



吴少林

善于总结，在反思中成长。个人技术博客 <https://codevvv9.github.io/>

27岁 | 山东青岛 | 15265249516 | 15265249516@163.com |

求职意向



初级前端开发工程师 全职 北京 9K-11K

教育背景



2011.9-2015.6	山东科技大学	交通工程(本科)
2015.9-2018.6	山东科技大学	交通运输工程(研究生)

职业技能



- 1、HTTP协议

了解

能够根据HTTP请求、响应、观察浏览器network的状态码去排除一些bug。深入一些HTTP的高级知识，例如缓存与持久化存储，理解Cookie，并在此基础上模拟理解了一个网站的登录注册的基本流程。
- 2、HTML5、CSS3、JavaScript

熟悉

熟悉W3C的主流规范，熟悉HTML5更加合理语义化标签（方便SEO）、合理的HTML页面布局，考虑浏览器兼容性并解决。

熟悉移动端开发。并在项目中实践以及进行相应的调试。

掌握CSS的选择器、常见样式、两种和模型、浮动与清楚浮动、定位、各种居中、媒体查询、动态REM、更好用的布局方式（Float+Flex布局）、CSS3的各种优秀的属性（animation、nth-child、border-radius、boxshaw、transform、transition）

掌握JavaScript的基本数据类型、基本的API操作、数组对象使用，函数的第一公民的理解（我的总结），原生JS的变量提升以及ES6的let、const，原型链的相关知识（我的总结）、引用类型、DOM操作、事件模型（冒泡、捕获、事件代理、取消默认事件）、闭包的使用，跨域的出现以及使用JSONP和CORS解决，AJAX的使用、MVC思想的实践、Promise的大量使用、使用debugger和console方式积极的调试。
- 3、JQuery

熟悉

为了简化原生JS操作DOM，使用JQuery来操作DOM、使用AJAX等相关API,能够使用JQ完成轮播图以及Tab页的切换。

4、模块化、面向对象的编程 熟悉

在项目里面大量使用MVC的编程思想并使用面向对象的思想提高代码的复用性，将不同功能的代码使用模块分离，降低耦合性。

5、前端自动化工具 熟悉

在前端自动化的使用中，前期使用过node-sass、babel、watch-CI等工具，考虑到要同时开启多个命令行，最后使用webpack静态打包器进行自动化，以及使用parcel。

6、寻找轮子提高工作效率 了解

一些业务需要使用现成的库去实现，例如Tween.js、Swiper.js、JQuery，根据文档来放到项目里面使其工作。

7、Vue.js、JSON、Node 熟悉

会使用Vue做一些组件，例如popover、轮播、tab切换，正在继续学习中。熟悉JSON的数据交换格式，开发中经常使用JSON.parse()、JSON.Stringify()。使用Node搭建静态服务器，模拟后端，可以提高自学中的真实性，具有很高的实用价值，githubNode仓库的server.js

8、Git 了解

常见的命令熟悉，在实际中使用git init、git add、git commit、git pull、git push、git diff、git log。为了简化操作，在.bashrc里面使用alias,并且注意commit的规范，配置git Message，作为commit的template。

9、Linux 熟悉

完全使用Linux进行前端开发，命令行操作占比较高

实习经验



2016.8-2017.9

青岛海信网络科技股份有限公司 软件开发实习生

- 1、负责平安城市深度学习自主样本库的建立。自学图像处理的基本代码以及批处理提高工作效率。
- 2、协助城交事业部检测监控点位是否标注完备。使用java解析XML向服务器发送报文，检验是否有标注。
- 3、协助城交事业部完成路段信息的级联信息的更新。使用原生JS以及jQuery操作DOM，局部刷新信息使用AJAX。

2017.12-2018.2

北京牛犇智能科技股份有限公司 前端开发实习生

- 1、与技术负责人一起完成新一代产品的需求调研与网站HTML分块以及CSS的修改。在维护老版本的客户需求的界面上，结合当前的需求进行兼容原来的IE版本，了解到了autoPrefixer,babel。学会用工程化的思路去看待一个项目。
- 2、使用规范的JSON发送数据。使用AJAX与后端进行交互，阻止某些form表单默认提交，改用AJAX。学会了跨域以及在经理的指导下，学习了JSONP和CORS。与后端相处的过程中，加深了自己的HTTP协议的理解。与后端的数据交互发生问题时，使用控制台的netWork查看请求与响应，及时定位bug所在。使用linux以及一些bash脚本提高工作效率。



