

第 1 章

走进 HTML

本章技能目标

- 掌握 HTML 标签的应用
- 掌握 CSS 的语法结构及其在网页中的应用

本章简介

在网络异常发达的今天，人们的生活、学习和工作基本上都离不开网络。大家经常浏览的新闻页面、微博和微信等各种从网上获得信息的途径，不论是 PC 终端，还是移动客户端，基本上都是以 Web 为基础来呈现的，因此由 Web 页面呈现信息已成为各种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，在有些书籍中也翻译为超文本标签语言）则是创建 Web 页面的基础。本章从 HTML 文件的基本结构开始，讲解如何通过各种标签编写一个基本的 HTML 网页，然后介绍如何使用基本的 CSS 来美化网页，希望通过基础内容的学习，大家能打下坚实的基础。



1 HTML 基础

在网络已完全融入人们生活的时代，从网络上获取信息或通过网络反馈个人信息，这些都离不开网页。图 1.1~图 1.3 分别代表了常规的宣传页面、用户反馈调查页面和电子邮箱页面。在这些各式各样的页面中，无论是漂亮的、丑的，还是文字的、图片的、视频的，都是以 HTML 文件为基础制作出来的。本节将介绍 HTML 文件的基本结构，在讲解之前，首先简单介绍什么是 HTML，以及它的发展历史。



图 1.1 宣传页面



图 1.2 反馈调查页面



图 1.3 电子邮箱页面

1.1 HTML 简介

在学习使用 HTML 之前，大家经常会问，什么是 HTML？HTML 是用来描述网页的一种语言，它是一种超文本标记语言（Hyper Text Markup Language），也就是说，HTML 不是一种编程语言，仅是一种标记语言（Markup Language）。

既然 HTML 是标记语言，那么 HTML 就是由一套标记标签（markup tag）组成的，在制作网页时，HTML 使用标记标签来描述网页。

在明白了什么是 HTML 之后，简单介绍一下 HTML 的发展历史，让大家了解 HTML 的发展历程，以及目前最新版本的 HTML，使大家在学习时有一个学习的目标和方向。图 1.4 为 HTML 的发展里程碑。

- 超文本置标语言——1993 年 6 月根据互联网工程工作小组的工作草案发布（并非标准）。
- HTML 2.0——1995 年 11 月作为 RFC 1866 发布，在 RFC 2854 于 2000 年 6 月发布之后被宣布过时。
- HTML 3.2——1997 年 1 月 14 日发布，为 W3C 推荐标准。

- HTML 4.0——1997 年 12 月 18 日发布，为 W3C 推荐标准。

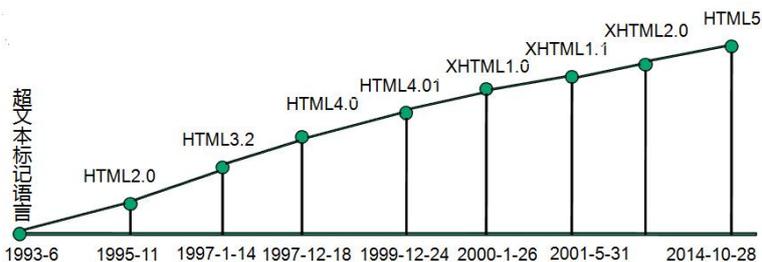


图 1.4 HTML 的发展

- XHTML 1.0——发布于 2000 年 1 月 26 日，是 W3C 推荐标准，修订后于 2002 年 8 月 1 日重新发布。
- XHTML 2.0——W3C 的工作草案，由于改动过大，学习这项新技术的成本过高而最终胎死腹中。因此，现在最常用的还是 XHTML 1.0 标准。
- HTML 5——于 2004 年被提出，2007 年被 W3C 接纳，随后，新的 HTML 工作团队成立，于 2008 年 1 月 22 日发布 HTML 5 第一份正式草案。2012 年 12 月 17 日，HTML 5 规范正式定稿。2013 年 5 月 6 日，HTML 5.1 正式草案公布。

HTML5 从字面上可理解的：HTML 技术标准的第 5 版。从广义上来讲，它是 HTML、CSS、JavaScript、CSS3、API 等的集合体。HTML 5 作为最新版本，提供了一些新的元素和一些有趣的新特性，同时也建立了一些新的规则。这些元素、特性和规则的建立，提供了许多新的网页功能，如使用网页实现动态渲染图形、图表、图像和动画，以及不需要安装任何插件直接使用网页播放视频等。

虽然 HTML 5 提供了许多新的功能，但是它目前仍在完善之中，新的功能还在不断地被推出，纯 HTML 5 的开发还处于尝试阶段。虽然大部分现代浏览器已经具备了某些 HTML 5 的支持，但是还不能完全支持 HTML 5，支持 HTML 5 大部分功能的浏览器仅是一些高版本浏览器，如 IE 9 及更高版本。

通过以上的介绍，相信大家已经明白了 HTML 是什么，以及它的发展历史和本门课程要学习的目标版本，下面我们开始使用 HTML5 的网页结构讲解基础的 HTML 和 CSS。

