



工业和信息化人才培养规划教材
高职高专计算机系列

◎ 温谦 周建国 练源 编著

网页设计与布局项目化教程 (HTML+CSS+DIV)



Marcello Brandi, Peter's Blue. Originally built in 1958 by architect Leopoldo Rodolfo. The night scene captures the building's illuminated facade and the palm trees in the foreground. The image is a high-quality photograph, likely from a professional photography portfolio.



光盘里包含
案例素材、效果文件



6 大完整商业案例逐步拆分详解

剖析摄影师个人网站、生物研究中心网站、教育公司网站、网上书店、汽车服务公司网站、橘汁仙剑游戏网站六大原创网站的布局设计，从原型设计到布局细节，逐步拆讲解，零基础读者也可一步步轻松实现专业的 CSS 页面布局。

4 大 CSS 核心技术精髓实践应用

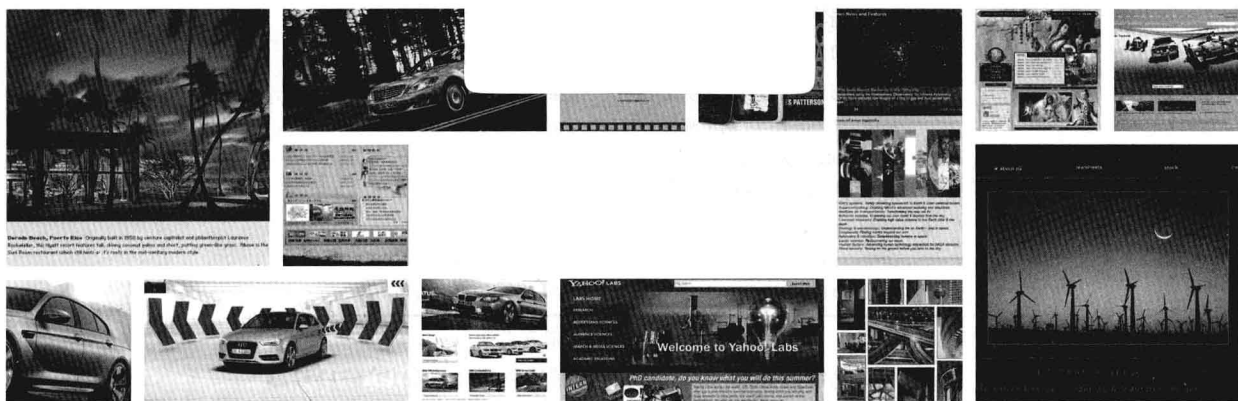
重点讲解 CSS 盒子模型、标准流、浮动、定位四大核心技术原理在实践中的应用，将不可或缺的 CSS 核心技术融会在实际案例中，直达技术精髓，加入实用技巧，明确工作流程，帮助读者真正做到理论与实践结合，提高页面布局能力。



工业和信息化人才培养规划教材
高职高专计算机系列

© 温谦 周建国 练源 编著

网页设计与布局项目化教程 (HTML+CSS+DIV)



人民邮电出版社
北京

图书在版编目(CIP)数据

网页设计与布局项目化教程: HTML+CSS+DIV / 温谦, 周建国, 练源编著. — 北京: 人民邮电出版社, 2013.9

工业和信息化人才培养规划教材. 高职高专计算机系列

ISBN 978-7-115-31867-1

I. ①网… II. ①温… ②周… ③练… III. ①网页制作工具—高等职业教育—教材 IV. ①TP393.092

中国版本图书馆CIP数据核字(2013)第207454号

内 容 提 要

本书以六大完整案例为线索, 紧密围绕在使用CSS+DIV布局制作网页过程中遇到的实际需要和应该掌握的技术, 全面介绍了CSS布局网页各方面的内容和技巧。

书中摒弃了案例书常见的一味罗列步骤, 轻视技术原理的做法, 把案例作为学习手段, 不但侧重讲解CSS盒子模型、标准流、浮动、定位四大核心技术原理, 更侧重讲解如何在实践中应用它们。读者通过案例, 既可以掌握CSS的核心技术要点, 也可以了解相关技术和工作流程, 从而做到不但知其然, 还知其所以然。

此外, 本书还结合流行的CMS内容管理系统, 讲解了使用CSS进行模板设计的方法, 使读者可以举一反三地掌握为各种内容管理系统设计模板的方法。

本书可作为高等职业院校网页设计类课程教材, 也适合具备一定HTML和网页设计制作基础的需要使用CSS的爱好者、Web设计人员和开发人员阅读。

◆ 编 著 温 谦 周建国 练 源

责任编辑 王 威

责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号

邮编 100061 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京艺辉印刷有限公司印刷

◆ 开本: 787×1092 1/16 彩插: 2

印张: 16 2013年9月第1版

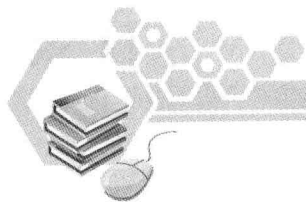
字数: 411千字 2013年9月北京第1次印刷

定价: 49.80元(附光盘)

