



工业和信息化“十三五”  
人才培养规划教材



# 动态网页

设计与制作 | HTML5+CSS3  
+JavaScript

第3版

Dynamic Web Design and Production

吴以欣 陈小宁 © 编著



作者拥有丰富的企业工作经验，书中案例来源于**真实项目**

**畅销图书**改版，广受院校教师和学生好评

根据 **HTML5** 规范重新打造，符合新版**技术标准**



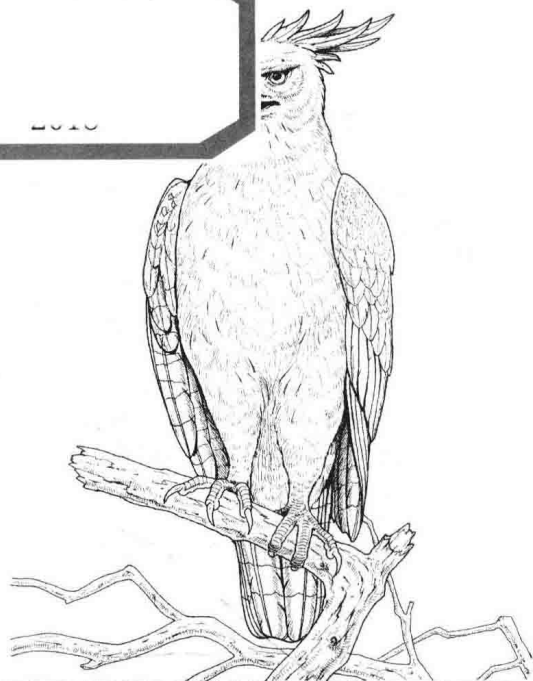
中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS



工业和信息化部“十三五”  
人才培养规划教材



# 动态网页

设计与制作 | HTML5+CSS3  
+JavaScript

第3版

Dynamic Web Design and Production

吴以欣 陈小宁 © 编著

人民邮电出版社  
北京

## 图书在版编目(CIP)数据

动态网页设计与制作 : HTML5+CSS3+JavaScript / 吴以欣, 陈小宁编著. — 3版. — 北京 : 人民邮电出版社, 2018.12

工业和信息化“十三五”人才培养规划教材  
ISBN 978-7-115-48775-9

I. ①动… II. ①吴… ②陈… III. ①超文本标记语言—程序设计—高等学校—教材②网页制作工具—高等学校—教材③JAVA语言—程序设计—高等学校—教材 IV. ①TP312②TP393.092

中国版本图书馆CIP数据核字(2018)第263485号

## 内 容 提 要

本书全面系统地介绍了用 HTML5、CSS3 和 JavaScript 制作网页的编程技术及方法。全书分为 HTML 基础、CSS 基础和实用技巧、JavaScript 的基本编程方法和实用技巧以及综合实训项目四大部分。读者通过学习本书的内容,可以制作出自己的动态网页,全面提高自己的网页设计基本技能。

本书可以作为多媒体技术专业的专业课程教材,也可以作为一般网页制作人员的自学用书。

- 
- ◆ 编 著 吴以欣 陈小宁  
责任编辑 左仲海  
责任印制 马振武
  - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号  
邮编 100164 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
固安县铭成印刷有限公司印刷
  - ◆ 开本: 787×1092 1/16  
印张: 17 2018 年 12 月第 3 版  
字数: 393 千字 2018 年 12 月河北第 1 次印刷

---

定价: 49.80 元

读者服务热线: (010) 81055256 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广登字 20170147 号



## 前 言

# FOREWORD

**早**期的 HTML 版本几乎涵盖了网页设计的全部内容，在网页的设计中，网页的内容和排版都是通过 HTML 文档完成的。但是，实际上网页的排版布局千变万化，因此，当需要改变网页的布局时，就必须大量地修改 HTML 文档，这给网页的设计开发带来了许多不便。

从 HTML 4.0 开始，为了简化程序的开发，HTML 已经尽量将“网页的内容结构”与“网页的排版布局”分开。它的主要原则：用标签元素描述网页的内容结构；用 CSS 描述网页的排版布局；用 JavaScript 描述网页的事件处理，即鼠标或键盘在网页元素上动作（如单击、双击、输入）后执行的程序。

