

服装品牌墙界面设计



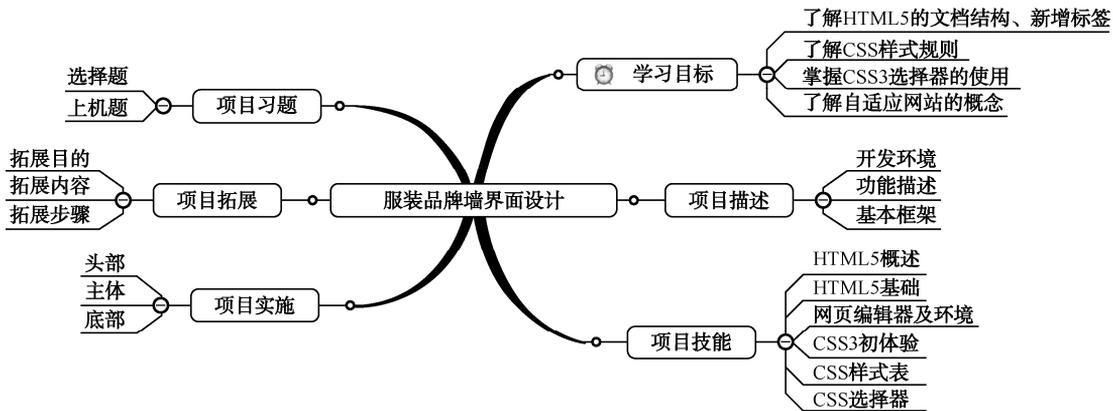
学习目标

通过实现服装品牌墙界面，学习 HTML5 与 CSS3 相关知识，了解 HTML5 与 CSS3 发展历史和基础标签的使用，以及自适应网站标签的使用。在项目实现过程中：

- 了解 HTML5 的文档结构、新增标签
- 了解 CSS 样式规则
- 掌握 CSS3 选择器的使用
- 了解自适应网站的概念



学习路径



项目描述



【情境导入】

随着智能手机的兴起，使用移动平台进行网上购物方兴未艾，其特点之一是操作具有针对性，如以服装品牌、性价比作为选择服装的条件，通过品牌服装墙显示服装所对应的牌子以及牌子所对应商品的链接。本项目主要是实现服装品牌墙界面的设计。



【功能描述】

- 使用响应式布局技术来设计服装品牌墙界面
- 头部包括服装品牌墙的标题
- 主体包括各种品牌的图标及说明
- 底部包括本站点的版权信息



【基本框架】

基本框架如图 1.1 所示，通过本项目的学习，能将框架图 1.1 转换成效果图 1.2。



图 1.1 框架图



图 1.2 效果图



【开发运行环境】

- 系统环境：Windows 7 及以上操作系统。
- 软件环境：Dreamweaver CS6。
- 服务器：Tomcat 7.0。
- 浏览器：电脑端——火狐浏览器、谷歌浏览器；
手机端——Webkit 内核浏览器、Android 手机内置浏览器。



1.1 HTML5 概述

1. HTML5 的发展

HTML 指超文本标记语言，主要是用来制作超文本的简单标记语言，HTML 是 1990 年被创建的一种标记性语言，1999 年推出 HTML4 后就停止了。HTML4 停止以后人们开始期待新的版本出现，为了推动 Web 前端的发展，一些公司联合起来开发了 Web 表单和应用程序，我们所熟悉的 W3C 主要专注于 XHTML 2.0 开发，在 2006 年两个公司进行了合作，开始创建一个新版本的 HTML，也就是 HTML5。HTML5 草案在 2004 年提出，W3C 接受 HTML5 草案是在 2007 年，同年成立了自己的 HTML 工作团队，第一份草案在 2008 年公布。

2. HTML5 的优点

(1) 取消过时标签，新增一些标签。

HTML5 诞生以后，为了简化和美化代码，取消了一些不常用的标签，在取消无用标签的同时新增了一些标志性的标签，现在可以通过 HTML5 中的头部标签<header>来定义，不再需要定义 DIV 标签之后再给 DIV 添加一个 class 或者 ID 标签，HTML5 中添加这些标签的原因是要改善文档的结构性功能。

(2) 解决浏览器兼容问题。

在 HTML5 诞生之前，制作的界面根据浏览器的不同，显示的效果也不太一样，为了能在每个浏览器中看到一样的效果，HTML5 诞生了，HTML5 分析了各个浏览器所使用的内核和它们所具备的功能，根据这些功能和需求制定了浏览器都可以使用的规范，从而达到浏览器兼容的问题。