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

广告经营许可证: 京崇工商广字第0021号



CSS和HTML一样,是一个非常基础性的规范,是任何一个Web站点设计过程中都离不开的“基础技术”。目前,我国很多高职院校都将“CSS”作为一门重要的专业课程。为了帮助高职院校的教师全面、系统地讲授这门课程,使学生能够熟练地使用CSS进行网页美化,我们组织长期在高职院校从事CSS教学的教师和专业网页设计公司经验丰富的设计师共同编写了本书。

本书以“案例”讲解为手段,使学生能够真正理解为什么要这么做,只有这样,才能使学生真正掌握。在各章中,我们把CSS的核心技术原理的讲解放在首位让学生明白如何在实践中使用某个原理;在实际制作时,需要注意的相关技术和工作流程方面重点有机地融合在一个案例中。通过案例的分析和学习,拓展学生的实际应用能力。在内容方面,力求细致全面、重点突出;在文字叙述方面,做到言简意赅、通俗易懂;在案例选取方面,强调案例的针对性和实用性。

本书还讲解了流行的CMS内容管理系统的模板设计方法。当前建立网站通常有两种方式,一种是完全根据需要定制开发,另一种是使用现成的内容管理系统,简称为CMS(Content Management System)。而当前也出现了很多非常流行的内容管理系统,很多网站正是使用CMS系统建立起来的,而要使用好CMS系统,就需要对CMS的模板机制和利用CSS来进行定制模板的方法熟练掌握。本书的后两章,就是专门针对使用CMS进行模板定制来讲解的。

本书配套光盘包含了书中所有案例的素材及效果文件。另外,为方便教学,本书配备了详尽的教学资源,任课老师可登录人民邮电出版社教学服务与资源网(www.ptpedu.com.cn)免费下载使用。本书的参考学时为29学时,各章的参考学时参见下面的学时分配表。

章 节	课 程 内 容	学 时 分 配	
		讲 授	实 训
第1章	从基础开始	2	1
第2章	摄影师个人网站布局	2	1
第3章	生物研究中心网站布局	2	1
第4章	教育公司网站布局	2	1
第5章	网上书店布局	2	1
第6章	汽车服务公司网站布局	3	1
第7章	橘汁仙剑游戏网站(静态)布局	3	2
第8章	橘汁仙剑游戏网站(动态)布局	3	2
课时总计		19	10