由此可知，网页制作的基本语言是 HTML，网页排版布局的基本技术是 CSS，开发动态网页的关键是 JavaScript 技术的应用，全面掌握这 3 项技术是动态网页开发与设计的基础。

本书主要分为四大部分，具体如下所述。

第 1 部分主要介绍了 HTML 基础（第 1 章）。

第 2 部分主要介绍了 CSS 基础和实用技巧（第 2 章和第 3 章）。

第 3 部分主要介绍了 JavaScript 的基本编程方法和实用技巧（第 4~9 章）。

第 4 部分是 HTML、CSS 和 JavaScript 的综合实训项目（第 10 章）。

本书由浅入深，配有大量示例，所有示例都可用于当前最为流行的两大浏览器——Microsoft Internet Explorer 和 Mozilla 的 Firefox 浏览器的较高级版本。

本书不仅全面介绍了 HTML、CSS 和 JavaScript 的基本编程技术，而且还将网页设计的常用技巧，如网页两列/三列排版、菜单制作、图像叠加以及运用 JavaScript 动态改变网页内容等，也进行了介绍。

本书的大部分示例为编者在工作中的实际应用，所有示例均经过验证。本书计划学时为 54 小时，如果能增加实验室上机实训，将会取得更好的效果。读者通过本书内容的学习及大量示例和实训项目的练习，可以很快地运用 HTML、CSS 和 JavaScript 进行动态网页设计。

编者的电子邮箱地址是 [syndia\\_wu@yahoo.com](mailto:syndia_wu@yahoo.com)，欢迎读者提出建议和意见。

编 者

2018 年 2 月

# CONTENTS

第 1 章 HTML 基础	1	3.1.5 鼠标指针移到元素上时的效果	82
1.1 HTML 简介	1	3.1.6 定义外部样式表的选项	86
1.1.1 网页与 HTML	1	3.2 CSS 用于响应式网页布局设计	87
1.1.2 编写及显示 HTML 文件	2	3.3 CSS 用于菜单设计	90
1.1.3 标签、元素和属性	4	3.4 CSS 其他设计原则	97
1.2 HTML 常用元素	7	第 4 章 JavaScript 简介	99
1.2.1 基本结构元素	7	4.1 什么是 JavaScript	99
1.2.2 常用块元素	7	4.1.1 JavaScript 的发展历史	99
1.2.3 常用列表元素	12	4.1.2 JavaScript 的特点	100
1.2.4 常用表格元素	14	4.1.3 JavaScript 的作用	101
1.2.5 常用行元素	16	4.2 编辑与调试 JavaScript	104
1.2.6 表单元素	22	4.3 第一个 JavaScript 示例	105
1.2.7 多媒体元素及内嵌元素	32	4.3.1 编写 JavaScript	105
1.2.8 特殊说明	35	4.3.2 运行 JavaScript 程序	107
第 2 章 CSS 基础	38	4.3.3 调试 JavaScript 程序	107
2.1 CSS 简介	38	第 5 章 JavaScript 编程基础	110
2.2 CSS 的基本语法	39	5.1 数据类型及变量	110
2.2.1 样式和样式表	39	5.1.1 数据类型	110
2.2.2 CSS 中的颜色和长度定义	40	5.1.2 常量与变量	111
2.2.3 常用的样式属性	41	5.2 表达式与运算符	114
2.2.4 定义样式表	60	5.2.1 表达式	114
2.2.5 内部样式表和外部样式表	67	5.2.2 运算符	114
2.2.6 层叠式应用规则	68	5.3 基本语句	119
第 3 章 CSS 实用技巧	69	5.3.1 注释语句	120
3.1 CSS 的常用技巧	69	5.3.2 赋值语句	120
3.1.1 网页内容的居中对齐	69	5.3.3 流程控制语句	120
3.1.2 网页内容的隐藏与显示	73	5.4 函数	125
3.1.3 处理长表格	74		
3.1.4 图片	77		

