Web 应用新架构研究

1. 前端架构

前端技术是集中在浏览器端的技术，最终使用的是 HTML、CSS和JavaScript。

1. HTML
   * 1. HTML： HTML是超文本标记语言，使用标签描述了网页的结构。HTML元素是HTML网页的基石，使用标签表示。这些标签标记了内容的片段，如标题使用 <h1 />, <h2>, …，而段落使用 <p />，表格使用 <table /> 等等。浏览器并不展示这些标签，而是使用这些标签来渲染页面的内容。
     2. HTML已经发展到HTML 5。HTML 5 的新特性：
        1. 最有趣的新元素是：
           1. 语义化的元素如 <header>, <footer>, <article>，以及 <section>



* + - * 1. 表单中新的属性如 number, date, time, calendar 和 range
        2. 图形元素如 <svg> 和 <canvas>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 元素 | https://www.w3schools.com/images/compatible_chrome.gif | https://www.w3schools.com/images/compatible_edge.gif | https://www.w3schools.com/images/compatible_firefox.gif | https://www.w3schools.com/images/compatible_safari.gif | https://www.w3schools.com/images/compatible_opera.gif |
| <canvas> | 4.0 | 9.0 | 2.0 | 3.1 | 9.0 |
| <svg> | 4.0 | 9.0 | 3.0 | 3.2 | 10.1 |

* + - * 1. 多媒体元素如 <audio> 和 <video>
      1. 新的 API：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| API | 说明 | https://www.w3schools.com/images/compatible_chrome.gif | https://www.w3schools.com/images/compatible_edge.gif | https://www.w3schools.com/images/compatible_firefox.gif | https://www.w3schools.com/images/compatible_safari.gif | https://www.w3schools.com/images/compatible_opera.gif |
| HTML Geolocation | 定位 | 5.0 - 49.0 (http)  50.0 (https) | 9.0 | 3.5 | 5.0 | 16.0 |
| HTML Drag and Drop |  | 4.0 | 9.0 | 3.5 | 6.0 | 12.0 |
| HTML Local Storage | Cookie 的强有力的替代品 | 4.0 | 8.0 | 3.5 | 4.0 | 11.5 |
| HTML Web Workers | 一个web worker是一个在后台运行的JavaScript程序，不会影响页面的性能。 | 4.0 | 10.0 | 3.5 | 4.0 | 11.5 |
| HTML SSE | Server-Sent Events | 6.0 | 不支持 | 6.0 | 5.0 | 11.5 |

* + - 1. 被删除的元素：

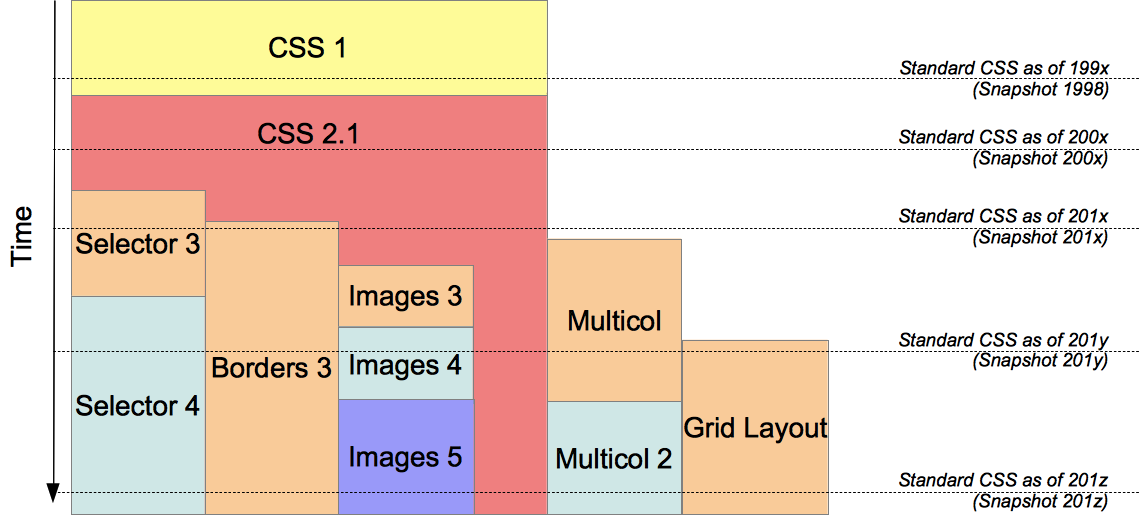
|  |  |
| --- | --- |
| 删除的元素 | 替代方案 |
| <acronym> | <abbr> |
| <applet> | <object> |
| <basefont> | CSS |
| <big> | CSS |
| <center> | CSS |
| <dir> | <ul> |
| <font> | CSS |
| <frame> |  |
| <frameset> |  |
| <noframes> |  |
| <strike> | CSS, <s>, 或者 <del> |
| <tt> | CSS |