1.2 HTML 开发工具

开发 HTML 页面非常的灵活，在任何操作系统下均可进行，如 Windows、Linux、Mac OS X。开发工具更是举不胜数，最简单的记事本就可以作为一个工具使用，而且一个得心应手的工具是非常有利于提高开发效率的，如十年之前就开始盛行的 Adobe Dreamweaver，以及 Adobe Edge、JetBrains WebStorm 等等，这几款开发工具都是近年来非常流行的前端开发工具。对于工具的选择本书不做硬性要求，读者可以根据自己的习惯选择开发工具。本书选择 JetBrains WebStorm 作为基本开发工具。

1.3 HTML 基本结构

HTML 的基本结构分为两部分,如图 1.5 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,如图片、文字等。

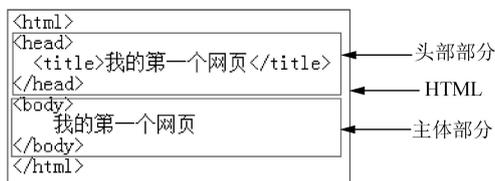


图 1.5 HTML 代码结构

- 页面的各部分内容都在对应的标签中,如网页以<html>开始,以</html>结束。
- 网页头部以<head>开始,以</head>结束。
- 页面主体部分以<body>开始,以</body>结束。

网页中所有的内容都放在<body>和</body>之间。注意 HTML 标签都以“< >”开始,以“</>”结束,要求成对出现,标签之间有缩进,能体现层次感,方便阅读和修改。

操作案例 1: 我的第一个网页

需求描述

编写一个网页并运行。

完成效果

打开网页,如图 1.6 所示。



图 1.6 我的第一个网页

技能要点

- HTML 网页结构。

实现步骤

- 打开开发工具,新建 HTML5 文件。

文件创建完毕后通常会自动生成网页结构,代码如下。若没有自动生成,请手动编写完成。

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

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

- 添加网页标题。

在<title></title>中，添加“我的第一个网页”。

- 添加网页主体内容。

在<body></body>中，添加“我的第一个网页”。

- 运行查看效果。

前面使用 IE 打开第一个网页显示正常，但是如果使用火狐浏览器打开，页面标题和网页内容可能均显示乱码，如图 1.7 所示。为什么会这样呢？

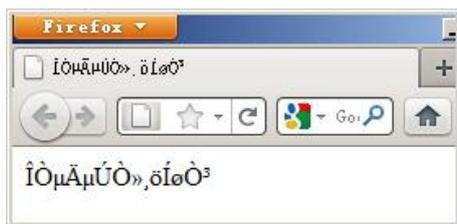


图 1.7 页面出现乱码

在前面的例子中只编写了网页的基本结构，实际上一个完整的网页除了基本结构之外，还包括网页声明、<meta>标签等其他网页基本信息，如示例 1 所示，下面进行详细介绍。

➔ 示例 1

```
<!DOCTYPE html>
<html>
<head lang="en">
<meta charset="UTF-8">
<title></title>
</head>
<body>
</body>
</html>
```

1. DOCTYPE 声明

从示例 1 中可以看到，最上面有关于“DOCTYPE”文档类型的声明，它约束 HTML 文档结构，检验其是否符合相关 Web 标准，同时告诉浏览器，使用哪种规范来解释这个文档中的代码。DOCTYPE 声明必须位于 HTML 文档的第一行，<!DOCTYPE html>表示声明本文档是 HTML5 结构文档。

2. <title>标签

使用<title>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，课工场网站主页对应的网页标题为：<title>互联网人都在学的在线学习平台</title>。打开网页后，将在浏览器窗口的标题栏显示网页标题。

3. <meta>标签

使用该标签描述网页具体的摘要信息，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，它采用“名称/值”对的方式描述摘要信息。

(1) 文档内容类型、字符编码信息

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

其中，属性“http-equiv”提供“名称/值”中的名称，“content”提供“名称/值”中的值，HTML 代码的含义如下。

- 名称：Content-Type（文档内容类型）。
- 值：text/html; charset 表示字符集编码。常用的编码有以下几种。
 - 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：国际性通用的字符编码，同样适用于中文和英文的页面，和 gb2312 编码相比，国际通用性更好，但字符编码的压缩比稍低，对网页性能有一定影响。

这种字符编码的设置效果，就类似于在 IE 中单击“查看”→“编码”菜单，给 HTML 文档设置不同的字符编码。需要注意，不正确的编码设置，将产生网页乱码。

实际上前面网页打开后出现乱码的原因就是没有设置<meta>标签字符编码造成的，从这里可以看出，一个网页的字符编码是多么的重要。因此在制作网页时，一定不要忘记设置网页编码，以免出现页面乱码的问题。

(2) 搜索关键字和内容描述信息

```
<meta name="keywords" content="课工场，在线教育平台" />
```

```
<meta name="description" content="互联网教育，在线学习平台，视频教程，课工场努力打造国内在线学习平台第一品牌" />
```