(3) 代码化繁为简。

HTML5 作为当下流行的语言，已经尽可能地简化了，严格遵循“简单至上”的原则，主要体现为以下几点。

- ① 简化的 DOCTYPE。
- ② 字符集声明。
- ③ 以浏览器原生能力替代复杂的 JavaScript。
- ④ 简单而强大的 API。

1.2 HTML5 基础

1. HTML5 文档的基本结构

每门语言都有自己特定的格式和规范，HTML5 也不例外。HTML5 文档的基本结构如下：

```
<!doctype html>  
<html>
```

```
<head>
<meta charset="utf-8">
<title>无标题文档</title>
</head>
<body>
</body>
</html>
```

HTML5 文档结构中包括以下四部分：

- (1) <!DOCTYPE>用于向浏览器说明当前文档使用哪种 HTML 标签。
- (2) <html>和</html>分别表示文档的开始和结束，用于告知浏览器其自身是一个 HTML 文档。
- (3) <head></head>为头部标签，用于定义 HTML 文档的头部信息，紧跟在<html>标签之后，里面包括的内容有<title>、<meta>、<link>和<style>等。
- (4) <body></body>为主体标签，用于定义 HTML 文档所要显示的内容，在浏览器中所看到的图片、音频、视频、文本等都位于<body>内。该标签中的内容是展示给用户看的。

2. HTML5 语法

HTML5 为了更加兼容各浏览器，在设计和语法方面发生了一些变化，语法变化的主要内容如下。

- (1) 标签不区分大小写。
- (2) 元素可以省略结束标签。
- (3) 允许省略属性的属性值。
- (4) 允许属性值不使用引号。

3. HTML5 新增标签

HTML5 和 HTML 以前版本相比，增加了结构标签、语义标签、特殊功能标签、audio 和 video 标签等。其中新增的标签如表 1.1 所示。

表 1.1 HTML5 新增标签

| 标 签 | 描 述 |
|-----------|--------------------------------|
| <article> | 用于描述页面上的一处完整文章 |
| <nav> | 用于定义导航条，包括主导航条、页面导航、底部导航等 |
| <aside> | 用于定义当前页面的附属信息，内容和 article 内容相关 |
| <hgroup> | 用于对网页或区段（section）的标题进行组合 |
| <figure> | 用于对元素进行组合 |
| <header> | 用于定义文档的页眉（介绍信息） |
| <footer> | 用于定义 section 或 document 的页脚 |



拓展

想了解或学习 HTML5 新增标签，可扫描图中二维码，获取更多信息。



1.3 网页编辑器及环境

HTML5 本身是十分简单的，可是要制作一个精美的网页却不容易，这需要较长时间的实践。在这个过程中，除了要多做之外，还要多看，看别人的网页是怎么设计、制作的。有时，同一种网页效果可以采用多方法来完成。

1. 网页编辑器

自从 Macromedia 的 Dreamweaver CS6 崛起之后，人们制作网页已经基本上不使用 Microsoft 的 FrontPage 了。除了 Dreamweaver 之外，还有许多专业制作网页的商业软件也十分实用。本书使用 Dreamweaver CS6 软件。实现的网页效果如图 1.3 所示。

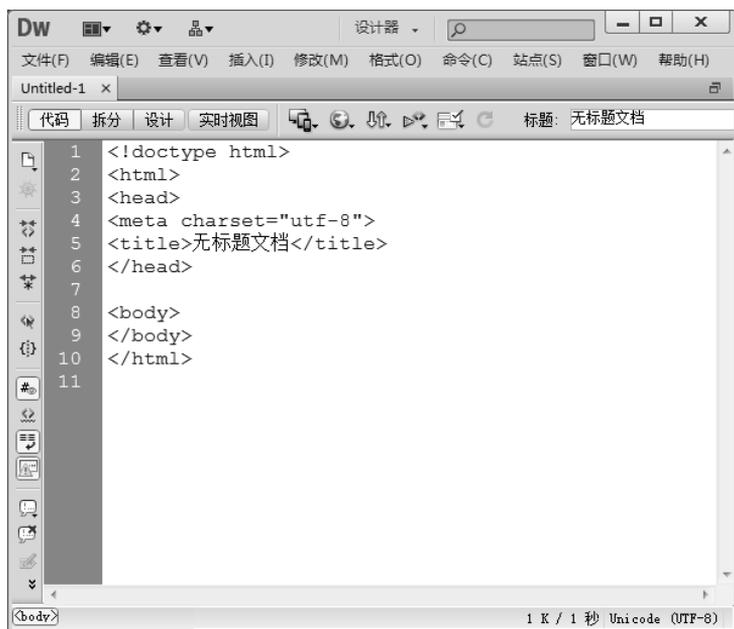


图 1.3 Dreamweaver CS6

2. 自适应网页设计

当使用 Dreamweaver CS6 进行网页编辑之后，打开浏览器就会看到想要的效果，随着智能手机的普及，设计的界面也需在手机端显示，为了能够在手机端正常显示，需要网页宽度自动调整好。