由于水平有限,书中难免存在错误和不妥之处,敬请广大读者批评指正。



目录

第1章 从基础开始 1

1.1 (X)HTML 与 CSS 2

1.1.1 DOCTYPE (文档类型) 的含义 与选择 2

1.1.2 XHTML 与 HTML 的重要区别 3

1.1.3 CSS 的基本思想 3

1.1.4 CSS 的引入 5

1.1.5 浏览器与 CSS 5

1.2 CSS 的基本选择器 6

1.2.1 理解 CSS 选择器的思想 6

1.2.2 标记选择器 7

1.2.3 类别选择器 8

1.2.4 ID 选择器 9

1.3 在 HTML 中引入 CSS 的方法 10

1.3.1 行内样式 10

1.3.2 内嵌式 11

1.3.3 外部样式表 11

1.3.4 链接式 12

1.3.5 各种方式的优先级问题 12

1.4 动手体验 CSS 14

1.4.1 从零开始 14

1.4.2 使用 CSS 设置标题 15

1.4.3 控制图片 16

1.4.4 设置正文 16

1.4.5 设置整体页面 17

1.4.6 对段落分别进行设置 17

1.4.7 兼容性检查 18

1.4.8 CSS 的注释 19

1.5 网页使用的编辑软件 20

1.5.1 可视化网页制作软件的优点 20

1.5.2 可视化软件的局限性 21

1.5.3 善于使用代码视图的功能 22

1.6 CSS 的复合选择器 24

1.6.1 交集选择器 24

1.6.2 并集选择器 25

1.6.3 后代选择器 27

1.7 CSS 的继承特性 29

1.7.1 继承关系 29

1.7.2 CSS 继承的运用 31

1.8 CSS 的层叠特性 33

1.9 本章小结 35

第2章 摄影师个人网站布局 36

2.1 案例描述 37

2.2 内容分析 38

2.3 HTML 结构设计 40

2.4 原型设计 41

2.5 页面方案设计 43

2.6 布局设计 43

2.7 CSS 技术准备——盒子模型 44

2.8 设置页面的整体背景 46

2.9 制作照片展示区域 48

2.10 设置网页标题的图像替换 50

2.11 CSS 技术准备——定位 52

2.11.1 理解标准文档流 52

2.11.2 认识定位属性 53

2.11.3 绝对定位 54

2.12 设置网页标题的位置 55

2.13 设置网页文本内容 56

2.14 本章小结 58

第3章 生物研究中心网站布局 59

3.1 案例描述 60

3.2 内容分析 61

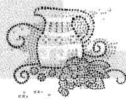
3.3 原型设计 62

3.4 页面方案设计 63

3.5 CSS 技术准备 ——盒子的浮动 63



3.5.1 设置浮动	64	4.8.3 顶部菜单	102
3.5.2 浮动的方向	66	4.8.4 主菜单	103
3.5.3 使用 clear 属性清除 浮动的影响	67	4.8.5 搜索框	104
3.5.4 扩展盒子的高度	69	4.8.6 页头部分的圆角框	105
3.6 布局设计	70	4.9 制作主体部分	107
3.7 CSS 技术准备—— 在 CSS 中设置边框	70	4.9.1 结构分析	107
3.7.1 对不同的边框设置 不同的属性值	71	4.9.2 面包屑导航	109
3.7.2 在一行中同时设置 边框的宽度、颜色和样式	71	4.9.3 设置正文标题	109
3.7.3 对一条边框设置与 其他边框不同的属性	72	4.9.4 设置页脚	110
3.7.4 同时指定一条边框的 一种属性	72	4.9.5 添加页面内容	110
3.8 制作页头部分	72	4.10 本章小结	112
3.9 制作主体部分	75	第 5 章 网上书店布局	113
3.9.1 主体的左侧部分	76	5.1 案例描述	114
3.9.2 主要内容区	79	5.2 内容分析	116
3.10 CSS 技术扩展——扩充布局	82	5.3 HTML 结构设计	118
3.11 本章小结	85	5.4 原型设计	121
第 4 章 教育公司网站布局	86	5.5 页面方案设计	122
4.1 两列布局	87	5.5.1 配色的技巧	123
4.2 案例描述	89	5.5.2 切片的技巧	125
4.3 内容分析	90	5.6 使用滑动门技术制作 导航菜单	128
4.4 原型设计	91	5.7 制作主体部分	131
4.5 CSS 技术准备—— 在 CSS 中使用背景图像	92	5.7.1 整体样式设计	131
4.5.1 设置平铺方式	92	5.7.2 内容部分的结构分析	131
4.5.2 设置背景图像的位置	94	5.7.3 设置右侧的主要内容列	133
4.5.3 背景的简写	96	5.7.4 制作左边栏	136
4.5.4 图像的固定设置	97	5.8 总结 CSS 布局的优点	139
4.6 制作标题图像	97	5.9 制作可以适应变化宽度的 圆角框	140
4.7 CSS 技术准备—— 实现圆角设计	99	5.10 CSS 技术扩展—— 从“网页”到“网站”	143
4.8 制作页头部分	100	5.10.1 历史回顾	143
4.8.1 搭建页头部分的 HTML 结构	100	5.10.2 不完美的办法	144
4.8.2 页面标题的图像替换	102	5.10.3 服务器出场	144
		5.10.4 CMS 出现	144
		5.10.5 具体操作	144
		5.10.6 CMS 的弊端	145
		5.11 本章小结	145
		第 6 章 汽车服务公司 网站布局	146



6.1 案例描述	147
6.2 内容分析	148
6.3 HTML 结构设计	151
6.4 原型设计	153
6.5 页面方案设计与切图	154
6.6 页面布局	155
6.6.1 切片	156
6.6.2 CSS 技术准备—— 行内元素与块级元素	157
6.6.3 布局	160
6.6.4 制作顶部菜单	160
6.6.5 制作标题图像	162
6.6.6 制作主体部分	162
6.7 实现超链接特效	164
6.7.1 技术准备—— 设置超链接的 CSS 样式	165
6.7.2 超链接效果	168
6.8 兼容性检查	169
6.9 本章小结	170