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。

1.4 HTML 基本标签

任何一个网页基本上都是由一个个标签构成的，网页的基本标签包括标题标签、段落标签、换行标签、水平线标签等，表 1-1 是对这些基本标签的概括介绍。

表 1-1 HTML 基本标签

名称	标签	示例
标题标签	<h1>~<h6>	<h1>静夜思</h1>
段落和换行标签	<p>...</p>、 	<p>床前明月光 疑是地上霜</p>
水平线标签	<hr/>	<hr/>
斜体	...	举头望明月
字体加粗	...	低头思故乡

1. 标题标签

标题标签表示一段文字的标题或主题，支持多层次的内容结构。例如，一级标题采用<h1>，

二级标题则采用<h2>，其他以此类推。HTML 共提供了 6 级标题<h1>~<h6>，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。示例 2 描述了各级标题对应的 HTML 标签。

➔ 示例 2

```
<!DOCTYPE html>
<html>
<head lang="en">
<meta charset="UTF-8">
<title>不同等级的标题标签对比</title>
</head>
<body>
<h1>一级标题</h1>
<h2>二级标题</h2>
<h3>三级标题</h3>
<h4>四级标题</h4>
<h5>五级标题</h5>
<h6>六级标题</h6>
</body>
</html>
```

在浏览器中打开示例 2，预览效果如图 1.8 所示。



图 1.8 不同级别的标题标签输出结果

2. 段落和换行标签

顾名思义，段落标签<p>……</p>表示一段文字等内容。例如，描述“北京欢迎你”这首歌，包括歌名（标题）和歌词（段落），则对应的 HTML 代码如示例 3 所示。

➔ 示例 3

```
<!DOCTYPE html>
<html>
<head lang="en">
```

```

<meta charset="UTF-8">
<title>段落标签的应用</title>
</head>
<body>
<h1>北京欢迎你</h1>
<p>北京欢迎你，有梦想谁都了不起！</p>
<p>有勇气就会有奇迹。</p>
</body>
</html>
    
```

在浏览器中打开示例 3，预览效果如图 1.9 所示。



图 1.9 段落标签的应用

换行标签
表示强制换行显示，该标签比较特殊，没有结束标签，直接使用
表示标签的开始和结束。

3. 水平线标签

水平线标签<hr/>表示一条水平线，注意该标签与
标签一样，比较特殊，没有结束标签，使用该标签在网页中的效果如图 1.10 所示。



图 1.10 水平线标签的应用

4. 斜体和字体加粗

在网页中，经常会遇到字体加粗或斜体字，字体加粗的标签是……，斜

体字的标签是……。例如，在网页中介绍徐志摩，其中人物简介加粗显示，介绍中出现的日期使用斜体，对应的 HTML 代码如下例 4 所示。

➔ 示例 4

```
<!DOCTYPE html>
<html>
<head lang="en">
<meta charset="UTF-8">
<title>字体样式标签</title>
</head>
<body>
<strong>徐志摩人物简介</strong>
<p>
<em>1910</em>年入杭州学堂<br/>
<em>1918</em>年赴美国克拉大学学习银行学<br/>
<em>1921</em>年开始创作新诗<br/>
<em>1922</em>年返国后在报刊上发表大量诗文<br/>
<em>1927</em>年参加创办新月书店<br/>
<em>1931</em>年由南京乘飞机到北平，飞机失事，因而遇难<br/>
</p>
</body>
</html>
```

在浏览器中打开示例 4，预览效果如图 1.11 所示。

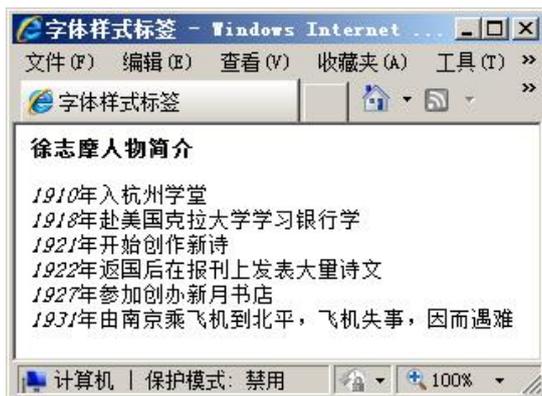


图 1.11 字体样式标签的应用

操作案例 2：静夜思网页制作

需求描述

编写一个网页并运行。

完成效果

打开网页，如图 1.12 所示。

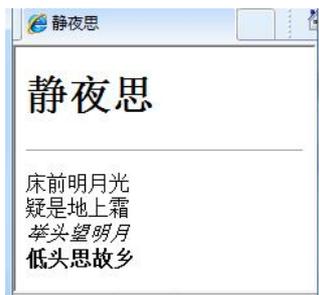


图 1.12 静夜思网页制作

技能要点

- 标题的设置。
- 段落和换行标签的使用。
- 水平线标签的使用。
- 字体样式标签的使用。

1.5 HTML 图像标签

在浏览网页时，随时都可以看到页面上的各种图像，图像是网页中不可缺少的一种元素，下面介绍常见的图像格式和如何在网页中使用图像。

1.5.1 常见的图像格式

在日常生活中，使用比较多的图像格式有 4 种，即 JPG、GIF、BMP、PNG。在网页中使用比较多的是 JPG、GIF 和 PNG，大多数浏览器都可以显示这些图像，不过 PNG 格式比较新，部分浏览器不支持。下面我们就来分别介绍这 4 种常用的图像格式。