* + 1. HTML简史：

|  |  |
| --- | --- |
| 年份 | 版本 |
| 1989 | Tim Berners-Lee 发明了 www |
| 1991 | Tim Berners-Lee 发明了 HTML |
| 1993 | Dave Raggett 起草了 HTML+ |
| 1995 | HTML工作组定义了 HTML 2.0 |
| 1997 | W3C Recommendation： HTML 3.2 |
| 1999 | W3C Recommendation： HTML 4.01 |
| 2000 | W3C Recommendation： XHTML 1.0 |
| 2008 | WHATWG HTML5 First Public Draft |
| 2012 | WHATWG HTML5 Living Standard |
| 2014 | W3C Recommendation： HTML5 |
| 2016 | W3C Candidate Recommendation：HTML5.1 |
| 2017 | W3C Recommendation：HTML5.1 第二版 |
| 2017 | W3C Recommendation: HTML5.2 |

* + 1. HTML5 浏览器支持情况：HTML5 能被所有现代浏览器支持。而且，不管是新的还是旧的浏览器，都会自动地将不认识的元素视为行内元素。
    2. 除了浏览器的原生支持的HTML，在后面会详细讲到有JavaScript框架中出现的新的标记语言：jsx，它看上去很类似HTML，但是需要由JavaScript来解析。它出现的原因是为了简化动态更新页面的工作，以及提高页面改变渲染呈现时的性能。

1. CSS
   1. CSS是一门用来描述HTML文档样式的语言，它定义了HTML元素该如何展示。
   2. CSS 已经发展到了CSS3：



* 1. CSS3 的新特性：
     1. 新的选择器：
        + [attr^=val]：匹配属性叫attr并且其值以val开头的DOM元素
        + [attr$=val]：匹配属性叫attr并且其值以val结尾的DOM元素
        + [attr\*=val]：匹配属性号attr并且其值包含val子串的DOM元素
     2. 圆角：可以使用border-radius来定义圆角，如

.box{ border: 2px solid orange; border-radius : 25px; width: 100px; padding: 10px; text-align:center; }

<div class="code"><div class="box">Submit</div></div>

产生如下效果：

http://webreference.com/authoring/css3/CSS3RoundedCorners.jpg

* + 1. border-image 图片边框：使用图片还代替边框
    2. box-shadow 边框阴影
    3. text-shadow 文本阴影
    4. CSS3 字体：通过CSS3可以指定任意字体，甚至是用户机器上并未安装的字体。
       - 只需要在网站上包含字体文件，当需要时它会自动下载。你只需要描述新的CSS3.0 @font-face规则。
       - 在@font-face定义里我们需要指定字体名称以及其URL。
       - 在CSS3.0里，设计师可以使用非网页安全的字体。
    5. CSS3 转场动画（Transition）：使用CSS3，不再需要JavaScript或者Flash来完成元素样式的转换。
       - 需要用CSS属性来指定应用哪种特效
       - 需要指定特效的时间。语法：transition: property duration
    6. CSS3 变形（Transforms）：我们可以用CSS3来拉伸、旋转、伸缩、移动和转动元素。元素可以被2D或者3D变形。

1. JavaScript:

JavaScript经常被缩写为JS，是一门高级解释型编程语言。也被归为动态语言、弱类型语言、基于原型的语言以及多范式的语言。JavaScript让网页具有交互性，所有是Web应用的核心部分。

随着JavaScript的发展，现在JavaScript不仅可以运行在浏览器端，也能运行在服务器端（NodeJs）。浏览器端JavaScript，曾经风靡一时的jQuery已经日渐式微，浏览器端JavaScript进化出三足鼎立的Angular、React以及Vue前端框架。除此之外，更多的开源框架层出不穷。

JavaScript简史：

ECMAScript 作为标准：由于商标问题，当作为标准时，使用了ECMAScript作为名称。术语ECMAScript也经常用来标记语言版本，如ECMAScript 5.

