



第一部分 *Part 1*

# 高效 Web 前端 开发综述



- 第1章 Web前端开发概述
  - 第2章 高效Web前端开发
- 



Chapter 1

第 1 章

# Web 前端开发概述

本章首先会简单介绍 Web 前端开发的历史由来，以及前端开发的概念及其范畴，让读者对前端开发有个整体的认识，同时还会对前端开发的一些错误认识进行更正。其次会介绍一下 Web 前端的现状，包括互联网整体环境、浏览器的发展、网站的设计和开发现状等，让读者认识到前端开发的现状和趋势，以便更好地提高自身的前端开发技能。

## 1.1 Web 前端开发的范畴

### 1.1.1 什么是 Web 前端开发

什么是 Web 前端开发？也许有人会说 Web 前端开发就是网页设计，就是美工，也有人会说 Web 前端的工作就是使用工具拖曳生成各种界面，然后导出为网页。其实这些都是对 Web 前端的误解，或者说还是停留在过去对前端开发的认识上。记得某个求职节目中，某互联网公司负责人对前来求职的前端工程师说他们理解的 Web 前端是需要用 Java 的，是需要用这种语言编写的。一个互联网公司的负责人竟然对自己公司使用的核心技术概念知之甚少，这实在是让人匪夷所思，更是闹了不小的笑话。此事件也激起了业内针对 Web 前端概念的大讨论，不过也从侧面说明 Web 前端这一概念还没有深入人心。的确，Web 前端兴起的时间还很短，这也就导致了很大一部分人不太了解这个新型的职业，所以我在这里再次解释一下 Web 前端这个词语的来龙去脉。

对于 Web 前端, 业内公认的说法是从 2005 年开始兴起的。2005 年以前, 因为 Web 网页主要以展示为主, 内容基本都是静态的, 所以客户端开发工作的目的就是让页面展现得更加整齐和漂亮, 没有太多花哨的内容。网站的用户也只是以浏览为主, 并不会会有复杂的交互。正因为如此, 一般的美工仅依靠 Photoshop 和 Dreamweaver 等工具就可以制作出外观漂亮的静态网页。2005 年之后, 互联网进入 Web 2.0 时代。Web 2.0 更注重用户的交互作用, 用户不再只是读者, 同时也是作者。用户不再是仅仅阅读静态的网页, 同时也为网站贡献内容。随着这一概念的发展, 人们开始重新审视网站的设计, 制作的网页慢慢变得生动起来, 页面也有了大量的交互, 不再是简单地展示静态的文字和图片。Google Gmail 的发布, 使得 AJAX 技术大红大紫, 这也把对 Web 2.0 的概念的认识推上了一个新的高度。AJAX 无刷新技术极大地增强了网页的用户体验, 使得用户操作页面更流畅, 操作体验更接近于本地应用。此外, 搜索引擎的普及使得网站的搜索引擎优化受到了重视。搜索引擎对网站的外观并不感兴趣, 只识别网站的 HTML 代码, 这就要求网站的设计者和开发者不仅要重视网站外在的用户体验, 还要重视网站内在的代码质量。

随着网站功能的丰富、设计风格的发展以及网站代码质量的要求, 网页端的开发也变得复杂起来, 其代码量和逻辑复杂度都增加不少, 同时还需要考虑网站的性能、浏览器兼容及网站安全性方面的问题。传统的网站开发者仅仅会使用网页制作工具已经不能够满足目前的需求了, 此时的网站开发更接近于后端开发, 需要有专门的软件开发工程师来做网站开发相关的工作, 于是原来的网页制作这一职业就演变成了 Web 前端开发。从职责上讲, Web 前端开发要涉及网站开发的方方面面, 从前端 UI 到和后端的数据交互都属于前端开发的范畴。因此, Web 前端开发是兼具艺术气息和逻辑思维的综合体, 既要考虑页面的美感和操作体验, 又要关注前端代码的质量。