1. JPG

JPG (JPEG) 图像格式是在 Internet 上被广泛支持的图像格式。JPG 格式采用的是有损压缩，会造成图像画面的失真，不过压缩之后的体积很小，而且比较清晰，所以比较适合在网页中应用。

2. GIF

GIF 图像格式是网页中使用最广泛、最普遍的一种图像格式，它是图像交换格式 (Graphics Interchange Format) 的英文缩写。GIF 文件支持透明色，使得 GIF 在网页的背景和一些多层特效的显示上运用得非常多。此外，GIF 还支持动画，这是它最突出的一个特点。因此，GIF 图像在网页中应用非常广泛。

3. BMP

BMP 在 Windows 操作系统中使用得比较多，它是位图 (Bitmap) 的英文缩写。BMP 图像文件格式与其他 Microsoft Windows 程序兼容。它不支持文件压缩，也不适用于 Web 网页。

4. PNG

PNG 是 20 世纪 90 年代中期开始开发的图像文件存储格式，它兼有 GIF 和 JPG 的优势，同时具备 GIF 文件格式不具备的特性。流式网络图形格式 (Portable Network Graphic Format,

PNG) 的名称来源于非官方的“PNG's Not GIF”，读成“ping”。唯一遗憾的是，PNG 是一种新兴的 Web 图像格式，还存在部分旧版本浏览器（如 IE 5、IE 6 等）不支持的问题。

1.5.2 图像标签的基本语法

图像标签的基本语法如下。

```

```

其中：

- src 表示图片路径。
- alt 属性指定的替代文本，表示图像无法显示时（如图片路径错误或网速太慢等）替代显示的文本。这样，即使当图像无法显示时，用户还是可以看到网页丢失的信息内容，如图 1.13 所示。所以 alt 属性在制作网页时和“src”配合使用。
- title 属性可以提供额外的提示或帮助信息，当鼠标移至图片上时显示提示信息，如图 1.14 所示。



图 1.13 alt 显示效果



图 1.14 title 属性显示效果

- width 和 height 两个属性分别表示图片的宽度和高度，有时可以不设置，那么图片默认显示为原始大小。

1.6 HTML 超链接标签

大家在上网时，经常会通过超链接查看各个页面或不同的网站，因此超链接标签在网页中极为常用。超链接常用来设置到其他页面的导航链接。下面介绍超链接的用法和应用场合。

1.6.1 超链接的基本用法

超链接的基本语法如下。

```
<a href="链接地址" target="目标窗口位置">链接文本或图像</a>
```

- href: 表示链接地址的路径。

- **target:** 指定链接在哪个窗口打开，常用的取值有 `_self`（自身窗口）、`_blank`（新建窗口）。

超链接既可以是文本超链接，也可以是图像超链接。例如，示例 5 中两个链接分别表示文本超链接和图像超链接，单击这两个超链接均能够在一个新的窗口中打开 `hetao.html` 页面。

示例 5

```
<!DOCTYPE html>
<html>
<head lang="en">
<meta charset="UTF-8">
<title>超链接的应用</title>
</head>
<body>
<a href="hetao.html" target="_blank">无漂白薄皮核桃</a><br/><br/>
<a href="hetao.html" target="_blank"></a>
</body>
</html>
```

在浏览器中打开页面并单击超链接，显示效果如图 1.15 所示。

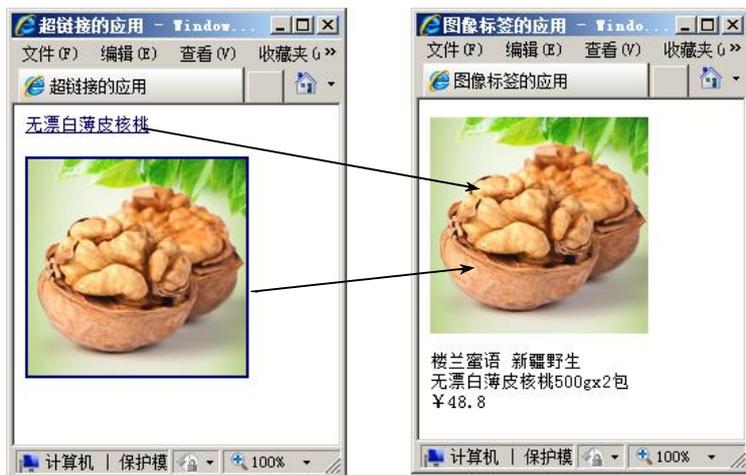


图 1.15 打开超链接示意图

示例 5 中超链接的路径均为文件名称，这表示本页面和跳转页面在同一个目录下，那么，如果两个文件不在同一个目录下，该如何表示文件路径呢？

网页中，当单击某个链接时，将指向万维网上的文档。万维网使用 URL（Uniform Resource Location，统一资源定位器）的方式来定义一个链接地址。例如，一个完整的链接地址的常见形式为 `http://www.bdqn.cn`。根据链接的地址是指向站外文件还是站内文件，链接地址又分为绝对路径和相对路径。

