有公众号里面只有情封的早读课和钠盐的科技美学我是天天都看的，因为推送的文章质量都很高（不过钠盐现在不发语音了，所以只剩下早读课了）。

另外近几年来中文前端圈里面有很多大大小小的分享会，我能接触到的有 D2 峰会、iWeb 峰会、CSS 大会等等，当然个个地区还有自己的小分享会（好像情封大大也在厦门组织了分享会，以后有机会一定要去听听），像一些大型的分享会一般都会有人录像，这些分享会上的内容质量都特别高，非常适合找来学习。真的非常感谢这些活动的组织者，给国内前端圈搭建了这么多优质的交流平台！

### 3) 在学习遇到的困难，如何解决？包括技术问题

我在学习的过程中主要遇到了这么几个比较大的问题：

知识点零散前端的知识点特别零散，很难找到一本书可以穿针引线，把所有的知识连接起来，这也让我慢慢理解了，为什么计子 winter 大大经常会总结一些脑图。其实跟着 winter 老师总结的脑图顺藤摸瓜，就能把前端的知识结构比较全面的过一遍。

学习新框架时，中文资料不全面由于最近在学习 React，发现中文资料是在是太少了，所以很多时候要硬着头皮去看英文。可以给浏览器装一个翻译的插件，慢慢看多了也就适应了。

容易沾沾自喜前端总给人一种入门很简单的错觉，学习的过程中很容易自以为已经触及到天花板，没有什么好学得了。其实这是大错特错的，每当发现自己飘飘然的时候，一定要多想想，自己已经学会的知识是不是真正掌握了，还是说需要实现某个功能时才去搜索引擎查找，事后不再做总结了。关于这点，张鑫旭前辈有一篇关于“瓶颈”的文章很好的解释出了问题。

而剩下的，关于具体技术问题（比如移动端的各种奇怪现象、怎么跨域、性能优化等等），只要你能很好的用语言描述出来，不论中文英文，网上总能找到答案的～

### 4) 在实习面试过程中有需要注意点的吗？

网上有大量的面试题、面经，我觉得只要你准备好了，面试的问题应该不会太大。这不是鼓励大家背面试题答案，正如百度 FEX 的面试题所说，如果你能把面试题都准备好，实际上你的学习能力已经到了公司需要你掌握的地步了。另外，毕竟自己在前端项目中遇到的问题还是比较片面的，所以如果能找一些题目来做，也能扩宽自己的知识面。

再之后就是要认清楚自己的强项了，想想看你相比其他同学有哪些优势。如果你看过 jQuery 源码分析、试图理解过为什么 angular 的性能一直被人诟病，都可以拿来说一说。

另外，讲一个面试的小技巧。一般体面的面试官都会在最后问一下你还有什么问题，这时候你可以稍微想一想，自己在这场面试里面的表现令人满意吗，如果觉得自己明明准备的很充分，但是整场面试下来都没有表现出来，你可以在这时候跟面试官多聊聊，我相信面试官也不愿意错杀任何一个适合他的同学～

### 5) 在公司实习中，要注意哪些特质或注意点



单纯从实习的角度来说，我觉得只要拼命的学东西就好啦，毕竟还不知道最终能否在公司留下来，所以学到一点算一点。

另外我觉得在公司里，理解好自己的导师和leader要你做什么是很重要的。最近在参与一向新技术的预研工作，发现同事和leader对预研的期待是不同的。leader的角度来说，更希望看到的是这项技术能否投入到生产环境里，所以会看重性能、网络开销等等和现有技术的比较；而同事可能更看重这个技术具体的实现细节、如何编码等等。知道了大家想要了解的东西，就能更有针对性得进行了研究。

## 6) 校招需要注意什么？

时间点很重要，不要错过投简历、笔试面试的时间；

找个基友一起喝酒聊天，正常来说校招肯定不会一帆风顺的，能有几个哥们一起吹吹水谈谈心，可以很好得释放压力。

## 7) 我的书单

这一两年来看了不少前端开发方向的书。一些很经典的《javascript高级程序设计》、犀牛书等等在这里就不再介绍了，推荐几本对我帮助比较大但很少被提起的书籍好啦～

### 《web性能》

这是一本详细介绍web中可能遇到的性能问题的书。算是我计算机网络的启蒙教程，哈哈。作者并不仅仅提出了 web 中可能会引起性能问题的地方，还对每种性能瓶颈做了详尽的分析。起初阅读起来还是会有点痛苦，但是随着逐步的深入，你会发现自己学到的不仅仅是性能优化的结论、而是一整套的分析思路。当然，如果你完全没有计算机网络基础，书中的一些概念可能会给阅读带来一些障碍，不过对于计算机专业的学生来说，这些概念迟早都是要掌握的，提前预习一下也无妨。

### 《javascript异步编程》

这是师兄买的一本书，我借来翻过。js 最被人津津乐道的内容之一就是异步回调函数的编程方式了，在 js 的世界里，通通都是单线程，这甚至也成了 Node.js 出现的契机。但是这种方式也会带来一些问题，比如我们常常说的“回调地狱”。这本书就是用来告诉你，怎么样才能优雅地解决回调地狱的问题。不过由于篇幅有限，这本书对每一个解决方式都只解释了最精华的部分，因此看完之后还要自己继续学习。其中的 promise 模式绝对是重点中的重点。看完书后可以参考新浪微博 @寒冬winter @十年踪迹在2015年4月10日左右发的微博，里面有他们关于“怎么用 promise 实现红绿灯”的争论（其中有大量干货代码，感谢几位前辈我们理解 promise）

### 《jQuery技术内幕》

阿里的高云前辈写的 jQuery 源码解析，这本书比网上的代码解析要讲的透彻很多，当然也是一本很难啃下来的书。

### 《你不知道的javascript》

原本看到标题和目录，觉得这本书好像也没啥，不就是把js里面的一些语言特性集中到一本书里面吗，我上网看看博客就行了。但是真正拿到书读下来，就会发现根本不是这样。这本书对每一个简单的知识点都做了非常详细的解释，而且配上了一些浅显易懂的插图。我感觉用来了解js的语言特性非常有帮助。

后语

最后的书单，是@chee是自己补充上去的，非常感谢@chee的支持！！虽然每个人的如何走上前端，如何学习的道路都不一样，但别人的经历，都值得参考。

最后非常再次感谢！！

## 【同说】机械转行前端，半年零基础自学的心路历程

---

前言

在入前端坑之前，有几个是科班的。今天来前端早读课之同说的是西安@Michael.Lu童鞋，在分享他这跨界学前端的经历。

正文从这开始~

### 背景介绍

关键词：前工作状况(机械)、离职经历、心态转变、目标确定

大家好，我是Michael.Lu，一枚前端萌新。早在2016年4月份之前，我还是一名大型石油装备制造公司的机械工程师，勤勤恳恳，工作强度不算低，即使在规定时间内完成任务但还是得常年形式加班(大概896)。生活节奏中规中矩，几乎泛不起一点点涟漪。变化的是眼花缭乱的三维机械模型和无穷无尽的生产蓝图，不变的是工作内容以及职位和薪资。虽然平时表现出了足够的上进心，但在这样的环境中，我开始慢慢意识到，在应该升职的时候总是会有奇人从天而降稳准狠的击中我的晋升目标。在应该加薪的时候总是能赶上国际油价下跌导致公司利润下滑开支紧缩。我。。。

从2013年7月大学毕业，全国包括我在内一共600多名大学生进入该公司，14年走了一半，15年还在坚持的不到100人，到16年3月还剩不到50个，想想是我也算持久坚挺的一员了。可是就在3月的某一天早晨，新空降的不知道第几任的领导来给我们打招呼，我的脑海里浮现并不是一片欣欣向荣的未来工作情景，而是一眼望穿的未来10年内的半养老生活格局，旧国企般的腐朽人际关系网，永恒不变的工作内容。再加上每逢佳节倍思亲和亲更思我的情节，离职的念头瞬间爆炸，当天就提出了离职。从提出辞职到真正变成无业游民花费了我整整一个月的时间，其中不乏各种HR画饼挽留、人心尽失工作交接障碍以及签署发射原子弹般的复杂离职签字流程。就在最后总裁拿起笔准备签名的一瞬间，仿佛高考最后一门考完交卷一般的清爽。皮皮虾，我们走！

之后4月到7月是我人生中最开心也是最失落，也是我永远无法忘却的时光。这算时间里，可谓是云游四海，心灵在天堂，身体在地狱，但却乐此不疲。孤身走遍大江南北去寻觅故友，虽然故友

多数都在上班没多少时间接待我，也没去多少景点，但是洒脱光环加身，怎样都开心。后半段时间一边返程一边旅游一边开始准备简历，落实下一个落脚点，这也是我心情俯冲直下的转折点。真是不看招聘信息不投简历不知道这个行业有多残酷，需求都是要么应届毕业生，要么5到10年工作经验的高级工程师，中不溜的待遇和要求根本无法直视，这个时候我才意识到我到底是有多年轻多不成熟。几经投递无果之后，带着比上坟还沉重却又不能表现出来的心情回家。继续投简历，要求底线不断降低，投递频率不断上升。接到面试通知电话的公司，不是远在天边的未知企业，就是连糊口都难的乡镇企业。落脚点毫无进展，慢慢感觉在家里呆着如同身陷囹圄，心情从冲动、激动、开心到气愤、失落、后悔甚至是绝望，心态逐渐瓦解崩坏，最后为自己贴上了loser的标签。

所以，三思而后行，当你面临是否跳槽或是否转行抉择时，无论答案是什么，一定要清楚这绝对不是一个瞬间解，而是一个经历过漫长的思想斗争和深思熟虑之后的产物。你需要先思考现在的工作到底是哪一方面差强人意，具体到关键点比如：薪水与付出严重失衡、晋升通道不清晰明朗、工作内容重复冗长、工作强度大身心俱疲、行业迅速衰退大背景不景气、还是人际关系不和谐领导不赏识？再想想针对以上情况可以通过什么具体的途径能够改变现状或者与现状共存从而活在当下。如果可以得到缓解，那应该坚决否决自己的辞职念头去改变去打破。如果不行，ok，要在辞职之前就未雨绸缪，尽量在落实好下一份工作的时候或者时机成熟的时候再提出离职。绝对不能像我这样头脑发热，自命清高，觉得之前工作那么努力也获得了多方肯定，找工作必然不是问题，升职加薪是迟早的事，真是too young too simple。这样的离职就如同炒中国的股市，一片绿的时候你一卖它就涨，一片红的时候你一买它就跌，气愤的认为是命运的不公，其实是自己根本没有看清现状，做决定不过脑子的结果罢了。

自己挖的坑，还是得自己填，解铃还须系铃人。6月下旬心情烦躁到极点的时候，转行的想法应运而生。于是再也不敢草率做决定的我决定找我爸商量一下，毕竟现在也老大不小了，良心迫使我不能啃老，但又迷失方向，希望能得到一个曲线救国的方法。一起喝点酒微醺之时，鼓起勇气告诉他我喜欢音乐弹的一首好吉他，楼下就有一家比较大的乐器店正在招聘，我可以教吉他可以卖乐器balabala一大堆。而他就好像没听进去一样，开始回忆我上大学前报志愿的情景，他说的话我到现在还记忆犹新：“当时石油行业确实如火如荼，所以我帮你报了这个专业(油气储运工程)，虽然行业兴衰轮转，却没想到你刚好赶上了退潮。你从小就喜欢玩游戏，为此没少去网吧抓你，也没少揍你。如果当时给你报计算机专业就好了。”我知道他根本不想让我去教什么吉他，也没听进去我说的，可是，重点是我听进去他说的了！！！真是一语点醒梦中人，我瞬间就想到了两个从事互联网行业的同学(一个纯前端&一个半前端半后端)，那时的心情难以用语言来表达，仿佛是行走在黑暗漫长走廊的人看到前方微弱光线的瞬间。不晚！绝对来得及！我疯狂的告诉自己。与此同时我向这两个朋友咨询了一下行业情况，于是确定了前端之路。And I take a deep breath and I get real high!

