



“十二五”江苏省高等学校重点教材

(编号: 2014-2-027)

SHI ERWU JIANGSUSHENG
GAODENGXUEXIAOZHONGDIANJIAOCAI

CSS+DIV页面布局技术

主 编 朱翠苗



中国水利水电出版社
www.waterpub.com.cn

要 目 录

“十二五”江苏省高等学校重点教材

编号: 2014-2-027

CSS+DIV 页面布局技术

主 编 朱翠苗

本书得到苏州吉利信信息科技有限公司、苏州工业园区支持，是一本定位教材，从实际案例入手，讲解CSS+DIV应用技术的同时掌握其精髓。各案例与实训按照由浅入深、循序渐进的原则，即使是初学者也可以轻松掌握CSS+DIV技术的布局。本书可作为高等院校计算机专业及相关专业的教材，也可作为从事网页设计工作的技术人员参考。 全书共分6个单元，单元一主要介绍网页的布局设计中的基本概念，网页类型，网页基本结构，单元二主要介绍网页的布局设计，各种表单元素的使用，单元三主要介绍网页的布局设计，使用表格实现图文布局，单元四主要介绍CSS设置文字、背景、样式属性，同时重点介绍了盒子模型，以及浮动和清除浮动等，单元五介绍CSS+DIV实现典型页面局部布局，并详细讲解了典型布局的例子，单元六介绍了一个用CSS+DIV实现页面布局技术，单元七主要介绍CSS+DIV实现页面布局技术。	本书可作为高等院校计算机专业及相关专业的教材，也可作为从事网页设计工作的技术人员参考。
本书可作为高等院校计算机专业及相关专业的教材，也可作为从事网页设计工作的技术人员参考。	本书可作为高等院校计算机专业及相关专业的教材，也可作为从事网页设计工作的技术人员参考。



中国水利水电出版社
www.waterpub.com.cn

2015年4月

内 容 提 要

本书为“‘十二五’江苏省高等学校重点教材”，文件编号为：2014-2-027。

CSS+DIV 页面布局技术是目前流行的网页制作核心技术。

本书由浅入深、循序渐进，通过典型的项目载体，采取工作过程项目化的模式，系统地介绍和训练了超文本标记语言 HTML 的基本标签、表单应用、表格应用和布局、CSS 层叠样式表、CSS+DIV 实现典型页面局部布局，重点讲述了 CSS+DIV 实现页面布局的技术和方法，为了便于读者全面掌握 CSS+DIV 技术和规范，深刻体会网页设计的乐趣，最后给出一个综合性的实战项目“CSS+DIV 页面布局实例”，全面巩固所学的技术和方法。

本书适合作为高职高专软件技术、计算机网络技术、计算机应用技术、电子商务类以及相关专业的教材，也可以作为网页设计技术培训教程，还可以作为 Web 网站开发人员、网页设计师等的参考书。

图书在版编目（C I P）数据

CSS+DIV 页面布局技术 / 朱翠苗主编. — 北京：中国水利水电出版社，2015.6

“十二五”江苏省高等学校重点教材

ISBN 978-7-5170-3251-9

I. ①C… II. ①朱… III. ①网页制作工具—高等学校—教材 IV. ①TP393.092

中国版本图书馆CIP数据核字(2015)第128840号

策划编辑：石永峰

责任编辑：陈 洁

封面设计：李 佳

书 名	“十二五”江苏省高等学校重点教材 CSS+DIV 页面布局技术
作 者	主 编 朱翠苗
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 68367658 (发行部)、82562819 (万水) 北京科水图书销售中心(零售) 电话: (010) 88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	184mm×260mm 16开本 10.5印张 257千字
版 次	2015年6月第1版 2015年6月第1次印刷
印 数	0001—2000册
定 价	24.00元

凡购买我社图书，如有缺页、倒页、脱页的，本社发行部负责调换

版权所有·侵权必究

前 言

CSS+DIV 页面布局技术目前已成为页面设计核心技术,采用 CSS+DIV 布局技术具有缩减页面代码、提高页面的浏览速度和稳定性、页面设计结构清晰、易被搜索引擎搜索等优势。但很多读者反映 CSS 类的图书偏重基础理论介绍,案例简单,学习之后很难在实际工作中应用。近年来,CSS+DIV 页面布局技术类课程在高职院校和本科院校普遍开设,但是此类教材却寥寥无几,尤其是高职高专教材几乎空白,很难满足教学需要,鉴于这种情况,教材编写小组决定编写该教材,力图帮助读者在 CSS 学习中快速进入实战状态,给专业教师一手参考资料,帮助读者提高页面设计技能。

本书得到苏州吉耐特信息科技有限公司、太仓智博软件科技有限公司大力支持,是一本校企合作教材,从实际案例入手,重点讲解典型的布局技术和方法,让读者在学习 CSS+DIV 应用技术的同时掌握其精髓。各案例与实例按照由浅入深、由易到难的顺序编排,即使是初学者也可以轻松掌握 CSS+DIV 技术的布局方式,制作出精美实用的网页作品。

