前端入门技术分享

路琳 基础应用组 2017.8

CONTENTS

01 前端开发的历史和趋势

02 前端知识框架

CONTENTS

01 前端开发的历史和趋势

前端开发的历史和趋势

前后端不分 -> 前后端分离 -> 全栈工程师

什么是前端

- 前端：针对浏览器的开发，代码在浏览器运行。
- 后端：针对服务器的开发，代码在服务器运行

前后端不分的时代

- 互联网发展的早期，前后端开发是一体的，前端代码是后端代码的一部分。

后端收到浏览器的请求 → 生成静态页面 → 发送到浏览器

前端工程师的角色

- Model（模型层）：提供/保存数据
- Controller（控制层）：数据处理，实现业务逻辑
- View（视图层）：展示数据，提供用户界面

前端只是后端 MVC 的 ***V***。

模板工程师，负责编写页面模板。

Ajax

- AJAX 即“Asynchronous Javascript And XML”（异步JavaScript和XML）
- 前端不再是后端的模板，可以独立得到各种数据。

Web 2.0

Ajax 技术促成了 Web 2.0 的诞生。

- Web 1.0：静态网页，纯内容展示
- Web 2.0：动态网页，富交互，前端数据处理

前端 MVC 框架

- 前端通过 Ajax 得到数据，因此也有了处理数据的需求。
- 前端代码变得也需要*保存数据*、*处理数据*、*生成视图*，这导致了前端 **MVC** 框架的诞生。

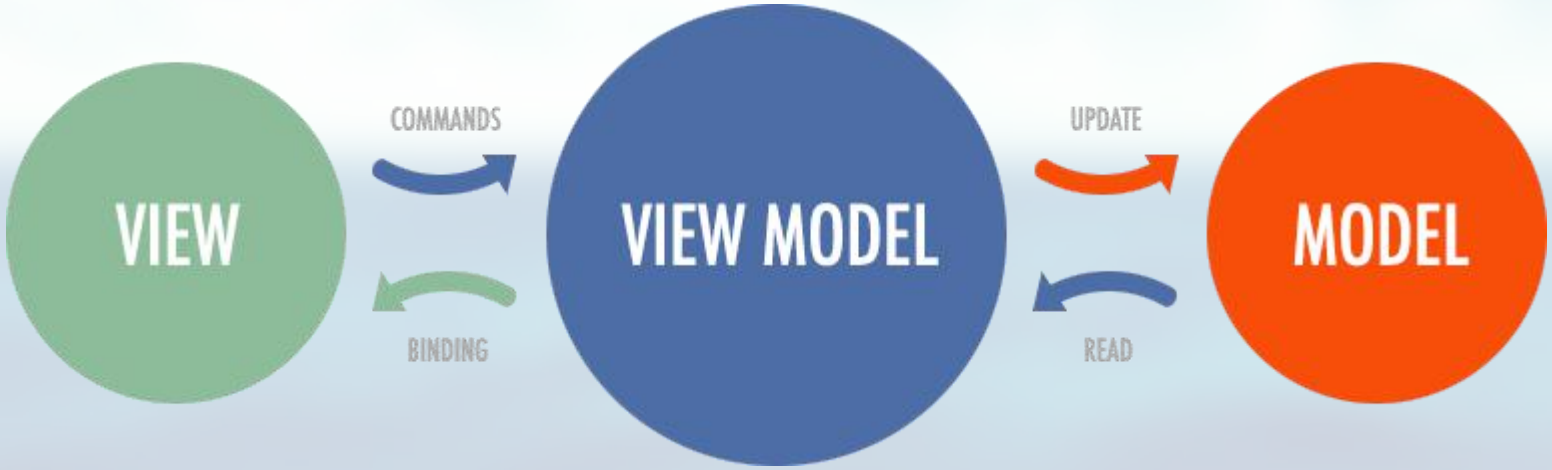
MVVM 模式

另一些框架提出 MVVM 模式，用 View Model 代替 Controller。

- Model
- View
- ***View-Model***：简化的 Controller，唯一作用就是为 View 提供处理好的数据，不含其他逻辑。

MVVM 模式

- 本质：view 绑定 view-model，视图与数据模型强耦合。数据的变化实时反映在 view 上，不需要手动处理。



Angular

- Google 公司推出的 Angular 是最流行的 MVVM 前端框架。
- 它的风格属于 HTML 语言的增强，核心概念是**双向绑定**。

示例：Angular 的双向绑定


```
<div ng-app="">
  <p>
    姓名 :
    <input
      type="text"
      ng-model="name"
      placeholder="在这里输入您的大名"
    >
  </p>
  <h1>你好，{{name}}</h1>
</div>
```

Vue

- Vue.js 是现在很热门的一种前端 MVVM 框架。
- 它的基本思想与 Angular 类似，但是用法更简单，而且引入了响应式编程的概念。

示例：Vue 的双向绑定

- Vue 的模板与数据，是双向绑定的。

Hello Vue!