### 1.1.2 Web 前端开发需要具备的技能

由于 Web 前端技术兴起的时间不长, 因此它还没有明确的界限定义, 不同的 Web 项目中可能要求的 Web 前端开发技术会有所不同。例如, 某些项目可能需要前端开发人员了解一些后端技术, 这样前端开发人员才可以更好地与后端开发人员配合, 如在页面上留下一些后端需要调用的“钩子”等, 而某些项目可能需要前端开发人员懂一些 UI 设计、Photoshop 工具的使用方法等, 以便于和 UI 设计师沟通和配合。虽然 Web 前端开发的范畴广泛, 并且界限模糊, 但是以下 7 点是 Web 前端开发必备的技能。

## 4 ❖ 第一部分 高效Web前端开发综述

### 1. 页面标记 (HTML)

由于页面 HTML 代码结构基本固定, HTML 的标签数量也不多, 因此, 从学习的难易程度来说, HTML 应该是前端技术中非常容易学习的技术。即使是一个新手, 也能在较短的时间里学会编写一个结构良好的页面。虽然说入门容易, 但是要编写语义良好、简洁整齐的 HTML 代码则需要大量的实践才能掌握。HTML 是页面的基本结构组成部分, 是网站的基础, 臃肿混乱的 HTML 代码不但会影响其本身的展现, 而且与其对应的 CSS 与 JavaScript 代码也会变得难以编写和维护。

### 2. 页面样式

CSS 是 Cascading Style Sheet (层叠样式表) 的简称。在标准页面设计中, 因为 CSS 负责网页内容的表现, 所以 CSS 也是前端开发需要掌握的核心内容之一。丰富的 CSS 样式能让平淡的 HTML 展现出绚丽的效果, 使得页面更为友好。好的样式可以让用户在页面上停留的时间更久一些, 也可以帮助用户更好地阅读网站内容, 同时, 还可以让用户在不同浏览器上有着相同的体验。CSS 和 HTML 代码一样, 没有复杂的逻辑, 上手也比较容易, 其主要的难点在于如何合理地利用 CSS 的组合和继承特性来编写简洁、可维护性好的 CSS 代码。以上这两项基本技能是前端 UI 开发必备的技能。

### 3. 前端编程

前端编程技能主要是指 JavaScript 编程。JavaScript 是一种基于对象和事件驱动的客户端脚本语言, 是页面实时动态交互的技术基础。相较于 HTML 和 CSS, 编写 JavaScript 代码更能让前端开发人员找到后端程序员的感觉。JavaScript 是非常灵活的脚本语言, 包含了高阶函数、动态类型以及灵活的对象模型等强大的语言特性, 当然, JavaScript 的灵活性也可能导致代码不易维护。此外, 浏览器的兼容性也增加了 JavaScript 编码的难度。同一个功能, 可能在不同的浏览器中有不同的实现。例如, 在 IE 浏览器中, 事件绑定使用的是 `attachEvent()` 方法, 但其他浏览器则使用的是 `addEventListener()` 方法。开发人员在熟悉 JavaScript 基本语法和基本的编码规范之外, 还应该了解并解决在不同浏览器中的 JavaScript 的兼容性问题。另外, 作为前端开发工程师, 必定会涉及后端的编程, 一些原因是目前流行的 Web 编程方式会有部分后端代码存在于前端页面中, 和前端的 HTML、JavaScript 等混合在一起, 如 PHP、JSP、ASP.NET 等, 因此, 前端开发工程师也有必要了解一些后端编程技术。

### 4. 跨平台, 跨浏览器

前端代码本来不存在跨平台方面的问题, 但是随着移动 Web 平台的兴起, 跨

平台的问题就逐渐显现出来了。移动设备（如智能手机和平板电脑）在近几年发展迅猛，用户通过移动设备访问 Web 站点的比率也是逐年增高。如何在众多移动平台、众多屏幕尺寸上展现友好的 Web 站点成为一项前端技能。不过，目前跨浏览器没有像几年前表现得那么突出了，这主要是因为 IE 6、IE 7 浏览器的占有率下降和众多浏览器对标准的重视，另外，目前流行的前端框架已经很好地解决了浏览器的兼容问题。尽管如此，但是还需要熟悉常见的浏览器兼容方法，主要包括：IE 7、IE 8 的兼容，HTML5 中新特性的兼容等。