全书共分 6 个单元,单元一主要介绍静态网页的开发环境、网页制作和设计中的基本概念、网页类型、网页基本构成元素,重点介绍了行级标签和块级标签的使用。单元二主要介绍表单的基本语法,各种表单元素的使用。单元三主要介绍表格的基本用法,其中包括跨行跨列的表格、使用表格实现图文布局,同时还介绍了框架的使用。单元四主要介绍了用 CSS 设置文字、背景等样式属性,同时重点介绍了盒子模型,以及浮动和清除属性等。单元五介绍 CSS+DIV 实现典型页面局部布局,并详细介绍了典型布局的例子。单元六介绍了一个用 CSS+DIV 页面布局技术布局网页的综合实战,介绍了商城网站的布局方法,通过综合实战训练学生技能,进一步提高学生应用实践能力,体现了“做中学、学中产”的实训教学思想。

本书采取“工程过程项目化”的编写模式进行编写,将团队多年教学经验完全融入到教材中,采用“引入项目场景提出工作问题,通过实例项目训练,解决工作问题并掌握相应的技术和方法,然后回到工作场景完成场景项目”的内容组织形式,实现了示例和项目相结合,尤其最后一单元通过一个综合的、商业化强的网站进行实战训练,提高了本书“工学结合、项目实战”特色,本书的编写得到了企业工程师深度参与和技术支持,内容更加标准、规范。

本书主要定位高职高专 CSS+DIV 布局技术类课程教学使用,也可以作为应用型本科和职业中专的参考教材,还可以作为工程技术人员的参考用书。

本书由朱翠苗、郑广成、沈蕴梅、金静梅、庾佳、孙伟、孔小兵(企业)、袁顺利(企业)等编写,朱翠苗负责统稿并担任主编,根据典型项目设计内容载体,项目开发流程组织教材内容,教材具有实战性、可操作性、新颖性、通俗性和项目过程化的特点,更加激发学生学习兴趣和主动性。

由于时间仓促,再加上编者水平有限,书中难免有错误和疏漏之处,敬请广大读者批评指正。

编 者

2015 年 4 月

目 录

前言

单元一 HTML 的基本标签	1
1.1 工作场景导入	1
1.2 技术与知识准备	2
1.2.1 HTML 文件的基本结构	2
1.2.2 编辑工具	4
1.2.3 网页的摘要信息	7
1.2.4 块级标签	10
1.2.5 行级标签	19
1.2.6 注释和特殊符号	25
1.2.7 W3C 标准	26
1.3 工作场景训练	29
1.3.1 实现工作场景 1 的任务	29
1.3.2 实现工作场景 2 的任务	30
1.3.3 实现工作场景 3 的任务	31
1.4 重点问题分析	32
1.5 小结	33
单元二 表单应用	34
2.1 工作场景导入	34
2.2 技术与知识准备	35
2.2.1 表单的基本语法	35
2.2.2 表单元素的介绍	36
2.3 工作场景训练	41
2.4 重点问题分析	42
2.5 小结	42
单元三 表格应用和布局	43
3.1 工作场景导入	43
3.2 技术与知识准备	44
3.2.1 表格基础	44
3.2.2 跨行和跨列的表格	46
3.2.3 表格的高级用法	50
3.2.4 表格的其他用法	56
3.2.5 框架技术	58

3.3 工作场景训练	63
3.3.1 实现工作场景 1 的任务	63
3.3.2 实现工作场景 2 的任务	64
3.3.3 实现工作场景 3 的任务	66
3.4 重点问题分析	68
3.5 小结	69
单元四 CSS 样式表	70
4.1 工作场景导入	70
4.2 技术与知识准备	71
4.2.1 CSS 基础	71
4.2.2 常用的样式修饰	77
4.2.3 盒子模型	90
4.3 工作场景训练	111
4.3.1 实现工作场景 1 的任务	111
4.3.2 实现工作场景 2 的任务	112
4.3.3 实现工作场景 3 的任务	113
4.4 重点问题分析	116
4.5 小结	116
单元五 CSS+DIV 实现典型页面局部布局	117
5.1 工作场景导入	117
5.2 技术与知识准备	118
5.2.1 div-ul-li 实现横向导航菜单	118
5.2.2 div-dl-dt-dd 实现图文混排	122
5.3 工作场景训练	128
5.3.1 实现工作场景 1 的任务	128
5.3.2 实现工作场景 2 的任务	130
5.4 重点问题分析	132
5.5 小结	133
单元六 CSS+DIV 页面布局实例	134
6.1 分块制作商城网站首页	134
6.1.1 利用 Dreamweaver 搭建站点	134
6.1.2 制作商城网站首页的总体布局	137

6.1.3 实现顶部布局.....	140
6.1.4 实现左部商品分类.....	143
6.1.5 实现中部主题布局.....	145
6.1.6 实现右部布局.....	147
6.1.7 实现底部信息.....	149
6.2 制作商城分支页——注册页面.....	150

6.3 制作商城分支页——登录页面.....	154
6.4 制作商城分支页——商品具体介绍页面.....	156
6.5 建立页面之间的链接和进行兼容测试.....	158
6.5.1 建立页面间的链接.....	158
6.5.2 网页的兼容性测试.....	158
参考文献.....	160