# 动态网页设计与制作 (HTML5+CSS3+JavaScript) (第3版)

5.4.1	定义函数	126	6.4.1	新建日期	157
5.4.2	使用函数	127	6.4.2	日期对象的属性和方法	157
5.4.3	函数的参数	127	6.4.3	日期对象的应用实例	159
5.5	对象	129	<b>第7章 JavaScript 常用文档对象</b>		163
5.5.1	什么是对象	129	7.1	HTML 文档对象结构	163
5.5.2	定义对象	130	7.1.1	文档对象的结点树	164
5.5.3	使用对象	132	7.1.2	得到文档对象中元素对象的一般方法	166
5.6	事件及事件处理程序	133	7.2	HTML 文档对象	167
5.6.1	网页中的事件	133	7.2.1	文档对象的属性和方法	167
5.6.2	用 JavaScript 处理事件	134	7.2.2	文档对象的 cookie 属性	169
<b>第6章 JavaScript 常用内置对象</b>		136	7.2.3	表单及其控件元素对象	171
6.1	数组对象	136	7.2.4	链接对象	182
6.1.1	新建数组	136	7.2.5	图像对象	183
6.1.2	数组中的序列号	136	7.3	动态改变网页内容和样式	187
6.1.3	引用数组元素	136	7.3.1	动态改变网页内容	187
6.1.4	动态数组	137	7.3.2	动态改变网页样式	189
6.1.5	数组对象的常用属性和方法	137	<b>第8章 JavaScript 常用浏览器对象</b>		193
6.1.6	排序数组	138	8.1	屏幕对象	193
6.1.7	关联数组	141	8.2	浏览器信息对象	194
6.2	字符串对象	142	8.3	窗口对象	196
6.2.1	使用字符串对象	142	8.3.1	窗口对象的常用属性和方法	197
6.2.2	字符串相加	142	8.3.2	多窗口控制	198
6.2.3	在字符串中使用单引号、双引号及其他特殊字符	143	8.3.3	输入/输出信息	203
6.2.4	比较字符串是否相等	143	8.4	网址对象	204
6.2.5	字符串与整数、浮点数之间的转换	143	8.4.1	网址对象的常用属性和方法	205
6.2.6	字符串对象的属性和方法	144	8.4.2	网址对象的应用实例	206
6.2.7	字符串对象的应用实例	145	8.5	历史记录对象	206
6.3	数学对象	150	8.5.1	历史记录对象的常用属性和方法	206
6.3.1	使用数学对象	150	8.5.2	历史记录对象的应用实例	207
6.3.2	数学对象的属性和方法	150	8.6	框架对象	208
6.3.3	特殊的常数和函数	151	8.6.1	框架对象的常用属性和方法	208
6.3.4	格式化数字	152			
6.3.5	产生随机数	154			
6.3.6	数学对象的应用实例	155			
6.4	日期对象	157			

8.6.2 框架对象的应用实例.....	209	第 10 章 实训项目.....	239
第 9 章 JavaScript 实用技巧.....	212	10.1 “第 1 章 HTML 基础” 实训.....	239
9.1 建立函数库.....	212	10.2 “第 2 章 CSS 基础”实训.....	242
9.2 读取与储存数据.....	217	10.3 “第 3 章 CSS 实用技巧” 实训.....	244
9.2.1 页面上数据的读取与 储存.....	217	10.4 “第 4 章 JavaScript 简介” 实训.....	247
9.2.2 浏览器上数据的读取 与储存.....	219	10.5 “第 5 章 JavaScript 编程 基础”实训.....	248
9.3 校验用户输入.....	221	10.6 “第 6 章 JavaScript 常用 内置对象”实训.....	253
9.4 设置弹出窗口.....	228	10.7 “第 7 章 JavaScript 常用 文档对象”实训.....	258
9.4.1 一般的弹出窗口.....	228	10.8 “第 8 章 JavaScript 常用 浏览器对象”实训.....	261
9.4.2 窗口中的“窗口”.....	228	10.9 “第 9 章 JavaScript 实用技巧”实训.....	263
9.5 设置下拉菜单.....	230		
9.6 事件冒泡处理.....	232		
9.7 动画技术.....	233		
9.7.1 动画网页对象的内容.....	234		
9.7.2 动画网页对象的尺寸.....	234		
9.7.3 动画网页对象的位置.....	236		