- **绝对路径:** 指向目标地址的完整描述，一般指向本站点外的文件。

例如，`搜狐`。

- **相对路径：**相对于当前页面的路径，一般指向本站点内的文件，所以一般不需要一个完整的 URL 地址的形式。

例如，“登录”表示链接地址为当前页面所在路径的“login”目录下的“login.htm”页面。假定当前页面所在的目录为“D:\root”，则链接地址对应的页面为“D:\root\login\login.htm”。

另外，站内使用相对路径时常用到两个特殊符号：“../”表示当前目录的上级目录，“../../”表示当前目录的上上级目录。假定当前页面中包含两个超链接，分别指向上级目录的 web1.html 及上上级目录的 web2.html，如图 1.16 所示。



图 1.16 相对路径

当前目录下 index.html 网页中的两个链接，即上级目录中 web1.html 及上上级目录中 web2.html，对应的 HTML 代码如下。

```
<a href=" ../web1.html">上级目录</a>
<a href=" ../../web2.html">上上级目录</a>
```



注意

当超链接 href 链接路径为“#”时，表示空链接，如首页。

1.6.2 超链接的应用场合

大家在网上时，会发现不同的链接方式，有的链接到其他页面，有的链接到当前页面，还有单击一个链接直接打开邮件。实际上根据超链接的应用场合，可以把链接分为 3 类。

- 页面间链接：A 页到 B 页，最常用，用于网站导航。
- 锚链接：A 页的甲位置到 A 页的乙位置或 A 页的甲位置到 B 页的乙位置。
- 功能性链接：在页面中调用其他程序功能，如电子邮件、QQ、MSN 等。

1. 页面间链接

页面间链接就是从一个页面链接到另外一个页面。例如，示例 6 中有两个页面间超链接，分别指向课工场在线学习平台的首页和课程列表页面，由于两个指向页面均在当前页面下一级目录下，所以设置的 href 路径显示目录和文件。

➔ 示例 6

```
<!DOCTYPE html>
<html>
<head lang="en">
```

```

<meta charset="UTF-8">
<title>页面间链接</title>
</head>
<body>
<p><a href="elearning/index.html" target="_blank">课工场在线学习平台</a></p>
<p><a href="elearning/courseList.html" target="_blank">课工场在线学习课程列表
</a></p>
</body>
</html>
    
```

在浏览器中打开页面，单击两个超链接，分别在两个新的窗口中打开页面。

2. 锚链接

常用于目标页内容很多、需定位到目标页内容中的某个具体位置时。例如，网上常见的新手帮助页面，当单击某个超链接时，将跳转到对应帮助的内容介绍处，这种方式就是前面说的从 A 页面的甲位置跳转到本页中的乙位置，做起来很简单，需要两个步骤。

首先在页面的乙位置设置标记：

```
<a name="marker">目标位置乙</a>
```

“name”为<a>标签的属性，“marker”为标记名，其功能类似古时用于固定船的锚（或钩），所以也称为锚名。

然后在甲位置链接路径 href 属性值为“#标记名”，语法如下。

```
<a href="#marker">当前位置甲</a>
```

明白了如何实现页面的锚链接，现在来看一个例子——聚美优品网站的新手帮助页面。当单击“新用户注册帮助”超链接时将跳转到页面下方“新用户注册”步骤说明相关位置，如图 1.17 所示。



图 1.17 锚链接

上面的例子对应的 HTML 代码如示例 7 所示。

示例 7

```
<!--省略部分 HTML 代码-->
<p>
[<a href="#register">新用户注册帮助</a>] [<a href="#login">用户登录帮助</a>]</p>
<h1>新手指南 - 登录或注册</h1>
<!--省略部分 HTML 代码-->
<h2><a name="register">新用户注册</a></h2>
<!--省略部分 HTML 代码-->
<h2><a name="login">登录</a></h2>
<!--省略部分 HTML 代码-->
```

上面这个例子是同页面间的锚链接，那么，如果要实现不同页面间的锚链接，即从 A 页面的甲位置跳到 B 页面的乙位置，如单击 A 页面上的“用户登录帮助”链接，将跳转到帮助页面的对应用户登录帮助内容处，该如何实现呢？实际上实现步骤与同页面间的锚链接一样，同样首先在 B 页面（帮助页面）对应位置设置锚标记，如 `登录`，然后在 A 页面设置锚链接，假设 B 页面（帮助页面）名称为 `help.html`，那么锚链接为 `用户登录帮助`，实现效果如图 1.18 所示。



图 1.18 不同页面间的锚链接

3. 功能性链接

功能性链接比较特殊，当单击链接时不是打开某个网页，而是启动本机自带的某个应用程序，如网上常见的电子邮件、QQ、MSN 等链接。接下来以最常用的电子邮件链接为例，当