50		1.2.1
51		1.2.2
52		1.2.3
53		1.2.4
54		1.2.5
55		1.2.6
56		1.2.7
57		1.2.8
58		1.2.9
59		1.2.10
60		1.2.11
61		1.2.12
62		1.2.13
63		1.2.14
64		1.2.15
65		1.2.16
66		1.2.17
67		1.2.18
68		1.2.19
69		1.2.20
70		1.2.21
71		1.2.22
72		1.2.23
73		1.2.24
74		1.2.25
75		1.2.26
76		1.2.27
77		1.2.28
78		1.2.29
79		1.2.30
80		1.2.31
81		1.2.32
82		1.2.33
83		1.2.34
84		1.2.35
85		1.2.36
86		1.2.37
87		1.2.38
88		1.2.39
89		1.2.40
90		1.2.41
91		1.2.42
92		1.2.43
93		1.2.44
94		1.2.45
95		1.2.46
96		1.2.47
97		1.2.48
98		1.2.49
99		1.2.50
100		1.2.51
101		1.2.52
102		1.2.53
103		1.2.54
104		1.2.55
105		1.2.56
106		1.2.57
107		1.2.58
108		1.2.59
109		1.2.60
110		1.2.61
111		1.2.62
112		1.2.63
113		1.2.64
114		1.2.65
115		1.2.66
116		1.2.67
117		1.2.68
118		1.2.69
119		1.2.70
120		1.2.71
121		1.2.72
122		1.2.73
123		1.2.74
124		1.2.75
125		1.2.76
126		1.2.77
127		1.2.78
128		1.2.79
129		1.2.80
130		1.2.81
131		1.2.82
132		1.2.83
133		1.2.84
134		1.2.85
135		1.2.86
136		1.2.87
137		1.2.88
138		1.2.89
139		1.2.90
140		1.2.91
141		1.2.92
142		1.2.93
143		1.2.94
144		1.2.95
145		1.2.96
146		1.2.97
147		1.2.98
148		1.2.99
149		1.2.100

1		1.1
2		1.2
3		1.3
4		1.4
5		1.5
6		1.6
7		1.7
8		1.8
9		1.9
10		1.10
11		1.11
12		1.12
13		1.13
14		1.14
15		1.15
16		1.16
17		1.17
18		1.18
19		1.19
20		1.20
21		1.21
22		1.22
23		1.23
24		1.24
25		1.25
26		1.26
27		1.27
28		1.28
29		1.29
30		1.30
31		1.31
32		1.32
33		1.33
34		1.34
35		1.35
36		1.36
37		1.37
38		1.38
39		1.39
40		1.40
41		1.41
42		1.42
43		1.43
44		1.44
45		1.45
46		1.46
47		1.47
48		1.48
49		1.49
50		1.50
51		1.51
52		1.52
53		1.53
54		1.54
55		1.55
56		1.56
57		1.57
58		1.58
59		1.59
60		1.60
61		1.61
62		1.62
63		1.63
64		1.64
65		1.65
66		1.66
67		1.67
68		1.68
69		1.69
70		1.70
71		1.71
72		1.72
73		1.73
74		1.74
75		1.75
76		1.76
77		1.77
78		1.78
79		1.79
80		1.80
81		1.81
82		1.82
83		1.83
84		1.84
85		1.85
86		1.86
87		1.87
88		1.88
89		1.89
90		1.90
91		1.91
92		1.92
93		1.93
94		1.94
95		1.95
96		1.96
97		1.97
98		1.98
99		1.99
100		2.00

单元一 HTML 的基本标签

在当今社会中,Web 页成为网络信息共享和发布的主要形式,而 HTML(Hyper Text Markup Language, 超文本标记语言或叫超文本标签语言)是用来描述网页的一种语言,是创建 Web 页的基础。HTML 不是一种编程语言,而是一种标记语言(Markup Language),标记语言是一套标记标签,HTML 使用标记标签来描述网页。HTML 标记标签通常被称为 HTML 标签,本单元将介绍 HTML 的基本结构、组成 HTML 文档的各类常用标签以及相关标准。本单元的重点是各类标签的基本语法,学习 HTML 最好的方式就是边学边做实验,多练习是记住这些标签及其语法的最好方法。

单元要点

- 静态网页的开发环境
- 行级标签和块级标签的使用

技能目标

- 能使用 Dreamweaver 编写 HTML 代码
- 能使用各种基本标签建立简单网页

1.1 工作场景导入

【工作场景 1】

实现李清照的如梦令和作者简介的网页效果,如工作场景图 1 所示。

【工作场景 2】

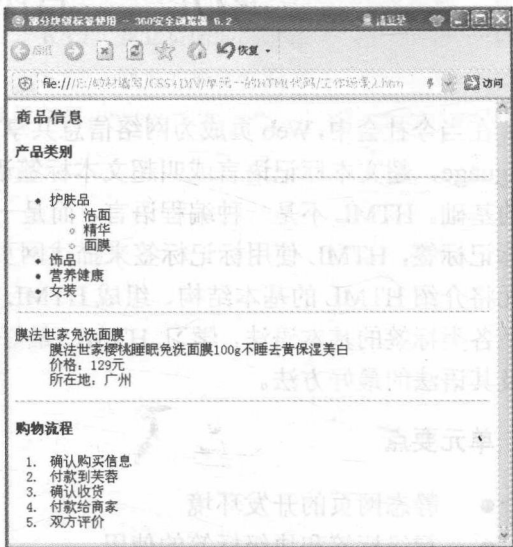
实现“产品信息”的网页效果,如工作场景图 2 所示。

【工作场景 3】

实现“促销信息”的网页效果,如工作场景图 3 所示。



工作场景图 1 李清照宋词赏析



工作场景图 2 商品信息



工作场景图 3 相宜本草促销信息

1.2 技术与知识准备

1.2.1 HTML 文件的基本结构

要深入学习 HTML，首先我们来了解一下什么是 HTML 以及 HTML 和浏览器的关系。

1. HTML 及其特点

HTML 被称为超文本标签语言,它包括很多标签,例如<p>段落标签、<h1>一级标题标签,告诉浏览器如何显示页面,它是网页制作的“核心语言”,是目前网络上应用最为广泛的语言,也是构成网页文档的主要语言。HTML 文档是由 HTML 标签和纯文本组成,HTML 标签可以说明段落、图形、表格、链接等,HTML 文档也被称为网页。它具备以下特点:

(1) 简易性:HTML 版本升级采用超集方式,从而更加灵活方便,并且各类 HTML 标签简单易学,方便网站制作者学习开发。

(2) 可扩展性:HTML 语言的广泛应用带来了加强功能、增加标识符等要求,HTML 采取子类元素的方式,为系统扩展提供保证。

(3) 平台无关性:这是 HTML 语言的最大优点,也是当今 Internet 盛行的原因之一。虽然 PC 机大行其道,但使用 MAC 等其他机器的大有人在,HTML 可以使用在广泛的平台上,它包括“硬件”平台无关性和“软件”平台无关性,不管你的计算机是普通的个人电脑,还是用于专业的苹果机,不管你的操作系统是常见的 Windows 还是高端 UNIX 或 Linux (一般用于服务器),HTML 文档都可以得到广泛的应用和传输。

2. HTML 和浏览器的关系

有了 HTML 源代码,还需要一个“解释和执行”的工具,而浏览器就是用来解释并执行显示 HTML “源码”的工具,随着市场的竞争,目前市场上的浏览器很多,主要有微软公司的 IE (Internet Explorer),网民大多数人都在使用 IE,这要感谢它对 Web 站点强大的兼容性。谷歌浏览器 Chrome 是由 Google 公司开发的网页浏览器,浏览速度在众多浏览器中走在前列,属于高端浏览器,谷歌浏览器在 2012 年 8 月份市场份额正式超过 IE 浏览器,跃居第一。Mozilla Firefox (火狐浏览器)现在是市场占有率第三的浏览器,仅次于微软的 Internet Explorer 和 Google 的 Chrome,搜狗浏览器是首款给网络加速的浏览器,本书使用 IE 8.0 和 360 安全浏览器进行介绍,关于浏览器的其他介绍学员可以上网获取更多资料。



图 1.1 常用浏览器图标

浏览器兼容性问题又被称为网页兼容性或网站兼容性问题,指网页在各种浏览器上的显示效果可能不一致而产生浏览器和网页间的兼容问题。在网站的设计和制作中,做好浏览器兼容,才能够让网站在不同的浏览器下都正常显示。而对于浏览器软件的开发和设计,浏览器对标准的更好兼容能够给用户更好的使用体验。规范化书写 HTML 可以解决浏览器兼容的问题,也可以避免给后台制作带来不必要的麻烦,随着课程的深入我们再予以介绍。

如果希望查看某个页面的“源代码”,我们可以通过点击浏览器(例如 IE)的菜单“查看”→“源文件”选项或右击,在弹出的菜单中选择“查看源文件”命令。下面将介绍 HTML 文档的基本结构。

3. HTML 的基本结构

HTML 文件的结构包括头部(head)、主体(body)两大部分,其中头部描述浏览器所需

的信息，而主体则包含所要说明的具体内容，如图 1.2 所示，整个 HTML 包括头部<head>和主体<body>两部分。头部包括网页标题（title）等基本信息，主体包括网页的内容信息（如所要呈现的图片、文字等）。注意大部分标签都以“<>”开始，以“</>”结束，要求成对出现，并且标签之间要有缩进，体现层次感，以便阅读和修改。