第 7 章 橘汁仙剑游戏网站

(静态) 布局 171

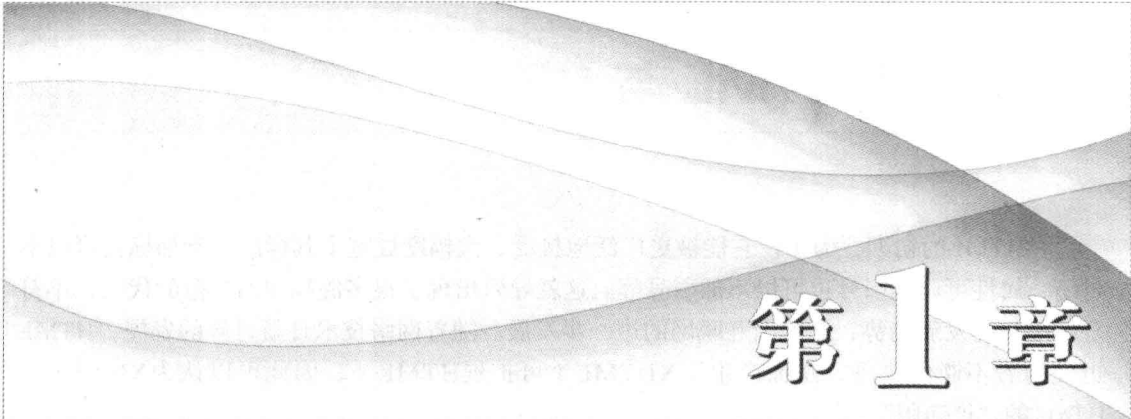
7.1 构思设计	172
7.1.1 站点分析定位	172
7.1.2 学习考察同类站点	173
7.1.3 构思规划站点	176
7.2 切片制作和生成	178
7.2.1 切片的制作	178
7.2.2 切片的生成	180
7.3 页面制作	180
7.3.1 整体框架的构建	181
7.3.2 头部的制作	182
7.3.3 首页左侧信息栏的制作	185
7.3.4 首页中部内容栏的制作	187
7.3.5 在页面右侧添加 百度搜索	188
7.3.6 在页面右侧添加广告位	192
7.3.7 分类目录中导航的制作	192
7.3.8 文章浏览区域的制作	194
7.3.9 页脚的制作	197
7.3.10 用户面板的制作	199

7.4 本章小结	202
----------------	-----

第 8 章 橘汁仙剑游戏网站

(动态) 布局 203

8.1 SupeSite 和 Discuz! 系统简介	204
8.2 系统安装	205
8.3 使用 SupeSite 系统	206
8.3.1 登录 SupeSite 后台设置	206
8.3.2 基本设置	208
8.3.3 资讯的发布和管理	210
8.3.4 资讯等级审核	212
8.3.5 资讯自定义字段	213
8.3.6 其他设置	215
8.4 制作 SupeSite 模板	215
8.4.1 SupeSite 模板系统	215
8.4.2 选择需要制作的模板	216
8.4.3 制作前的准备	218
8.4.4 首页头部信息的制作	220
8.4.5 首页头部导航的制作	221
8.4.6 体验 SupeSite 模块设置	223
8.4.7 在头部导航中加入 资讯分类	226
8.4.8 首页主体内容的制作	228
8.4.9 深入探究 SupeSite 模块系统	230
8.4.10 首页页脚的制作	232
8.4.11 站点头部及页脚文件的 制作	234
8.4.12 分类目录页的制作	236
8.4.13 文章浏览页面的制作	237
8.4.14 用户面板的制作	239
8.4.15 SupeSite 模板制作小结	241
8.5 模块系统的高级应用	242
8.5.1 SupeSite 模块系统的 语法格式	242
8.5.2 条件判断语句	244
8.5.3 自定义广告显示函数	246
8.6 完成测试	247
8.7 Discuz! 模板系统简介	247
8.8 本章小结	250



第1章

从基础开始

制作网页的基础是使用HTML语言。实际上使用HTML非常简单，其核心思想就是需要设置什么样式，就使用相应的HTML标记或者属性。然而仅仅依靠HTML会遇到很多难以解决的问题，为此HTML逐步发展到了XHTML，CSS也应运而生。因此在本章中，将简单介绍HTML，XHTML和CSS三者之间的关系，以及CSS的基础知识，读者须重点理解使用CSS的核心原理。

课堂学习目标

- (X)HTML与CSS
- CSS的基本选择器
- 在HTML中引入CSS的方法
- 动手体验CSS
- 网页使用的编辑软件
- CSS的复合选择器
- CSS的继承特性
- CSS的层叠特性



1.1 (X)HTML 与 CSS

在HTML的初期,为了使它能被更广泛地接受,大幅度放宽了其标准,例如标记可以不封闭、属性可以加引号也可以不加引号等。这就导致出现了很多混乱和不规范的代码,不符合标准化的发展趋势,影响了互联网的进一步发展。随着网络技术日新月异的发展,HTML也经历着不断的改进,从而产生了XHTML(可扩展HTML),因此可以认为XHTML是HTML的“严谨版”。

XHTML是由W3C组织(World Wide Web Consortium,全球万维网联盟)负责制定的。W3C的主要职责是研究Web规范和指导方针,推动Web发展,确定万维网的发展方向,并且制定相关的建议。它负责制定了CSS,XML,XHTML和MathML等多种网络技术规范。

