

高职高专“十二五”规划教材

Web 前端开发基础

谢钟扬 郑志武◎编著



重庆大学出版社
<http://www.cqup.com.cn>

内容提要

本书主要介绍了 Web 前端开发的现状;HTML 相关的基础知识,包括 HTML 的基本标签、表格、表单的设计和制作;CSS 相关的基础知识,包括 CSS 的基本属性、CSS 的选择器、CSS 的定位机制、CSS 的框模型;JavaScript 的基础知识,包括 JavaScript 的基础语法、对象、数组、浏览器对象、HTML 对象;JQuery 的基础知识,包括 JQuery 的概念、JQuery 的选择器、JQuery 对象的主要方法和使用方式。

本书可作为高职高专软件类专业的教材或教学参考书,也可供其他相关专业和技术人员参考。

图书在版编目(CIP)数据

Web 前端开发基础/谢钟扬,郑志武编著. —重庆:重庆大学出版社,2016.6

ISBN 978-7-5624-9810-0

I. ①W… II. ①谢… ②郑… III. ①超文本标记语言—程序设计—高等职业教育—教材②网页制作工具—程序设计—高等职业教育—教材 IV. ①TP312②TP393.092

中国版本图书馆 CIP 数据核字(2016)第 112679 号

高职高专软件技术专业系列规划教材

Web 前端开发基础

谢钟扬 郑志武 编 著

策划编辑:曾显跃

责任编辑:文 鹏 版式设计:曾显跃

责任校对:张红梅 责任印制:赵 晟

*

重庆大学出版社出版发行

出版人:易树平

社址:重庆市沙坪坝区大学城西路 21 号

邮编:401331

电话:(023)88617190 88617185(中小学)

传真:(023)88617186 88617166

网址:<http://www.cqup.com.cn>

邮箱:fxk@cqup.com.cn(营销中心)

全国新华书店经销

重庆升光电力印务有限公司印刷

*

开本:787mm×1092mm 1/16 印张:15.75 字数:393 千

2016 年 6 月第 1 版 2016 年 6 月第 1 次印刷

印数:1—2 000

ISBN 978-7-5624-9810-0 定价:33.00 元

本书如有印刷、装订等质量问题,本社负责调换

版权所有,请勿擅自翻印和用本书
制作各类出版物及配套用书,违者必究

前言

随着互联网越来越深地融入到社会生活的方方面面,其所涉及的功能、需求日益复杂,面向的用户群体日益广泛,对 Web 前端的复杂程度、用户友好性提出了巨大的挑战;另一方面,近年来移动互联网的发展,也让 Web 前端开发需要兼顾符合移动设备的使用习惯和功能要求。所以,无论是业务上的发展,还是技术上的进步,都对 Web 前端开发形成了巨大的挑战。Web 前端技术面对这个挑战,也通过不断完善自身、发展新的内容来应对挑战,抓住了一个高速发展的机遇。随着 HTML5、CSS3 标准的更新,JQuery 框架的推出以及随之而来的大量技术框架和解决方案的涌现,Web 前端技术领域在近年来呈现出井喷式的发展势头。

Web 前端开发入门难度并不高,但是初学者如果没有一个很好的学习和编码习惯,则开发水平的提高速度会变得很慢。本书综合了笔者多年来积累的各种 Web 前端开发经验以及各种高效的 Web 前端开发实践,详细介绍了 Web 前端开发所需的核心知识和实用的解决方案,力图用简明扼要的语言、翔实具体的实例,让读者从原理上理解和掌握参与 Web 前端开发所需的技术。