在此先行感谢听我啰嗦的朋友们，这也是一种倾诉，压抑半年的感情没想到以这种形式得以释放，喜悦之情难以言表。随后就是开了挂一样的自学经历了。

## 自学经历与经验分享

关键词：编辑器选择、自学方法选择、笔记、写代码、时间管理

这是我自学前端时的第一个页面。当时还在纠结到底用什么编辑器好，于是到网上看各种编辑器的优缺点，从Vim、Sublime text这种轻量级编辑器到Dreamwaver、WebStorm等大型IDE，有同样经历的朋友应该也知道，网上褒贬不一，众说纷纭，看的是一头雾水。现在想想当时真是可爱，所以在这里告诉还在为编辑器纠结的刚入门的朋友，唯一的择偶标准就是颜值(顺眼)和身材(顺手)。因为处于启蒙阶段的我们实在是没有什么必要去关心它们多么强大的功能，不仅用不到，还会在分散注意力。如果硬要我推荐一个，那就Sublime Text3，不为别的，就是启动快！而到后面能力有所提升，视野渐渐开阔以后，编辑器的选择也就没有那么困难了，需要用到什么功能就用什么编辑器，公司需要你用什么就用什么，不需要你自己选择。而且就我目前的经验来看，随便哪个编辑器都能很快上手，因为基础功能都一样，就是熟悉界面、快捷键、特殊功能的问题了。你听过跨浏览器有兼容性问题，但你听过前端代码跨编辑器有什么问题吗？

从第一个html文件到现在为止，去掉期间各种节假日、推不掉的旧友小聚以及春节前自制力失控玩了近一个月的游戏，粗略估算自己真正在自学的时间只有半年左右，很多人都说我进步快。讲真，如果你也有我上面阐述的类似经历，你肯定比我快得多。因为我不仅自制力差，而且从小就看不进去白纸黑字的书。所以无论你现在是在上学，还是准备进击前端行业在自学，亦或是已经工作但是想通过自学来提升，那就行动起来吧，You can do it！

很多人认为玄学说错了是自学有什么好说的，不就是摒弃一切杂念，埋头苦读奋笔疾书嘛？其实不然，在我看来自学这种形式的学习最容易缺少动力、鞭策力，没有达到自己既定目标而中途放弃的人不胜枚举，尤其是对那些自学时间少，条件有限的人群而言。最形象的例子就是自学好比健身，锻炼少没任何效果健身卡垫桌脚，锻炼多伤筋动骨身体吃不消，而且在没有任何指导的情况下还容易练的左右肌肉不对称，自己看着都难受。所以，来不及解释了，快上车。

自学接受知识基础方式从大方向上来讲个人认为主要分为死磕专业书籍和死磕专业视频两种。只有最合适自己的，没有大众最佳款。所以一定要在自学前先对自己有个客观评价，然后对号入座，这样才能更高效的自学。这非常重要，试想如果看书或者看视频的时候都能看睡着了，那还谈什么自学。如果能清晰的明确自己接受知识的习惯，就可以有效的避免这些不经意的厌倦或者困意，从而在自己可利用的时间内，最大程度的缩小不必要的时间流失。随着时间的步伐，慢慢的注意力开始高度集中，心无旁骛。不知道大家有没有这样的感受，一边工作一边一个耳朵听歌，不知不觉等自己完成了自己的任务或者回过神来才意识到，上一首歌是what？想不起来了？这也许就是所谓的进入状态，这也与我说的自学过程不谋而合。正是因为我深知自己看书犯困，所以我选择看视频，并且整个自学过程中也主要是以看视频(秒味、慕课)为主。群里也有很多群主分享的相关电子书籍，如果你感兴趣，尽情去下载，不用排队。如果你能选择适合自己的自学方式，那么恭喜，火锅底料有了。

好记性不如烂笔头，自学不仅要汲取知识，还得转化知识，其中一种方式就是勤做笔记。这一点我做的还算不错，半年来，一本pc端一本移动端合起来也差不多一整本了，反正你们也看不见。不过重点是，准备好一个档次稍微高一点的笔记本，当你面对这么好的一个笔记本，心想总不能在上面胡写乱画吧，那就老老实实的好好写字，写什么呢？照搬书籍原文？或者是照搬视频里面的ppt？肯定不是，不然书和打印机要收你的智商税了。所以肯定是写经过自己的脑回路之后，提炼的精华部分，这样以后再翻看的时候才能体现出笔记的意义。All right，好习惯你这么被你轻易的养成了。而且我在看视频的时候会在另一个草稿本上刻意模糊的记录知识点，看完视频后一个

一个回忆，就像背单词，如果这个记得很清晰，ok next one，如果不行，那就再看一下视频相关的知识点，直到所有本节知识点clear，抄起笔头就是一顿记。这点对于看书的朋友同样适用。菜品有了！

眼过千遍不如手过一遍，是时候该展现真正的技术了，真枪实弹的写代码，这就是所学知识的终极输出手段。这也是一个老生常谈的问题，也确实是一个答案只能是量变产生质变的问题。对于一个复杂抽象知识点或者体系的记忆程度而言，如果你不想很快的忘记，那至少写一遍；想长一点时间忘记，那至少在若干时日后再写一遍；什么？你想要通过某种方式达到倒背如流刻骨铭心的地步？对不起，她只是一个人生过客。曾经我有就这样的困惑，一个功能不是特别复杂的自定义拖拽，几个月前我来回写了两三遍，但是在我近期学OOP的重构这个拖拽的时候，第一次我竟然把把onmousemove和onmouseup这两个事件绑定到了被拖拽元素而不是document身上，其中禁用拖拽时的系统默认事件return false也忘记写了，为此懊恼不已。这也引申出两个问题：我为什么没记住和我需不需要记这么清楚？第一个问题反映出我虽然写过几次，但短期记忆使然，并没有彻底理解他的底层原理，于是我又重新记忆了一遍，一个是让元素离开可视区域还能回来，另一个是拽图片和带文字的元素也能拽的动。第二个问题自己没想明白，向朋友请教，他的答复是：无论是工作还是不刻意刁难你的面试过程中，只要能对答如流并理清清晰的脉络就行，不至于卡壳和无从下手。所以，我的理解是勤写代码对我们最大益处就是更容易的接近底层原理去理解，并且有助于永久性的思路建设，日积月累，善于重构，于是一个业界大牛粉墨登场了，你就是你，不要往别处看，说的就是你！那么，火锅点火走你！

自学那些小事

仓库<https://github.com/codefine/practise> (别因为我是一枚小鲜肉而怜惜我，用star粗暴的来吧)

Reading、Watching、Writing、Noding、Reading、Wa.....Wait！误解太深了，我们不是钱多话少死的早的人肉代码器，也不会陷入这样的死循环，什么时间做什么事情我们比谁都清楚！不过确实，如果停止学习，那将被这个爆炸式发展的行业所淘汰，干这一行就要付出和牺牲更多，才能不说造福世界起码出人头地，这也很容易积劳成疾，所以不论工作还是自学，我们都要学会时间管理。先来个没错就是我的反面教材：由于原来没有意识到这个问题，还有各种压力，加之对这个行业的热情，于是我自学的自我管理就是没有管理。除了吃饭睡觉其余的时间几乎全部用来学习，半年以来平均每天10小时毫不夸张。晚上睡觉的时间由写没写完代码决定，早上起床的时间由还有没有意识决定，期间最长纪录是将近一个月没下楼没出家门，也得益于家人的悉心照料和督促，要不然头上都能长出蘑菇来。也许科比在下一次说只有他能看到凌晨四点的太阳的时候我会义正词严的反驳他。当然这是我年前自制力失控之前的时候，也是一个彻头彻尾的反面教材，千万别学我。而正面教材是：连续写代码一个小时无论写没写完，至少休息5到10分钟，一方面可以整理思路从而更高效的完成余下任务，另一方面你对身体好点身体也会对你余生请多指教。以后的工作和学习生涯中，我将坚决执行这条原则。劳逸结合，拒绝猝死，从你我做起。锅里的菜快熟了，我们去调一下油碗儿吧。

自学期间如果遇到自己解决不了的问题怎么办？这个问题问得好，这也正是我们这个群存在的意义。我们要充分贯彻群内不懂就问精神的同时，跟随群主以及众管理理解先想、再查、后问的三步走思想，切实做到有的放矢的提问方式，拒绝假大空的无理取闹，把问题。。。好了，你不要说了，我选择去吃火锅，广告你都打到自家门口儿了。



## Demo页制作心得分享

关键词：整站思路建议、素材来源

个人Demo页地址：<https://codefine.github.io/>，各位先睹为快。

对于github这个网站而言，我也是懂得pages的皮毛而已，而且pages功能也只是github全站功能的冰山一角，最重要的功能个人认为还是他的共同协作主线分枝系统，并且网站本身也是全球最重量级的开源社区，希望大家多多关注，多多学习，早日贡献自己的开源作品而造福一方。这个网站的存在，对于我们而言，就是一笔无价的财富。

如果了解pages的功能，那我建议先去看看官方说明 <https://pages.github.com/>，看不明白的话百度一下，相关攻略也比较多。如果还是没研究明白，那也没关系，这也是我下一次分享的主题——如何通过github的pages功能轻松搭建个人主页。我将带着大家从github注册、用于展示个人主页的仓库建立、简易个人主页搭建、克隆和同步这四个环节，一步一步由浅至深的完成。而且，图文并茂，过程中只要有新的页面或者界面都有对应的全新截图和说明，个人认为是迄今为止最为细致的攻略了，细致到令人发指！力争全民轻松搭建个人主页！届时希望大家不要错过，精彩等着你。