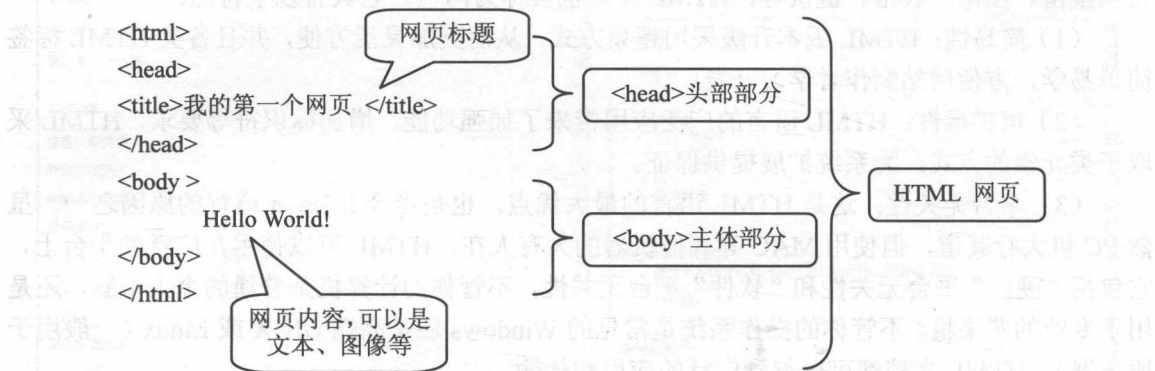


图 1.2 HTML 的基本结构

1.2.2 编辑工具

了解了 HTML 文档的基本结构以后，下面介绍常用的 HTML 代码编辑工具。

1. 记事本

记事本是 Windows 自带安装的编辑附件，使用简单方便，实际项目开发中常用于代码较少的编辑或维护。使用记事本编辑 HTML 文档的步骤如下：

- (1) 在 Windows 中打开记事本程序：开始→所有程序→附件→记事本。
- (2) 在记事本中输入 HTML 代码，如图 1.3 所示。



图 1.3 在记事本里编辑 HTML 网页

(3) 单击菜单“文件”→“保存”命令，弹出“另存为”对话框，如图 1.4 所示，将上述文档保存为*.html 的 html 文档，如 my-firstpage.html。需要注意，因记事本默认保存文档后缀名为“*.txt”，所以需要英文的双引号（"）将文件名括起来。既可以使用.htm 也可以使用.html 扩展名。两者没有区别，完全根据自己的喜好。

(4) 双击保存的 html 文档，Windows 将自动调动浏览器软件（如 IE）打开 html 文档，如图 1.5 所示，也可以先启动浏览器，然后选择“文件”→“打开文件”命令。



图 1.4 “另存为”对话框

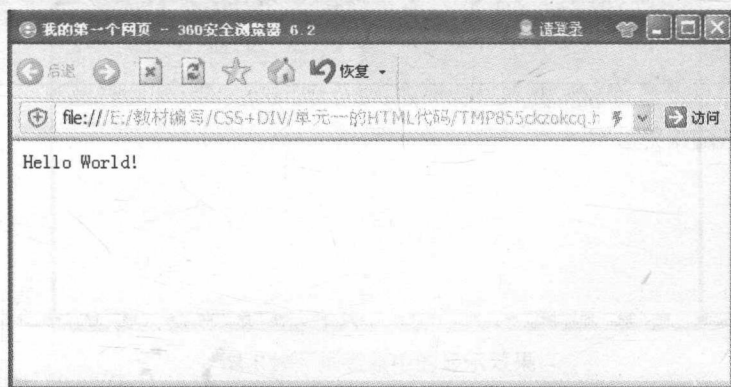


图 1.5 我的第一个网页

2. Dreamweaver

可以使用专业的 HTML 编辑器来编辑, 如 Adobe Dreamweaver、Microsoft Expression Web、CoffeeCup HTML Editor。Dreamweaver 是由 Macromedia 公司开发的一款所见即所得的网页编辑器, 与二维动画设计软件 Flash, 专业网页图像设计软件 Fireworks, 并称为“网页三剑客”。它有“所见即所得”的网页编辑器的优点, 即直观、使用方便、容易上手, 本书所有单元都将采用 Dreamweaver 作为 HTML 文档的编辑工具, 启动方法为: 单击桌面左下角“开始”→“程序”→“Macromedia”→“Macromedia Dreamweaver CSS”选中后单击就可以打开 Dreamweaver 了, 或者下载绿色版, 绿化后直接双击 Dreamweaver.exe 运行。

对于上述第一个网页在 Dreamweaver 中的编辑如下: 选择“文件”→“新建”命令, 单击“常规”选项卡, 如图 1.6 所示, 单击“创建”按钮, 弹出如图 1.7 所示的窗口, 一般情况选用拆分视图的编辑环境, 编辑好且保存之后如图 1.8 所示, 然后根据图 1.9 选择一种浏览器进行浏览, 浏览效果如图 1.10 所示。