# 第 1 章 HTML 基础

本章主要内容:

- HTML 简介
- HTML 常用元素

## 1.1 HTML 简介

### 1.1.1 网页与 HTML

当通过浏览器浏览网页时, 用户会看到文字信息、超链接、图表、图片等各种内容, 如图 1-1 所示。浏览器是如何显示这些网页内容的呢? 通过浏览器查看这些网页的源代码, 如图 1-2 所示, 可以发现这些源代码就是浏览器可以“理解”的一种计算机语言——HTML。



图 1-1 网页内容



图 1-2 查看网页源代码

HTML 是 Hyper Text Markup Language 的缩写, 中文翻译为“超文本标记语言”, 是制

作网页的最基本语言, 它的特点正如它的名称, 具体如下。

- Hyper (超): “超 (Hyper)” 是相对于 “线性 (Linear)” 而言的, HTML 之前的计算机程序大多是线性的, 即必须由上至下顺序运行, 而用 HTML 制作的网页可以通过其中的超链接从一个网页 “跳转” 至另一个网页。

- Text (文本): 不同于一些编译性的程序语言, 如 C、C++ 或 Java 等, HTML 是一种文本解释性的程序语言, 即它的源代码不经过编译, 在浏览器中运行时直接被 “翻译”。

- Markup (标记): HTML 的基本规则就是用 “标记语言” —— 成对尖括号组成的标签元素来描述网页内容的, 进而在浏览器中显示。

HTML 最早作为一种标准的网页制作语言是在 20 世纪 80 年代末由科学家蒂姆·伯纳斯李 (Tim Berners-Lee) 提出的, 当时他定义了 22 种标签元素。发展至 1999 年 12 月, 由万维网联盟 (W3C) 出版的 HTML 4.01 规范中还保留着其中的 13 种标签元素。2000 年 5 月, HTML 已成为一项国际标准 (ISO/IEC 15445:2000)。2014 年 10 月, 万维网联盟正式发布了 HTML5 规范的推荐版, 包含大约 107 种标签元素, 以满足现代网络发展的需要。目前, 大多数最新版本的浏览器都已经支持 HTML5 规范。

早期的 HTML 版本不仅用标签元素描述网页的内容结构, 而且还用标签元素描述网页的排版布局。在网页设计中, 网页的内容结构一般变化较小, 但是网页的排版布局可能千变万化, 当需要改变网页的布局时, 就必须大量地修改 HTML 文档, 这给网页的设计开发工作带来了很大不便。从 HTML 4.0 开始, 为了简化程序的开发, HTML 已经尽量将 “网页的内容结构” 与 “网页的排版布局” 分开, 它的主要原则如下。

(1) 用标签元素描述网页的内容结构。

(2) 用 CSS 描述网页的排版布局。

(3) 用 JavaScript 描述网页的事件处理, 即通过鼠标或键盘在网页元素上进行动作 (如单击、双击、输入) 后执行的程序。

本书将以 HTML5 规范为标准进行讲解, 因此, 本章将主要讲述原则 (1) 的内容, 原则 (2) 的内容将在第 2 章和第 3 章中讲述, 原则 (3) 的内容将在第 4~9 章中讲述。

## 1.1.2 编写及显示 HTML 文件

在计算机中, 用 HTML 语言编写程序, 并保存为文件, 然后在浏览器 “地址栏” 中输入该文件名, 包括文件所在的文件夹名, 例如 C:/project/ch1\_01.html, 如图 1-3 所示, 浏览器就会显示出 “翻译” 后的网页效果。



图 1-3 在浏览器中查看计算机中的 HTML 文件

如果将该文件放在一个网页服务器上，并在浏览器“地址”栏中输入该服务器的地址或指向该服务器的域名及该文件名，如图 1-4 所示，就可以通过 Internet 浏览这个网页的内容。这也就是当初科学家蒂姆·伯纳斯李发明 HTML 的目的——大家共享文件内容。