单击“联系我们”邮件链接时，将打开用户的电子邮件程序，并自动填写“收件人”文本框中的电子邮件地址。

电子邮件链接的用法是“mailto:电子邮件地址”，完整的 HTML 代码如示例 8 所示。

➔ 示例 8

```
<!DOCTYPE html>
<html>
<head lang="en">
<meta charset="UTF-8">
<title>邮件链接</title>
</head>
<body>
<p>
  [<a href="mailto:bdqnWebmaster@bdqn.cn">联系我们</a>] </p>
</body>
</html>
```

在浏览器中打开页面，单击“联系我们”链接，弹出电子邮件编写窗口，如图 1.19 所示。



图 1.19 电子邮件链接

操作案例 3: 超链接的应用

需求描述

完成页面之间的链接设置。点击页面 index.html 的不同页面文字，分别跳转到 indexOnline.html 页面和 courseList.html 页面。

完成效果

打开网页 index.html，点击后跳转到不同页面，如图 1.20 所示。



图 1.20 页面间跳转链接

技能要点

- 超链接的使用。
- 相对路径。

实现步骤

- 下载素材文件，打开 index.html。
- 找到<a>标签，添加 href 属性，添加链接地址。

1.7 注释和特殊符号

HTML 中的注释是为了方便代码阅读和调试。当浏览器遇到注释时会自动忽略注释内容。HTML 的注释格式如下。

```
<!-- 注释内容 -->
```

当页面的 HTML 结构复杂或内容较多时，需要添加必要的注释方便代码阅读和维护。同时，有时为了调试，需要暂时注释掉一些不必要的 HTML 代码。例如，将示例 4 中的一些代码注释掉，如示例 9 所示。

示例 9

```

<!DOCTYPE html>
<html>
<head lang="en">
<meta charset="UTF-8">
<title>字体样式标签</title>
</head>
<body>
<strong>徐志摩人物简介</strong>
<p>

```

```

<!--<em>1910</em>年入杭州学堂<br/>-->
<em>1918</em>年赴美国克拉大学学习银行学<br/>
<em>1921</em>年开始创作新诗<br/>
<em>1922</em>年返国后在报刊上发表大量诗文<br/>
<!--<em>1927</em>年参加创办新月书店<br/>
<em>1931</em>年由南京乘飞机到北平，飞机失事，因而遇难<br/>-->
</p>
</body>
</html>
    
```

在浏览器中打开示例 9 的预览效果，如图 1.21 所示，被注释掉的内容在页面上不显示。

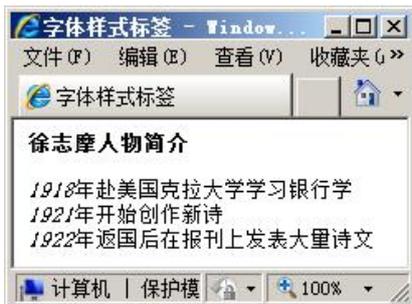


图 1.21 注释的应用

由于大于号 (>)、小于号 (<) 等已作为 HTML 的语法符号，所以，如果要在页面中显示这些特殊符号，就必须使用相应的 HTML 代码表示，这些特殊符号对应的 HTML 代码被称为字符实体。

在 HTML 中常用的特殊符号及对应的字符实体如表 1-2 所示，这些实体符号都以“&”开头，以“;”结束。

表 1-2 特殊符号

特殊符号	字符实体	示例
空格	 	百度 Google
大于号 (>)	>	如果时间>晚上 6 点，就坐车回家
小于号 (<)	<	如果时间<早上 7 点，就走路去上学
引号 (")	"	W3C 规范中，HTML 的属性值必须用成对的"引起来
版权符号 (@)	©	© 2013-2016 课工场

2 揭开 CSS 的神秘面纱



在前面的学习中，介绍了简单地使用 HTML 语言编辑网页，那么大家看图 1.22 所示 QQ 页面中的推荐红钻特权页面，然后回答一个问题：使用前面学习过的 HTML 知识能实现这样的页面效果吗？答案是否定的，单纯地使用 HTML 标签是不能实现的，要实现这样精美的网页需要借助于 CSS。到底什么是 CSS 呢？



图 1.22 推荐红钻特权部分页面

2.1 什么是 CSS

CSS 全称为层叠样式表 (Cascading Style Sheet), 通常又称为风格样式表 (Style Sheet), 它是用来进行网页风格设计的。例如, 在图 1.22 所示页面中就使用了 CSS 混排效果, 将图片和文本结合在一起, 非常漂亮, 并且很清晰, 这就是一种风格。

通过设立样式表, 可以统一地控制 HTML 中各标签的显示属性, 如设置字体的颜色、大小、样式等, 使用 CSS 还可以设置文本居中显示、文本与图片的对齐方式、超链接的不同效果等, 这样层叠样式表就可以更有效地控制网页外观。

使用层叠样式表, 还可以精确地定位网页元素的位置, 美化网页外观, 如图 1.23 和图 1.24 都是使用了 CSS 来控制设计的页面, 它们看起来是不是条理清晰, 配色清新, 页面结构非常赏心悦目呢?



