前端自学路线之js篇

level 1

首先你要对js的基础知识进行系统的学习，脑海中先有一幅知识蓝图。我们现在说的js其实包含三部分：ECMAScript规范、DOM规范、BOM规范，你要知道这三部分都有哪些内容。其中ECMA规范定义了js作为一门编程语言的标准，包含变量基本类型、对象、函数、作用域、运算符、流程控制语句等。DOM规范则规定了js如何与网页进行交互，包括访问与操作DOM节点、不同类型的节点都有哪些特性和方法、事件的监听与传播等。BOM规范定义了js如何与浏览器进行交互，包含window对象、location对象、navigator对象、history对象等。

基础知识的学习最好就是看书了，像犀牛书、蝴蝶书、高程，都是经典的教材。

当你掌握了以上基础知识的时候，差不多能写出像弹出、移动、删除节点等动效，你体会到了所谓交互其实就是用DOM操作来模拟出各种“假象”。这个时候你会开始接触jquery，或许接触的还更早一些。你开始逐个尝试jquery封装好的各个API，发现用jquery书写代码确实简单了很多。jquery的学习和原生js的学习其实也没必要有严格的先后顺序，我一开始也是穿插的来学的。你只要清楚哪些是js的内容，哪些是jquery给你包装出来的快捷方法就行。

尝到jquery的甜头之后，你会发现一个大宝藏，那就是jquery插件。你要用到的各种组件网上基本上都有现成的，你开始在网上扒一些插件下来，自己连猜带蒙改改代码，改出一个自己能用的版本。你这个时候也会开始用jquery-ui，这一套比较完善的插件库，能解决你很多需求。

这就是第一阶段，你对js有了一个基本了解，自己摸索着能改改别人插件，差不多能写出一个完整的“玩具页面”了。

level 2

在上面的基础上写了一段时间代码后，你会感觉到自己的调试效率很低。具体表现为，你写的代码有时候总是运行不出自己想要的结果，代码报错了你却怎么调也调不对。这时候你就要对js进行更深一轮的理解了，你要对js中的一些概念做到真正的理解，比如作用域链、原型、闭包。比如你要知道js运算的自动转化机制，为什么2+“1”是string，而2-“1”是number。typeof和typeof()前者是操作符后者是内置函数。如何判断一个变量是否是数组，等等这些细节问题，都是有对应的概念和原理能解释的，你要做到知其所以然。

这个阶段你会开始与服务端进行交互了，原生的ajax要理解并能手写，jquery的ajax要熟练使用。你会遇到ajax的跨域问题，开始掌握发送跨域请求，知道JSONP是个什么东西，从原理上。

在改了无数别人的插件后，你需要开始自己动手写一个jquery插件了，因为别人的难免有坑，自己写的才最可控。另外对于自己的逻辑和代码组织能力也是一个锻炼。自己尝试写插件吧，从简单到复杂。比如一个弹框插件，要具备可以配置宽高、标题、内容、回调函数等功能。或者是一个焦点图滑动插件，不要看别人的代码，自己完全从头开始构思。

当你能理解jquery的插件机制，并能自己写出项目所需的插件时，你的js水平又上了一个等级。做到以上这些，你基本可以应付一个项目的前端需求了。也就是说，你写出来的页面不再是玩具了，可以上线使用了。

level 3

接下来你应该开始接触一些更复杂的系统了，就是所谓的“富客户端”，前端代码量上了一个档次，一个js文件动辄成白上千行。你会发现js代码还像以前那样从上往下堆着，太乱了。你写的ajax返回的数据越来越复杂，用拼字符串的方式进行局部更新太费力了。

这个时候你会开始用前端模板引擎来进行局部更新，比如handlebars、artTemplate等等。你需要熟练使用他们，并尝试去理解这些模板引擎的运行机制。

然后你要开始使用mvc模式来组织你日益复杂的代码了，典型的框架就是backbone。但是backbone现在已经过时了，虽然用的不多了，但如果你还是新手，起码去了解一些它的思维，知道用逻辑层次来划分代码结构是怎么个搞法。知道前端路由是个什么机制。