## 5. 前端框架

各种前端框架的出现，在很大程度上降低了前端开发的难度。框架统一了编码的方式，封装了浏览器兼容问题并添加大量的扩展功能。如今的 Web 项目中前端框架应用非常广泛，在开源社区 GitHub 上排名靠前的开源框架也是以前端框架居多。优秀的前端框架可以在很大程度上缩短项目开发的周期，尤其是 jQuery，几乎成为 Web 项目默认的前端框架。但是，前端框架的接口众多，各种框架的使用方式和编码方式也不尽相同，作为前端开发工程师，需要熟悉一些常用框架的使用方法，并且要了解如何编写常用框架的扩展插件，如 jQuery、YUI、Ext JS 等。

## 6. 调试工具

对于前端代码，在调试过程中需要查看页面的 HTML 结构变化、CSS 渲染效果、JavaScript 代码的执行情况以及 HTTP 请求和返回的数据，并且要了解网站各个部分的性能等，甚至需要动态更改 HTML、CSS 代码来查看预期的效果，模拟发起 HTTP 请求来查看后端返回的数据。主流浏览器都会有对应的浏览器插件来辅助完成这些工作，如 IE 中的 IE Dev Toolbar、Chrome 中的 Developer Tools、Firefox 中的 Firebug 等，此外还有 HTTP 请求监控和模拟工具，如 Fiddler 等。开发工程师需要熟练使用这些工具来辅助完成前端代码的调试。

## 7. 沟通能力

沟通是开发人员必备的一项基本技能，尤其是对于前端开发工程师来说。Web 前端开发介于 UI 和后端逻辑开发之间，因此，Web 前端工程师在开发过程中必定会和 UI 设计师及后端工程师合作：前端工程师需要和 UI 设计师沟通，确定效果是否可以实现以及实现的代价，并对 UI 设计提出建议；还需要和后端工程师沟通，确定前后端交互的接口以及传输的数据实体的结构等，良好的沟通会让这些过程变得轻松许多。



## 1.2 Web 前端开发现状

前端开发虽然起步时间晚，但是发展势头迅猛，在各种新技术、新标准的推动下，各大互联网公司也开始重视 Web 产品的前端重构与开发，如淘宝、腾讯、新浪、百度、搜狐等都对自己的网站进行了重构并同时使用了 HTML5 中的新特性。现在开发的 Web 新产品的页面交互也越来越丰富，视觉效果也越来越绚丽。互联网公司除了重视前端开发之外，还在积极推动互联网技术的发展，2001 ~ 2012 年，腾讯、百度及奇虎 360 公司先后加入了万维网联盟（W3C），共同参与互联网技术标准的研究和制定，促进了国内互联网产业的发展。

在 Web 前端发展的过程中，浏览器的发展起着至关重要的作用。浏览器厂商在这场 Web 浪潮中表现突出，具体表现在浏览器的版本升级周期短、对前端标准的支持也越来越好、浏览器的性能也有大幅的提升等。IE 浏览器也开始发力，最新发布的 IE 10 浏览器无论是标准的支持还是性能都表现突出。浏览器的快速发展使得低端的浏览器如 IE 6、IE 7 等快速退出市场。

下面看一下全球范围内浏览器的占有率情况。根据 StatCounter 的统计，2012 年 10 月份，全球浏览器占有率前 5 位的分别是 Chrome、IE、Firefox、Safari 及 Opera，占有率分别为 34.77%、32.08%、22.32%、7.81% 和 1.63%。按照浏览器的版本统计，在低端浏览器中，IE 8 占有率为 12%，IE 7 的占有率仅为 0.88%，IE 6 的占有率则更低。

此外，移动设备上的浏览器在整个浏览器占有率中上升速度很快。根据 StatCounter 的统计，Safari iPad 在浏览器的占有率已经达到了 2.76%。目前移动设备的普及率极高，用户已经逐渐习惯于通过移动设备来浏览网页。主流的移动平台主要为 Android 和 iOS 系统，对应的浏览器都是以 WebKit 为核心的，性能和对标准的支持都不错。