HTML

```html
<div id="app">
  {{ message }}
</div>
```

JS

```js
var app = new Vue({
  el: '#app',
  data: {
    message: 'Hello Vue!'
  }
})
```

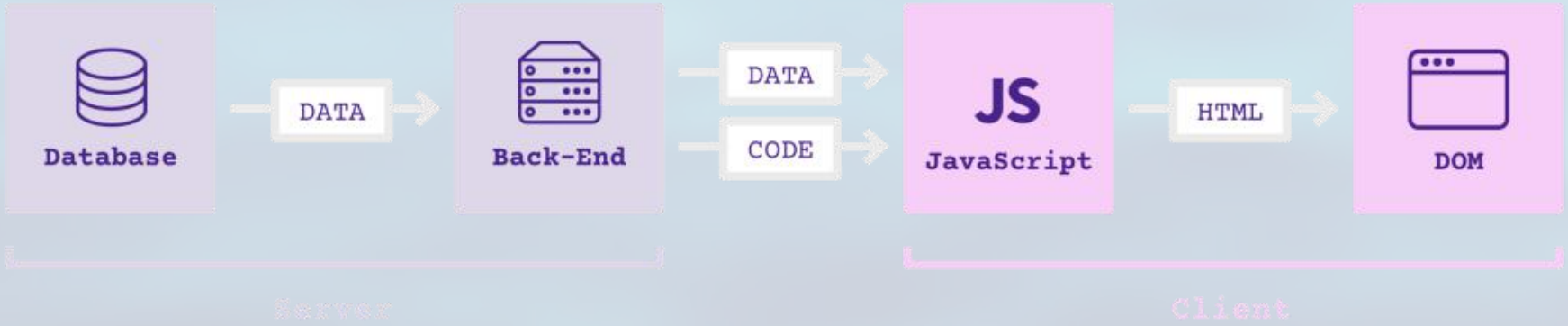
SPA

- SPA = Single-page application
  - 读写数据
  - 切换视图
  - 用户交互

- 2010年后，前端工程师从开发页面，变成了开发“前端应用”（跑在浏览器里面的应用程序）。

单页应用的架构

传统架构：



单页应用的架构：

前后端分离

- Ajax -> 前端应用兴起
- 智能手机 -> 多终端支持

这两个原因，导致前端开发方式发生根本的变化。

前端不再是后端 MVC 中的 V，而是单独的一层。

REST 接口

- Resource Representational State Transfer
- 前后端分离以后，它们之间通过接口通信。
- 后端暴露出接口，前端消费后端提供的数据。
- 后端接口一般是 REST 形式，前后端的通信协议一般是 HTTP。

Node

- 2009年，Node 项目诞生，它是服务器上的 JavaScript 运行环境。
- Node = JavaScript + 操作系统 API

Node 的意义

- JavaScript 成为服务器脚本语言，与 Python 和 Ruby 一样
- JavaScript 成为唯一的浏览器和服务器都支持的语言
- 前端工程师可以编写后端程序了

全栈工程师

前后端不分 -> 前后端分离 -> **全栈工程师**

- 传统前端技能：HTML、JavaScript、CSS
- 一门后端语言
- 移动端开发：iOS / Android / HTML5
- 其他技能：数据库、HTTP 等等

CONTENTS

02 前端知识框架

前端知识框架

前端知识框架

理论知识

理论知识

硬知识:

- http标准
- W3C标准
- ECMAScript标准

http标准

浏览器要从服务端获取网页，网页也可能将信息再提交给服务器，这其中都有http的连接。

- http请求的过程
- http状态码的意义
- http头部信息
- cookie状态管理
- 方法　Get Post
- https

《图解http》

W3C标准

- *html* html5
- *css* css3
- *javascript*
- json(一种轻量级的数据交换格式)
- xml(可扩展标记语言)
- websocket（是基于TCP的一种新的网络协议）
- .....