在这里我先放出自己绘制的简明流程图，整个过程中不要你具备后端知识。配合官方文档说明食用更佳。

关于个人demo展示的这个网站，由于时间仓促，制作过程从头天中午吃完饭一直到次日凌晨4点左右，仅仅用了不到1天的时间，UI、代码优化等方面做的确实不够好，但是目前这个页面对于我而言，也仅仅只是陈列我自学期间的积累，能在面试的时候加分就行了。到后期如果有好的idea并认认真真做个人网站的时候会考虑买域名和云服务器。整站用bootstrap搭建，不得不说这款框架真心适合不懂设计，又不想太难看，还想做到响应式的人群了，而且这种UI框架API文档相对简单易懂，边查边写，几乎不用对原框架进行改写就能做出一个比较优雅的网站了。在此也推荐各位小伙伴，如果没有什么好的idea而且想快速搭建demo页，那这款框架是不二之选。

网站中的demo素材均来源于自学期间的积累，但不是全部不积累，只是很少的一部分可以拿得出手的，剩下的家丑不可外扬。其中最得意的就是那个移动端安踏去打破的demo，年前做到年后，陪我跨了个年，而且所用到的较为新颖的思路比如css3构建的伪圆柱结构，以及原生js陀螺仪相关api实现的手机位置与场景交互，都是一边研究一边写代码。也非常感谢秒味网站的相关导师和群友，在不同时期给了我不同程度的启发和指引。所以为什么我的整个学习过程几乎是加速状态，这全都得益于那些看得见和看不见的良师益友，在此特别感谢期间为我答疑解惑的前辈和群友们。

对未来的展望

转眼就到了快要和大家说再见的时候了。这个时候如果有一首《What's Up - 4 Non Blondes》来作为离场时的bgm，那就再好不过了。回首整篇文章，虽然文风略显戏谑，但均是真情流露，本人也不擅长写一板一眼的教条式文章。所以希望这篇文章给大家的是一种不一样的感受，全当是听了故事会里的一篇小故事，也是极好的。



我想，互联网行业在未来几年内的发展，肯定会持续这种爆炸的态势。那么，既然我们选择了这条道路，无论你是被迫还是主动都已经无所谓了，因为经历过隔行如隔山的我已经深深的意识到，我可能，终于将要走向了一条正确的道路，而已经在这条路上的你们就应该珍视眼前的一切。宝剑锋从磨砺出，梅花香自苦寒来，持之以恒，身体力行，尽自己最大的努力在还算可控的几年内鹤立鸡群，这样，未来的自己和家人一定会为现在你而骄傲自豪。虽然我还处于没找到工作没入门的阶段，但经历了这么多，我的心态已经摆出了一个大大的正，就等着去工作，去拼搏，去证明我辞职转行是一个多么正确的选择了。我愿意相信这是一个可以通过付出百倍努力就能遨游的时代，而互联网行业就是我们的星辰大海！

## 【第684期@同说】let\_Scott：4 年前端狗，2 年 CTO

前言

Hi，又一周了，八月份又快结束了，今天下午厦门有场没有主题的主题分享会。

另外这篇来自@let\_Scott的投稿。昨天7点多在上班的公交车上看到这篇文章，立马被吸引，CTO这是走上人生巅峰的节奏。但从文中流露出的成长，选择，以及对自己未来的思考，是否有值得你思考呢？

正文从这开始～

我，Scott，一家创业公司的 CTO。

从业 6 年却很少写文章，近一年来接触了几十个刚毕业的前端新人，也面试了 100 多个前端工程师和 Nodejs 工程师，对于前端发展的这个职业算是有些感触吧，打算陆续写一些从业经验也好，技术分享也好，对自己前 6 年的经历做一些文字上的沉淀，按照我工科背景不善修辞的尿性，这些文章很可能凌乱也许更会烂尾，还请误入的童鞋谅解。

这是一篇准鸡汤文，谨献给工作 0～3 年的前端工程师，内容都是我的亲身经历，不精彩但接地气。

2010 年毕业于一所普通 211 大学，电气学院自动化专业，了解这个专业的大概知道自动化是弱电方向，就业高不成低不就，不读研仅靠本科的技术积累很难进入好单位，而我又是那种一进大学就迷失自我的“逃课生”，四年逃课挂科无数，不仅荒废了专业，也虚度了青春，彼时年少而又轻狂，骄傲却不知路在何方，唯一的收获是大一便早早的搞定了老婆，恋爱 8 年后领了证，认识到现在已经整整 10 年，所以说码农恋爱要趁早啊。

以上是典型的屌丝工科男背景介绍。。。。。

如下的一切，都源自毕业实习到了杭州钢铁集团，说是实习其实是见习，带着安全帽到各个生产线上看看钢材生产流程做个笔记，仅此而已。初到杭州，得知我有一个学霸级的高中女同学在杭州淘宝的广告部门实习，就在实习间隙登门拜访，学习求经。

入行以前

## 贵人点拨改换行业

见到高中女同学的同时，也结识了她学霸级的男朋友，跟她都同在淘宝，是一名资深的 Java 工程师，他俩知道我对互联网很感兴趣后，便向我口述了淘宝种种开放的职场氛围，有趣的江湖文化，每天发生的每一个故事都深深打动了，激发起了我对于互联网的兴趣和想要进入淘宝的动力，后面两位开始给我出谋划策，鼓励我从事编程方向，比如 Java 编程，可以从自学开始。其实，早在 2009 年，就有朋友建议我从事前端开发方向，并且帮我选了月影的《Javascript 王者归来》这本大厚书，但是因为不够了解加上没有兴趣支撑，一直放角落里风干积灰，直到如今我也没看完。

## 初学 Java 编程

当时对于 HTML，CSS，Javascript，我其实一无所知，于是决定学习 Java，6 月份毕业后在杭州租了个小房间，每天啃杭州图书馆的 Java 书籍，同时在网上看马士兵老师的 Java 视频教程，陆续跟下来一个小聊天室和仿 QQ 通讯的小软件，但也只是照葫芦画瓢，并没有吃透技术点，心急如焚！加上这两个月窘迫到口袋里只剩下几十块钱，开始每天只吃一顿挂面，也是在这个时期留下了胃溃疡等一堆胃病，快撑不下去了，找工作就势在必行了。

## 拖了很久去做的胃镜

## 寻找 Java 岗工作

从 8 月中旬开始逼自己投简历找 Java 工程师的工作，面过的 2 家公司都发了 offer，但薪资都是 1500，不仅是学徒工，更要签 4 年合同且不许毁约，想想有些后怕就跑去找之前提到的学霸女同学求建议，她男朋友建议我可以先考虑试试前端这个岗位，上手快并且我活泼的性格更适合设计相关的工作，回头想想，人生有时候不仅需要运气，更需要贵人点拨，于是当天下午回到住处便开始了为期 2 周的废寝忘食背书模式。

## 放弃 Java 投奔前端

所谓背书模式，其实就是把能搜到的前端知识都手抄到纸上，然后开始背诵，一天到晚脑海中不停的像放电影一样，熟记块状元素和行内元素的区别、绝对定位和相对定位的区别、jQuery 二级下拉菜单的 Javascript 插件代码实现。。。甚至背下了DOCTYPE、几种文档声明的写法，陈年小抄中，还能隐约看到 垂直居中 几个字：

如果说前面的两个月是信心百倍，有的放矢的充电 Java，那么这两周就是完全是饥不择食，慌不择路的学习前端，在快要被房东赶出去的时候，再次逼自己开始又一轮的找工作。

## 面试上岗

\*\*

\*\*

## 面试前端岗位

精心准备了什么都精通的简历，也是第一份前端简历，投给了口碑网（淘宝当时内部新成立的项

目)来练手,结果面试官电话里第一个问题就把我问懵逼了:你知道 IE6 的双边距怎么解决么?

我嘞个大擦!这是什么鬼?!我听都没听过!!!

此时的我,根本不了解原来浏览器竟然有兼容性问题!也根本不知道前端开发是需要对 IE6 的样式和行为写 Hack 脚本进行兼容的(包括 IE7 和 IE8)!

再往后面又问到不同文档类型下的盒模型...,我就像是一个傻子一样,在电话这头大气都不敢出一个,真心想找地缝儿钻进去,气氛尴尬的要死。

面试官看出我啥也不会,便心平气和的给我普及了一下前端开发需要关注的知识点,不出意外的礼貌拒绝了我。

掉头重新学习前端

那次面试之后,我的信心受到了巨大打击,也发现了自己和一个哪怕是及格的前端开发工程师,相差的距离是如此的遥远,梦想看着就在眼前,而你臂展不够,各种不好的心理暗示开始浮上心头:

淘宝肯定进不去了.....

杭州我白来一趟了.....

工作肯定找不到了.....

爸妈没法骗下去了.....

此时已是 9 月出头,我亲爱的同学们都已经转正上岗,而我还在原地踟蹰。我只能安慰自己:你是 0 编程基础的新人(大学计算机都是挂的,对于挂计算机课还是略后悔的),不要拿自己和别人比较,这个阶段是正常的,挺过去就好了!

于是我努力回忆面试官的问题,重点突击 IE 兼容性问题,办法依然是背,感谢上苍赋予了我强大的临时抱佛脚能力,从小学到大学,都是以临考突击的方式应对,短期记忆量可以很大,但说忘就忘的一干二净,幸好有面试官的耐心提点,我才有的放矢,后来面试官我们私底下成为了很好的朋友。这一次背书只经历了 3 天 3 夜,对的,不眠不休的记忆,我便又开始新一轮的求职经历。

第三次求职面试

这一次花了一下午做简历,只投给了两家小公司,其中一家 HR 效率极高,几个小时后就电话联系面试。去了后做了一套试卷,我就和公司老总在办公室聊起来了,没想到越聊越嗨,为什么呢,因为我俩聊的是足球,我大学一直是院队主力左后卫,踢了四年校联赛,于是跟这个老总在足球这个话题上开始投机。

有时人生真是捉摸不定的曲折离奇啊,第二天中午我便拿到了这家公司的 offer,但 offer 的内容是:

把你外派到淘宝当外包,你愿意不愿意?工资会比较低

当下才知道这家公司是外包公司，思前想后，决定还是接受这个 offer，即便薪资低的可怜，勉强够交房租。

不过这只是第一关，还需要到淘宝去参加第二轮面试，被淘宝面试选中才能最终获得外派资格，于是屁颠屁颠跑到华星科技面试，开始依然是做了一套笔试题，面试官看完我答的卷子后是这样跟我讲的：果然是刚毕业的学生，卷子填的满满的，但是题目全部做错了，又问了我 apply 和 call 的区别，我自然是不会的，甚至连上下文都搞不清楚是啥意思，只知道它俩传参的区别，于是面试官问我其他方面的，比如为啥从事这个职业之类的，我拼尽全力声情并茂讲述我的求职渴望，这也是我唯一可以讲的东西了。

走出华星科技大厦，心情是灰色的，感觉就要和淘宝真正的 say goodbye 了，背的知识点遇到资深的面试官随便一挖，我便原形毕露伪装失败，越想越伤心，此时竟然雷声大作瓢泼大雨没头没脑哗啦啦的下起来了，我把档案塑料袋顶在头上，默默的往家走，躲雨的心情都没有，脑海中一片空白，雨越下越大，迷住了前面的道路，迷住了我的视线...