以上介绍的是浏览器在全球范围内的占有率情况。从上面的统计可以看出，在全球范围内，低端浏览器的占有率已经很低了，但国内浏览器占有率的状况则比较复杂，低端浏览器如 IE 8、IE 7 和 IE 6 等还占有不小的份额。根据 StatCounter 的统计，2012 年 10 月份，国内浏览器占有率前 5 位分别为 IE 8、IE 9、IE 6、搜狗浏览器和 360 安全浏览器，占有率分别为 48.7%、14.09%、8.29%、5.74% 和 2.72%。其中低端浏览器 IE 8、IE 7 和 IE 6 的总占有率大约为 58%，低端浏览器依然占有“半壁江山”，而高级浏览器如 Chrome、Safari、Firefox 等占有率则很低。值得称道的是，360 安全浏览器和搜狗浏览器这两款浏览器都是基于开源的浏览器内核，对标准支持较好，性能表现突出。这两款浏览器的占有率目前虽然不高，但发展势头迅猛，是推动老旧浏览器退出市场的中坚力量。

除各大互联网公司和浏览器厂商的积极推动外，Web 前端开源社区的参与热

情也极为高涨。在著名的网络代码托管系统 GitHub 上，开源的项目仍然是以前端相关项目居多，按语言排名，前 3 位分别为 JavaScript、Ruby 和 Python；按项目排名，靠前的大部分也是前端项目，如 Prototype、Bootstrap、jQuery 及相关插件等，具体内容可参考 GitHub 官方网站。国内的一些前端社区则发展较为缓慢，多是以推广 HTML5 为主，靠一些酷炫的效果来吸引眼球，反而关注前端基础的较少。国内 Web 互联网公司（如淘宝、百度、腾讯等）也有部分优秀的开源前端框架，这些公司的前端技术分享氛围不错。整体来讲，国内前端技术发展还是比较迅速的，也不乏顶级的前端技术人员，但普遍缺乏的是开放的心态和与国内外同行交流的能力，国内前端技术的发展任重道远。

Web 前端整体技术的发展和前端工程师个人的能力是相辅相成的。目前，前端工程师很多是“半道出家”，一部分是从页面 UI 开发转行为 Web 前端开发的，一部分则是由后端工程师转行而来，所以前端工程师普遍自学成才，并没有受过足够的专业训练，也缺乏实际的项目经验。

除了以上这些 Web 前端开发的外在环境之外，Web 前端在技术方面也存在着大量的挑战，大量旧的网站需要重构来提高网站用户体验和性能等。这些网站的前端代码普遍存在的问题有：代码组织混乱，CSS 代码和 JavaScript 代码混合在 HTML 代码中；代码的格式问题突出，不够整洁；页面布局随意，HTML 代码不符合标准；网站整体性能差，还没有意识到要去应用诸如缓存、动态加载、脚本压缩、图片压缩等提高性能的技术。

此种状况并非是个案，查看目前访问量颇高的某网站的首页源代码，可以很直观地看出其中不规范的内容：页面没有 body 的闭合标签，页面代码缩进随意，页面中大量使用内联样式并且页面中还在使用诸如 <marquee>、<font> 等标准不推荐的标签。以下是其中的代码片段：

```
<marquee width="260" ONMOUSEOUT="this.start()"
ONMOUSEOVER="this.stop()" scrollamount=3 scrolldelay=100 >
    <font color=blue style="font-size:14px;">
    <a href="http://www.china-pub.com" target="_blank"><font color=blue
style="font-size:14px;">
    </font></a>
</marquee>
```

在这段代码中，<marquee> 和 <font> 标签已经不再被 W3C 制定的标准推荐使用，<marquee> 所呈现的效果完全可以通过 JavaScript 代码实现，并且实现的效果可以更好。<font> 标签完全可以通过 CSS 样式代码实现。