W3C标准 --HTML

- HTML:超文本标记语言。

```
<html>

<head>
 这里是文档的头部 ... ...
 ...
</head>

<body>
 这里是文档的主体 ... ...
 ...
</body>

</html>
```

W3C标准 --CSS

- CSS 规则由两个主要的部分构成：*选择器*，以及一条或多条*声明*。

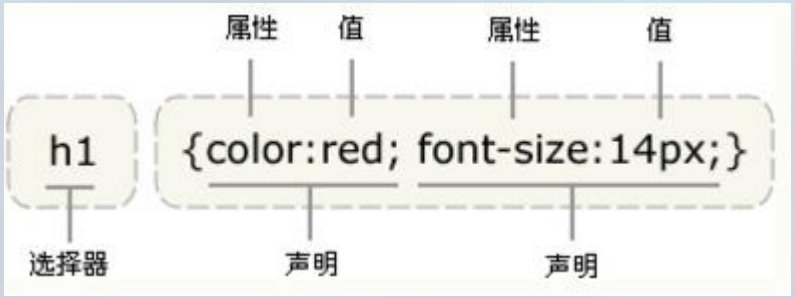
```
selector {declaration1; declaration2; ... declarationN }
```

- 每条***声明*** 由一个属性(property)和一个值(value)组成。

```
selector {property: value}
```

- 举个栗子：

```
h1 {color:red; font-size:14px;}
```



W3C标准 --JavaScript

- JavaScript是一种**动态类型**、弱类型、基于原型的脚本语言。

- 组成部分：
  - ECMAScript：描述了该语言的语法和基本对象。
  - 文档对象模型（DOM）：描述处理网页内容的方法和接口。
  - 浏览器对象模型（BOM）：描述与浏览器进行交互的方法和接口。

ECMAScript

简称 **ES**

- 语法
- 对象
- 原型链 继承
- 上下文环境
- 作用域 闭包
- 正则表达式
- 严格模式

前端知识框架

类库框架

框架和类库

- jQuery
- BootStrap
- requirejs seajs
- angular
- react
- ***vue***
- ...

前端知识框架

编码开发

编码开发 --开发工具

编辑器

- sublime
- webstorm
- ......