雨过天晴

人生是如此的曲折离奇，第二天便接到通知，淘宝竟然要我了！于是跑到外包公司签了就业合同便去淘宝报道了，此时是 2010 年 9 月 6 号，到了淘宝才意外发现，我竟然跟之前的学霸女同学在同一个大部门 - 淘宝广告事业部，人生处处是惊喜，一言不合就相遇。

自此我开始了 4 年的前端工程师之旅。

自知才疏学浅，我进入部门后，也是尽量的谦虚做人，不懂就问，跟同事好好相处，同事帮我解决了一个问题，我就基于这个问题，晚上到家后，根据他的解决思路，把所有相关的知识点，全部在百度搜一遍（那时候还不会科学上网），如此坚持了一年，这第一年技术也成长迅速，在外包公司的工程师队伍里已经是走在最前列了，那么外包公司也是给出较大幅度的涨薪，但是跟阿里正式的前端工程师待遇依然不可同日而语，毕竟是外包，要赚你的人头费，不过之后没多久淘宝觉得我已经完全达到正式工程师的水准了，便赔了外包公司一笔钱（这个我没考证，是听外包同事说的），于是我就欢天喜地的转正成为淘宝正式员工，但和外包公司老总依然成为了很好的朋友，同时也组建了这家公司成立以来的第一支球队，作为队长我带领外包公司的球队在杭州城参加各种比赛，虽然胜负参半也没拿到太好的成绩，但是跟这一波外包公司的队员一起风吹日晒，建立了珍贵的革命友情，后来随着转正淘宝正式员工并且加入了淘宝的球队之后，跟老球队一起踢球的次数越来越少了...

## 职业沉淀

### 4 年阿里前端生涯

铺垫了那么多转行求职历程，其实只是为了说明，我是非常普通的一个人，跟所有的求职者经历不同但起点类似，有提心吊胆，有惊喜连连，有彷徨失落，有坚决果敢...，但选择了前端开发，无论它是不是不归路，都是我自认为的一条很适合自己的路，认识到这一点，就要铁了心走下去。

在阿里的四年，最大的收获是见识到了什么是大牛。在公众视野中不少活跃的各种圈内牛人红人（包括我自己，有些新人认为我很牛），接触下来其实根本比不上大公司里那些低调做事的大

牛，他们做人做事，对自己的定位、要求和产出，总是能让我意识到自己的种种不足——不够谦虚不够努力。如果说这是榜样的力量，ok，那我承认与优秀的人共事总是十分难忘的经历，而我自己作为一个普通的工程师，自认够努力，但还远不够优秀，也许是天分不足，也许是尘世扰心，我的职业瓶颈还是早早出现，而要迈过去需要更长时间。

在工作 2 年左右，我发现自己在技术深度上很难有专业造诣的时候，我开始把眼光放的更长远，我不再把自己仅仅定位成一个前端工程师，而是会开始关心互联网的发展趋势与整体格局，关心产品从 0 到 1 诞生过程中的盈利模式，关心在产品迭代中工程师与运营多方参与的角度和结合的效率，关心工作流程和团队文化，这一切其实都是我潜意识不自觉的关注，是一种不安分，过后很久才逐渐意识到自己早就在默默铺路，有时候一个选择早就做了，只是你还未意识到。

那么对于技术这个领域，我也开始去关注更多非前端的生态，从 Linux 生产环境配置部署到域名解析，到软文策略和 SEO 推广，从移动端、PC 端的前端分离和架构，去折腾 PHP 的 Wordpress，Codeigniter...，折腾 Ruby on Rails，直到接触到 Nodejs，尝试借助 Nodejs 拉上 Mongodb 做数据接口，和模板输出的工作，从数据库的安装到主从灾备，不求精通只求通，下班的无数的晚上和周末，在屏幕前消耗青春，通过乱七八糟的折腾这一系列有的没的，我逐渐意识到自己可以创造的价值不再是仅仅前端页面的输出，这个价值不单单是指可以一己之力搭建一个完整的网站，而是有了前后的动手经验，能从更高的一个层次上去把握一个项目以什么样的形式去设计，以什么样的思路去迭代会更加的省时省力，这个省时省力其实能更好的促成产品，拿到商业目标，而有了这个意识，就能更好的配合产品经理实现他脑海中的想法，如果你认为是不切实际海阔天空的想法，没关系，来小成本验证一下，验证之后说不定就真的是海阔天空，也是这种想法促使我业余时间结交了许许多多的朋友，以兼职的方式参与了至少 3 个小创业项目，自己的技术面和技术深度也得到了很大的拓展。

## 发掘自己的更大价值

当聚焦的点不再单单是前端开发的时候，我发现了更多的乐趣和更多的可能性，从前，我会花好几天反复测试一个广告投放模板，对于里面的一个模块加载器或者特效组件如何去设计才能最小化体积，减少展示延迟，最大化优化动画流畅度，针对移动端的广告投放用哪些效果兼容性最好最省电这些细碎的点占据着我工作的大部分时间，必须承认，解决这些技术点是很有成就感的，但是技术的革新速度实在太快了，昨天还在纠结的几 K 大小带来的网络延迟、代码执行效率带来的性能消耗，今天突然在百兆宽带甚至是 4G 大水管高规格硬件时代变得不那么敏感了，昨天还是小心翼翼使用 radius 圆角图片两套并存的兼容方案，今天 CSS3 属性放开随便用，这些当然是我们需要不断更新的知识栈，但仔细想来，让我花费数倍的时间去研究底层的优化细节，跟实现我的个人价值放到一起的时候，是不是最佳的途径呢？很多类似的技术问题，我可能需要花费 2 天，而天分好的同事，只需要花费 1 天甚至更少，从这一点上，我越来越不认可自己的技术方向和优势，于是我摘掉带有职业界限的眼镜去看待彼此的分工，带着合作共赢的心态去接触更多的领域，包括前面我提到的小创业团队，他们都没有能够生存下来，有的是因为技术原因，而有的是因为资源问题，参与这些项目也给了我更多的感触，让我逐渐明白想要做好一件事情，是要先定位实现的程度也就是预期目标，这个不同时期的目标完全取决于不同时期的资源配置，平衡追求这个性价比才是做事的关键，说白了，就是知道什么可为不可为，外加一些方法论。

## 创业萌芽

## 职业波动期

在阿里工作的第 3 年节点上，我萌生了创业的想法，但并没有机会也没有胆量迈出这一步，并且就在这个第三年中，我在阿里迎来了职业生涯中最心旷神怡的几个月，被分配到了技术非常优秀的主管，也开始参与更有挑战性的项目，个人技术成长越来越快，无论是前端开发能力，还是后端开发能力，然而好光景持续了仅仅几个月，便被分派去支持双 11、双 12 推广会场。在阿里这种变化是常态，业务支持永远是第一位的，因为客户永远是第一位的，所以抽调出来支持一个活动是很正常的，我起初也是很平静的参与这个项目，然而这个活动的历程却充满了心酸，耗尽了我的耐心。

成长受阻，陷于无序的需求算是一个促使我做出离职创业的导火索吧，另外一个重要原因则是遵从内心的创业冲动，后文会写到。

关于这个项目呢，原本是提前了 2 个月筹备这个活动，从后端数据到前端和运营，全部到位了，虽然前端在里面扮演的角色很关键，但技术难度却并不高，工作量也不大，就是把页面过来时候带的参数，按照规则梳理一下，发一些请求到后端，拿到加密值然后重新替换到页面里的所有 a 标签 href 中，也就是刷页面参数，然后把这套代码抽象成组件，集成到一个大系统里面，根据不同的会场类型加载不同的刷参逻辑。

原本是 3 周做完的项目，用了整整 7 周的时间才最终落定下来，主要原因是产品经理更换了 3 个，需求方大 Boss 介入了好几次，项目经理风险意识不够导致工作不断重做，解释成本也因此居高不下。每一次联调出问题，都先会把责任全部抛过来丢给前端，也就是我，然后每一次我都要反过来充当半个项目经理的角色，去 push 各个参与的团队来配合我核查问题的来源，而每一次查出来后要么是后端数据环境更改异常，要么是数据引擎切换引发不兼容，要么是产品设计规则不全出现参数漏洞，总之就是项目负责人的不专业，加上团队协作的成本高，导致参与的人都做的不开心，其实在这之前也参与过类似这样的扯皮的延期项目，然而这一次对于我却尤其难熬。

## 挣扎在白天黑夜边缘

在双 11 项目远没开始的时候，我接触到了一个创业项目，兼职赚外快补贴家用，平时的晚上和周末断断续续在做，大概到了 10 月份，这个项目准备去找融资，因此希望我在 10 ~ 12 月份能尽快把项目原型开发上线，我评估了一下剩余的工作量，差不多需要 15 人日，同时算了一下时间，10 月份至少有 3 周的空闲周末，还不算晚上，再加上 11 月份甚至 12 月份的晚上和周末大概能有 20 多天的时间，时间简直不能再充裕了，于是便也一口答应了下来。

然而这个从双 11 前就启动，一直持续到双 12 的公司项目，却频频上线出问题，而每一次出问题，都需要前端工程师介入联调，于是几乎所有的周末全部都被公司征用，不止如此，白天只能调试，不能发布，需要熬夜发布，出问题后还要配合回滚，回滚后，还是不能走，要等测试工程师测试完，再重复以上过程，于是就有了连续的周末通宵，这个答应在 12 月份上线的创业项目，便搁浅了，毕竟公司事大，孰轻孰重我还是分的清楚。

总算双 12 上线一周后，我下班回到久违的家中，眼看要延误掉这个创业项目的融资窗口，我想既然承诺了，无论如何都要拿下！于是，白天去公司工作，幸好是双 12 之后，这个项目不再变动了，每天只需要配合调整策略参数就行，不需要投入大精力去开发。晚上回家以后，7 点开始写代



码赶这个创业项目，敲代码一口气通宵到第二天早上 6 点，洗漱一下出门到公司吃个早餐，然后在公司躺椅上睡到 9 点，起来洗把脸，冲上浓咖啡，配合这个双 12 的项目继续做沟通啊，邮件通报啊，参数更正啊这些琐碎的事情，中午吃过饭，又在躺椅上睡一个小时，下午跟进一些常规的部门工作，晚上回家后继续写代码，如此反复，经历了大概 12 天，期间至少有 2 次半夜敲着敲着代码就一头栽在桌子上睡着了 zzz。要预发布创业项目的那个晚上，需要跑通发布环境，我买了一箱的红牛，一夜喝了 5 罐，凌晨 4 点，突然一阵恍惚，有一种灵魂出窍的感觉，手脚冰麻，说不出一句话，视力急速下降，那一刻，我想，我怕是要猝死了。