本书以 HTML、CSS、JavaScript 和 JQuery 公开发布的技术标准和文档为基础,结合实际开发和教育教学过程中的实践积累编纂而成。

由于作者水平有限,书中难免会出现一些错误或者不准确的地方,恳请读者批评指正。

编者
2016年1月

目 录

第 1 章 前端开发概述	1
1.1 前端开发综述	1
1.1.1 页面标记(HTML)	1
1.1.2 页面样式	1
1.1.3 前端编程	2
1.1.4 跨平台,跨浏览器	2
1.1.5 前端框架	2
1.1.6 调试工具	2
1.1.7 沟通能力	3
1.2 界面布局	3
1.2.1 页面尺寸	3
1.2.2 整体造型	3
1.2.3 页头	4
1.2.4 文本	4
1.2.5 图片	4
1.2.6 多媒体	4
1.2.7 页脚	4
1.3 界面配色	5
1.3.1 网页界面配色的重要性	5
1.3.2 网页界面配色的一般原理	5
第 2 章 HTML5	7
2.1 HTML 语言的语法特点	8
2.2 HTML DOM	9
2.3 基本标签	12
2.3.1 HTML5 网页结构	12
2.3.2 HTML5 页面规划	14
2.3.3 HTML5 基本标签	24

2.4	列表标签	35
2.4.1	有序列表	36
2.4.2	无序列表	36
2.4.3	自定义列表	38
2.5	表格标签	39
2.5.1	定义表格的基本语法	39
2.5.2	表格 <table> 标签的属性	39
2.5.3	表格的边框显示状态 frame	40
2.5.4	设置分隔线的显示状态 rules	40
2.5.5	表格行的设定	41
2.5.6	单元格的设定	41
2.5.7	表格的表头	42
2.5.8	表格中的空单元格	42
2.5.9	跨行与跨列的单元格	43
2.5.10	表格分组标记	46
2.6	表单标签	46
2.6.1	form 标签	46
2.6.2	input 标签	46
2.6.3	select 标签	52
2.6.4	textarea 标签	52
2.6.5	HTML5 新增表单元素	54
2.6.6	HTML5 新增表单属性	55
2.7	框架标签	59
2.7.1	基本结构	59
2.7.2	<Frameset> 标签	59
2.7.3	<frame> 标签	60
2.7.4	<noframes> 标签	61
2.7.5	iframe 语法和用法	64
2.7.6	透明 iframe	65
2.8	多媒体	65
2.8.1	HTML 5 视频	65
2.8.2	HTML 5 音频	67
2.9	画布	68
2.9.1	canvas 的属性和方法	69
2.9.2	绘制线条	71
2.9.3	绘制圆形	72
2.9.4	绘制渐变	73
2.9.5	绘制图像	74

第3章 CSS3	75
3.1 CSS3 简介	75
3.1.1 CSS3 的更新	75
3.1.2 CSS 工作原理	76
3.1.3 CSS 规则	77
3.2 样式添加方式	78
3.2.1 行内样式	78
3.2.2 内嵌样式	79
3.2.3 外部样式	79
3.3 CSS3 选择器	81
3.3.1 基本选择器	82
3.3.2 属性选择器	86
3.3.3 伪类选择器	91
3.4 盒子模型	97
3.5 CSS 样式属性	101
3.5.1 布局与定位属性	101
3.5.2 文本属性	110
3.5.3 字体属性	116
3.5.4 背景属性	119
3.5.5 列表属性	122
3.5.6 边界属性	125
3.5.7 边框属性	127
3.5.8 浮动和清除	129
3.5.9 显示属性	135
3.6 动画及 3D 转换	135
3.6.1 CSS3 @keyframes 规则	135
3.6.2 创建 CSS3 动画	137
3.6.3 CSS3 动画属性	139
3.6.4 3D 转换	141
第4章 JavaScript	143
4.1 JavaScript 简介	143
4.2 JavaScript 语法基础	144
4.2.1 添加 JavaScript 脚本	144
4.2.2 变量	146
4.2.3 数据类型	147
4.2.4 运算符	147
4.2.5 注释	149
4.3 流程语句	149