(1) 加入元标签。在网页代码的头部，加入一行 viewport 元标签。

```
< meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=yes, width=device-width" />
```

其中：

width=device-width 表示宽度是设备屏幕的宽度；

initial-scale=1.0 表示初始的缩放比例；

minimum-scale=1.0 表示最小的缩放比例；

maximum-scale=1.0 表示最大的缩放比例;

user-scalable=yes 表示用户是否可以调整缩放比例。

(2) 不使用绝对宽度。所谓不使用绝对宽度就是说 CSS 代码不能指定像素宽度, 如 width: xxx px;

只能指定百分比来定义列宽度, 如 width: xx%; 或者 width: auto; , 或者使用最大宽度和最大高度 max-width、max-height。

(3) Media Query 模块。Media Query 模块可自动探测屏幕宽度, 然后加载相应的 CSS 文件。

例如, media="screen and (max-device-width: 300px)" href="tiny.css" />表示如果屏幕宽度小于 300 像素 (max-device-width: 300px), 则加载 tiny.css 文件。media="screen and (min-width: 300px) and (max-device-width: 600px)" href="small.css" />表示如果屏幕宽度在 300 像素和 600 像素之间, 则加载 small.css 文件。

(4) @media。@media 规则用于同一个 CSS 文件, 根据不同的屏幕分辨率, 选择不同的 CSS 规则。

例如, @media screen and (max-device-width: 400px) {.column {float: none;width:auto;} #sidebar {display:none;}}表示如果屏幕宽度小于 400 像素, 则 column 块取消浮动(float:none)、宽度自动调节 (width:auto), sidebar 块不显示 (display:none)。



拓展

想了解更多自适应网页设计的标签和方法, 可扫描图中的二维码, 获取更多信息。



3. 手机端访问网页环境部署

在 Dreamweaver CS6 制作完之后, 单击浏览器就能出现效果, 要想要在手机上访问, 不仅需要头部添加响应式布局所对应的代码, 还需配置服务器的环境 (本处以 Tomcat 7.0 为例进行说明)。

(1) 下载 Tomcat 软件, 网址为 <http://tomcat.apache.org/download-70.cgi>。

(2) 配置 Tomcat 环境。



拓展

配置相关的服务器 Tomcat 软件以及 JDK 的安装和配置, 可扫描图中的二维码, 获取更多信息。



(3) 启动 Tomcat 软件: 运行 Tomcat 中 bin 目录下的 startup.bat。Tomcat 启动成功后的效果如图 1.4 所示。

(4) 启动成功后在网页上输入 localhost:1010, 效果如图 1.5 所示。(1010 为 Tomcat 端口号, 默认端口号为 8080。)