在这种症状持续的 2 分钟时间里，我心里怕到了极点，想了许许多多，如果我猝死了，父母怎么办？外婆怎么办？我的事业怎么办？这个项目怎么办？我闭上双眼，让自己使劲喘气，慢慢的回过神来，然后起来走动了一下，狠了狠心，继续折腾发布环境，差不多 7 点多，环境弄好后，就洗个脸到了公司。这一次到公司没敢睡觉，担心躺下了会彻底体力崩溃，就咬牙坚持了 1 天，晚上回家又熬夜大概到 2 点，成功发布上线，整个心才真的像大石头一样放了下来，顿时困倦无比，迷迷糊糊正打算去睡，突然又发现一个环境异常问题，赶紧屏住呼吸紧急调试，这一次调试的每一秒钟我都记忆深刻，因为我需要把已经休眠的身体和意识拉回来，大概 5 分钟左右重新修复上线，我给公司主管发了短信上午在家请假休息，就混沌沌的睡去了。

大概 2 周后，这个上线的创业项目拿到了 200 多万的天使投资，这个就是促使我做出出来创业的重要原因吧，这也是我目前正在创业的项目。

## 离职创业

在大公司当一颗螺丝钉，其实我并没有什么不爽，但当发现一些不好的兆头或者流程大问题的時候，由于我人微言轻，再好的技术，或者说再好的工作态度都无法改善这个情况，我做了反馈但所有人都摇头，我做了推动但所有人都表示无能为力，这种无力感越来越强，我发现除了被我抓在手上的这些前端技能和可怜的工资，我什么也没有，什么也做不到，我开始怀疑人生，是坚持死磕到公司上市死磕到自己在公司和团队有更好的能力和话语权，还是选择离开。

这时候已经是 2014 年春天，我重新开始考虑去向，这时候年前参与的创业项目投资到位了，我也基本了解了它的业务模式，可能是年前拼命赶项目那次太投入，导致我对它也有很深的感情，毕竟是自己一手写出来的代码，是自己一手养大的孩子，这时候对方希望我以技术合伙人的身份加入，我最终想了想便答应了，其实还是有点冲动的，因为恰恰是那个时候是我在公司做项目最不爽的时候，在公司的无力感和在创业项目上的认同感成就感形成了鲜明的对比。

Anyway，我是希望通过我的力量，证明一些或者是实现一些我的想法，我希望可以有一些事情是可以在我影响和控制范围之内的，于是到了 6 月份，也就是阿里巴巴上市前夕，我递交了辞职书，这次走的虽有留恋但是毫无遗憾。

## 创业狗的日常

创业到今天，已经两年多，走的并不轻松，而创业两个字跟 CTO 一样，都是听起来真的很高大上，但是做的事情可能是灰突突的并无新意，唯一不同的时候，这是一条踩在我自己脚下的路，我走快一点就看到目标近一些，我走慢一些，就看到远方模糊一些。

创业有太多的不确定性，最大的不确定性就是能否兼顾好自己的身体和家庭，因为无论是时间投入还是资金回报，都是看公司状况和市场反应，所以像 i5ting 这样的 CTO 也要出书立作，也要有在线课堂，在拓宽人脉影响力和技术布道修炼的同时，也可以赚点小钱贴补家用，没错，就是贴补家用，不要以为创业公司合伙人都是高富帅，白富美，其实我认识的不少也仅仅是够花，甚至是穷屌丝，我也不例外，所以和慕课网合作讲授自己的前端实战经验，以下是我 2015 年底业余时间鼓捣的一个课程——[《7天搞定 Nodejs 微信公众号开发》](#)，刚入行的新人可以参考我这个老人开发一个小项目的切入方向，开发功能所涉猎的角度，以及推进的方式是怎样的，如果你资金尚且充足，对知识又充满渴望，不妨看看这个视频课程，支持下我这个所谓的创业狗 - 草根 CTO，这是个硬广，对，说到硬广，我这一年来招进来的员工，还真的的确是看过我慕课网 Nodejs 课程入门这个行业的，所以我还是很欣慰的，其他的一些 Nodejs 课程是 2 年前创业前后边做项目边录制的，其实比较过时了，居然还可以帮助到入行的新人，我已经很满足了。

## 写在后面的话 - 所谓 CTO 的称谓

对外面我总是宣称自己是这家创业公司的 CTO，而对朋友和同事，我总是希望大家把我当成一个普通的程序员看待。因为对外，可以树立公司的形象和减小解释成本，这个 CTO 的 title 有明显的代入感，而对身边朋友，包括对我自己来讲其实它意义不大，我丝毫不认为有一顶 CTO 的帽子在头上，技术就能牛逼几分，身价就能增加几分，相反，有这顶帽子在头上，带给我更多的是一种责任，而恰恰，我慢慢成为了一个有责任感爆棚的人。在技术方面，我始终不认为自己可以配上 CTO 这个称谓，只不过对当下这个团队这样的业务规模，我尚可以完全 handle 住，但至于未来会怎样，依然还是个未知数。

## 【同说】小天童鞋：“不断学习，不断成长”

### 前言

这个同说计划，早读君一直没忘，在邮箱中还收藏着几个童鞋分享他们的经历。今天分享的是@小天童鞋，有一段时间看他在朋友圈写了很多感性的文章，那个时候非常有兴趣邀请他来写一写他的经历。

正文从这开始～

### 写在前面的话

今天睁开眼睛，习惯性的打开微信朋友圈查看早读课的推送，发现今天的头条既不是大大的招聘贴也不是干货的分享，看到【同说】以为是XXX，怪我想多了，打开一看是一个活动的召集贴啊，看介绍看的似乎一大堆小伙伴朝我走来一样，随机立马留言“支持”表示支持。

细细看完之后发现这个活动真心的不错，他就像婚恋活动中的“非诚勿扰”一样，让我们在芸芸众生中能有机会找到身边的那个志同道合的小伙伴，听起来就很nice的一个想法。

然后就收到了“情封”（早读君）大大的微信，说让我试着也写写参与这个活动，分享一下自己。真的很谢谢早读君的邀请，给了我莫大的勇气，因为我本来持观望态度的，打算是先看看别人的分

享再决定是否动手，因为自己道行不够，可能也没有那么多的干货分享。但是，当我看到征稿贴下面的那句“勇敢秀出你自己，对孤单say no!”时，加上早读君的诚邀，于是我决定本周就赴约写一篇关于自己的前端之旅，就当抛砖引玉吧！接下来是我的旅程，我在南京，期待你能看到！

## 自我介绍

首先来个自我介绍，网上一直称自己为“小天同学”，男，90后的尾巴，喜欢写作摄影，对于设计很感兴趣，但是本职是“前端家族”中的一员。有着文艺的性格，却没有文艺的胆子。

我目前就职在美丽的人文古都·南京。这个城市一样有着像北上广那样的梦想，这个城市一样有着生机和活力，这个城市一样有着朝气蓬勃的年轻人。每天我们可以坐着不像北京那样拥挤的地铁上班，每周我们可以约3两好友去公园景区郊游，每月我们可以找到城市那头的朋友小聚喝酒。我们可以一起去拍照，去长江大桥看黄昏的江景，去总统府感受民国风情，去中山陵瞻仰国父的天下为公，去夫子庙探寻金陵十三钗的遗迹，去明城墙感受古时巍峨的城防.....这里还有很多美好，这里就是南京期待和你一起发现更多的美好。

我的工作前面也说了是前端工程师，入行已经快3年了，目前一边写代码，一边在公司负责了一个不到10人的小团队做着前端的事情，有时候也会思考怎么能让大家更好的工作，怎么能让大家都保持一颗学习进步的心，怎么能让我们一起都在成长。

## 前端之旅的开始

关于入行，我不知道该从什么时候开始算起，因为我在知道大学被录取的时候（2009年），在入学前就开始在家里自学html和css了，虽然说只是简单的基础的基础。那时候觉得用简单的几行代码几个标签在浏览器里就能出来一个很酷的效果或者一个简单的网站，当时觉得很神奇，真的很让我着迷。在学习了简单的基础之后，我就到网上找各种免费的建站和免费的博客折腾起来，这也许算是启蒙的入门吧，从这里我发现我的兴趣。

然后大学学的是计算机的本科，你们懂的前三年都是枯燥的理论课，该学的不该学的都一起来了，什么：信息技术导论、程序设计基础、数据结构、离散数学、面向对象程序设计(Java)、计算机组成原理、网络基本原理、数字逻辑、数据库原理与应用、操作系统、网络安全与管理、TCP/IP协议与网络编程、操作系统、计算机网络、面向对象程序设计(C++)、软件工程、软件测试技术等等。

这和我预想的不一样，我本来以为就是学习如何开发制作网站，可能是那时太小还不懂，这些都是基础吧。但是随着理论课程学习的结束，实践课程的增多，我们开始实际上手一些简单的业务系统的开发，最基本的像“新闻发布系统”、博客系统等等。我在完成功能的基础上总会对前端花更多的时间去写，把大家都是千篇一律的东西做的不一样，做美美的优化，每次也都能在视觉上更受老师的亲赖，因此就更有成就感和更加有兴趣去做去学。这也是鼓励着我去自学的动力吧。

毕业前夕，外出找实习，因为那时对于毕业后自己该从事什么样的工作不是那么清晰，又因为对于前端设计和开发又更有兴趣，于是在实习的时候就在朋友的推荐下找了一份设计的工作，没错就是通俗的美工。每天的工作就是简单的页面设计，每天对着PS，AI等等。但是三个月下来总感觉做的有点吃力，可能这才意识到喜欢的未必适合当作工作那么简单，当时又感觉不是专业对

口，有种辜负自己4年的学习成果的感觉，于是实习结束后就从新找了专业对口的工作，就是现在的这个工作单位。

刚进我们公司我是以“web开发”的身份进来的，因为我大学学的是后端的开发，前一个月还是做着后端的工作，每天都是Java、数据库、增删查改、SQL等等，虽然做的也不错，但是发现同事做的前端的开发我更加的感兴趣，这也是我第一次接触到真正的前端开发工作的内容，于是就申请转到前端开发上来。然后就开始了我的前端之旅。

## 如何成长

在刚开始做前端的时候，说真的虽然有点基础，但是可谓也是小白一枚，基本的切图都不能算得上会啊。因为我之前的那点基础，与面临的工作相比真的是九牛一毛啊。面对工作有了挑战，就开始着手努力学习吧。

刚开始，为了完成工作，就不停的请教同事，不厌其烦的，从基本的页面布局和各种熟悉又陌生的html标签和css属性开始，每一次都是看同事做好的页面去研究，那块是怎么写的用了什么标签，这块怎么写的用了什么css属性，为什么这块这样写更好，为什么行内元素里不要套块级元素.....从这些基本的开始，从简单的页面开始，慢慢的上手，慢慢的在实战中练习，遇到问题首先就百度按照搜到的方法去试，管用的话就先记在自己的小本子上，搜不到的就去找同事请教，学到后再记在小本子上。因为那时公司还没有完全意义上的前端团队所以也就没有导师会带，只能自己主动的去请教。遇到浏览器兼容问题，这个确实是需要经验来处理的，同样知道解决方案后就把经验记下来，就这样日复一日的积累着，慢慢的把那些不会的变成自己的，再内化到自己的知识体系中。那段时间工作比较忙，再加上自己新手知道自己水平不行，就努力学习抓住一切实战的机会去练习去感受，几乎那段时间每天都会加班到很晚，但是那段时间真的是进步最快的一段时间。