云音乐（移动端）后台和用户使用界面

项目预览|代码链接

功能介绍：

- 1、后台界面admin.html功能，a.点击+拖曳上传歌曲 b.显示歌曲列表 c.修改歌曲相关信息 d.与数据库同步 e.上传歌曲的loading的动画 f.上传歌曲时使用CSS属性和JS加锁的方式阻止用户操作。
- 2、前台用户界面index.html与song.html，a.使用SVG与HTML5、CSS3做出推荐歌曲、热歌榜、搜索三个tab页，可切换 b.可以在推荐歌曲内部模块同步数据库歌曲列表 c. 最新音乐可以播放歌曲 d. 在song.html实现播放与暂停的功能。

- 技术细节：1、参考各大音乐播放器完成项目的需求分析。明确管理员页面和用户页面的不同功能。前后台完全使用模块化与MVC思想。
- 2、在上传音乐的时候。使用node搭建静态服务器server.js获得uptoken密钥，上传歌曲的时，有加载动画。
 - 3、在index.html推荐音乐的模块内部使用原生JS完成模块间的相互引用。
 - 4、使用JQuery简化DOM操作。使用svg显示各种图标、使用ES6的常见语法，例如解构赋值、object.assign()、避免全局变量的{}方式替代原生的立即执行函数。
 - 5、使用发布订阅模式Event Hub.js模块来进行模块间的通信，解耦。避免了以前使用全局app.js来完成这模块间通信。
 - 6、用[博客](#)记录项目遇到的一些bug与搭建

技术栈：HTML5+CSS3(阻止用户操作)+原生JS（部分ES6）+jQuery+自学轮子+MVC+面向对象+钩子思想+发布订阅模式通信+Promise+原生的模块引用+AJAX

个人在线简历

项目预览|代码链接

功能介绍：

- 1、具有加载动画
- 2、顶部导航栏的sticky+hover高亮+二级菜单显示
- 3、内容自动定位+线性滚动
- 4、作品集的轮播
- 5、各个模块的hover的阴影效果，简历的下载功能
- 6、数据库的留言上传功能。

技术细节：1、使用HTML5+原生JS+webpack完成前端自动化使用HTML5+原生JS+webpack完成前端自动化，使用各种CSS3的属性完成阴影、动画效果、过渡效果，使用更好的选择器完成DOM操作。

2、在代码重构的过程中，学习使用组件Tween.js以及Swiper.js。

3、使用AJAX+LeanCloud讲用户的留言保存到数据库。

4、使用立即执行函数避免各种全局函数的污染+闭包+全局JS+MVC思想+面向编程的思想，提高代码复用性，解耦。最后使用Webpack打包发布项目。

技术栈：HTML5+CSS3+原生JS+部分ES6+webpack+loader+plugins+自学轮子+AJAX+MVC+面向对象

CSS3动态绘制皮卡丘

[项目预览](#)[代码链接](#)

功能介绍：支持 PC 端和移动端展示，完全使用 CSS 3实现皮卡丘的绘制，并使用动画的效果显示，并三个速度可调节播放速度。

技术细节：1、先用纯CSS3绘制出静态的皮卡丘，综合运用position、伪元素before、after、Flex布局、box-shadow、border-radius、transform 和渐变等 CSS 3技术。

2、使用部分ES6语法，比如模板字符串，使用原生js将写好的CSS代码插入到Style标签里面。

3、使用setTimeout和JQ完成动画的显示,使用立即执行函数避免全局变量的污染。

技术栈：HTML5+style标签的全新理解+CSS3+ES6语法+setTimeout+switch+立即执行函数

PC端页面导航

[项目预览](#)[代码链接](#)

功能介绍：1、键盘点击+鼠标点击，跳转到预设的网站

2、可以点击键盘的编辑按钮，修改网址，会弹出提示界面。

3、具有持久化存储的功能，可以一直保留在客户端。

技术细节：1、使用了 HTML5、CSS3、原生 JavaScript 和 Animation 等技术，使用 CSS3 的阴影 + 动画，使得页面更加美观

2、使用 [localStorage](#) 将用户的设置保存在本地，可以做到持久化存储。

技术栈：HTML5+CSS3+原生JS+localStorage+CSS的布局优化

移动端Canvas画板

[项目预览](#)[代码链接](#)

功能介绍：1、支持 PC 端和移动端的操作，最好是在移动端操作。

2、包含 画笔、橡皮擦、九宫格颜色选取、画笔粗细、清空和下载、用户使用手册等功能

技术细节：1、使用了 HTML5+canvas API + 原生 JS 进行开发

2、使用事件检测的思想与函数封装，通过ontouchstart来判断是否是手机端

技术栈：Canvas+原生JS+事件检测思想+ontouchstart、ontouchmove