编码开发 --前端构建工具

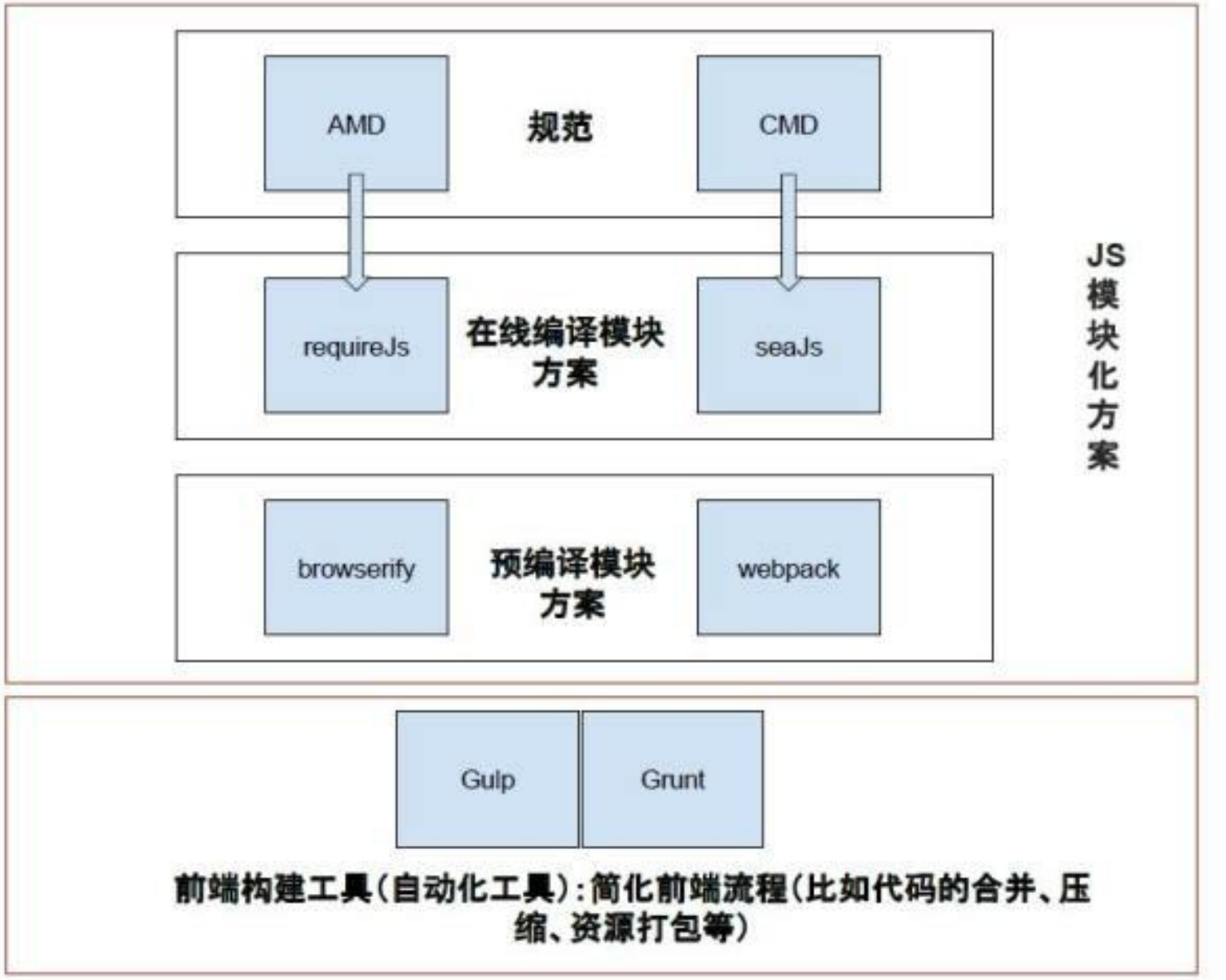
- gulp
- grunt

优化前端工作流程。比如自动刷新页面、combo、压缩css、js、编译less等等。简单来说，就是使用Gulp/Grunt，然后配置你需要的插件，就可以把以前需要手工做的事情让它帮你做了。

编码开发 --模块化方案

- seajs / require：是一种在线"**编译**" 模块的方案，相当于在页面上加载一个CMD/AMD 解释器。
- browserify / webpack：是一个**预编译**模块的方案，（不需要在浏览器中加载解释器）相比于上面，这个方案更加智能。

提升开发效率



Gulp/Grunt是一个**工具**，而Webpack、Browserify 等是**模块化方案**。

Gulp也可以配置seajs、requirejs甚至webpack的插件。

编码开发 --Css预处理器

what ?

- CSS 预处理器用一种专门的编程语言，进行 Web 页面样式设计，然后再编译成正常的 CSS 文件，以供项目使用。
- 为 CSS 增加一些编程的特性，无需考虑浏览器的兼容性问题，例如可以在 CSS 中使用变量、简单的逻辑程序、函数等在编程语言中的一些基本特性，可以让 CSS 更加简洁、适应性更强、可读性更佳，更易于代码的维护等。

编码开发 --Css预处理器

- Sass（SCSS）*注意：SASS依赖于Ruby，安装前必须先安装Ruby。*
- LESS
- Stylus

LESS的基本语法和CSS差不多，SASS和Stylus都可以利用缩进代替花括号，并且空格有重要的意义。SASS保存为".sass"是缩进格式，保存为".scss"是非缩进格式。SASS一般使用".scss"扩展名。LESS的扩展名为".less"，Stylus的扩展名为".styl"。

编码开发 --版本管理

- **Git** : 本地版本管理的机制. github、gitlab
- SVN : 远程中心的版本管理机制.

编码开发 --调试

- Chrome控制台
- firebug

编码开发 --生态系统

- npm
- bower

编码开发 --生态系统 NPM

NPM 是随同NodeJS一起安装的包管理工具，能解决NodeJS代码部署上的很多问题，常见的使用场景有以下几种：

- 允许用户从NPM服务器下载别人编写的第三方包到本地使用。
- 允许用户从NPM服务器下载并安装别人编写的命令行程序到本地使用。
- 允许用户将自己编写的包或命令行程序上传到NPM服务器供别人使用。

编码开发 --生态系统 Bower

- Bower是一个客户端技术的软件包管理器，它可用于搜索、安装和卸载如JavaScript、HTML、CSS之类的网络资源。其他一些建立在Bower基础之上的开发工具，如YeoMan和Grunt。

编码开发 --自动化测试

在业务较为稳定的情况下，可以通过自动化测试来减少测试的事件，但需求较多的时候，维护测试用例的成本会很高，可能用自动化测试会起到反效果。

- Mocha
- Karma
- Jasmine

前端知识框架

运行环境

运行环境

- 浏览器环境
- Node环境

运行环境 --浏览器环境

- 浏览器兼容性
- 响应式布局
- web安全
- 性能优化

运行环境 --浏览器环境 web安全

- 同源策略
- XSS跨站脚本攻击
- CSRF跨站点请求伪造
- 点击劫持
- SQL注入
- ...

运行环境 --浏览器环境 性能优化

- 压缩
- css sprites
- 合并 减少http请求
- 缓存
- CDN
- 避免重定向
- ...

Thanks for listening

路琳