让我从一个前端小白慢慢的开始有了感觉，对于一般的切图和交互都能够不那么费劲的实现了。但是我又发现，这只是能够应付工作了，但是有些知识还是不扎实，不成体系。对于Javascript也只是能看懂别人的代码去修改，而让自己去从0开始写的话，还是很费劲的，于是开始计划着，把相关的HTML、CSS和Javascript的基础知识再成体系的去看一遍。于是在经典的W3school上把对应的课程看了一遍。然后我又买了一本很厚的犀牛书，然后又开始买了“锋利的jQuery”，开始啃书。期间不断的在网上看各种大神的经验分享，关注前端知名的博客阅读他们的文章，其中张鑫旭大大的看的最多了，因为很多基础性的和进阶的都结合例子讲的很通俗易懂。随着自己的知识的增加，慢慢的也开始思考代码的质量和可维护性，开始关注命名规范、编码规范、团队协作规范等等的事情，现在有个习惯就是看到一个网站的效果自然而然的就F12查看源码，估计就是那时学习时留下的习惯，那时候看到别人的效果是怎么写的就会去看别人的html怎么写的，CSS怎么写的，class是怎么命名的等等。

之前同事们用的编辑器是DW，那个很重的编辑器，因为当时边做可以边看到效果用起来感觉还不错，但是DW太重了，而且代码补全等功能不完善，所以慢慢的放弃了他投入到高效的编辑器，现在用的最多的还是Sublime，轻·强·快用着顺手，效率自然也就高了。我就倡导大家改用这个编辑器，毕竟好东西要分享。

就这样在2014年的6月份，随着项目的工作量增大，前端同学的增多，公司考虑成立一个小团队，来形成一个规范的战斗集体，更好的工作。可能是我的认真好学，可能是我进步的神速，领导居然让我负责组建和建设这个团队，这不仅是对我工作的肯定更是对我的一种信任。于是我就带着更多的责任带起了这个小团队，开始规划项目中的基本规范、基本类库、基本的协作方式、基本的学习培训计划等等。

随着学习的深入，我慢慢知道了前端自动化，AMD、CMD等的模块化加载器，sass、less等css预处理器，bootstrap前端框架等等的一些更加新的更加能够提高我们工作效率的工具和框架。随着移动端的到来，也慢慢的开始涉足移动端的开发，接触到了HTML5结合CSS3的开发，接触到了响应式开发等等。开始研究学习，不单单是为了自己提高，也是为了团队未来的发展方向。因为我们是在公司的产品基础上做前端开发的，所以不免有所限制，这也是我在探索如何能结合我们目前环境使用这些新的工具和框架。

现在随着这几年的前端行业的发展，新的框架和工具层出不穷，特别是node出现之后，基于node的工具和框架真的如雨后春笋般涌现出来。虽然我们来不及探索和学习所有的框架和技术，但是这些新的出现正是激励着我和我的团队往前走，不断的进步，不断的追随。

目前，我对于技术的研究也就是看看那些新的东西，不能说深入只是略懂有待实践，探索一下对于团队哪些是可以用的哪些是我们目前还不能在工作中适用的。因为现在带领一个小小的团队，平时对于团队的建设，对于学习培训，对于管理等方面的知识学习的多了一点，思考的多了一点。无论怎么说我自己也才入行3年多左右，对于技术仍然不能松懈，无论是为了自己还是为了团队，让我们都保持一颗热爱学习的心态，共同进步成长。

因为只有成长才能挣得在这个社会更好生活的资本。

## 期待·你

期待你是开朗活泼的，你是技术大牛或者有一颗努力学习进步的心，这样我们可以相互切磋然后共同进步。

期待，与大家的面基，带着你的抱怨、你的心得、你的感悟、你的经验！我在南京，希望你能看到。

## 书籍推荐

其实看书不是很多，主要是在网上学习，所以推荐的书也就经典的入门款吧：

- 《CSS权威指南》
- 《Javascript高级程序设计》
- 《Javascript权威指南》
- 《锋利的jquery》

## 【同说】白树：做一点有回报的事情

## 前言

还记得之前的那个【[同说](#)】我的前端路上期待有你！吗？早读君有做过专访前端界的前辈们，分享他们的经验，那么在我们身边呢其实也存在非常多低调的前端大神，只是有时候我们不知道而已。那么同说计划就是为你设立的。

曾经也有人问我，我不是计算机专业的能做前端吗？能做好吗？

曾经也有人留言，所服务的公司就一个前端，加了不少前端群，但是没什么用，技术也没成长。

那么同说就是为你而来的～

正文从这开始～

首先感谢情封对前端圈的辛苦付出，能坚持做好一件事真的很不容易，敬佩这份精神。2天前看到早读课的【同说】活动计划，我觉得想法真的很好，忽然情封微信邀请我参与，听到后我是心虚的，作为一枚技术宅男，步入前端也不过4年，谈不上有很好的工作经验和学习心得分享给大家。

## 简单介绍一下自己

大家好，我的网名是白树，英文名是peunzhang。2011年毕业后加入腾讯财付通设计中心，2014年转岗到腾讯微信支付设计中心，主要负责H5的前端开发。没错，鬼哥就是我的组长。

## 关于写博客

每个人的时间不多，写博客除了因为兴趣外，更是希望做一点对自己有回报的事情，一来可以记录工作方法和生活感悟，方便查阅和回忆，利于思考和发现自己的不足；二来希望自己给前端行业做点小贡献，促进良性发展，对每个前端从业者都有好处。博客地址：<http://peunzhang.cnblogs.com/>

## 怎么入前端这个行业的

2007年读大学，专业是数字媒体技术，学设计也学技术，期间认真做过UI设计，作品被嫌弃后很受打击，深刻意识到有些能力不是努力就能获取，所谓的『天赋』很重要，现在回头看以前的作品确实丑爆了。

一次偶然的会接触大四的师兄用Dreamweaver做网页设计，想要是自己可以使用它来做个人博客就觉得很厉害。于是在图书馆借了一本Dreamweaver入门教程的书籍自己玩着看，那时候还有配套的光碟，看视频照着案例做，居然可以完成一个网站，很简单的入门引起了我的『兴趣』，于是买书学习，记得最早在亚马逊买朱印宏写的《精通css网页布局》和《JavaScript征途》，那时候很火，然后常跑图书馆闭关学习『基础』知识。在2009年底跟几位同学创建了广州启凡设计团队，边做项目边学习，遇到不懂的就利用百度谷歌，技术进步得特别快。那段时期网站建设的市场需求量还满大的，我更加认定这个方向，后面也顺利走入这个行业。

## 如何学习前端

记得群里有人问我现在开始学习前端还来得及吗，种一棵树最好的时间是十年前，其次是现在，



想做什么就马上去做，并坚持下去。

对于刚步入前端的同学来说，最重要的应该是学习『基础』知识，像CSS、JavaScript的基础原理看多几遍也不会过时，最好找两本书系统的学习或者上网找教程如w3school在线教程，然后如我前面提到的找项目或者做demo去实践，将知识转化为经验，并坚持下来，这种学习方式最简单，进步也最明显。

有的同学说工作忙没有太多时间学习，其实可以挤出来，例如我通常会利用每天上班前和下班后的时间，大概有一个钟在公交或地铁上，看书或者拿着手机学习，特别是早上上班那段路上，学习效率会很高。

我们知道现在前端的水很深，为了解决各种业务问题，提高生产效率，技术创新特别快，那么具备快速的学习能力是你的核心竞争力之一，并不意味着你每样新技术都要学，应该根据公司的业务需求选择适合的框架，其它的了解下，用到时再学习也不迟，更何况使用新框架的学习门槛不会太高，容易上手，这点上看，刚步入前端的新人是特别有优势的，至于想要掌握它的深沉原理，需要花费很多的功夫去学习，这个阶段你可能达到资深工程师的高度。

每个人的学习方法可能对自己进步很大但不适合别人，人都是独一无二的，要结合自己的生活习惯，通过实践中思考，找到属于自己的方法。

### 在学习的过程中，遇到问题是怎么解决的？

在组内有个毕业生妹子，有一次问我3D旋转动画的问题，在说完简单原理后我想把发个例子给她参考，她拒绝了，说要自己思考怎么做，我笑着给她点赞。

学习遇到问题懂得『独立思考』去解决是一项最基础的能力，这种能力完全可以培养并形成习惯，不管是在哪个行业工作，对个人的提升大有帮助。很可惜，我在博客或者群里见过很多刚步入前端的同学，遇到问题就马上提问，甚至是要求提供现成的demo。

思考后解决不了问题可以百度或谷歌，例如stackoverflow，需具备一点英语能力。实在解决不了再上群等方式『提问』，可以参考张鑫旭写的《如何提问才能进阶成为前端大神》

### 现在在研究的技术是哪个方向？

余果写的《全栈工程师的自我修养》这本书让我有挺大的感触，也正在往一专多长的线路走，如产品、视觉、交互、后台等都可以在平时的合作中去关注和思考，然后像鬼哥说的保持『空杯』心态学习，定些计划慢慢来吧。近期技术上主要研究nodejs这块，希望后面做一点对团队有意义事情。

## 【同说】@小武：跨界的成长

### 前言

遇见他，是在某个周六的下午，在公司看完产品经理直播后，无意间看到@小武的博客中自我介绍跟个人成长经历（岁月轮回），又联想有一段时间有几个已经工作几年后的童鞋跨行来做前

端，早读君想通过实例来分享给大家，你们的选择，你们的坚持应该是不会错的。

正文从这开始～

大家好，我叫谢耀武，网名叫前端小武，目前就职于百度网页搜索部，主要负责结果页的模板开发和优化，平时喜欢折腾(90%)、玩游戏(10%)，属于技术宅。

## 转型做前端

大家可能会说上面介绍为什么没有学历和专业，是的，我并没有这些，在2004年刚上初三时学习很烂的我就辍学了，2005年去天津工地做装修，主要做刷油漆、刮腻子、刷涂料，常常工作于外墙高空作业（6-35层），我去过了天津很多区，几乎每年都要转天津近十几个小区。

工作几年后的我发现这个行业可能没有什么出路，也没有什么盼头，但由于小时候就酷爱电脑，2009年一个偶然的机会认识了有“网页制作”这门技术，于是在三线城市一个电脑培训班学了一年多点～

由于平时比较爱折腾，学校的冰点还原、路由MAC限制上网都被我一一搞定，还记得写了几个破解上网专用的bat批处理，索性后来就被学校主管当成了学校的网管，包括平时电脑的维修、内网网站维护、上网白名单维护等日常工作，当时也算小有名气～