(5) 把相应的项目放到 Tomcat 目录下的 webapps 文件中。

(6) 配置局域网, 使手机和计算机在同一局域网中。

(7) 打开手机浏览器, 输入 localhost:端口号/文件夹/文件.html, 即可访问计算机端的网页。

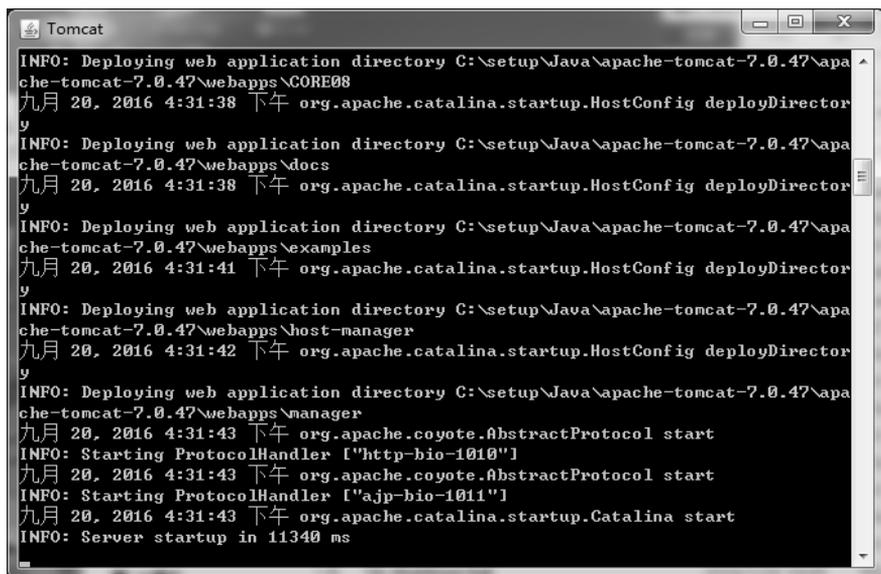


图 1.4 Tomcat 启动成功

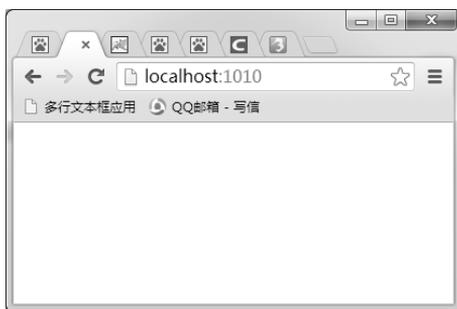


图 1.5 Tomcat 运行效果

1.4 CSS3 初体验

1. CSS3 概述

CSS 即层叠样式表 (Cascading StyleSheet)。在网页制作时采用层叠样式的技术,可以有效地对页面的布局、字体、颜色、背景和其他效果进行更精确的控制。CSS3 是 CSS 技术的升级版。CSS3 将完全向后兼容,网络浏览器将继续支持 CSS。

CSS3 的特点如下。

- (1) 更加灵活地控制网页中文字的字体、颜色、大小、间距、位置。
- (2) 灵活地设置一段文本的行高、缩进,并可以为其加入三维效果的边框。
- (3) 方便为网页中任何元素设置不同的背景颜色和背景图像。
- (4) 精确地控制网页中各元素的位置。
- (5) 为网页中的元素设置各种过滤器,从而产生阴影、模糊、透明等效果。

(6) 与脚本语言相结合，从而产生各种动态效果。

2. CSS 样式规则

学习任何一样新的知识或技能，首先要学习它的规则，然后在这个框架内充分发挥其效果。CSS 样式规则具体格式如下。

```
选择器{属性 1:属性值 1; 属性 2:属性值 2}
```

在上面的规则中，选择器表示希望进行格式化的元素，大括号内是对该元素设置的具体样式，属性是对指定的对象设置样式属性，如文字大小、颜色、字体等。属性和属性值之间用英文的“:”连接，多个属性之间用英文的“;”进行区分。

例如，`p{font-size:10px;color:red}`表示 `p` 元素的字体大小为 10 像素，字体颜色为红色。

1.5 CSS 样式表

在 CSS 中可以使用如下 4 种方法，将样式表的功能加到网页里。

- (1) 定义标记的 `style` 属性。
- (2) 定义内部样式表。
- (3) 嵌入外部样式表。
- (4) 链接外部样式表。

1. 定义标记的 `style` 属性

将 CSS 样式定义在 HTML 标记内，这是最简单的样式制定方法。采用这种方式的弊端是效果只能控制该 HTML 标记，无法做到通用和共享。基本语法如下。

```
<标记 style= "样式属性:属性值...." >
```

该语法格式中，`style` 是标记的属性，实际上任何 HTML 标记都拥有 `style` 属性，用来设置行内式。其中，属性及值的书写规范和 CSS 样式规则相同。

使用标记的 `style` 属性实现效果如图 1.6 所示。



图 1.6 标记的 `style` 属性

为了实现图 1.6 的效果，新建 CORE0101.html，代码如 CORE0101 所示。

```
//代码 CORE0101: 标记的 style 属性
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<meta content="width=device-width, initial-scale=1.0, minimum-scale=1.0,
maximum-scale=1.0,user-scalable=no" name="viewport" />
<meta name="format-detection" content="telephone=no"/>
<meta name="apple-mobile-web-app-status-bar-style" />
<title>标记的 style 属性</title>
</head>
<body>
<p style="font-size:20px;color:red">p 元素的字体大小为10 像素,字体颜色为红色</p>
<p>此行文字未定义 style 属性</p>
</body>
</html>
```