ECMA-262：

ECMA’s Technical Committee 39（TC39）

ECMAScript 3 （1999年12月）：这是今天多数浏览器支持的ECMAScript版本。它引入了很多新特性最终成为语言的内在部分：正则表达式、更好的字符串处理、新的控制语句、try/catch 异常处理、错误的紧凑定义、数字输出的格式化以及其他增强。

ECMAScript 4（于2008年7月废弃）

ECMAScript 5（2009年12月）

ECMAScript 6（2015年）：ES6 是下一代JavaScript，由ECMA Technical Committee 39管理ECMA规格。 ES6一般不能直接在浏览器中执行，所以需要转译器将ES6转译成ES5，比如babel就能从ES6产生出浏览器支持的JavaScript。新特性：

箭头函数

类

增强的对象字面量

模板字符串

解构

默认参数+不定数量参数+展开

定义变量的let和const

新的循环for..of表达式

生成器

Unicode

模块

模块加载器

新的数据结构：map+set+weakmap+weakest

代理

符号

Math、number、string、array、object的新API

二进制和八进制字面量

反射API

尾部调用：函数尾部的递归调用能够被保证不会无限增长以至于堆栈溢出。

ECMAScript 7（2016年）：TC39决定从2015年开始每年发布一个ECMAScript新版本，从而避免再次一下发布像ES6如此大的发动。新特性：

Array.prototype.includes()

指数操作符Exponentiation operator

ECMAScript 8（2017年），主要新特性：

异步函数

共享内存与原子操作

1. 目前流行的前端框架：

MVX = MVC + MVP + MVVM

* MVC：Model（模型）+View（视图）+Controller（控制器），主要是基于分层的目的，让彼此的责任分开。  
  View通过Controller来和Model联系，Controller是View和Model的协调者，View和Model不直接联系，基本联系都是单向的。用户通过Controller来操作Model从而达到View的变化。
* MVP：是从MVC模式演变而来，都是通过Controller/Presenter负责逻辑的处理+Model提供数据+View负责显示。  
  在MVP中，Presenter完全把View和Model进行了分离，主要的程序逻辑在Presenter里实现。并且，Presenter和View是没有直接关联的，是通过定义好的接口进行交互，从而使得在变更View的时候可以保持Presenter不变。
* MVVM：MVVM是把MVC里的Controller和MVP里的Presenter改成了ViewModel。Model+View+ViewModel。  
  View的变化会自动更新到ViewModel，ViewModel的变化也会自动同步到View上显示。这种自动同步是因为ViewModel里的属性实现了观察者模式，当属性变更时都能触发对应的操作。

常见框架对比：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 框架 | 模式 | 说明 | 特点 | 缺点 |
| Riot js | MVP |  |  |  |
| Angular | MVVM | 优秀的前端JS框架，已经被用于Google的多款产品当中 | 支持指令（内置、自定义）  支持过滤器（内置、自定义）  双向数据绑定  良好的应用程序结构  HTML模板  可嵌入、注入和测试  坚实的基础和社区支持 | 学习成本高  Watcher 越多，性能越差  不支持低端浏览器 |
| Vue | MVVM | 严格地说它不是一个框架，因为它只聚焦视图层，是一个构建数据驱动的Web界面库 | 轻量级  双向数据绑定  支持指令（内置、自定义（内置、自定义）  插件化  不支持低端浏览器  基于依赖追踪的观察并且使用异步队列更新，性能好  .vue 特殊文件格式需要编辑后使用  一切都是组件  支持混入  DOM 模板 | 新生儿，不如Angular成熟  影响度不大  不支持IE8 |
| React | V | 主要用于构建UI。 | 采用特殊的JSX语法，需要编译后使用  一切都是组件  插件化  支持混入  依赖Virtual DOM  声明式设计  高效  灵活  速度快  跨浏览器兼容  模块化  单向数据流  同构、纯粹的JavaScript | 只是一个V而已，不是一个完整的框架。  如果是大型项目，基本要加上ReactRouter和Flux等。 |
| Knockout js | MVVM |  |  |  |
| Ember js | MVVM |  |  |  |

1. 推荐的现代Web开发前端架构：

JAMstack：基于客户端JavaScript、可复用的API以及预编译好的标签。这是一种新型的构建网站与应用的方式，它交付更好的性能、更高的安全性、更低成本的扩展以及更好的开发体验。

1. 后端架构
   1. SOA（Service-Oriented Architecture）Hub（服务总线）
   2. MicroService（微服务）

微服务是一种架构风格，是为了满足现代业务需求而演化出的软件开发途径。它能提升组织敏捷度，交付速度和扩展能力。微服务给你一种开发更多的物理区隔的模块化应用的开发方式。

微服务与传统的面向服务架构有很多类似的地方。

在很多应用微服务的组织里，成功应用了著名的“分而治之”的技术，将它们的单一应用分区成更小的原子性单元。每一个单元执行单一职能——一个服务。这些组织解决了在单一应用时代碰到的大量的问题，于是越来越多的组织开始在重构他们的单一应用时应用此模式。后来这个模式被正式称为微服务架构。

* 1. BFF（Backend for Frontend）

1. 整体架构