到了这个阶段，你已经告别了原先的纯“刀耕火种”时代了，你开始使用各种框架来帮助你更好的完成功能。

level 3 plus

为什么有个3 plus阶段呢？因为你此时需要进行的是一个横向的扩展，js的学习本来就是一个纵横交叉的网络嘛。

此时是学习HTML5的最佳时机，一方面你对js的基础知识有了一定的储备，另一方面你也可能开始接触移动端的页面了。

HTML5新增的标签倒是小菜一碟，像<header>、<footer>、<section>、<video>之类的你在之前也多多少少接触了。更多的内容在HTML5新增的js API这块。比如新增的File API，localStorage/sessionStorage、canvas API、histroy的扩展，xhr的扩展等等。这些在移动端都是可以放心使用的，需要你对这些新特性一一学习。

你也应该在这个阶段尝试移动端的页面，了解它与PC页面开发的不同点。比如你应该在移动端抛弃jquery，开始使用zepto。了解touch事件，了解移动端click 300毫秒延迟的问题并找到解决方法。尝试写移动端的组件，如无限滚动、左右滑动插件，并解决移动端的性能问题。在网上找找司徒正美、张代平、叶小钗等总结的移动端兼容问题的解决方案等等。

总之，在这个阶段，你要能拿得下移动端页面的开发，并熟练使用HTML5的新特性。

level 4

经历了上面阶段，你的单兵作战能力已经算可以了。这个时候你会面临团队协作以及模块化开发。这个时候你就有必要了解模块化的规范了。你需要了解commonjs、AMD、CMD都是包含哪些内容，他们是什么关系，有哪些区别。

你起码得使用一下requirejs，知道老牌经典AMD模块化工具是个什么思维，解决了哪些开发中的痛点。seajs你也需要了解，知道seajs对AMD进行了哪些改进和包容。最后，你得知道commonjs规范是如何借助打包工具（browserify、webpack）一统天下，通吃服务端(nodejs)以及浏览器端的。

模块化开发与打包是分不开的，这个阶段，你也应该自己尝试使用grunt、gulp、webpack对项目进行构建和打包。尽管grunt已经没落，眼看未来是webpack的天下。

level 5

上面几个阶段的内容，其实已经是三年前乃至更早时候的东西了。在这个阶段，你才真正与时代接轨了。那就是mvvm。

你应该首先了解mvvm模式的开发思想，它是如何用双向绑定的方式来解脱我们的DOM操作，又是如何用组件化的思维来更好组织我们的代码。

老牌的mvvm框架如ember、knockout，你估计能用上的概率不高了。但是从2013年开始火起来的avalon和angular你起码得用一个。做到能用mvvm架起一个项目的程度。关于angular我曾经写过一个系列的文章，有兴趣的同学可以在我博客搜索，或者百度关键字「走近angularjs」。

最新也是目前最火的vuejs你可以得开始了解，它面向未来，只兼容高级浏览器，性能极佳而且API极简，是移动端mvvm方案的不二选择。

学习mvvm，会对你的开发习惯和开发思维进行一次转变，相对于jquery时代。可能一开始会不习惯，但这个转变是你必须经历的，因为，它会变成你未来学习其他框架的基石。

level 6

在与时代接上轨之后，你还不能停，还得更近一步。因为未来已经离我们很近了。没错，我说的就是ES2015。ES2015，乳名ES6，已于去年发布正式版，虽然浏览器的支持程度还在慢慢完善，但是我们现在使用babel进行编译后，已经完全可以工作了。

所以你现在就应该开始了解并使用ES6了。事实上，我们项目组也是今年才开始使用ES6的，对于新特性，越早上手肯定是越好，毕竟还需要一定时间的熟悉，以及开发模式的架构。

另外一块新技术就是React，以及React Native、淘宝weex这样的混合app开发框架。这又是一套全新的开发理念，在未来两三年也必定会占领越来越多的份额，所以，他们的学习也是不能落下的。其实在我们公司，目前还未开始React的使用，我相信在不久后的某个时间节点，也就自然上场了。

到了这个阶段，js方面的主流知识和框架，你也已经掌握的差不多了。其实这个时候也就不需要我来指点学习路线了，因为你到了这个阶段，自然也就明白了行业的发展趋势，能够自己把控该学什么该用什么了。

level 7

源码深度研究以及研发框架。应该明白是什么意思了哈。因为我自己目前也还不到这个阶段，所以也就不误人子弟了。放在这里，当作奋斗目标好了。

今天要说的就这么多了，这差不多也是我这几年的学习路线，基本和行业发展的进度吻合。但这也只是一条主线而已，前端领域的知识是网状的，这中间又会穿插各种大小知识，需要在学习的过程中点滴积累了。关于主线之外的其他知识点，我会在后续的文章中再给大家叨叨。敬请期待下一篇：前端自学路线之综合篇。