从HTML到XHTML,经历了若干版本。目前HTML最高到4.01版,XHTML到1.1版。

1.1.1 DOCTYPE (文档类型) 的含义与选择

由于同时存在不同的规范和版本,因此为了使浏览器能够兼容多种规范,规范中规定可以使用DOCTYPE指令来声明使用哪种规范解释该文档。目前,常用HTML或者XHTML作为文档类型,而规范又规定HTML和XHTML又各自有不同的子类型,例如包括“严格类型”和“过渡类型”。

其中,“过渡类型”是兼容以前版本定义,而在新版本已经废弃的标记和属性,“严格类型”则不兼容已经废弃的标记和属性。

目前,建议读者使用XHTML 1.0 transitional类型(XHTML 1.0过渡类型),这样设计师就可以按照XHTML的标准书写符合Web标准的网页代码,同时在一些特殊情况下还可以使用传统的做法。具体声明方法如下面的这段代码所示。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title> 无标题文档 </title>
  </head>
  <body>
  </body>
</html>
```

可以看到最上面有两行关于“DOCTYPE”(文档类型)的声明,它就是告诉浏览器,使用XHTML 1.0的过渡规范来解释这个文档中的代码。在第3行中,<html>标记带有一个xmlns属性,它被称为“XML命名空间”,其具体含义不用深究、不用修改,只要照抄即可。



如果读者觉得这些代码难以记忆,可以使用Dreamweaver或者Expression Web等网页制作辅助软件,在新建文档的时候选择使用哪种文档类型,这些文档类型代码就会自动生成,不需要记住具体代码。

例如,在Dreamweaver的【新建文档】对话框中,在右下方有一个【文档类型】下拉框,如图1.1所示。

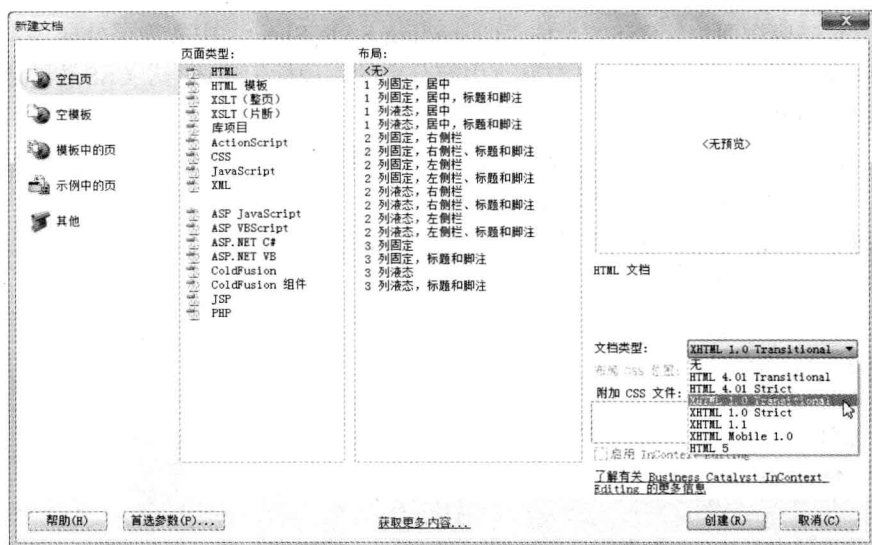


图 1.1 在 Dreamweaver 中选择文档类型

1.1.2 XHTML 与 HTML 的重要区别

XHTML和HTML最重要的区别是许多HTML中定义的标记和属性已经被废止。此外还有以下注意事项。

- (1) 在XHTML中标记名称必须小写。
- (2) 在XHTML中属性名称必须小写。
- (3) 在XHTML中标记必须严格嵌套。
- (4) 在XHTML中标记必须封闭。
- (5) 在XHTML中空元素的标记也必须封闭。
- (6) 在XHTML中属性值用双引号括起来。
- (7) 在XHTML中属性值必须使用完整形式。

1.1.3 CSS 的基本思想

CSS出现的根本原因在于,HTML中的内容与表现代码混杂在一起,导致出现了以下几点仅靠HTML难以解决的问题。

(1) 维护困难。为了修改某个特殊标记(如<h2>标记)的格式,需要花费很多时间,尤其对于整个网站而言,后期修改和维护的成本很高。



(2) 标记不足。HTML本身的标记十分少,很多标记都是为网页内容服务的,而关于美工样式的标记(如文字间距、段落缩进等)在HTML中很难找到。

(3) 网页过“胖”。由于没有对各种风格样式进行统一控制,因此HTML的页面往往体积过大,占用了很多宝贵的带宽。

(4) 定位困难。在整体布局页面时,HTML对于各个模块的位置调整显得捉襟见肘,过多的其他标记同样也导致页面的复杂和后期维护的困难。

为此,CSS应运而生。CSS的全称是Cascading Style Sheet,中文为层叠样式表,它是用于控制网页样式并允许将样式信息与网页内容分离的一种标记性语言。CSS最早是1996年由W3C审核通过并推荐使用的。

XHTML与CSS的关系就是“内容结构”与“表现形式”的关系,由XHTML确定网页的结构内容,而通过CSS来决定页面的表现形式。

为了理解CSS的用法,在具体使用CSS之前,请读者先思考一个生活中的问题:通常我们是如何描述一个人的?我们可以为一个人列一张表:

```
李逵 {  
    身高: 185cm;  
    体重: 105kg;  
    性别: 男;  
    性格: 莽撞;  
}
```

这个表实际上是由3个要素组成的,即“姓名”、“属性”和“属性值”。通过这样一张表,就可以把一个人的基本情况描述出来了。表中每一行分别描述了一个人的某一种属性以及该属性的属性值。

CSS的作用就是设置网页各个组成部分的表现形式。因此,如果把上面的表格换成描述网页上一个标题的属性表,可以设想大致会是这个样子:

```
2 级标题 {  
    字体: 宋体;  
    大小: 15 像素;  
    颜色: 红色;  
}
```

再进一步,如果我们把上面的表格用英文写出来,则如下:

```
h2 {  
    font-family: 宋体;  
    font-size: 15px;  
    color: red;  
}
```

这就是完全正确的CSS代码了。由此可见,CSS的原理实际上非常简单,对于使用英语的人来说,写CSS代码几乎和使用自然语言一样简单。而对于我们中国人,只要理解了这些属性的含义,就并不复杂,相信每一位读者都可以掌握它。

CSS的思想就是首先指定对什么“对象”进行设置,然后指定对该对象哪个方面的“属性”进行设置,最后给出该设置的“值”。因此,概括来说,CSS就是由“对象”、“属性”和“值”3个基本部分组成的。



1.1.4 CSS 的引入

下面看一个具体的页面，代码如下：

```
<html>
<head>
  <title> 演示 </title>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</style>
p{
  color:blue;
}
</style>
</head>
<body>
  <h2> 这是标题文本 </h2>
  <p> 这里是正文内容 </p>
  <p> 这里是正文内容 </p>
</body>
</html>
```

可以看到，这个页面由1个标题和2个文本段落构成，在HTML中没有设置任何font属性，而在页面的head部分，使用了<style>标记以及其中对<p>标记的定义，即：