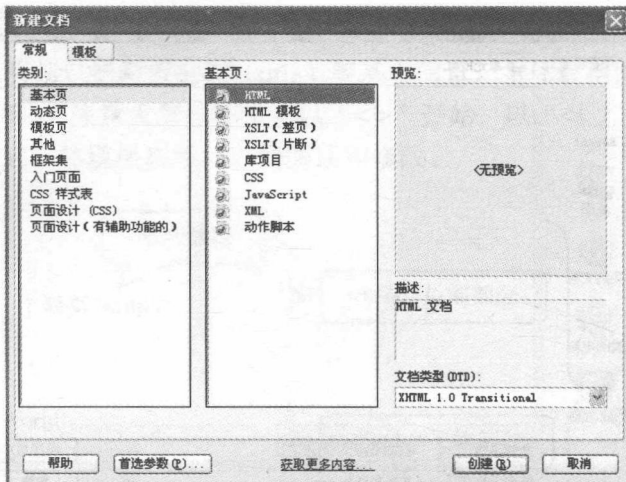


图 1.6 “常规”选项卡

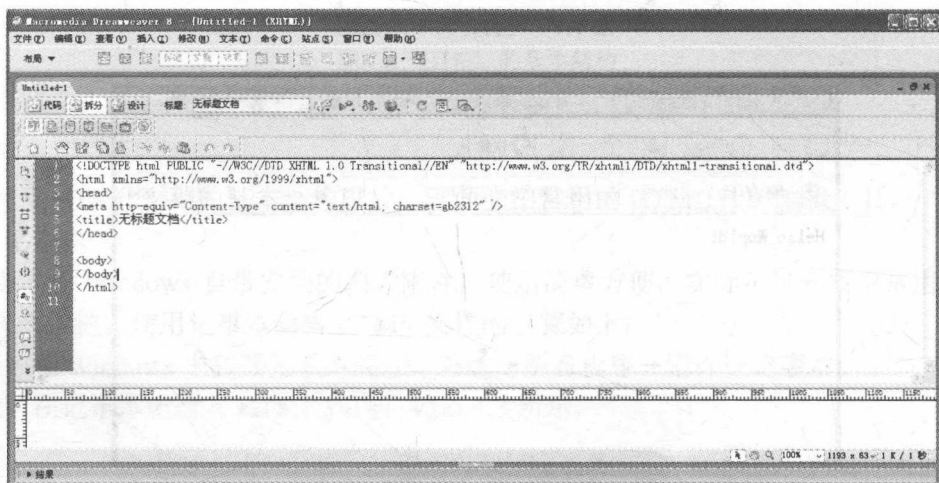


图 1.7 未编辑、保存的默认网页

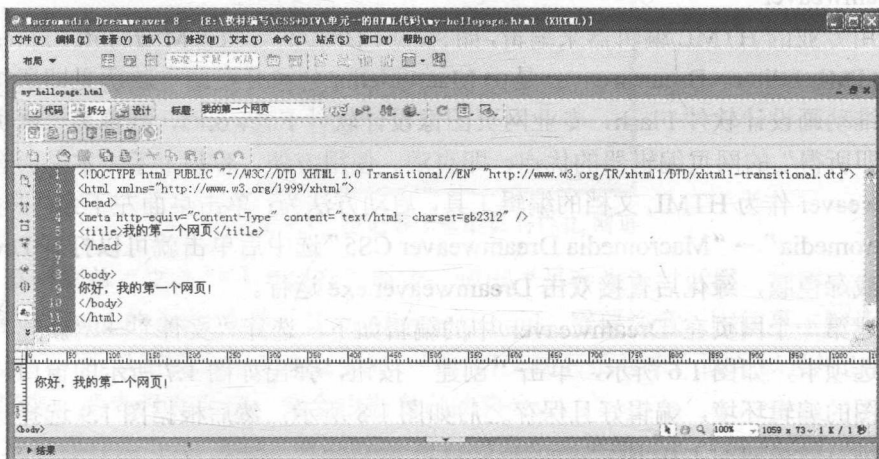


图 1.8 编辑、保存好的网页

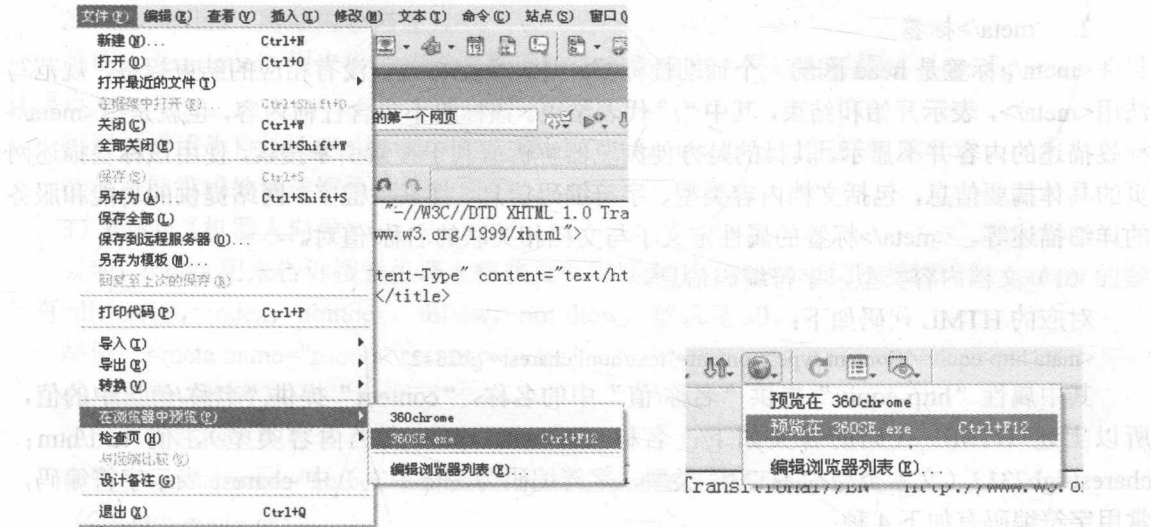


图 1.9 浏览器的选择

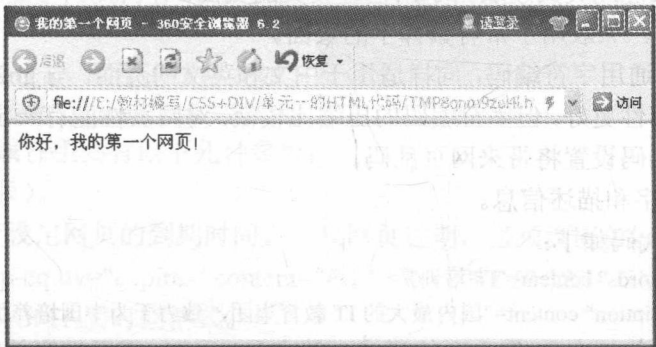


图 1.10 浏览器中的显示效果

3. 其他编辑器

除了记事本, Notepad (PC) 或 TextEdit (Mac) 也是文本编辑器。Notepad是在微软视窗下的一个纯文本编辑器。TextEdit (Mac) 是苹果电脑 Mac OS X 操作系统自带的 TextEdit (文本编辑器)。