4.3.1	条件语句	149
4.3.2	循环语句	150
4.3.3	break 和 continue	151
4.4	函数与事件	153
4.4.1	函数(Function)	153
4.4.2	事件(Event)	155
4.4.3	事件与函数的关联	160
4.5	内置对象	161
4.5.1	数组对象(Array)	161
4.5.2	字符串对象(String)	167
4.5.3	数学对象(Math)	172
4.5.4	日期对象(Date)	173
4.6	浏览器对象模型(BOM)	174
4.6.1	Window 对象	175
4.6.2	Navigator 对象	180
4.6.3	Screen 对象	180
4.6.4	History 对象	181
4.6.5	Location 对象	182
4.7	JavaScript DOM 对象	183
4.7.1	document 对象概述	184
4.7.2	DOM 节点操作	185
4.8	异常处理	188
4.8.1	利用 try-catch-finally 处理异常	188
4.8.2	利用 throw 处理异常	189
第 5 章 JQuery		190
5.1	JQuery 简介	190
5.1.1	JQuery 的优势	190
5.1.2	JQuery 库类型说明	192
5.1.3	JQuery 环境配置	192
5.1.4	JQuery 语法	192
5.1.5	第一个 JQuery 程序	193
5.2	选择器	194
5.2.1	选择器的定义	194
5.2.2	基本选择器	194
5.2.3	使用 JQuery 基本选择器选择元素	194
5.2.4	层次选择器	196
5.2.5	过滤选择器	196
5.2.6	表单选择器	199

5.2.7 选择器实例	199
5.3 DOM 节点操作	205
5.3.1 元素节点操作	207
5.3.2 元素属性操作	212
5.3.3 元素内容操作	214
5.3.4 获取或设置元素值	215
5.3.5 遍历节点	215
5.3.6 元素样式操作	217
5.3.7 DOM 节点操作实例	217
5.4 事件与动画	220
5.4.1 事件机制	221
5.4.2 页面载入事件	221
5.4.3 绑定事件	222
5.4.4 hover()方法	222
5.4.5 toggle()方法	223
5.4.6 事件对象的属性	223
5.4.7 移除事件	224
5.4.8 JQuery 动画	225
5.5 JQuery Ajax 编程	227
5.5.1 Ajax 的优势	228
5.5.2 Ajax 的 XMLHttpRequest 对象	228
5.5.3 JQuery 中的 load()方法	228
5.5.4 全局函数 getScript()	229
5.5.5 \$.get()方法和 \$.post()方法	230
5.5.6 \$.ajax()方法	232
5.5.7 JQuery 中的 Ajax 全局事件	237
参考文献	239

第 **I** 章

前端开发概述

1.1 前端开发综述

由于 Web 前端技术兴起的时间不长,因此对其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。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能更好地与后端开发人员配合,比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。尽管 Web 前端开发的范畴广泛,并且界限模糊,但是以下 7 个方面则是 Web 前端开发的主要内容。

1.1.1 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,所以从学习的难易程度来说,HTML 应该是前端技术中非常容易上手的技术。即使是一个新手也能在较短的时间里学会并编写出一个结构良好的页面来。虽说入门容易,但是要编写出语义良好、简洁整齐的 HTML 代码则需要大量的实践学习才能掌握。HTML 是页面的基本结构组成,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 和 JavaScript 代码也会变得难以编写和维护。

1.1.2 页面样式

CSS 是 Cascading Style Sheets(层叠样式表)的简称。在标准页面设计中,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。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 开发必备技能。

1.1.3 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,而其他浏览器使用的是 `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,还应该掌握不同浏览器中 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,比如 PHP、JSP、ASP.net 等,所以前端工程师也有必要了解一些后端编程技术。

1.1.4 跨平台,跨浏览器

前端代码本来是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 站点成为了一项前端技能。不过,目前跨浏览器没有像几年前表现得那么突出了,这要归功于 IE6、IE7 浏览器的占有率快速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处理了浏览器的兼容问题。尽管如此,但还是需要熟悉常见的浏览器兼容方法,主要包括 IE7、IE8 的兼容,HTML5 中新特性的兼容等。

1.1.5 前端框架

各种前端框架的出现,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减少项目开发的周期,尤其是 JQuery,几乎成为了 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、ExtJs 等。

1.1.6 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 代码的执行情况及 HTTP 请求和返回的数据,并且还要查看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。各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。

1.1.7 沟通能力

软件工程师向来是一个不善于沟通的群体,但是沟通却又是一项必备的基本技能,尤其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,所以 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。良好的沟通能力会让这些过程变得轻松许多。

1.2 界面布局

刚接触网页设计时,初学者应该了解大多数访问者浏览网站的习惯,因为只有在这样的基础上再加上设计才思,才能创造出既有自己特色又能符合访问者喜好的设计方案。作为网页设计的初学者,最好要了解网页布局的基本概念,只有了解了它的重要性,才能控制一切能控制的东西,制作出精美的网页。网页布局主要有下面几个方面:

1.2.1 页面尺寸

由于页面尺寸和显示器大小、分辨率有关系,网页的局限性就在于设计者无法突破显示器的范围,而且因为浏览器也将占去不少空间,留下的页面范围变得越来越小。一般分辨率在 800×600 的情况下,页面的显示尺寸为 780×428 个像素;分辨率在 640×480 的情况下,页面的显示尺寸为 620×311 个像素;分辨率在 $1\,024 \times 768$ 的情况下,页面的显示尺寸为 $1\,007 \times 600$ 个像素。从以上数据可以看出,分辨率越高,页面尺寸越大。

浏览器的工具栏也是影响页面尺寸的原因。目前的浏览器的工具栏都可以取消或者增加,那么当全部工具栏显示或关闭时,页面的尺寸是不一样的。

在网页设计过程中,向下拖动页面是唯一能给网页增加更多内容(尺寸)的方法。但除非站点的内容能吸引访问拖动,否则最好不要让访问者拖动页面超过三屏。如果需要在同一页面显示超过三屏的内容,那么最好能做上页面内部链接,方便访问者浏览。

1.2.2 整体造型

什么是造型?造型就是创造出来的物体形象。这里是指页面的整体形象,这种形象应该是一个整体,图形与文本的接合应该是层叠有序的。虽然,显示器和浏览器都是矩形,但对于页面的造型,可以充分运用自然界中的其他形状以及它们的组合:矩形、圆形、三角形、菱形等。

对于不同的形状,它们所代表的意义是不同的。比如矩形代表着正式、规则,很多 ICP 和政府网页都是以矩形为整体造型;圆形代表着柔和、团结、温暖、安全等,许多时尚站点喜欢以圆形为页面整体造型;三角形代表着力量、权威、牢固、侵略等,许多大型的商业站点为显示它的权威性常以三角形为页面整体造型;菱形代表着平衡、协调、公平,一些交友站点常运用菱形作为页面整体造型。虽然不同形状代表着不同意义,但目前的网页制作多数是结合多个图形

加以设计,其中某种图形的构图比例可能占得多一些。

1.2.3 页 头

页头又可称为页眉,页眉的作用是定义页面的主题。比如一个站点的名字多数都显示在页眉里。这样,访问者能很快知道这个站点是什么内容。页头是整个页面设计的关键,它将牵涉后面的更多设计和整个页面的协调性。页头常放置站点名字的图片、公司标志以及旗帜广告。

1.2.4 文 本

文本在页面中都以数行或者块(段落)出现,它们的摆放位置决定着整个页面布局的可视性。过去因为页面制作技术的局限,文本放置的灵活性非常小,而随着 DHTML 的兴起,文本已经可以按照自己的要求放置到页面的任何位置。

1.2.5 图 片

图片和文本是网页的两大构成元素,缺一不可。如何处理好图片和文本的位置成了整个页面布局的关键。而设计者的布局思维也将体现在这里。

1.2.6 多媒体

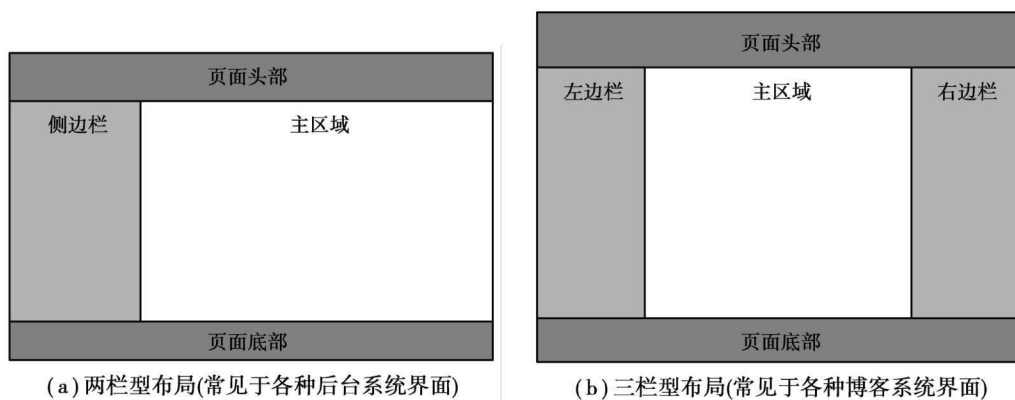
除了文本和图片外,还有声音、动画、视频等其他媒体。虽然它们不是经常能被利用到,但随着动态网页的兴起,它们在网页布局上也将变得越来越重要。