2010年中毕业后找工作是相当的坎坷啊，因为学的全是在DW 8.0设计视图中使用表格...经过努力终于拿到一个面试机会了，由于是我人生一个转折点，我很激动，第一次踏入公司型的工作，第一次见那么多“公司型的人”。我异常的激动，我就给我媳妇打电话，她一直在给我打气，于是我进入面试公司的大门了。他们只问了我几个问题就让我回去等消息了。

- 你DIV+CSS精通吗？
- 你手写代码吗？
- 你兼容性熟练吗？

我的回答是：

- 不会
- 不会
- 没了解

“诚实”是我至今还存在的缺点，其实我完全可以这样回答：

- 不太熟悉，我会修电脑
- 有时会，我会修电脑
- 了解的少，我会修电脑

然而入职的消息迟迟不给我，由于又迫于现实经济的压迫只能还原老本行：装修。

但我的内心并没有放弃，2011年在天津装修时由于有点网页制作基础，晚上我就用手机上网看那些个代码说明、特效之类（当然非常感谢当时的Nokia 6730塞班S60V3手机），并在脑海记下来自己想实现的效果，每隔一段时间就去网吧把这些写下来，记得当时写的最多的就是jQuery特效，也因此有些网吧网管会一直盯着我，那时候我知道了：这就是代码～

2011年五月在家里休息时，在家注册了域名，写了个博客，很简单的asp，并在淘宝上接了些修改模板之类的活，后来我的博客也由asp->伪静态->asp生成静态->asp cms的过程，在这个过程中学到了很多，也因此我找到了人生第一份it工作：网站程序员～

后来在邯郸亿搜人才网做程序员的过程算是一个转折点，是由装修转成it工作，当时在三线城市工资不高，2012年时是1200不管吃住，而我一家三口（我老婆+我女儿）在邯郸市里算是勤俭节约的生活着，很多朋友、亲属劝我趁早放弃这份工作，因为养不住家，但我的内心在想，我再坚持下，争取混的好点～

现在想想，也正是因为那份坚持，才使我走到现在，也非常感谢我老婆一直支持我～

## 快速成长

2012年时，针对网站使用的常用组件，我都是自己写，前期各种是抄，当时我把360buy的幻灯片、tab组件、延迟加载、放大镜等效果代码全部抄下来，然后自己改，最后熟悉后整个重构，再结合自己的思想统一设计接口，也正是因为这样的抄插件、改插件、重构插件的过程让我对使用jquery更熟练，也认识到了`.fn.extend`和`extend`的区别。也让我养成了看源码、写代码的习惯，直到今天还有这个习惯～

后来在网站改版时自己尝试设计整个架构，包括代码模式和加载方式、打包方式等，学到了很多，当然也给后面的人带来了许多坑～

2013年来到北京美食天下，负责一个新频道的改版，这对我是一个机会，我负责了前端整个架构的设计，包括用到的插件，由于之前的经历，这次我重点强调了规范文档、代码风格相关的事情，也在该频道的应用上做到了正向收益，为后续推动整个网站重构做了很多沉淀。乃至现在美食天下的很多通用组件还是之前我写的，比如全站登录弹出层(用户模块)、通用评论、延迟加载等，现在想想还是有点小自豪～

后来产品总监还玩笑的说要成立UED小组，让我来负责，当时我对这些没有概念，也可能错过了一个机会，现在我才知用户体验、交互设计是那么的重要～

记得2014年grunt横行的年代，我内心里对这些很抵触，认为自己学不会，后来跟layer作者@贤心聊天，他说：人不能被工具所吓到。我就抱着试一试的态度去用grunt，后来我可以熟练的把她当成项目工程的脚手架。最后我发现，没有学不会，只有自己学不学，不能被自己打倒～

有个话说的好：不怕你不努力，就怕比你聪明的人比你更努力。这句话激励着我过去、现在和未来。

我认为其实成长就是积累经验的过程，那么通过项目的洗礼、代码的沉淀使自己充实起来，从刚开始的调研->尝试->做出来->做好->做优的转变，平时只看不去实践应用（只看不写）是不管用

的，只有把想法实际落地才能有收获～

当然在整个转变中也有很多因素困扰我，比如：

不会英语怎么办？

由于辍学很早，我的英语一向都不好，刚接触的时候看到满屏幕的细小而琐碎的单词时我都差点放弃，但我知道如果放弃那么可能就一辈子做苦力活了，我不甘于现状，于是自己拿本记、用电脑记，最开始的时候我的命名是.liebiao，后来逐渐的尝试使用翻译工具，把中文->英文，我的做法是把常用的英文写到一个记事本里，然后打上中文注释，每次用到单词时就去里查找，然后看着记事本打出来。

然后在大量的单词进入记事本后发现有些是有特征的，比如：

- animate => 爱你骂他
- change => 嫦娥
- getElementById => get element by id
- attribute => attri bu te

根据这些特征，再加上工作中大量的应用，对于常用的慢慢的基本不用翻我的记事本了。

我个人感觉如果英语好对编程肯定是有帮助的，而英语不好的可以多加练习加强对英语的拼写，始终坚信一点：英语不是瓶颈。当然对于更高层面的技术可能有一定的限制，比如英语文章的阅读能力等。

如何使自己更靠谱

其实想想现在前端这么多，而很多企业都说招不到人，其实现状并不是这样的，如果他们对人没有要求，我想实习生的数量肯定可以满足，而他们是想招那些“靠谱”的人，那么如何才能使自己更靠谱呢？

- 多总结 - 常常总结工作中遇到的问题和解决思路、结果
- 质量高 - 不只是快速的把页面做出来，还要会考虑页面的整体质量、对上游、下游的配合，比如前端列表循环如何让后端更省心？如何和设计师更好的沟通反馈？
- 更贴心 - 日常工作要注意细节，也多优化细节，比如有日期的地方就添加周几:08-08(周一)这样的格式、汇报里有移动网站链接时带上二维码、汇报里有长链接时考虑短网址替换等
- 会的多 - 额，说的好像很俗，其实现实正是如此，要多去接触其他有涉及到的技术，比如：正则、shell、nodejs、nginx、php等，这些会让你的工作更加如鱼得水
- 发挥特长 - 找出自己的优点，并应用于工作中
- 稳定性高 - 严格对自己的代码负责，测试不只是Qa的任务，必须自测完才走下一环节，最大的减少bug的数量，有问题时也勇于承担责任

- 提出好的建议 - 在项目评审阶段，可以适当的从用户体验、代码性能、前端环境等因素考虑，提出些优质的建议，比如页面做动画优先使用css3是否考虑低端浏览器降级处理呢，移动端fixed元素的一些常见bug，移动端滚动事件的延迟等

遇到瓶颈怎么办

在学习、工作中难免会遇到技术的瓶颈，如果只是遇到问题可以 搜索->群/社区->朋友/同事，要多培养自我解决能力和提问题的能力。

如果遇到大的瓶颈，可以总结下自己目前的处境、遇到的难题、自己的期望状态、自己是否试着解决和进展等因素，找前辈沟通下，请教问题前自己最好对其进行相关的尝试，而不只是一句话的: 前端好学好？怎么学好js啊？

我通常是把目前遇到的问题列个表，邮件相关的前辈，一般都会得到满意的答复。很多前辈都是可以通过邮箱联系到的，并且可以及时的收到回复邮件。

或者说遇到方向的问题，迷茫了，不仿出去找前同事转一转，吃个饭，不要用已经浑浊的头脑去做大的决定。当然不只是跟同职位，跨职位的沟通往往可以达到意想不到的。

最后

每个人的成长故事都不一样，早读君期待你的参与～～

## 【同说】@吕大豹：有计划地踏实前进

---

前言

好久没看到同行的成长记录了，正在每次面试的时候都会问一个经典的问题，选择前端这个职业，你是怎么考虑，基于什么机缘进入的，每个人的问题都不一样，但每个人都很精彩。这篇是前端早读课专栏作者@吕大豹在前端之路三年的成长与回顾。

正文从这开始～

弹指一挥间，毕业已经三年了，昔日的小鲜肉变成上了年纪的小伙子，真是有些猝不及防。站在时间的分岔口，思绪万千。从事了三年的前端工程师，这一路是如何走来的呢，在此做个总结，也给初入前端的新手一点参考。

### 在国企晃了一年

在大学期间，我的技术并不算突出，资质平平。以至于参加百度、360的校招都被刷了。所以毕业后就进入了北大方正这样的大型国企。希望能从大企业中习得一套软件开发之法。

这期间我完成了从学生到软件工程师的转变，知道了标准的软件开发流程是怎么回事。简单说就是“步入社会”了。

我的岗位是前端工程师。在2013年，seajs已经小有名气了，但是我们并没有用，整体开发模式还

是原始的刀耕火种。一个jQuery作为基础库，整站的js都用script标签来引入。当时的后端是PHP，也没有前后端分离啥的，代码的目录结构就按照PHP框架的来组织，模板当然是后端模板。

至于代码的压缩合并，那时候也还没有grunt/gulp，我记得是用了一个客户端小工具来压缩过代码。

页面也都是我切的，这个值得说一说，因为当时我们的项目要同时支持PC访问和平板电脑访问。所以就用了当时比较流行的响应式设计，即不同设备使用同一套页面。那个时候就和Android/iOS的webview开始打交道了。踩的坑真是不少。

为了学习流式布局，我每晚都抱着张鑫旭的博客看，他css方面的文章很多，我基本都看了不止一遍。根据学到的东西，用百分比单位+inline-block流式布局+媒体查询，我们的项目在我一个新手的尝试下，成功的兼容了各种设备。那个时间是成长最快的，也是很充实的。

但是过了新鲜期，我很快就感受到了与国企的“水土不服”，国企的工作节奏是拖沓的，而我是血气方刚的，真是水火不容。有段时间我一直为项目的事情着急上火，但是其他人却不为所动，任凭项目一再延期。要知道在互联网公司，延期是很严重的事情，而在国企，简直是家常便饭。

另外一方面让我不爽的就是，我没有可交流的对象。部门只有我一个人做前端，作为前端新人的我此时是很渴望汲取知识的，经常跑去参加w3ctech组织的分享会。然而在公司内却没有一个可以交流技术的人。我甚至不知道别人家公司的前端都是如何工作的，我到底算不算一个真正的前端。

叫你一声前端你敢答应吗？我不敢。所以在不满一年的时候，我就离职了。

## 在大型互联网公司的一年

离职后我的目标很明确，就是互联网公司，工作强度越高越好。于是就加入了当时正处于虎狼之年的360。

之后我的工作是很充实的，看到了更大的世界，知道了大型互联网公司的前端都是怎么搞的。一开始有成银大神带着，后来又跟@墨尘频繁接触交流，成长很快。尤其是奇舞团的每周分享，我基本每期不落，吸收到了很多知识。这个时候，我才敢说自己是一名前端工程师了。