1.2.3 网页的摘要信息

网页一般包含大量文字及图片等信息内容, 和报纸一样, 它需要一个简短的摘要信息, 方便用户浏览和查找。如果希望自己的网页能被百度、Google 等搜索引擎搜索, 或提高在搜索结果中的排名, 那么在制作网页时更需要注意编写网页的摘要信息。网页的摘要信息一般放在 HTML 文档的头部 (head) 区域内容, 主要通过如下两个标签进行描述。

1. <title>标签

使用该标签描述网页的标题, 类似一篇文章的标题, 一般为一个简洁的主题, 并能吸引读者有兴趣读下去。例如, 搜狐网站的主页, 对应的网页标题为: <title>搜狐-中国最大的门户网站</title>, 打开网页后, 将在浏览器窗口的标题栏显示网页标题。

2. <meta/>标签

<meta/>标签是 head 区的一个辅助性标签，是一个空标签，没有相应的结束标签，规范写法用<meta/>，表示开始和结束，其中“/”代表结束。该标签不包含任何内容，也就是指<meta/>标签描述的内容并不显示，其目的是方便浏览器解析或利于搜索引擎搜索，使用该标签描述网页的具体摘要信息，包括文档内容类型、字符编码信息、搜索关键字、网站提供的功能和服务的详细描述等。<meta/>标签的属性定义了与文档相关联的名称/值对。

(1) 文档内容类型，字符编码信息。

对应的 HTML 代码如下：

```
<meta http-equiv="Content-type" content="text/html; charset= gb2312"/>
```

其中属性“http-equiv”提供“名称/值”中的名称，“content”提供“名称/值”中的值，所以上述 HTML 代码的含义如下：名称 Content-Type（文档内容类型），值 text/html；charset=gb2312（文本类型的 HTML 类型，字符编码为简体中文）中 charset 表示字符编码，常用字符编码有如下 4 种：

- gb2312：简体中文，一般用于包含中文和英文的页面。
- ISO-885901：纯英文一般用于只包含英文的页面。
- big5：繁体，一般用于带有繁体字的页面。
- utf-8：国际通用字符编码，同样适用于中文和英文的页面。与 gb2312 编码相比，utf-8 的国际通用性更好，但字符编码的压缩比较低，对网页性能有一定的影响。需要注意，不正确的编码设置将带来网页乱码。

(2) 搜索关键字和描述信息。

对应的 HTML 代码如下：

```
<meta name="keywords" content="IT 培训"/>
```

```
<meta name="description" content="国内最大的 IT 教育集团，致力于为中国培养优秀的 IT 技术人才"/>
```

实现的方式仍然为“名称/值”对的形式，其中 keywords 表示搜索关键字，description 表示网站内容的具体描述。通过提供搜索关键字和内容描述信息，方便搜索引擎的搜索。

meta 标签共有两个属性，它们分别是 name 属性和 http-equiv 属性，不同的属性有不同的参数值，这些不同的参数值就实现了不同的网页功能。

(1) name 属性。

name 属性主要用于描述网页，与之对应的属性值为 content，content 中的内容主要是便于搜索引擎机器人查找信息和分类信息用的。

meta 标签的 name 属性语法格式如下：

```
<meta name="参数" content="具体的参数值">
```

其中 name 属性主要有以下几种参数：

1) keywords（关键字）。

说明：keywords 用来告诉搜索引擎你网页的关键字是什么。

举例：<meta name="keywords" content="政治,经济,科技,文化,卫生,情感,心灵,娱乐,生活,社会,企业,交通">，按照搜索引擎的工作原理，搜索引擎首先派出机器人自动检索页面中的 keywords，设置好这些关键字后，搜索引擎将会自动把这些关键字添加到数据库中，并根据这些关键字的密度来进行合适的排序。因此，我们必须设置好关键字，来提高页面的搜索点击率。

2) description (网站内容描述)。

说明: description 用来告诉搜索引擎网站的主要内容。网站内容描述 (description) 的设计要点:

①网页描述为自然语言而不是罗列关键词 (与 keywords 设计正好相反)。

②尽可能准确地描述网页的核心内容。

3) robots (机器人向导)。

说明: robots 用来告诉搜索机器人哪些页面需要索引, 哪些页面不需要索引。content 的参数有 all, none, index, noindex, follow, nofollow, 默认是 all。

举例: `<meta name="robots" content="none">`。

4) author (作者)。

说明: 标注网页的作者。

举例: `<meta name="author" content="zys666,zys666@21cn.com">`

(2) http-equiv 属性。

http-equiv 顾名思义, 相当于 http 的文件头作用, 它可以向浏览器传回一些有用的信息, 以帮助正确和精确地显示网页内容, 与之对应的属性值为 content, content 中的内容其实就是各个参数的变量值。

meta 标签的 http-equiv 属性语法格式如下:

`<meta http-equiv="参数" content="参数变量值">`

其中 http-equiv 属性主要有以下几种参数:

1) Expires (期限)。

说明: 可以用于设定网页的到期时间。一旦网页过期, 必须到服务器上重新传输。

用法: `<meta http-equiv="expires" content="Fri,12 Jan 2001 18:18:18 GMT">`。

注意: 必须使用 GMT 的时间格式。

2) Pragma (cache 模式)。

说明: 禁止浏览器从本地计算机的缓存中访问页面内容。

用法: `<meta http-equiv="Pragma" content="no-cache">`。

注意: 这样设定, 访问者将无法脱机浏览。

3) Refresh (刷新)。

说明: 自动刷新并指向新页面。

用法: `<meta http-equiv="Refresh" content="2;URL=http://www.root.net">` (注意后面的引号, 分别在秒数的前面和网址的后面)。