1.2.7 页 脚

页脚和页头相呼应。页头是放置站点主题的地方,而页脚是放置制作者或者公司信息的地方。

当初学者掌握了这些基本的网页布局,就可以依据这些设计一个新的网页,看看是不是与之前的设计有所不同,新设计的网页是不是更加符合浏览者的习惯,也更受浏览者喜爱。

常见的页面布局模型如图 1.1 所示。



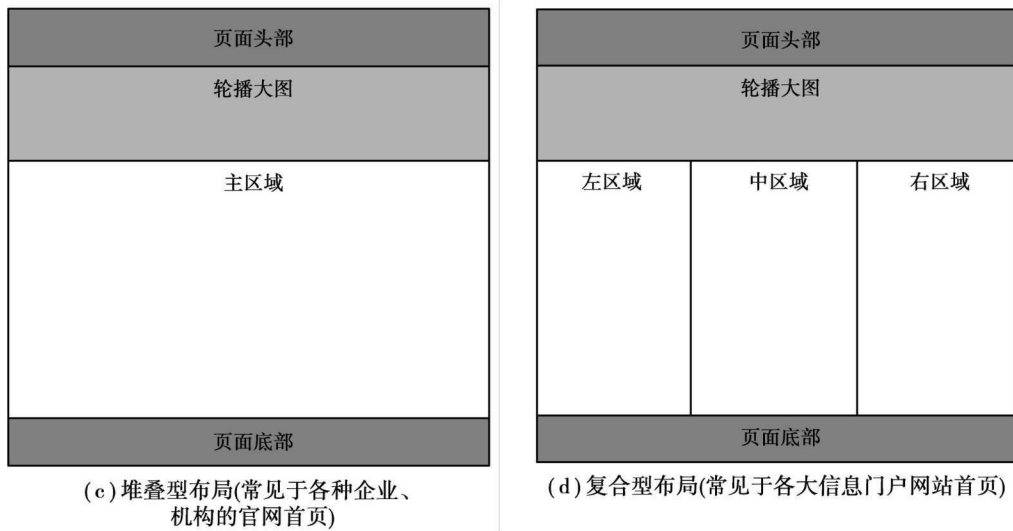


图 1.1 页面布局

1.3 界面配色

1.3.1 网页界面配色的重要性

色彩在网页设计中是一个重要的表现要素,在无以计数的网页中,要使一个网页明显区别于其他网站的页面,更具有吸引浏览者的魅力,就离不开色彩的设计与运用。成功的网页界面配色方案,能够将页面上的文字、图片以及排版恰到好处地融合在一起,从而使网页界面的作用得到最大的发挥。

1.3.2 网页界面配色的一般原理

(1) 色彩三属性

电脑屏幕显示的色彩均是由 R、G、B 三种色光组成,由于网页色彩是通过电脑显示器屏幕显示的,因此,网页色彩的三原色即红、绿、蓝三色。色相、明度和纯度叫做色彩的三属性,色彩由这三个要素的差异来区分。

色相即色彩的相貌,是色彩最基本的特征,也是色与色彼此互相区分最明显的特征。可见光谱不同波长的辐射在视觉上就表现为不同的色相,即一个特定波长的色光给人特定的色彩感受。