公司有很多大牛，也有很多现成的工具可以用。而我当时只是业务线上的一名小小前端，没有机会能够和大牛们一起造轮子。从技术框架上看，我当时开始接触了angular，requirejs和seajs也有所研究，当时由于微信H5游戏开始火爆了一阵，还接触了一下开发H5小游戏，学了学cocos2d-js，不过后来H5游戏的风似乎是刮过了，我也就没再深入cocos2d。

当时也有些需求是要做手机上的页面的，所以移动端的项目也做了一些，当时也没有现在比较成熟的rem方案什么的，切图需要有些额外的技巧，比如字号大小就是约摸着写，看上去差不多就行。

mvvm开发模式以及模块化开发，是前端发展历程中比较重要的两个里程碑，有幸自己没有落下这两个关键节点。



在我敢承认自己是一名真正的前端工程师之后，又开始有了别的想法，总在大公司做一个零件有点没劲。此时我迫切想干的事情就是，靠自己的力量去架构起一个项目的前端。所以除了各个流行框架，我还研究了一些前端工程化方面的东西，grunt尝试了一把。

感觉准备的差不多之后，我又开始蠢蠢欲动了。

\*\*\*

\*\*\*

## 在创业公司的一年

出于一些很现实的原因，我加入了现在的公司，一家比较成熟的创业公司。两方面原因：1.开得起工资，2.前景不错。毕竟，北漂了两年了，我也该为自己在北京落脚做一些准备了。

加入现在的公司后，我的定位不再是大型机器的一个零件，我需要对公司的前端担负更多的责任，因为我已经是一个老鸟了。于是我的思考方式也必须发生变化，以前不求甚解的知识现在也必须搞明白，并且更多的从项目整体上来考虑问题。

我开始承担面试工作，开始带新人。时刻思考着如何能利用多余的时间来做点事情，给公司的前端工作带来实际收益。

在技术方面，自由度也比较大了，可以尝试一些新的东西。首先是mvvm的大规模使用，如果以前使用angular只是蜻蜓点水的话，那么现在是在千万级pv的大站点施工了。由于我们的网站还需支持IE6（主要用户是老师学生，没办法~），所以采用了能够支持IE6的knockout。外加requirejs以及knockout的amd-helper插件，也搭建起一套模块化开发框架。

在移动端项目上，其他同事已经开始使用vuejs，使用webpack打包。尴尬的是，我对这两者兴趣极大，但是一直没能参与到移动端项目中去。

去年开始在公司推广sass，因为之前还一直是手敲css的。另外构建流也用上了gulp，来做一些项目上线的构建工作。

由于ES6也日渐成熟，而且我们有些项目也是不兼容低浏览器的。所以前段时间开始在项目中使用ES6，配置babel来编译，虽然现在使着还不是很熟练，但也总得跟上潮流不是。另外，我们项目中jQuery代码已经几乎没有了。

关于nodejs，自己业余也在学，今年年初用express框架写了一个团队内部用的小项目。话说nodejs这个玩意吧，虽然我觉得前端都得会点，但是如果没有项目驱动，要写nodejs的机会还是很少的。

现在比较流行的东西，没用上的一个就是react了，因为现在还没有合适的场景，而且目前的框架也能满足业务需要。可能得以后有功夫了再尝试了。只是不要等我用的时候就发现过时了就行。

这就是创业公司的诱人之处，可以尽情尝试各种新技术。总之现在的状态我还是挺享受的，手中忙碌，心中平静，脑中计划未来。

## 关于前端学习

发现这么多年自己折腾的还是挺多的，自己还在成长的道路上慢慢前行，学习的脚步一刻不能停。

如果要我给新手同学们写点建议的话，我还是很乐意的。首先前端的知识体系多而杂，所以点滴积累是必要的。在我是新手的时候，有一个习惯帮了我很多，那就是写博客。毕业三年，在博客园开博也三年了，写博客对你的学习有很大的驱动力，而且要想准确得表达出你的理解，你必须对某个知识点了然于胸，这反过来能增加你的学习深度。

另外就是知识路线，得有一个规划，大体上来讲，这条主路径不要偏移：原生js、使用jQuery及编写jQuery扩展、前端模板、前端mvc (backbone)、模块化开发 (amd/cmd)、mvvm开发模式、react为代表的redux模式。

另外css也是前端的一项基本功，有一些从后端转到前端的同学不想写css，我觉得这样不行，理解css有助于你去理解前端整体。所以关于css的学习，大体有这些就够：运用css完成页面、css基本原理（看w3c标准文档）、sass/less等预编译、架构可维护易扩展的整站css。至于css界新出的一些东西，比如OOcss、css module，我也还未研究，这些感觉是锦上添花的，可以先放一放。

nodejs的话，倒也不是最迫切需要学的，我看一些新手同学，js/css还未熟练，就扑到nodejs上，这就有点舍本逐末了。如果有余力，在学好前端的基础上再去碰nodejs。

## 【同说】@曼青：前端折腾之路

### 前言

还记得之前的那个[【同说】我的前端路上期待有你！](#)吗？早读君有做过专访前端界的前辈们，分享他们的经验，那么在我们身边呢其实也存在非常多低调的前端大神，只是有时候我们不知道而已。那么同说计划就是为你设立的。

正文从这开始

@曼青，广州华农某宿舍，妹子一枚，19岁（读书比较早还在奔二的路上啊哈哈）。曾参加过[【活动】第二期21天技术书籍阅读分享](#)

在前端摸爬滚打已经两个年头，说长不长说短不短，一直想写点什么记录一下自己的成长，刚好看到早读君的活动，起身动笔。

两年半前，带着想进IT行业想学计算机的想法入了华农的网络工程专业（然后回家都会被人问..呀跑去耕田了么...），稀里糊涂的开始了各种计算机基础理论课。然而一行行的代码，仅仅是在黑白屏之间跳跃，作为一个从小喜欢美术自认为有些艺术细胞的我，实在觉得枯燥对于课程也只是处于好好学习并未深究的阶段。

再说真正接触前端，从大一的一次网页比赛设计开始，为了响应学校的号召发挥自己多年的设计才能（我才不会说是看中他的奖金呢（咳咳...），设计了一个当时自认为不错的网站然后一两星期在w3cschool上速成了html+css，当然....结果是跟大奖无缘。不过也正是这一次经历，让我发现了

原来语言不仅仅是运行在控制台输出结果用了，原来有这么神奇的东西存在，原来我想的我画的东西可以真实的显示在网页中，可以那么酷炫的动起来。接着的剧情可想而知....在漫漫前端路上一发不可收拾，白天上课，晚上时常看文档看教程看博客到一两点（这点大家不要学还是身体重要）。

因为学校并没有前端系列化的知识，所以学习也基本都是靠着自学。认识一些同在前端路上的师兄，开始跟我介绍文本编辑工具（刚开始一两个星期被建议是用写字本的然后也因此对于常用的属性全拼都没多大问题，当然后来就换成了大名鼎鼎的sublime），开始按着F12去模仿网站去了解布局去判别绝对定位相对定位有什么不同。期间，主要的学习方法就是不停的翻文档，练习，不懂的百度查google查各种问，大概这种状态花了一两个月基本能做一些基本排版布局还能做点简单特效了，大一暑假跟着老师做了一个[关于艺人签约演出（“贩卖人口”](#)的项目（此刻，前端入门的新手任务完成）

再说说js，从w3cschool的相关内容和《Javascript Dom编程艺术》开始入门的，因为有c和java的编程基础，所以学起来并不是很困难，不用多久就可以发现能实现一些基本的动画可以写一些小玩意了（比如倒计时之类的），此时成就感max的像是看着自己的小孩学会了走路一样（男朋友都没拿来小孩...摔...）不过跟游戏一样，当你成功闯完了一关打败了boss，往往下面的关卡难度等级也开始提高。又继续看文档学了jq之后，发现整个人跟开了外挂一样，腰不酸了腿不疼了，不用每天去写document.getElementById(id)这么长的东西了。继续做项目没项目的时候就自己找事做，不过因为大二的课不是一般多（一星期从早上八点到晚上十点就三节空的，而且大多老师喜欢点名喜欢不定时堂测），技能点并没有飞速发展。利用零散的时间学会了ajax的使用，看着教程书用canvas写了一个俄罗斯方块，开始接触css3，学会了响应式布局，再有就是jq的使用开始开始熟练（常用函数基本记得，不记得的去翻文档也知道大概在哪个位置）暑假的时候在一个公司实习了一个月，再有接了一个外包的项目（内部成员冗杂，海内海外学生大触都有）也就是这个项目让我发现了人与人之间的差距），开始思考人生，遇到了传说中的瓶颈。一直的学习，只是纯粹的像小孩子闹过家家那样，虽然是兴趣的驱使，但就是这边玩玩那边玩玩，没有什么精通的。会重构会jq，基本在学校老师给的网站项目基本实现都不会有多大问题，有种穷途末路的感觉。然而，身边的挺多都还停留在课本阶段，都觉得你是大神，也找不到多少可以吐槽的对对象。就这样迷茫的纠结着，跟老师做做项目，重复的机械的排版布局，继续前行....

再到大三，关注了很多关于前端的公众号，微博，博客，每天去浏览，也许是视野宽阔了，突然发现满世界开始大谈less,sass,postcss，谈grunt到glup，谈node.js，谈Angular谈React，谈性能优化谈各种适配。也许刚开始会被吓到，但也因此有种清风徐来，柳暗花明又一村的感觉。跟着教程搭了hexo的博客开始记录新的生活，开始在github上提交自己的代码，开始学会分享，跟着师兄建立了w3ctrain的平台。也慢慢的发现，前端的发展很快，但是所有看起来酷炫听起来逼格很高的框架，都是别人封装好了的东西，基础好的话上手也只是花时间去了解语法思想而已，并不用太过畏惧。真正需要学习的，是去了解他们为什么要这么写，这么写和那样写有什么差之类之类。一方面回过头开始翻js红宝书和js设计模式，开始真正的去了解链式，了解对象，另一方面开始研究css3，html5的一些新属性，跟着一些主流博客的脚步去认知flex布局去对做的项目进行性能优化，再有一直说想学习一门后端语言的我开始接触了node.js，正在尝试去写点什么好玩的东西。

流水账的废话了那么多，总结一下：

- 看过的相关书籍:
- 《css禅意花园》
- 《JavaScript DOM编程艺术》
- 《锋利的jQuery》
- 《图解CSS3:核心技术与案例实战》
- 《响应式Web设计:HTML5和CSS3实战》
- 《高性能网站建设指南》
- 《单页Web应用JavaScript从前端到后端》
- 《图解http》
- 《web全栈工程师的自我修养》
- 《白帽子讲Web安全》
- 经常上的网站:
- <http://www.w3cplus.com/>
- <http://www.w3cfuns.com/>
- <https://css-tricks.com/>
- 

大致这样吧有些也不是太记得，更多整理于：<http://collect.w3ctrain.com/>

## 后语

早读君希望你也参与同说计划，分享你的前端故事，告诉Ta，你的学习路上有我~~

## 最后

---

如果你愿意分享你的故事，早读君在这期待你的到来。



公众号：前端早读课