图 1-4 在浏览器中查看网页服务器上的 HTML 文件

HTML 文件具有以下特点。

- HTML 文件是一种包含成对标签元素的普通文本文件。因此，可以用任何一种文本编辑器来编写，如 Windows 中的记事本、写字板等应用软件；也可以使用任何一种工具软件编辑 HTML 文件，如 Macromedia 的 Dreamweaver 和 Microsoft 的 FrontPage 等。

- HTML 文件必须以 .htm 或 .html 作为扩展名，两者并没有太大的区别。本书示例中统一使用 .html 作为扩展名。

- HTML 文件可以在大多数流行的浏览器上显示，例如 Microsoft 的 Internet Explorer（以下简称 IE）以及 Google 的 Chrome（以下简称 Chrome）等。本书将在 Windows 操作系统中使用 Chrome 浏览器显示示例。如果计算机中没有 Chrome 浏览器软件，可以到 Google 网站下载。

示例 1-1 是一个最简单的 HTML 程序，首先在文本编辑器中输入该程序（注意，其中的行号用于本书的讲解之便，程序中不要输入行号），然后保存文件名为 ch1\_01.html。

**示例 1-1 第一个 HTML 程序。**

目的：初步了解 HTML 程序。

程序文件名：ch1\_01.html。

```

1  <!DOCTYPE HTML>
2  <html>
3    <head>
4      <title>我的网页</title>
5    </head>
6    <body>
7      <h1>欢迎进入我的网页</h1>
8      <hr>
9      <p>这是我的<b>第一个</b>网页。</p>
10   </body>
11  </html>

```



(1) 编写 HTML 程序时, 格式上没有要求, 例如, 示例中的 11 行程序也可以写成一行, 这是因为浏览器“翻译”HTML 程序时是通过其中的标签内容进行识别的, 与程序的格式无关。但是, 为了便于程序的阅读和维护, 编程时应该根据标签的结构适时换行。

(2) HTML 的文件名最好不要有空格, 为了能符合大多数网页服务器的要求, HTML 的文件名最好只包含英文小写字母和有效的字符, 例如 a~z、0~9 以及下划线 ( \_ ) 和减号 ( - ), 并且文件名的长度不要超过 31 个字符 (包括扩展名)。

(3) 按下述方法进行操作, 就可以通过浏览器查看示例 1-1 所编写的 HTML 程序效果。

① 打开浏览器 Chrome, 按【Ctrl+O】组合键。

② 在打开的对话框中单击“浏览”按钮, 然后在打开的对话框中选择示例 1-1 中保存的文件 ch1\_01.html。

③ 单击“确定”按钮, 就会得到图 1-3 右图所示的效果。

(4) 按下述方法进行操作, 可以调试 HTML 程序。

① 在计算机中同时打开文本编辑器和浏览器 Chrome, 并且在文本编辑器中打开 HTML 程序, 如 ch1\_01.html, 然后按 (3) 介绍的内容进行操作, 在浏览器中显示该网页内容。

② 在文本编辑器中修改程序内容, 例如, 将示例 1-1 中的“第一个”修改为“第二个”, 并保存该文件, 然后反复按【Alt+Tab】组合键直到切换到浏览器窗口, 再按【F5】键或【Ctrl+R】组合键刷新浏览器窗口。

③ 不断重复步骤②直到调试结束。

## 1.1.3 标签、元素和属性

从示例 1-1 中可以看出, HTML 文档是由各种 HTML 元素组成的, 如 html (HTML 文档) 元素、head (头) 元素、body (主体) 元素、title (题目) 元素和 p (段落) 元素等, 这些元素都采用了尖括号组成的标签形式。实际上, HTML 程序的内容就是标签、元素和属性。

### 1. 标签

HTML 标签是由一对尖括号 <> 及标签名称组成的。标签分为“起始标签”和“结束标签”两种, 两者的标签名称是相同的, 只是结束标签多了一个斜杠“/”。如图 1-5 所示, <b> 为起始标签, </b> 为结束标签, 它是英文 bold (粗体) 的缩写。标签名称对大小写是不敏感的, 即 <html>...</html> 和 <HTML>...</HTML> 的效果是一样的, 但是为了便于程序的阅读和维护, 万维网联盟推荐使用小写字母书写标签。

