

1. 基本JS语法

学习资料:

MDN为Web最权威的API接口及规范的参考书

MDN的JS资料: <https://developer.mozilla.org/zh-CN/docs/Web/JavaScript>

2. ES6语法 (新一代JS语法, 增加很多特性, 可以认为是JS的新版本, 目前很多浏览器不兼容, 需要babel或typescript转换, 稍后会说)

推荐读: <http://es6.ruanyifeng.com/>

其中重点看 前九章及Promise, Class, Module

备注: es6可以认为写现代单页面web应用的基本语法。

3. npm (node package manager 包管理器, 类比iOS的cocoapod, 是node相关的控制器)

主要是npm的基本用法, 包括package.json文件的使用

在官网: <https://www.npmjs.com/> 可以搜索到想用的第三方包

备注: 学会如何安装包即可

4. node.js (一个语法与现代JS相似的语言)

其是一门脚本语言, 是一门服务器语言, 安装node环境后即可通过 `node *.js` 执行js脚本

api网址: <https://nodejs.org/docs/latest-v5.x/api/>

备注: 前期不用学, 当充分理解webpack及单页面应用的原理, 可通过node.js 写服务器

特点: 轻量级

5. webpack (最难的, 最复杂的, 最让人崩溃的, 单页面应用最需要了解的, 需要长期的查阅, 分析, 是一个前端工程师能力的一个重要体现)

webpack为一个单页面应用模块管理工具, 目前推荐使用webpack2

官网参考资料: <https://webpack.js.org/configuration/>

需要了解webpack基本使用方法包括配置文件, loader的理解, 插件的理解等, 可通过Vue或React的脚手架来理解webpack。

备注: 理解了webpack的流程, 前端就已经进入了高阶, 同时已经有了可以通过node.js写简单服务器的能力。

6. Vue.js (一个目前流行的前端框架, 尤雨溪大神的作品)

学习教程: <https://cn.vuejs.org/v2/guide/>

awesome: <https://github.com/vuejs/awesome-vue>

问题解决: <https://github.com/vuejs/vue/issues>

备注: 上面的知识熟悉后, 看一遍教程就差不多了, vue.js本身不难, 建议建一个小项目边学变做, 常见问题可以去issues看。

进阶: vuex的学习 (一个状态管理模式的基于vue的实现)

前端比较流行的状态管理模式是基于redux的, <http://redux.js.org/>。

Vuex: <https://vuex.vuejs.org/zh-cn/>

备注: 建议后期进阶必须学习。

7. React及ReactNative (FaceBook搞的前端框架)

React是Web端的 ReactNative是APP端的, 两者都使用JSX语法使用, 学习成本比较大, 需要充分理解前5个知识并理解Vue的思想才好学。

React: <https://reactjs.org/>

ReactNative: <http://facebook.github.io/react-native/>

备注: 如果学习, github上有托管的项目, 但是私有库, 需要加你的ssh密钥方可clone。