

此简历在线地址: <https://floatsyi.com/resume/>

个人信息

- 喻智勇/男/1998
- 学历: 大专 (计算机科学与技术)
- 博客: <https://floatsyi.com>
- github: <https://github.com/doublethinkio>
- 阅读书签: <https://www.gettoby.com/p/0qhnlxhp4lfh>
- 工作经验: 四年
- 期望职位: 前端高级开发工程师
- 期望城市: 深圳
- 期望薪资: 面谈
- 当前状态: 在职

技术栈

以下为我熟悉的技术栈(使用 react hooks 较多):

- 基础: [html5](#) & [css3](#) & [javascript\(esnext\)](#) & [node-lts](#)
- 预编译语言: [typescript](#) & [scss](#) & [pug](#)
- 前端库: [react hooks](#) & [vue-next](#)
- 脚手架: [create-react-app](#) & [vue-cli](#)
- 工程化: [webpack](#) & [babel](#)
- 单元测试: [jest](#) & [react-hooks-testing-library](#)
- 跨端开发框架: [taro](#)
- 编辑器: [vscode](#) & [neovim](#)
- 代码规范: [eslint](#) & [stylelint](#) & [prettier](#) & [husky](#) & [commitlint](#) & [lint-staged](#)
- 文档: [markdown](#)
- 版本管理: [git](#)
- 包管理工具: [npm](#) & [yarn](#) & [scoop](#) & [choco](#) & [pip](#) & [cargo](#) & [pacman](#) & [yay](#)
- 操作系统: [windows 10](#) & [wsl](#) & [archlinux](#) & [mac](#)
- 终端: [windows-terminal](#) & [kitty](#)
- shell: [zsh](#) & [fish](#)

以下为我了解的技术栈

- 微前端: [qiankun](#)
- 跨平台桌面应用框架: [electron](#) & [react.node-gui](#)
- webassembly: [rust wasm-pack](#)

更多可能(我对我的技术栈没有限制, 因为看完文档都能写)

个人优势

自学前端并积累了整套前端自学资源, 正在兼职带徒, 五年编码经验, 三年开发经验, 涉猎广泛, 熟悉前端框架选型, 脚手架搭建, 自动化测试, 持续集成持续部署, gitlab 私服搭建, 还能写得一手同事都夸赞的干净代码。

联系方式

- 手机: 13606640872
- 邮箱: floatshuyin@gmail.com

工作经历

- 公司名称: 深圳观麦科技有限公司
 - 职位类型: 前端开发
 - 工作内容: 负责面试、erp 项目的进销存与分拣模块、以及 PC 端 electron 分拣软件的开发与维护
 - 在职时间: 2021.04 - 至今
-

- 公司名称: 西安猫兜灵智能科技有限公司
 - 职位类型: 前端组长
 - 工作内容: 框架选型, 脚手架搭建, 组件开发, 前端自动化测试
 - 在职时间: 2019.12 - 2020.12
-

- 公司名称: topcoder
- 职位类型: 前端开发
- 工作内容: 前端 web 页面开发
- 在职时间: 2018.06 - 2019.11

项目与开源库

PC

分拣软件

分拣软件 是一个桌面软件, 可通过串口和蓝牙连接电子秤与电子秤通信, 还可以连接打印机打印分拣标签是分拣系统的重要环节。

技术栈方面使用 [electron](#) & [react](#) & [typescript](#) & [mobx](#).

我使用 [html-react-parser

] 重构了打印页面模块, 接入了 [react](#) 生态, 解决了在单个 html 文件里开发导致的许多问题, 并使用 promise 队列解决打印过快时打印机漏打的问题。

Web

分拣系统 ERP

分拣系统 ERP 是一个将物流、财流、信息流集成化管理的应用系统, 包含采购、销售、库存、客户、财务等模块。

技术栈方面使用 [gm-react-app](#) & [react hooks](#) & [typescript](#) & [tailwindcss](#) & [react-router](#) & [mobx](#).

主要负责 erp 项目的进销存模块需求评审, 开发以及维护。

学习观

学习观是一个应用多模态学习的在线学习平台。

技术栈方面使用 [create-react-app](#) & [react hooks](#) & [typescript](#) & [emotion](#) & [react-router](#) & [easy-peasy](#), 并大量使用了 [react-use](#) 库中的 hooks.

我主要负责框架选型, 工程化配置, 代码规范, 单元测试, 以及视频模态区的编写。

具体工作内容如下:

- 使用 [create-react-app](#) 脚手架创建项目, 并使用 [craco](#) 自定义 [create-react-app webpack](#) 等配置, 避免了 eject.
- 配置 [eslint](#) & [stylelint](#) & [prettier](#) & [husky](#) & [commitlint](#) & [lint-staged](#) 以保障代码质量与规范 git commit message. 配置了 [vscode](#) 编辑器 `"source.fixAll": true` 配合 [prettier](#) 和 [eslint](#) 实现保存代码自动格式化, 在提高代码质量的同时提高编码体验.