此外，此站点首页 HTML 代码中包含大量不必要的标签和多余的样式设置，代码的可读性较差。再来看一段代码：

## 8 ❖ 第一部分 高效Web前端开发综述

```
<!--第一段代码片段 -->
<div id="slideshow">
  <div class="current">
</div>
  <div></div>
  <div></div>
</div>
<!--第二段代码片段 -->
<div><br/>
  <a href="./gljd/gywm/">关于我们</a> | <a href="./gljd/wzls/">网站
声明</a>
  <br />
  版权所有 &copy; 2004-2012 北京博彦科技股份有限公司 中国最佳IT外包企业 <br /><a href=
"http://www.beyon.com.cn/" target="_blank">北京博彦科技</a>
</div>
```

在上面的代码段中，第一段代码中的 `<img>` 标签外的 `<div>` 标签是可以省略的，`<img>` 标签内的 `width` 和 `height` 属性可以统一通过 CSS 代码设置，`<img>` 标签内需要添加必要的 `alt` 属性来说明图片的信息。在第二段代码中，依靠 `<br>` 标签来增大距离是不合语义的做法，应该通过 CSS 样式来设置。因此，以上两段 HTML 代码至少可以简化为：

```
<!--第一段代码片段 -->
<div id="slideshow">
  
  
  
</div>
<!--第二段代码片段 -->
<div>
  <a href="./gljd/gywm/">关于我们</a> | <a href="./gljd/wzls/">网站声
明</a>
  版权所有 &copy; 2004-2012 北京博彦科技中 国最佳IT外包企业
  <a href="http://www.beyon.com.cn/" target="_blank">北京博彦科技
</a>
</div>
```

另外，站点中的 CSS 样式文件和 JavaScript 脚本文件，没有任何文件被压缩和合并。推荐的做法是分别压缩合并样式文件和脚本文件，在缩小文件大小的同时也减少了文件的 HTTP 请求次数，提高了性能。

综合以上的这些信息，Web 前端开发的现状可以概括为：前端技术发展迅速，但起步较晚，基础薄弱；前端工程师热情高涨，但缺乏足够技能培训，对代码规范重视不足，对一些基础原理的理解不够深刻。



### 1.3 规范的 Web 前端代码：更易维护、更高性能和更安全

规范的代码，这是所有软件开发中对代码的基本要求，前端开发也是一样的，要求编写规范的 HTML、CSS 和 JavaScript 代码。

什么样的前端代码才能称得上规范的代码？探讨这个问题之前，首先需要强调的是规范不是标准，不是放之四海而皆准的，不同的项目中的代码规范是有可能有差异的，比如命名，有些项目规定 HTML 标签的 id 必须要以控件的缩写名作为前缀，如按钮的 id 名以“btn”作为前缀，有些只是规定命名有意义就可以。再比如有些项目规定 JavaScript 代码语句结尾必须添加分号，但是有些项目并不要求，大名鼎鼎的 Bootstrap 源代码中并没有给 JavaScript 语句结尾添加分号。因此，规范的作用只是让同一个团队代码风格统一，减少协作时的复杂性，确保后续的维护和修改方便。不同团队中遵循的规范有可能存在部分差异，但是在同一个团队中规范必须是统一的，团队中的成员应该严格遵循。澄清了规范的作用后，接着上面的问题来探讨什么才是规范的前端代码。

Web 前端的代码规范主要针对的是 HTML、CSS 和 JavaScript 代码。尽管前端代码规范在不同场合会有差异，但是规范的前端代码应该具有如下特征。

#### (1) 符合标准

所谓的标准指的是 W3C 制定的 Web 标准，这是从百度百科找到的关于 W3C 的介绍：

---

万维网联盟（World Wide Web Consortium, W3C），又称 W3C 理事会。1994 年 10 月在麻省理工学院计算机科学实验室成立。建立者是万维网的发明者蒂姆·伯纳斯-李。

万维网联盟是国际著名的标准化组织。1994 年成立后，至今已发布近百项相关万维网的标准，对万维网发展做出了杰出的贡献。目前，万维网联盟拥有来自全世界 40 个国家或地区的 400 多个会员组织，已在全世界 16 个地区设立了办事处。2006 年 4 月 28 日，万维网联盟在中国内地设立首个办事处。

