拥有轻量级的库

丰富的DOM选择器

基本兼容了现在主流的浏览器

有着丰富的第三方的插件,并可以根据自己需要去改写和封装插件

实现脚本与页面的分离

对面向对象的编程思想把握不是很到位,结构很松散

JS中所有的东西都由Object衍生,所有东西原型链的终点指向Object.prototype

Prototype

JQuery

不会额外产生内存,所有实例化后的对象都会从原型上继承这个方法

数据驱动+组件化的开发模式

核心是一个响应的数据绑定系统

双向数据的绑定

使用基于依赖追踪的观察并且使用异步队列更新

支持指令:内置指令和自定义指令

支持过滤器:内置过滤器和自定义过滤器

使用基于依赖追踪的观察并且使用异步队列更新

MVVM框架,映射出模块化的特点

学习成本相对vue来说比较高,比如增加了Dependency Injection特性

angular.js

React

Vue.js

性能上,AngularJS依赖对数据做脏检查,所以Watcher越多越慢

采用特殊的JSX语法,需要编译后使用

中心思想和vue相同,一切都是组件,组件实例之间可以嵌套

React依赖Virtual DOM,不总是直接操作DOM

并没有像vue一样提供了指令,过滤器,不如vue快捷

支持客户端与服务器渲染

是一个基于V8的,可以快速构建网络服务及应用的平台

框架 (封装库)

遵循ES6的规范,从核心语言方面和类概念的相相当于JS的DSL,伪编译(语法处理)后转化

超集typescript

语言本身

JS

通过类型注解提供编译时的静态类型检查

引入了模块的概念,可以把声明、数据、函数;

安全子集 (这里不做介绍)

不再使用多进程、线程来处理多个web请求,使用事件驱动,非堵塞IO模式

node.js

平台 (运行于服务端)

是一种由JS、CSS等组成的技术,基于JS但没有对JS进行封装

AJAX

使用非同步的HTTP请求,使用客户端脚本与Web服务器交换数据