2. 定义内部样式

内部样式表在所应用的 HTML 文档的头部设置，然后在整个 HTML 文件中直接调用该样式的标记。基本语法：

```
<style type="text/css">
选择符 1{样式属性: 属性值; 样式属性: 属性值}
选择符 2{样式属性: 属性值; 样式属性: 属性值}
.....
</style>
```

该语法格式中，<style>标记一般位于<head>标记中，也可以把它放在 HTML 文档的任何地方。但浏览器是从上到下解析代码的，把 CSS 代码放在头部便于提前被下载和解析。

定义内部样式效果如图 1.7 所示。

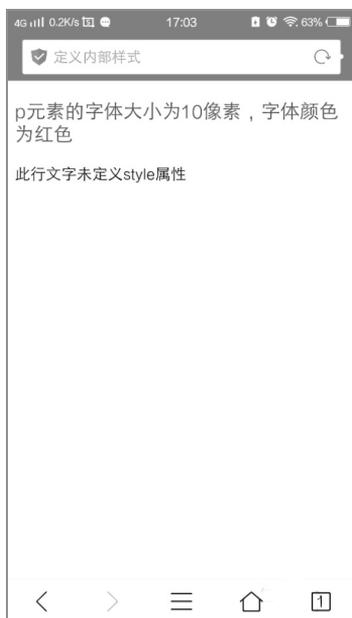


图 1.7 定义内部样式

为了实现图 1.7 的效果，新建 CORE0102.html，代码如 CORE0102 所示。

```
//代码 CORE0102: 定义内部样式
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<meta content="width=device-width, initial-scale=1.0, minimum-scale=1.0,
maximum-scale=1.0,user-scalable=no" name="viewport" />
<meta name="format-detection" content="telephone=no"/>
<meta name="apple-mobile-web-app-status-bar-style" />
<title>定义内部样式</title>
<style type="text/css">
.p1{font-size:20px;color:red}//字体大小为 10 像素，字体颜色为红色
</style>
</head>
<body>
<p class="p1">p 元素的字体大小为 10 像素，字体颜色为红色</p>
<p>此行文字未定义 style 属性</p>
</body>
</html>
```

3. 嵌入外部样式表

嵌入外部样式表就是在 HTML 代码中直接导入样式表。基本语法：

```
<style type="text/css">
@import url("外部样式表的文件名称");
</style>
```

该语法格式中 import 语句后的“;”一定要加上。

为了实现图 1.7 的效果，新建 CORE0103.html，代码如 CORE0103 所示。

```
//代码 CORE0103: 嵌入外部样式
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<meta content="width=device-width, initial-scale=1.0, minimum-scale=1.0,
maximum-scale=1.0,user-scalable=no" name="viewport" />
<meta name="format-detection" content="telephone=no"/>
<meta name="apple-mobile-web-app-status-bar-style" />

<title>定义内部样式</title>
<style type="text/css">
@import url("test.css");
</style>
</head>

<body>
<p class="p1">p 元素的字体大小为 10 像素，字体颜色为红色</p>
<p>此行文字未定义 style 属性</p>
</body>
</html>
```

test.css 代码如 CORE0104 所示。

```
//代码 CORE0104: 嵌入外部样式
.p1{font-size:20px;color:red}
```

4. 链接外部样式表

除了以嵌入外部样式表的方法达到在 HTML 文件中引用样式表的目的之外，还可以用链接的方式使用外部 CSS 样式。基本语法：

```
<link type="text/css" rel="stylesheet" href="外部样式表的文件名称">
```

该语法中，<link>标记要放在<head>头部标记中。

要想实现图 1.2 的效果，只需把代码中的

```
<style type="text/css">  
@import url("test.css");  
</style>
```

换成<link type="text/css" rel="stylesheet" href="test.css">即可。

1.6 CSS 选择器

要想将 CSS 样式应用于特定的 HTML 元素，首先需要找到该目标元素，在 CSS 中，执行这一任务的样式规格部分被称为选择器。

1. 类选择器

类选择器根据类名来选择前面以“.”标志的选择器，使用类选择器设置样式的效果如图 1.8 所示。



图 1.8 类选择器

为了实现图 1.8 的效果，新建 CORE0105.html，代码如 CORE0105 所示。

```
//代码 CORE0105：类选择器  
<!doctype html>
```

```
<html>
<head>
<meta charset="utf-8">
<meta content="width=device-width, initial-scale=1.0, minimum-scale=1.0,
maximum-scale=1.0,user-scalable=no" name="viewport" />
<meta name="format-detection" content="telephone=no"/>
<meta name="apple-mobile-web-app-status-bar-style" />
<title>类选择器</title>
<style>
.p1{font-size:20px;color:red}
.p2{
    color:blue;}
</style>
</head>
<body>
<p class="p1">p 元素的字体大小为 10 像素，字体颜色为红色</p>
<p class="p2">此行文字字体为蓝色</p>
</body>
</html>
```