---

W3C 制定的标准包括使用语言的规范、开发中使用的原则和解释引擎行为等，主要由 3 个部分组成：结构（Structure）标准、表现（Presentation）标准和行为（Behavior）标准。结构标准包括 XML 标准、XHTML 标准和 HTML 标准，目前使用的标准版是 HTML 4.01 标准，HTML5 是 HTML 和 XHTML 的最新标准，还没有发布最终版；表现标准主要指的是 CSS 样式标准，目前使用的标准版是 CSS2，CSS3 尚未发布最终版；行为标准主要包括 ECMAScript 标准和 DOM 标准。ECMAScript 是 ECMA（European Computer Manufacturers Association）制定的标准脚本语言（JavaScript）。DOM 是文档对象模型（Document Object Model）的缩写，是一个中立于语言的应用程序接口，允许程序访问并更改页面的内容、结构

## 10 ❖ 第一部分 高效Web前端开发综述

和样式。目前推荐遵循的 ECMAScript 标准是 ECMAScript 262 第 5 版, DOM 标准是 DOM 级别 2, DOM 级别 3 目前还没有发布正式版本。W3C 推荐的这些标准受到了各浏览器厂商和 IT 互联网公司的欢迎, 前端代码遵循标准则可以保证网页在不同浏览器上正常展现。

### (2) 格式规范统一

前端代码的格式主要包括命名、代码缩进、空格和空行的使用以及代码注释。命名主要有 HTML 元素的 id 和 class 名, JavaScript 中函数和变量的命名; HTML、CSS 和 JavaScript 代码中都需要通过代码的缩进来体现代码的层次关系; 空格和空行主要用在 CSS 和 JavaScript 代码中, 用来提高代码可读性, 如操作符前后添加空格、不同代码段逻辑之间添加空行等; CSS 和 JavaScript 代码中都需要添加必要的注释来解释说明代码文件及代码段, HTML 代码中使用注释的情况较少。和其他代码格式规范一样, 前端代码格式规范也是为了提高代码的可读性和可维护性。

### (3) 高性能

前端性能体现在浏览器的响应速度上, 包括网页的加载速度和页面的交互响应速度。网页加载所占用的时间包括后端处理请求的时间、代码文件从服务器端传输的时间、HTML 和 CSS 的组合展现的时间以及 JavaScript 加载和运行的时间。除了第一条, 其余都和前端代码有直接的关系, 减少文件传输时间的最直接的方式就是减小文件的大小, 越少的代码文件相对传输就会更快; 简洁和符合标准的 HTML 和 CSS 代码能减少浏览器解析的时间, 加快浏览器渲染过程; 页面中请求数量越少相对页面加载时间也会越快; 在 JavaScript 代码中选择性能更好的实现方案, 如延迟加载、动态加载等技术, 会让页面加载更快和交互更流畅。规范的前端代码应该针对这些方面来编写高性能的前端代码, 提高用户的前端体验。

### (4) 高安全性

网站的安全有时很难引起一些互联网公司的足够重视, 他们更看重的是站点的用户体验、性能等这些更直观的方面。2011 年, 多个网站用户信息泄露风波震惊整个互联网界, 网站安全也再次引起业内的重视。从技术上讲, 网站的安全瓶颈主要在后端, 但是随着前端技术的发展, 富客户端应用越来越多, 前端安全问题也随之增多, 如跨站点攻击、Cookie 劫持等。这些攻击通过设置 JavaScript 变量、HTML 标签的值和属性、CSS 属性值等方式伪造恶意代码来达到攻击的目的, 因此, 规范的前端代码至少要针对这些方面做必要的安全校验和编码, 提高代码的安全性。

前端代码如果能遵守如上的几个规范点, 则基本上能称为高质量的代码。需要强调的是, 酷炫的技术只是“浮云”, 良好的编码习惯和意识才是真功夫。本书后续的章节将会针对如上的代码规范点展开详细讨论。