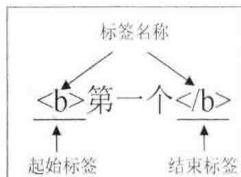


图 1-5 HTML 标签

## 2. 元素

HTML 元素是组成 HTML 文档的最基本的部件，它是用标签来表现的，一般情况下，“起始标签”表示元素的开始，“结束标签”表示元素的结束。

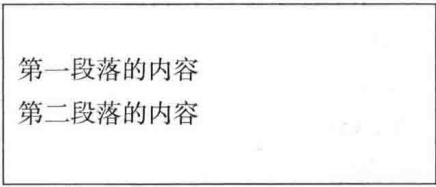
HTML 元素分为“有内容的元素”和“空元素”两种。“有内容的元素”是由起始标签、结束标签以及两者之间的元素内容组成的，其中，元素内容既可以是需要显示在网页中的文字内容，也可以是其他元素。例如，示例 1-1 中起始标签**<b>**与结束标签**</b>**组成的元素的内容是文字“第一个”；而起始标签**<head>**与结束标签**</head>**组成的元素的内容是另外一个元素——**title** 元素。“空元素”则只有起始标签，而没有结束标签和元素内容，如示例 1-1 中的 **hr**（分隔线）元素就是空元素。

HTML 元素还可以分为“块元素”和“行元素”。“块元素”在网页中的效果是该元素中的内容相对于其前后元素的内容另起一行，图 1-6 左侧所示的 **p** 元素就是块元素，图 1-6 右侧所示为 **p** 元素的网页显示效果。而“行元素”的网页效果则是，行元素中的内容与其前后元素的内容在同一行中，图 1-7 左侧所示的 **a** 元素就是行元素，图 1-7 右侧所示为 **a** 元素的网页显示效果。

HTML 程序：

```
...
<p>第一段落的内容</p>
<p>第二段落的内容</p>
...
```

网页显示效果：



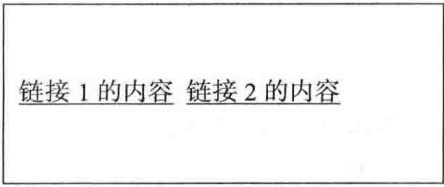
第一段落的内容  
第二段落的内容

图 1-6 块元素程序及其网页显示效果

HTML 程序：

```
...
<a href="a.html">链接 1 的内容</a>
<a href="b.html">链接 2 的内容</a>
...
```

网页显示效果：



[链接 1 的内容](#) [链接 2 的内容](#)

图 1-7 行元素程序及其网页显示效果

## 3. 属性

在元素的起始标签中，还可以包含“属性”，用来表示元素的其他特性，它的格式如下。

```
<标签名称 属性名="属性值">
```

例如，下述代码中 **img**（图像）元素中的 **src="my\_picture.jpg"** 就是 **img** 元素的属性，表示“该图像元素的图像源文件是 **my\_picture.jpg**”。

```

```

像标签名称一样，属性名对大小写也是不敏感的，但是为了便于程序的阅读和维护，编程时建议使用小写字母。另外值得注意的是，虽然 HTML5 并没有要求属性值一定要有双引号，但是，为了养成良好的编程习惯，还是应该统一地在属性值外面加上双引号（英文输入模式下的双引号）。

## 4. 元素的嵌套性

除了 HTML 文档元素 `html` 外, 其他 HTML 元素都是被嵌套在另一个元素之内的。例如示例 1-1 中, `head` 元素和 `body` 元素是嵌套在 `html` 元素中的, 而 `title` 元素是嵌套在 `head` 元素中的。值得注意的是, HTML 中的元素可以多级嵌套, 但是不可以互相交叉。

如下代码为不正确的写法, 因为 `p` 元素的起始标签在 `b` 元素的外层, 而它的结束标签却放在了 `b` 元素结束标签的里面。

```
<p>这是我的<b>第一个</p>网页。</b>
```