```
p{
  color:blue;
}
```

还可以看到，页面中的两个文本段落都是用蓝色显示的，这就是CSS产生的作用，如图1.2所示。

从这个很简单的例子中可以明显看出，CSS对于网页的整体控制较单纯的HTML语言有了突破性的进展，并且它的后期修改和维护都十分方便。不仅如此，CSS还提供了各种丰富的格式控制方法，使得网页设计者能够轻松地应对各种页面效果，这些都将在后面的章节中逐一讲解。

最核心的变化就是，原来由HTML同时承担的“内容”和“表现”双重任务，现在分离开了，“内容”仍然由HTML负责，而“表现形式”则是通过<style>标记中的CSS代码负责的。当然，由于还没有介绍CSS的具体用法，因此以上代码的具体内容读者可能还无法清晰地理解，但是读者只要明白其中的原理即可。

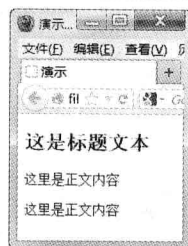


图 1.2 设置 CSS 样式后的效果

1.1.5 浏览器与 CSS

网上的浏览器各式各样，绝大多数浏览器对CSS都有很好的支持，因此设计者不用担心其设计的CSS文件不能显示。但目前主要的问题在于，各个浏览器在对CSS很多细节的处理上存在差异，设计者在一种浏览器上设计的CSS效果，在其他浏览器上的显示效果很可能会不一样。就目前主流的两大浏览器IE（Internet Explorer）与Firefox而言，在某些细节的处



理上就不尽相同。IE 6与IE 7对相同页面的浏览效果也存在一些差异。图1.3所示分别是IE和Firefox的标志。

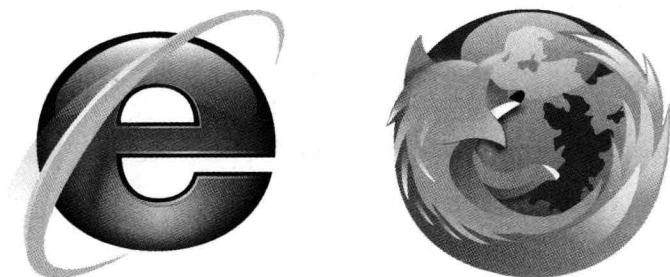


图 1.3 IE 和 Firefox 的标志

就目前而言，使用最多的3种浏览器是IE 6、IE 7和Firefox，制作网页后应该进行调整，使页面在IE6、IE7和Firefox这3个浏览器中都显示正确，这样可以保证99%以上的访问者正确浏览该网页。

但出现各个浏览器效果上的差异，主要是因为各个浏览器对CSS样式默认值的设置不同，因此可以通过对CSS文件各个细节的严格编写，使得各个浏览器达到基本相同的效果。这点在后续的章节中都会提到。

经验

使用CSS制作网页时，一个基本的要求就是网页在主流的浏览器中的显示效果要基本一致。通常的做法是一边编写HTML和CSS代码，一边在两个不同的浏览器上进行预览，以便及时地调整各个细节，这对深入掌握CSS也是很有好处的。

另外Dreamweaver的“视图”模式只能作为设计时的参考来使用，绝对不能作为最终显示效果的依据，只有浏览器中的效果才是大家所看到的。

1.2 CSS 的基本选择器

在CSS的3个组成部分中，“对象”是很重要的，它指定了对哪些网页元素进行设置，因此，它有一个专门的名称——选择器（selector）。

选择器是CSS中很重要的概念，所有HTML语言中的标记样式都是通过不同的CSS选择器进行控制的。用户只需要通过选择器对不同的HTML标签进行选择，并赋予各种样式声明，即可实现各种效果。

1.2.1 理解 CSS 选择器的思想

为了理解选择器的概念，可以以“地图”作为类比。在地图上都可以看到一些“图



例”，如河流用蓝色的线表示，山峰用三角形表示，省会城市用黑色圆点表示等，如图1.4所示。

本质上，这就是一种“内容”与“表现形式”的对应关系。而在网页上，也同样存在着这样的对应关系，例如h1标题用蓝色文字表示，h2标题用红色文字表示。因此为了能够使CSS规则与HTML元素对应起来，就必须定义一套完整的规则，实现CSS对HTML的“选择”，这就是被叫作“选择器”的原因。

在CSS中，有几种不同类型的选择，本节先来介绍基本选择器。它是相对于下一节中要介绍的复合选择器而言的，也就是说复合选择器是通过对本选择器进行组合而构成的。

基本选择器有标记选择器、类别选择器和ID选择器3种，下面依次介绍。



图1.4 地图中的“图例”

1.2.2 标记选择器

一个HTML页面由很多不同的标记组成，而CSS标记选择器就是声明哪些标记采用哪种CSS样式，因此，每一种HTML标记的名称都可以作为相应的标记选择器的名称。如p选择器，就是用于声明页面中所有<p>标记的样式风格。同样可以通过h1选择器来声明页面中所有<h1>标记的CSS风格，如下所示：

```
<style>
h1{
    color: red;
    font-size: 25px;
}
</style>
```

以上这段CSS代码声明了HTML页面中所有的<h1>标记，其文字的颜色都采用红色，大小都为25px。每一个CSS选择器都包含选择器本身、属性和值，其中属性和值可以设置多个，从而实现对同一个标记声明多种样式风格，如图1.5所示。

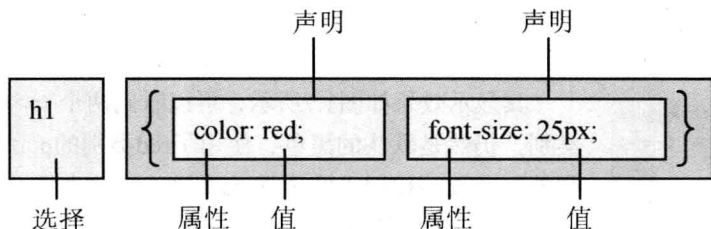


图1.5 标记选择器

如果希望所有<h1>标记不再采用红色，而是采用蓝色，这时仅仅需要将属性color的值修改为blue，即可全部生效。



1.2.3 类别选择器

在上一节中提到的标记选择器一旦声明,那么页面中所有的该标记都会相应地发生变化。例如当声明了<p>标记为红色时,页面中所有的<p>标记都将显示为红色。如果希望其中的某一个<p>标记不显示为红色而是蓝色,仅依靠标记选择器是不够的,还需要引入类别(class)选择器。

类别选择器的名称可以由用户自定义,属性和值跟标记选择器一样,也必须符合CSS规范,如图1.6所示。



图 1.6 类别选择器

例如当页面中同时出现多个<p>标记,并且希望它们显示的颜色各不相同,就可以通过设置不同的类别选择器来实现。一个完整的案例如下所示。

```
<html>
<head>
<title>class 选择器 </title>
<style type="text/css">
.red{
    color:red;          /* 红色 */
    font-size:18px;     /* 文字大小 */
}
</style>
</head>