图 1.23 某下载首页



图 1.24 某网站游戏下载页面

使用 CSS 的优势如下。

- 内容与表现分离，也就是使用前面学习的 HTML 语言制作网页，使用 CSS 设置网页样式、风格，并且 CSS 样式单独存放在一个文件中。这样 HTML 文件引用 CSS 文件就可以了，网页的内容（XHTML）与表现就可以分开了，便于后期的 CSS 样式的维护。
- 表现的统一，可以使网页的表现非常统一，并且容易修改。把 CSS 写在单独的页面中，可以对多个网页应用其样式，使网站中的所有页面表现、风格统一，并且当需要修改页面的表现形式时，只需要修改 CSS 样式，所有的页面样式便同时修改。
- 丰富的样式，使得页面布局更加灵活。
- 减少网页的代码量，增加网页的浏览速度，节省网络带宽。在网页中只写 HTML 代码，在 CSS 样式表中编写样式，这样可以减少页面代码量，并且页面代码清晰。同时一个合理的层叠样式表，还能有效地节省网络带宽，提高用户体验量。
- 运用独立于页面的 CSS，还有利于网页被搜索引擎收录。

其实使用 CSS 远不止这些优点，在以后的学习中，大家会深入地了解 CSS 在网页中的优势，现在进入本章的重点内容，学习 CSS 的基本语法。

2.2 CSS 基本语法

CSS 和 HTML 一样，都是浏览器能够解析的计算机语言。因此，CSS 也有自己的语法规则和结构。CSS 规则由两部分构成：选择器和声明。声明必须放在大括号 {} 中，可以是一条或多条；每条声明由一个属性和值组成，属性和值用冒号分开，每条语句以英文分号结尾。如图 1.25 所示，h1 表示选择器，“font-size:12px;”和“color:#F00;”表示两条声明，声明中 font-size 和 color 表示属性，而 12px 和 #F00 则是对应的属性值。

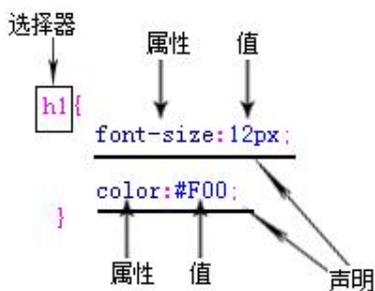


图 1.25 CSS 基础语法

了解了 CSS 的基础语法,在网页中如何定义 CSS 呢?在 HTML 中通过使用<style>标签引入 CSS 样式。<style>标签用于为 HTML 文档定义样式信息,位于<head>标签中,它规定浏览器中如何呈现 HTML 文档。在<style>标签中,type 属性是必需的,它用来定义 style 元素的内容,唯一值是“text/css”,示例 10 简单的展示了如何在网页中对 h1 进行样式设定。

➔ 示例 10

```
<!DOCTYPE html>
<html>
<head lang="en">
<meta charset="UTF-8">

<title>CSS 引入到网页</title>
<style type="text/css">
h1 {font-size:14px;
font-family:"宋体";
color:red; }
</style>
</head>
<body></body>
</html>
```

掌握了如何在 HTML 中编辑 CSS 样式,那么如何把样式应用到 HTML 标签中呢?首先需要学习 CSS 选择器。

2.3 CSS 选择器

选择器(selector)是 CSS 中非常重要的概念,所有 HTML 语言中的标签样式,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,就可以对不同的 HTML 标签进行选择,并赋予各种样式声明,即可以实现各种效果。

在 CSS 中,有 3 种最基本的选择器,分别是标签选择器、类选择器和 ID 选择器,下面分别进行详细介绍。

1. 标签选择器

一个 HTML 页面由很多的标签组成,如<h1>~<h6>、<p>、等,CSS 标签选择器就是用来声明这些标签的。因此,每种 HTML 标签的名称都可以作为相应标签选择器的名称。

每个 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，从而实现了对同一个标签声明多种样式风格，CSS 标签选择器的语法结构如图 1.26 所示。

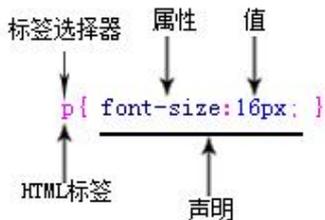


图 1.26 标签选择器

示例 11 声明了<h3>和<p>标签选择器，h3 选择器用于声明页面中所有<h3>标签的样式风格，p 选择器用来声明页面中所有<p>标签的 CSS 风格。

➔ 示例 11

```
<!DOCTYPE html>
<html>
<head lang="en">
<meta charset="UTF-8">
<title>标签选择器的用法</title>
<style type="text/css">
h3 {color:#090;}
p {
    font-size:16px;
    color:red;
}
</style>
</head>
<body>
<h3>北京欢迎你</h3>
<p>北京欢迎你，有梦想谁都了不起！</p>
<p>有勇气就会有奇迹。</p>
</body>
</html>
```

示例 11 中 CSS 代码声明了 HTML 页面中所有的<h3>标签和<p>标签。<h3>标签中字体颜色为绿色；<p>标签中字体颜色为红色，大小都为 16px。在浏览器中打开页面，即可观察到标题字体颜色为绿色；文本字体颜色为红色，并且字体大小为 16px。