2. 标签选择器

一个完整的 HTML 页面是由很多不同的标签组成的，而标签选择器决定哪些标签采用相应的 CSS 样式。使用标签选择器设置样式的效果如图 1.9 所示。

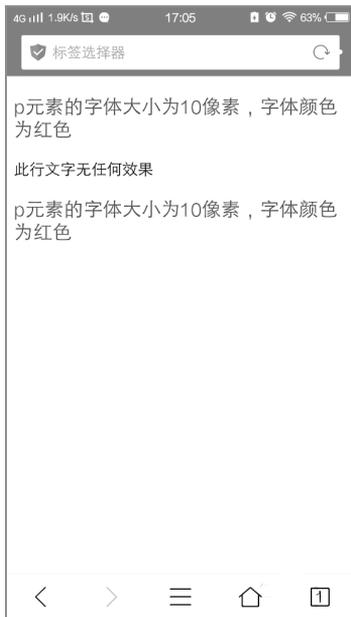


图 1.9 标签选择器

为了实现图 1.9 的效果，新建 CORE0106.html，代码如 CORE0106 所示。

```
//代码 CORE0106：标签选择器
<!doctype html>
<html>
<head>
<meta charset="utf-8">
```

```
<meta content="width=device-width, initial-scale=1.0, minimum-scale=1.0,
maximum-scale=1.0,user-scalable=no" name="viewport" />
<meta name="format-detection" content="telephone=no"/>
<meta name="apple-mobile-web-app-status-bar-style" />

<title>标签选择器</title>
<style>
p{font-size:20px;color:red}

</style>
</head>

<body>
<p>p 元素的字体大小为 10 像素，字体颜色为红色</p>
<div>此行文字无任何效果</div>
<p>p 元素的字体大小为 10 像素，字体颜色为红色</p>
</body>
</html>
```

3. ID 选择器

ID 选择器可以为标有特定 ID 的 HTML 元素指定特定的样式。根据元素 ID 来选择元素具有唯一性，这意味着同一 ID 在同一文档页面中只能出现一次，其前面以“#”来标志，如要实现图 1.9 的效果，只需设置样式为

```
p1{font-size:20px;color:red}
```

4. 后代选择器

后代选择器也称为包含选择器，用来选择特定元素或元素组的后代，将对父元素的选择放在前面，对子元素的选择放在后面，中间加一个空格隔开。使用后代选择器实现图 1.10 的效果。

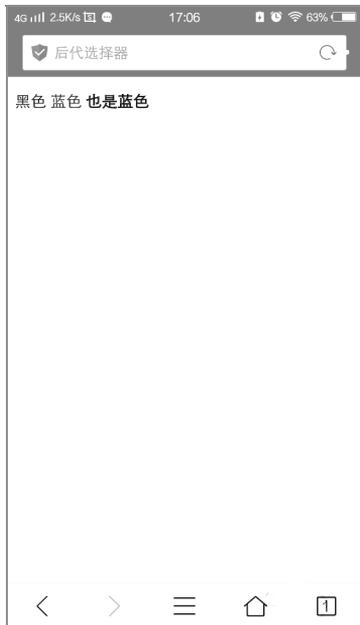


图 1.10 后代选择器

为了实现图 1.10 的效果，新建 CORE0107.html，代码如 CORE0107 所示。

```
//代码 CORE0107: 后代选择器
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<meta content="width=device-width, initial-scale=1.0, minimum-scale=1.0,
maximum-scale=1.0,user-scalable=no" name="viewport" />
<meta name="format-detection" content="telephone=no"/>
<meta name="apple-mobile-web-app-status-bar-style" />
<title>后代选择器</title>
<style>
.father .child{
color:#0000CC;
}
</style>
</head>
<body>
<p class="father">
黑色
<label class="child">蓝色
<b>也是蓝色</b>
</label>
</p>
</body>
</html>
```

5. 子选择器

请注意子选择器与后代选择器的区别，子选择器仅指它的直接后代；而后代选择器作用于所有子后代元素。后代选择器通过空格来进行选择，而子选择器是通过“>”进行选择的使用子选择器的效果如图 1.11 所示。