明度是指色彩的亮暗程度。明度按一定的间隔划分,就构成了明暗尺度。有彩色的明暗,以无彩色灰调的相应明度来表示其相应的明度值。

在网页中,纯度主要是指色彩对人的视觉刺激程度,纯度越高,色越纯,越容易给人艳丽鲜亮的感觉;纯度越低,色彩越弱。如果纯度足够低,就变成了没有颜色的无彩色。

(2) 网页色彩搭配方法

当根据方案选定一种主打色相后,调整其明度和纯度,将颜色加深或减弱,则可以产生新的色彩。这种色彩的搭配可以使页面颜色统一,增加页面层次感。

色相环中相邻的颜色称为邻近色。如红色和黄色,绿色和蓝色等。使用相邻色进行配色,能使页面产生和谐统一的效果。

在色相环中,对立面的两个颜色,如红色和绿色,黄色和紫色等,称为互补色。在网页配色时,使用互补色进行配色,将会刺激人的色彩神经,给人以鲜明的印象,产生强烈的视觉效果。在进行页面配色时,选用一种主色调进行配色,添加其对比色进行点缀,能起到画龙点睛的功效。

第 2 章

HTML5

HTML (Hyper Text Markup Language, 超文本标记语言) 是一种用来制作超文本文档的简单标记语言。它利用各种标记 (tags) 来标识文档的结构以及标识超链 (Hyperlink) 的信息。

目前, HTML 语言的版本是 5.0, 它是基于 SGML (Standard Generalized Markup Language, 标准广义置标语言, 是一套用来描述数字化文档的结构并管理其内容的复杂的规范) 的一个子集演变而来。

HTML5 是一种组织 Web 内容的语言, 其目的是通过创建一种标准的和直观的 UI 标记语言来把 Web 设计和开发变得容易起来。HTML5 提供了各种切割和划分页面的手段, 它允许设计者创建的切割组件不仅能用来逻辑地组织站点, 而且能够赋予网站聚合的能力。HTML5 可谓是“信息到网站设计的映射方法”, 因为它体现了信息映射的本质, 划分信息, 并给信息加上标签, 使其变得容易使用和理解。这是 HTML5 富于表现力的语义和实用性美学的基础, HTML5 赋予设计者和开发者各种层面的能力来向外发布各式各样的内容, 从简单的文本内容到丰富的、交互式的多媒体, 无不包括在内。

HTML5 提供了高效的数据管理、绘制、视频和音频工具, 促进了 Web 上的和便携式设备的跨浏览器应用的开发。HTML5 是驱动移动云计算服务方面发展的技术之一, 因为其允许更大的灵活性, 支持开发非常精彩的交互式网站。它还引入了新的标签 (tag) 和增强性的功能, 其中包括了一个优雅的结构、表单的控制、API、多媒体、数据库支持和显著提升的处理速度等。

HTML5 中的新标签都是能高度关联唤起的, 标签封装了它们的作用和用法。HTML 过去的版本更多的是使用非描述性的标签, 然而, HTML5 拥有高度描述性的、直观的标签, 提供了丰富的、能够立刻让人识别出内容的内容标签。例如, 被频繁使用的 <div> 标签已经有了两个增补的 <section> 和 <article> 标签。<video>、<audio>、<canvas> 和 <figure> 标签的增加也提供了对特定类型内容的更加精确的描述。

HTML5 提供了:

- 确切描述了其旨在要包含的内容的标签;
- 增强的网络通信;
- 极大改善了的常用存储;
- 运行后台进程的 Web Worker;

- 在本地应用和服务器之间建立持续连接的 WebSocket 接口；
- 更好的存储数据检索方式；
- 加快的了的页面保存和加载速度；
- 对使用 CSS3 来管理 GUI 的支持,这意味着 HTML5 可以是面向内容的；
- 改进了的浏览器表单处理；
- 基于 SQL 的数据库 API,其允许客户端的本地存储；
- 画布和视频,可在无须安装第三方插件的情况下添加图形和视频；
- Geolocation API 规范,其通过使用智能手机定位功能来纳入移动云服务和应用；
- 增强型的表单,其降低了下载 JavaScript 代码的这种必要性,允许在移动设备和云服务之间进行更多高效的通信。

HTML5 创建了一种更吸引用户的体验:使用 HTML5 设计的页面能够提供类似于桌面应用的体验。HTML5 还通过把 API 功能和无处不在的浏览器结合起来的方式来增强多平台开发。通过使用 HTML5,开发者能够提供一种顺畅地跨越各个平台的现代应用体验。