如下代码为正确的写法。

```
<p>这是我的<b>第一个</b>网页。</p>
```

由于元素的嵌套性, 编写 HTML 程序时一般先写外层的一对标签, 然后逐渐往里写, 这样既不容易忘记结束标签, 也可以减少交叉标签的错误。例如, 示例 1-1 应该按下列步骤完成。

① 编写最外层标签, 示例如下。

```
<html>  
</html>
```

② 添加内层标签, 示例如下。

```
<html>  
  <head>  
  </head>  
  <body>  
  </body>  
</html>
```

③ 继续添加内层标签, 示例如下。

```
<html>  
  <head>  
    <title></title>  
  </head>  
  <body>  
  </body>  
</html>
```

④ 添加元素内容, 示例如下。

```
<html>  
  <head>  
    <title>我的网页</title>  
  </head>  
  <body>  
  </body>  
</html>  
...
```

以此类推。

## 1.2 HTML 常用元素

HTML5 相对于旧版本新增加了 32 种元素，去掉了 12 种元素，一共包含 107 种元素。本节将主要介绍其中一些最常用的元素。

### 1.2.1 基本结构元素

HTML 的基本结构元素主要有 3 种，它们分别是 html（HTML 文档）元素、head（头）元素和 body（主体）元素。每个网页文件中一般都包含这 3 种元素，而且它们只能出现一次。

#### 1. html 元素

html 元素是网页文件中最外围的、紧接着文档元素的一对标签，它告诉浏览器整个文件是 HTML 格式，并且从<html>开始，至</html>结束。

#### 2. head 元素

head 元素包含的是网页的头部信息，它的内容被浏览器所用，而不会显示在网页正文中。head 元素中可以包含下述元素。

- title 元素，它的内容将在浏览器的标题中出现。例如，示例 1-1 中的 title 元素内容是“我的网页”，如图 1-3 右图所示，它显示在浏览器的标题栏。

- link 元素和 style 元素，常用于 CSS，将在第 2 章和第 3 章中讲解。

- script 元素，常用于 JavaScript，将在第 4 章中讲解。

- meta 元素，特殊元素将在“1.2.8 特殊说明”小节中讲解。

#### 3. body 元素

body 元素是 HTML 文件的主体元素，它包含所有要在网页上显示的各种元素。下面几小节内容中讨论的元素，都是 body 元素可以包含的内容。

### 1.2.2 常用块元素

块元素的内容总是沿网页垂直方向另起一行显示。HTML5 中的常用块元素有以下几种。

#### 1. 标题块元素

标题块元素主要有 h1、h2、h3、h4、h5 和 h6。h 是 header（标题）的简写，数字 1~6 表示标题的级别，h1 的标题级别最大，h6 的标题级别最小，如图 1-8 所示。

HTML 程序：

...

<h1>第一级标题</h1>

<h2>第二级标题</h2>

<h3>第三级标题</h3>

<h4>第四级标题</h4>

<h5>第五级标题</h5>

<h6>第六级标题</h6>

...

网页显示效果：

第一级标题

第二级标题

第三级标题

第四级标题

第五级标题

第六级标题

图 1-8 标题块元素程序及其网页显示效果

## 2. 段落块元素

最常用的段落块元素有 p (段落) 元素和 pre (原样显示文字) 元素。

### (1) p 元素

p 是 paragraph (段落) 的简写, p 元素内一般包含段落文字, 浏览器将自动在 p 元素前后加一行空行, 如图 1-6 右图所示。值得注意的是, HTML 元素内的文字内容都不要用空格来排版 (pre 元素除外), 因为浏览器会将标签内的所有空格或换行符看作一个空格。下面的示例 1-2 要说明的就是这个问题。

**示例 1-2** 将李白的诗《静夜思》显示在网页上。

目的: 了解 p 元素及 HTML 程序中的空格特点。

程序文件名: ch1\_02.html。

```

1  <!DOCTYPE HTML>
2  <html>
3    <head>
4      <title>静夜思</title>
5    </head>
6    <body>
7      <h1>静夜思</h1>
8      <hr>
9      <p>
10         床前明月光,
11         疑是地上霜。
12         举头望明月,
13         低头思故乡。
14     </p>
15 </body>
16 </html>
    
```