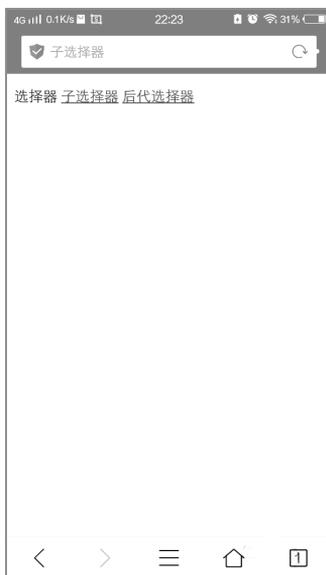


图 1.11 子选择器

为了实现图 1.11 的效果，新建 CORE0108.html，代码如 CORE0108 所示。

```
//代码 CORE0108: 子选择器
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<meta content="width=device-width, initial-scale=1.0, minimum-scale=1.0,
maximum-scale=1.0,user-scalable=no" name="viewport" />
<meta name="format-detection" content="telephone=no"/>
<meta name="apple-mobile-web-app-status-bar-style" />

<title>后代选择器</title>
<style>
#links a {color:red;}
#links > a {color:blue;}
</style>
</head>

<body>
<p id="links">
<a>选择器</a>
<span><a href="#">子选择器</a></span>
<span><a href="#">后代选择器</a></span>
</p>
</body>
</html>
```

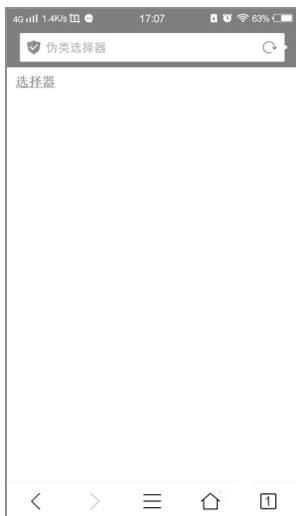


提示

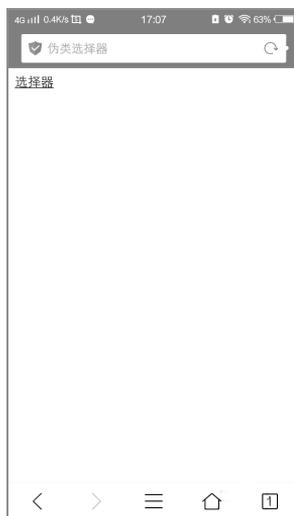
子选择器 (>) 和后代选择器 (空格) 的区别: 都表示“祖先-后代”的关系, 但是>必须是“爸爸>儿子”, 而空格不仅可以是“爸爸儿子”, 还可以是“爷爷儿子”、“太爷爷儿子”。

6. 伪类选择器

有时候还会需要用文档以外的其他条件来应用元素的样式, 如鼠标悬停等。这时就需要用到伪类了。使用伪类选择器实现图 1.12 的效果。



(a) 伪类选择器访问时



(b) 伪类选择器被单击时

图 1.12 伪类选择器

为了实现图 1.12 的效果，新建 CORE0109.html，代码如 CORE0109 所示。

```
//代码 CORE0109: 伪类选择器
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<meta content="width=device-width, initial-scale=1.0, minimum-scale=1.0,
maximum-scale=1.0,user-scalable=no" name="viewport" />
<meta name="format-detection" content="telephone=no"/>
<meta name="apple-mobile-web-app-status-bar-style" />

<title>伪类选择器</title>
<style type="text/css">
a:link{
color:red; /*链接未点击时红色*/
}
a:visited{
color:green; /*已经被访问时为绿色*/
}
a:hover{
color:blue; /*鼠标悬停为蓝色*/
}
</style>
</head>

<body>
<a href="#">选择器</a>
</body>
</html>
```