<body>
<p class="red">class 选择器 1</p>
<p>class 选择器 2</p>
<h3 class="red">h3 同样适用 </h3>
</body>
</html>
```

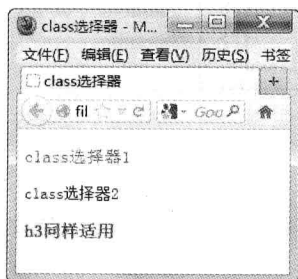
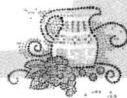


图 1.7 类别选择器示例

其显示效果如图1.7所示,可以看到两个<p>标记中没使用red类别,仍然是默认的黑色,使用了red类别的p元素显示为红色。另外一个<h3>标记也使用了red类别,因此也显示为红色。

在图1.7中仔细观察还会发现,最后一行<h3>标记显示效果为粗体字,这是因为在red类别中没有定义字体的粗细属性,所以h3标题仍然采用其自身默认的显示方式,显示为粗体字,而<p>标记仍默认为正常粗细。



上面的案例说明，同一个类别可以应用于多个标记。此外，还可以同时给一个标记运用多个类别选择器，从而将两个类别的样式风格同时运用到一个标记中，如下例所示。

首先定义两个样式，分别定义了颜色和文字的大小，代码如下。

```
.blue{
    color:blue;          /* 颜色 */
}
.big{
    font-size:22px;      /* 字体大小 */
}
```

然后可以对同一个元素同时使用这两个类别。

```
<body>
    <h4 class="blue big"> 两种 class， 同时使用 blue 和 big。 </h4>
</body>
```

将两个类别同时作为class的值，两个类别名称之间用空格分隔即可。

1.2.4 ID 选择器

ID选择器的使用方法跟类别选择器基本相同，不同之处在于ID选择器只能在HTML页面中使用一次，因此其针对性更强。在HTML的标记中只需要利用id属性，就可以直接调用CSS中的ID选择器，其格式如图1.8所示。

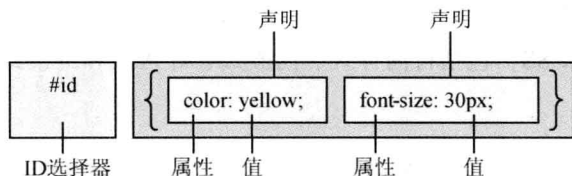


图 1.8 ID 选择器

下面举一个实例，例如先定义两个ID选择器及其样式。

```
#bold{
    font-weight:bold;    /* 粗体 */
}
#green{
    font-size:30px;      /* 字体大小 */
    color:#009900;      /* 颜色 */
}
```

然后分别将它们应用到各自的元素上。

```
<body>
    <p id="bold">ID 选择器 1</p>
    <p id="green">ID 选择器 3</p>
</body>
```

类别选择器与ID选择器在直观上的区别是：在定义时选择器的前缀不同，ID选择器用“#”，类别选择器用“.”；在HTML使用时，前者用id属性，后者用class属性。

二者更深层次的区别在于：一个ID选择器的样式只能用于一个HTML元素，一个



HTML元素只能使用一个ID选择器。而类别选择器则没有这个限制，如下面的代码是完全正确的。

```
<p class="class-a"> 使用类别选择器 </p>
<p class="calss-a"> 使用类别选择器 </p>
<p class="class-a class-b"> 使用类别选择器 </p>
```

而下面的代码是错误的，因为id-a赋给了两个元素，第3个元素同时使用了两个ID选择器定义的样式。

```
<p id="id-a"> 使用 ID 选择器 </p>
<p id="id-a"> 使用 ID 选择器 </p>
<p id="id-c id-d"> 使用 ID 选择器 </p>
```

那么在实际工作中，如何确定是使用类别选择器还是ID选择器呢？这要根据元素的具体情况确定。有的内容，如要指定页面中元素的样式（如制定页头、页脚的样式），在一个页面中只出现一次，就应该使用ID选择器；而有的元素要出现多次，则应该使用类别选择器。

1.3 在 HTML 中引入 CSS 的方法

在对CSS有了大致的了解之后，便可以使用CSS对页面进行全方位的控制。本节主要介绍如何在HTML中使用CSS，包括行内样式、内嵌式、导入式和链接式等，最后探讨各种方式的优先级问题。

1.3.1 行内样式

行内样式是所有样式方法中最为直接的一种，它直接对HTML的标记使用style属性，然后将CSS代码直接写在其中，如下面的代码所示。

```
<html>
<head>
<title> 页面标题 </title>
</head>
<body>
    <p style="color:#FF0000; font-size:20px; text-decoration:underline;">正文内容 1</p>
    <p style="color:#000000; font-style:italic;">正文内容 2</p>
    <p style="color:#FF00FF; font-size:25px; font-weight:bold;">正文内容 3</p>
</body>
</html>
```



其显示效果如图1.9所示。可以看到在3个<p>标记中都使用了style属性，并且设置了不同的CSS样式，各个样式之间互

图 1.9 行内样式