2.1 HTML 语言的语法特点

HTML 的标记总是封装在由小于号(<)和大于号(>)构成的一对尖括号之中。

(1) 单标记

单标记只需单独使用就能完整地表达意思,这类标记的语法是:

< 标记 >

最常用的单标记是
,它是 HTML 语言里的回车换行标记。

(2) 双标记

双标记由“始标记”和“尾标记”两部分构成,必须成对使用。其中,始标记告诉 Web 浏览器从此处开始执行该标记所表示的功能,而尾标记告诉 Web 浏览器在这里结束该功能。始标记前加一个斜杠(/)即成为尾标记。这类标记的语法是:

< 标记 > 内容 </标记 >

其中,“内容”部分就是要被这对标记施加作用的部分。例如,如果想突出对某段文字的显示,就将此段文字放在一对 标记中:

```
<EM > text to emphasize </EM >
```

可见,双标记能够对包含在其中的文本等对象的颜色、字体等各种属性进行设置。双标记也是 HTML 语言中应用最广泛的标记,是 HTML 语言的一大特点。

(3) 标记属性

许多单标记和双标记的始标记内可以包含一些属性,其语法是:

< 标记 属性 1 属性 2 属性 3 ... >

各属性之间无先后次序,属性也可省略(即取默认值),例如单标记 <HR > 表示在文档当前位置画一条水平线(horizontal line),一般是从窗口中当前行的最左端一直画到最右端。此标记可以带有一些属性:


```
<HR SIZE = 3 ALIGN = LEFT WIDTH = "75%" >
```

其中,SIZE 属性定义线的粗细,属性值取整数,缺省为 1;ALIGN 属性表示对齐方式,可取 LEFT(左对齐,缺省值),CENTER(居中),RIGHT(右对齐);WIDTH 属性定义线的长度,可取相对值(由一对"%"号括起来的百分数,表示相对于充满整个窗口的百分比),也可取绝对值(用整数表示的屏幕像素点的个数,如 WIDTH = 300),缺省值是 100%。

注意:双标记必须成对使用。

除一些个别标记外,HTML 都可以嵌套使用,如:

```
<p > <font color = red > RED </font > <font color = blue > BLUE </font > </p >
```

除少数几个转义序列之外,HTML 标记不区分大小写,即 <title > 等价于 <TITLE >。

并非所有的 World Wide Web 浏览器都支持所有的标记,如果一个浏览器不支持某个标记,它通常会将其忽略。

2.2 HTML DOM

DOM 是“Document Object Model”(文档对象模型)的首字母缩写。如果没有 Document(文档),DOM 也就无从谈起。当创建了一个网页并把它加载到 Web 浏览器中时,DOM 就在幕后悄然而生。它将根据设计者编写的网页文档创建一个文档对象。

DOM 把一份文档表示为一棵树(这里所说的“树”是数学意义上的概念),这是我们理解和运用这一模型的关键。更具体地说,DOM 把文档表示为一棵家谱树。

家谱树本身又是一种模型。家谱树的典型用法是表示一个人类家族的谱系并使用 parent(父)、child(子)、sibling(兄弟)等记号来表明家族成员之间的关系。家谱树可以把一些相当复杂的关系简明地表示出来:一位特定的家族成员既是某些成员的父辈,又是另一位成员的子辈,同时还是另一位成员的兄弟。

类似于人类家族谱系的情况,家谱树模型也非常适合用来表示一份用(X)HTML 语言编写出来的文档。

请看如图 2.1 所示的网页,它的内容是一份购物清单。

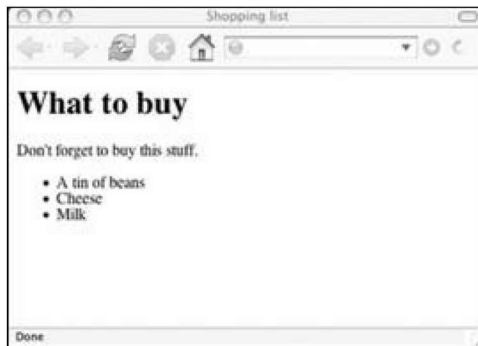


图 2.1 一份购物清单