项目 实施

通过下面七个步骤的操作，实现图 1.3 所示的服装品牌墙界面的效果。

第一步：打开 Dreamweaver CS6 软件，文档类型选择“HTML5”选项，如图 1.13 所示。

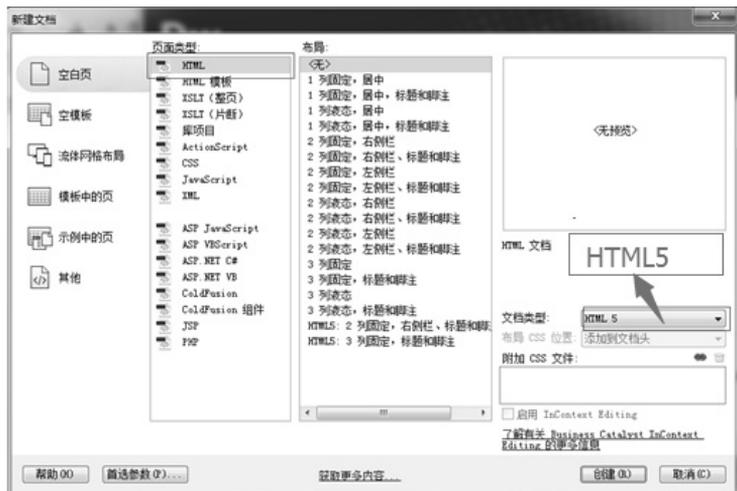


图 1.13 使用 Dreamweaver CS6 新建 HTML5 界面

第二步：创建并保存 CORE0110.html 文件。

第三步：新建 state.css 文件，通过外联方式引入到 HTML 文件中，如图 1.14 所示。

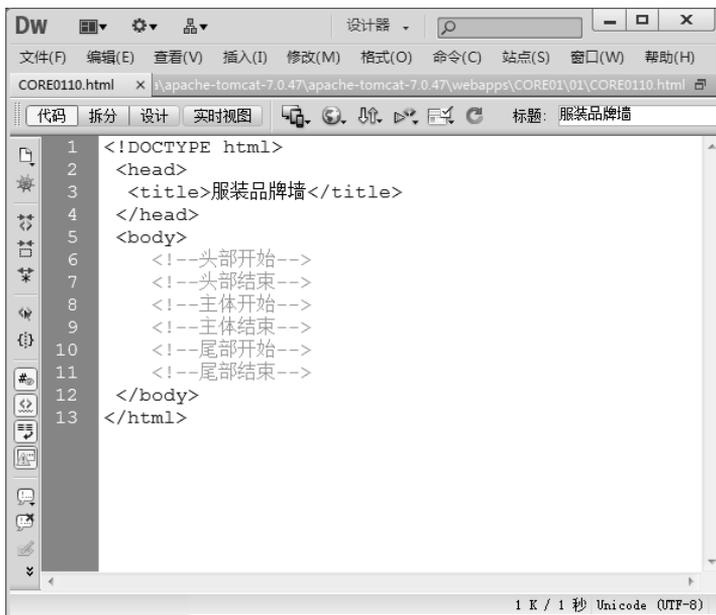


图 1.14 新建 HTML5 并引入 CSS 文件

第四步：在<head>中添加<meta>标签，使网页适应手机屏幕宽度。代码如 CORE0110 所示。

```

//代码 CORE0110:<meta>标签
<meta content="width=device-width, initial-scale=1.0, minimum-scale=1.0,
maximum-scale=1.0,user-scalable=no" name="viewport" />
<meta name="format-detection" content="telephone=no"/>
<meta name="apple-mobile-web-app-status-bar-style" />

```

第五步：头部制作。

现在来制作新浪导航图头部，Logo 部分为新浪的 Logo，用标签表示，代码 CORE0111 如下，效果如图 1.15 所示。

```

//代码 CORE0111:头部 HTML 代码
<header>
  <!--定义顶部锚点-->
  <div class="screen">
    <div class="t_cen ">品牌墙</div>
  </div>
</header>

```

设置头部样式代码 CORE0112 如下，效果如图 1.16 所示。

```

//代码 CORE0112:头部 CSS 代码
* {
  margin: 0px; /*清除外边距*/
  padding: 0px; /*清除内边距*/
}

```

```

.screen {
    background-color: #b52221;/*背景颜色*/
    height: 45px;/*高度*/
    color: #fff;/*颜色*/
    line-height: 45px;/*行高*/
    position: relative;/*位置相对*/
}
.screen a:link,.screen a:visited,.screen a:active {
    color: #fff;/*字体颜色*/
}
.t_cen {
    text-align: center;/*文字居中*/
    position: absolute;/*相对定位*/
    width: 100%;/*宽度100%*/
}

```



图 1.15 头部设置样式前



图 1.16 头部设置样式后

第六步：主体部分制作。

主体部分包括各种品牌的图标及说明，需要用到列表样式<dl>标签，代码 CORE0113 如下，效果如图 1.17 所示。

```

//代码 CORE0113:主体 HTML 代码
<section class="main">
  <div id="channel">
    <div class="test_box">
      <div class="list">
        <dl>
          <dd>
            <a href="#"></a>
          </dd>
          <dt>才子</dt>

```