注意: 其中的 2 是指停留 2 秒钟后自动刷新到 URL 网址。

4) Set-Cookie (cookie 设定)。

说明: 如果网页过期, 那么存盘的 cookie 将被删除。

用法: `<meta http-equiv="Set-Cookie" content="cookievalue=xxx; expires=Friday,12-Jan-2001 18:18:18 GMT; path=/">`。

注意: 必须使用 GMT 的时间格式。

5) Window-target (显示窗口的设定)。

说明: 强制页面在当前窗口以独立页面显示。

用法: `<meta http-equiv="Window-target" content="_top">`。

注意: 用来防止别人在框架里调用自己的页面。

6) Content-Type (显示字符集的设定)。

说明: 设定页面使用的字符集。

用法: `<meta http-equiv="Content-Type" content="text/html; charset=gb2312">`

7) Content-Language (显示语言的设定)。

用法: `<meta http-equiv="Content-Language" content="zh-cn" />`

在团队开发中, 需要注意养成良好的习惯。书写顺序应该成对书写, 从外层写到内层, 结构清晰, 利于阅读和调试排错。例如编写:

```
<html>
  <head>
    <title>标题</title>
  </head>
</html>
```

推荐书写顺序如下:

① `<html></html>`

② 写入 `head` 再排版缩进(默认缩进为 2 个空格或 4 个空格, 各个团队的具体规定不一样)。

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

③ `title` 先写标签, 再写里面的内容:

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

④ 同理再编写 `<body>` 中的其他标签。

1.2.4 块级标签

介绍了头部 (`head`) 的常用标签后, 下面再介绍主体 (`body`) 内常用的各类标签, 如图 1.11 所示, 从页面布局和显示外观的角度看, 一个页面的布局就类似一篇报纸的排版, 需要分为多个区块, 大的区块再细分为小区块。块内为多行逐一排列的文字、图片、超链接等内容, 这些区块一般称为块级元素, 而区块内的文字、图片或超链接等一般称为行级元素, 页面这种布局结构, 其本质上是由各种 HTML 标签组织完成的, 因此本文将 HTML 标签分为相应的块级标签和行级标签 (有的教材也称为块级元素和行级元素), 以方便理解后续单元讲解的页面布局。

顾名思义, 块级标签显示的外观按“块”显示, 具有一定的高度和宽度, 例如 `<div>` 块级标签、`<p>` 段标签等, 行级元素显示的外观按“行”显示, 类似文本的显示, 例如 `<img>` 图片标签、`<a>` 超链接标签等。和行级标签相比, 块级标签具有如下特点:

(1) 前后断行显示, 如图 1.11 所示。块级标签比较“霸道”, 默认状态占据一整行。

(2) 具有一定的宽度和高度, 可以通过设置块级标签 `width`、`height` 属性来控制。

块级标签常用于作容器,即可再容纳其他块级标签和行级标签,而行级标签一般用于组织内容,即只能用于“容纳”文字、图片或其他行级标签。



图 1.11 页面中的块级元素和行级元素

从页面布局的角度,块级标签又可细分为基本的块级标签和常用于页面布局的块级标签。

1.2.4.1 基本的块级标签

基本的块级标签包括标题标签、段落标签及水平线标签。

1. 标题标签<h1>~<h6>

在 HTML 文档中,标题很重要,是通过 <h1>~<h6> 等标签进行定义的。<h1> 定义最大的标题,<h6>定义最小的标题,浏览器会自动地在标题的前后添加空行,默认情况下,HTML 会自动地在块级元素前后添加一个额外的空行,比如段落、标题元素前后。因为用户可以通过标题来快速浏览您的网页,所以用标题来呈现文档结构是很重要的。应该将 h1 用作主标题(最重要的),其后是 h2 (次重要的),再其次是 h3,以此类推。例如,一级标题采用<h1>,如还有二级标题采用<h2>,HTML 共提供了六级标题,并赋予了标题一定的外观:所有标题字体加粗。<h1>字号最大,<h6>字号最小。这 6 个标题标签都有对应的结束标签,通过结束标签来关闭。HTML 清楚地标记某个元素在何处开始,并在何处结束,不论对您还是对浏览器来说,都会使代码更容易理解。例如,如图 1.12 所示,其所对应的 HTML 代码如示例 1.1 所示,代码在浏览器中的效果就是图 1.12 的效果。

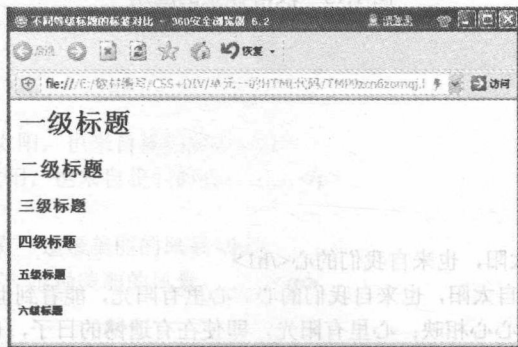


图 1.12 不同级别的标题输出的结果