- 编写 README 文档, 主要是写明了一些开发环境的需求(Requirements), 以及快速开始运行项目的步骤(Getting started), 还有就是推荐一些提升编程体验的编辑器插件(Enhance your development experience), 最后是 QA 部分会写明一些常见问题的解决办法, 比如包管理工具 [yarn](#) 的淘宝镜像设置, 帮助新人快速融入到项目开发中.
- 使用 [react-testing-library](#) 和 [jest](#) 编写前端路由的快照测试.
- 配置 [webpack postcss](#) 插件, 使用 [purgecss](#) 剔除未被使用的 css class , 以减少编译后的 css 文件大小.
- https 请求库选用支持 [node](#) 与 promise 的 [axios](#), [axios](#) 还有拦截器插件机制, 通过编写拦截器很方便的实现了 [jsonwebtoken](#) 的获取与自动续期.
- 编写 useAxios hook, 并使用 [swr](#) 增强 useAxios, 还使用 [react-hooks-testing-library](#) 编写了 useAxios hook 测试.
- 编写 loading 和 404 页面, 页面路由时 loading 使用的 react suspense fallback, 配合 `react.lazy` 异步加载页面, 优化页面请求大小.
- 搜集资料做 react hooks 全局状态管理库的选型, 并输出了一篇[比对文章](#): 最终选用了 [easy-peasy](#).
- [css-in-js](#) 库的选型时参考 [awesome-css-in-js](#) 后选用了 [emotion](#).
- 视频模态区的播放器组件使用 [videojs](#), 进度条使用的 [react-compound-slider](#), 并编写了 usePlayOrStop / usePlaybackRates/ useVideoQuality / useVideoTime /useVolume /useFFFR 等 hook.
- form 组件和 form 验证选用 [formik](#).
- i18n 选用 [react-i18next](#).
- 动画库选用 [react-spring](#), 对于简单的动画还使用了 [animate.css](#) 库, 对于 [animate.css](#) 我封装了一个 hook: [use-animate-css](#).
- 编写 useSocialShare hook, 这个 hook 封装了各个社交网站的分享 api.
- 更新项目依赖项并跟进 Breaking changes.
- ...

学习观后台管理系统

技术栈方面使用 [create-react-app](#) & [react hooks](#) & [typescript](#) & [emotion](#) & [react-router](#) & [easy-peasy](#) & [ant design pro](#).

我主要负责框架选型, 工程化配置, 代码规范, 单元测试.

具体工作内容如下:

- 编写 useRouter hook 实现配置式动态路由, 路由守卫等功能.
- 配置 [webpack](#) 插件 [mocked-api](#).
- ...

TC3.0 Marketing Site Prototype

[topcoder](#) 的 marketing site 页面

- 为 marketing site 添加 svg 动效
- 为 topcoder 主页导航栏原型添加交互动画

线上地址: <https://www.topcoder.com/>

项目地址: <https://gitlab.com/FloatingShuYin/tc3-marketing-site-prototype>

connect-app

参与 topcoder 的 connect-app 项目开发

- 使用 React 编写 [draft](#) 编辑器的插件实现类似 [google docs](#) 链接编辑模块的功能

线上地址: <https://accounts.topcoder.com/connect/registration>

项目地址: <https://github.com/doublethinkio/connect-app>

开源库

zaza

[zaza](#) 是一个使用 [typescript](#) 来编写 [neovim](#) 配置的项目。

[neovim](#) 现已支持使用 lua 来编写配置，而 [typescripttolua](#) 则可以将 [typescript](#) 代码转换为 [lua](#) 代码，因此使用 [typescript](#) 来编写 [neovim](#) 配置是可能的。

如果要使用 [typescript](#) 来编写 [neovim](#) 配置，首先要做的就是提供 [neovim lua](#) api 的类型文件。此处参考 [lua-dev.nvim](#) 的[做法](#)，从 [neovim](#) 获取到 message pack 文件，并从 [vim-lsp](#) 获取到 [builtin-docs.json](#) 文件，这些文件描述了 [neovim](#) 的 [lua](#) api，我们可以将其作为输入，通过 [typescript](#) 的编译器 api 生成并输出 d.ts 文件。

message pack 的获取通过编写一个 [archlinux](#) 的 dockerfile，git clone neovim 然后执行生成 message pack 的 [python](#) 脚本，最后将生成的 message pack 导出 [docker] 到宿主机即可。

由于 message pack 与 [builtin-docs.json](#) 描述的信息不全面，因此生成的 [neovim lua](#) api 的函数参数类型多为 any，需要人工的在此基础上对常用 api 做一层封装，提供类型安全的便捷的上层 api。

use-animate-css

使用 [typescript](#) 编写的 [react hooks](#)，封装了 [animate.css](#)

🌟 Live Demo: <https://codesandbox.io/s/useanimatecss-tyklw?file=/src/A.tsx>

项目链接: <https://github.com/doublethinkio/use-animate-css>

文章

[development-environment-manual](#)

这是一份使用 [windows](#) 系统包管理工具 [scoop](#) 搭建开发环境的教程, 特点是几乎所有操作都是通过纯命令行的方式去做的, 这意味着在 [windows](#) 系统上也可以通过自动化脚本一键配置开发环境。

包括以下编程语言:

- [c/c++](#)
- [golang](#)
- [java](#)
- [javascript](#)
- [python](#)
- [rust](#)

开发环境的搭建涉及四个方面, 软件安装, 环境变量, 编辑器, 等宽字体

软件安装方面, 使用 [windows](#) 包管理工具 [scoop](#)

通过包管理工具 [scoop](#) 下载安装软件将会自动处理软件依赖, 添加环境变量等琐碎事情

编辑器则推荐的开源编辑器 [vscode](#)

除此之外还推荐了支持 [ligatures](#) 的 [monospaced](#) 字体: [FiraCode](#), 对中文与 [powerline](#) 的支持都很好的:

项目链接: <https://github.com/doublethinkio/development-environment-manual>

致谢

感谢您花时间阅读我的简历，期待能有机会和您共事。