在浏览器中打开 ch1\_02.html, 得到的效果如图 1-9 所示, 可以看到, 虽然在程序中的每行诗句后面都有换行符, 但显示结果中的所有诗句都在一行中, 只是每个句子之间有一个空格。

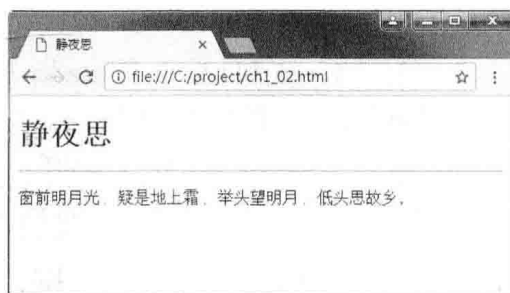


图 1-9 示例 1-2 的网页显示效果

如果将第 9~14 行代码改为如下格式, 刷新浏览器后, 将得到同样的效果。由此可以看出, 在元素内容中用空格或换行符进行排版都是无效的。

```
<p>床前明月光, 疑是地上霜。 举头望明月, 低头思故乡。 </p>
```

### (2) pre 元素

pre 是 preformatted 的简写, pre 元素与 p 元素的作用基本相同, 唯一区别是, 该元素

中的文字内容将保留空格和换行符，并且元素中的英文字符都将统一用等宽字体，以便对齐。例如，将示例 1-2 中的第 9~14 行改为如下格式，刷新浏览器后，将得到图 1-10 所示的效果。

```
<pre>
  床前明月光，
  疑是地上霜。
  举头望明月，
  低头思故乡。
</pre>
```



图 1-10 使用 pre 元素改写示例 1-2 中的网页效果

值得注意的是，pre 元素一般只用于在网页中显示诗句、计算机程序等需要使用空格和换行来排版的文字内容。根据 HTML5 的规范设计原则，网页的排版布局应尽可能地使用 CSS，详见“第 2 章 CSS 基础”和“第 3 章 CSS 实用技巧”。

### 3. 通用块元素

div 元素是最常用的通用块元素，它用于包住一些元素和文字内容，以便进一步用 CSS 进行排版处理。下面通过示例 1-3 介绍 div 元素的用途。

示例 1-3 修改示例 1-2 的 ch1\_02.html 代码如下，并保存为 ch1\_03.html。

目的：了解 div 元素的用途。

程序文件名：ch1\_03.html。

```
1  <!DOCTYPE HTML>
2  <html>
3    <head>
4      <title>我喜爱的李白诗</title>
5    </head>
6    <body>
7      <h1>我喜爱的李白诗</h1>
8      <div>
9        <h2>静夜思</h2>
10       <hr>
11       <pre>
12         床前明月光，
13         疑是地上霜。
14         举头望明月，
```

```

15     低头思故乡。
16     </pre>
17 </div>
18 <div>
19     <h2>下江陵</h2>
20     <hr>
21     <pre>
22         朝辞白帝彩云间，
23         千里江陵一日还。
24         两岸猿声啼不住，
25         轻舟已过万重山。
26     </pre>
27 </div>
28 </body>
29 </html>
    
```

在浏览器中打开 ch1\_03.html，得到的效果如图 1-11 左图所示，第 8、17、18 及 27 行的 div 元素在浏览器中看不出什么效果。这时将第 8 行修改为如下格式。

```
<div style="float:left;width:200px;margin-right:10px">
```

将第 18 行修改为如下格式。

```
<div style="float:left;width:200px">
```

然后将文件保存为 ch1\_03\_1.html，在浏览器中打开该文件，就会得到图 1-11 右图所示的效果。修改上述两行代码的目的是通过 CSS 语句将 div 元素所包含的内容由纵向排版变为横向排版。



图 1-11 示例 1-3 及其修改后的显示效果

## 4. 语义块元素

HTML5 新增了一系列语义块元素，也就是说，这些元素的标签名定义了应用的意义，例如，header（顶部）元素一般用于一个区域或整个页面的顶部，footer（底部）元素一般用于一个区域或整个页面的底部，等等。语义块元素的使用，主要是为了搜索引擎读取页