1. react的前世今生

1.1 刀耕火种的年代

1.2 web应用的出现

1.3 react的诞生

2. 预备知识

2.1 npm

2.1.1 什么是npm

2.1.2 理解npm script

2.2.3 dependencies和devDepencies

2.2 webpack

2.2.1为什么需要webpack

2.2.2 webpack 入口和出口

2.2.3 webpack loader

2.2.3 webpack plugin

2.3 ES6

2.3.1 函数的扩展

2.3.2 对象的扩展

2.3.3 class

3. 初探react

3.1 本地环境搭建

3.1.1 npm与node安装

3.1.2 打造属于你的编辑器

3.2 编写第一个react应用

3.3 与传统jquery对比

3.3.1 写法的异同

3.3.2 命令式与声明式

3.4 react调试

3.4.1 安装chrome插件

3.4.2 chrome插件的使用

4. react 组件

4.1 理解组件化开发思想

4.2 组件之间的通信

4.2.1 state

4.2.2 props

4.2.3 子父组件通信

4.2.4 同级组件通信

4.3 组件生命周期

4.3.1 组件的挂载

4.3.2组件的卸载

4.3.3 组件的更新

4.3.4 总览组件生命周期

5. 漫谈react事件系统

5.1 剖析react事件系统

5.2 react事件系统实际应用

6. 深入react原理

6.1 JSX

6.2 dom-diff

6.3 setState

6.4 fiber

7. react 组件编写实践

7.1 react 组件分类

7.1.1 木偶组件

7.1.2 智能组件

7.1.3 功能组件

7.2 组件设计思路

7.3 react组件组织思想

8 redux数据管理

8.1 总览react数据管理

8.1.1 flux的出现

8.1.2 redux应运而生

8.1.3 mobx

8.2 redux核心概念

8.2.1 store

8.2.2 action

8.2.3 reducer

8.2.4 总结

8.3 redux生态

8.3.1 redux-logger

8.3.2 redux middleware

8.4 redux进阶

8.4.1 理解Middleware原理

8.4.2 动手实现Middleware

9. react架构

9.1 css方案

9.1.1 css module

9.1.2 postcss

9.2 状态管理

9.2.1 你可能不需要redux

9.2.1 redux

9.3 路由管理

9.4 文件结构

10 react服务端渲染

10.1 服务端渲染的意义

10.2 理解服务端渲染原理

10.3 动手实现服务端渲染

11 编写测试

11.1 测试驱动开发

11.1.1 测试驱动开发好处

11.1.2 测试驱动开发现状

11.1.3 定义属于自己的测试原则

11.2 react测试工具

11.2.1 ject

11.2.2 enzyme

11.3 动手测试我们的代码

11.4 测试之外

11.4.1 proptype

11.4.2 flow

11.4.3 typescript

12 性能优化

12.1 不要过早优化

12.2 性能优化的瓶颈

12.3 react优化手段

12.3.1 shoudComponentUpdate

12.3.2 immutable

13 代码规范

13.1 代码规范的价值

13.2 代码规范工具