注意

标签选择器声明之后，立即对 HTML 中的标签产生作用。

标签选择器是网页样式中经常用到的，通常用于直接设置页面中的标签样式。例如，页面中有<h1>、<h4>、<p>标签，如果相同的标签内容的样式一致，那么使用标签选择器就非常方便了。

2. 类选择器

标签选择器一旦声明，那么页面中所有的该标签都会相应地发生变化。例如，当声明了<p>标签都为红色时，页面中所有的<p>标签都将显示为红色。但是，如果希望其中的某个<p>标签不是红色，而是绿色，仅依靠标记选择器是不够的，还需要引入类（class）选择器。

类选择器的名称可以由用户自定义，属性和值跟标签选择器一样，必须符合 CSS 规范，类选择器的语法结构如图 1.27 所示。

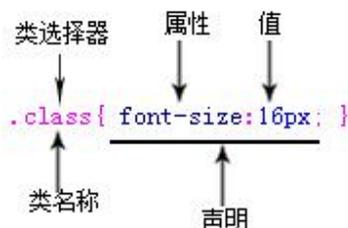


图 1.27 类选择器

设置了类选择器后，就要在 HTML 标签中应用类样式。使用标签的 class 属性引用类样式，即<标签名 class="类名称">标签内容</标签名>。

例如，要使示例 11 中的两个<p>标签中的文本分别显示不同的颜色，就可以通过设置不同的类选择器来实现，代码如下例 12 所示，增加了 green 类样式，并在<p>标签中使用 class 属性引用了类样式。

➔ 示例 12

```
<!DOCTYPE html>
<html>
<head lang="en">
<meta charset="UTF-8">
<title>类选择器的用法</title>
<style type="text/css">
h3 {color:#090;}
p {
    font-size:16px;
    color:red;
}
.green {
    font-size:20px;
    color:green;
}
</style>
</head>
<body>
<h3>北京欢迎你</h3>
<p>北京欢迎你，有梦想谁都了不起！ </p>
```

```
<p class="green">有勇气就会有奇迹。</p>
</body>
</html>
```

在浏览器中打开页面，效果如图 1.28 所示，由于第二个<p>标签应用了类样式 green，它的文本字体颜色变为绿色，并且字体大小为 20px；而由于第一个<p>标签没有应用类样式，因此它直接使用标签选择器，字体颜色依然是红色，字体大小为 16px。

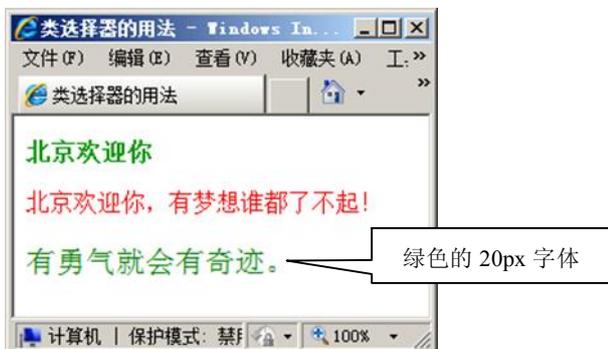


图 1.28 类选择器的效果图

类选择器是网页中最常用的一种选择器，设置了一个类选择器后，只要页面中某个标签中需要相同的样式，直接使用 class 属性调用即可。类选择器在同一个页面中可以频繁地使用，应用起来非常方便。

3. ID 选择器

ID 选择器的使用方法与类选择器基本相同，不同之处在于 ID 选择器只能在 HTML 页面中使用一次，因此它的针对性更强。在 HTML 的标签中，只要在 HTML 中设置了 id 属性，就可以直接调用 CSS 中的 ID 选择器。ID 选择器的语法结构如图 1.29 所示。

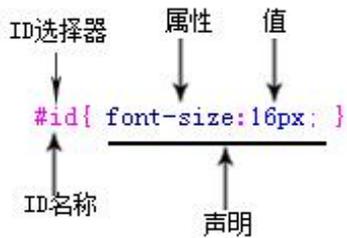


图 1.29 ID 选择器

下面举一个例子说明 ID 选择器在网页中的应用。设置两个 id 属性，分别为 first 和 second，在样式表中设置两个 ID 选择器，代码如下例 13 所示。

➡ 示例 13

```
<!DOCTYPE html>
<html>
<head lang="en">
```

```
<meta charset="UTF-8">
<title>ID 选择器的应用</title>
<style type="text/css">
#first{font-size:16px;}
#second{font-size:24px;}
</style>
</head>
<body>
<h1>北京欢迎你</h1>
<p id="first">北京欢迎你，有梦想谁都了不起！</p>
<p id="second">有勇气就会有奇迹。</p>
<p>北京欢迎你，为你开天辟地</p>
<p>流动中的魅力充满朝气。</p>
</body>
</html>
```

在浏览器中打开的页面效果如图 1.30 所示，由于第一个<p>标签设置了 id 为 first，它的字体大小为 16px；第二个<p>标签设置了 id 为 second，它的字体大小为 24px。由此例可知，只要在 HTML 标签中设置了 id 属性，那么此标签就可以直接使用 CSS 中对应的 ID 选择器。

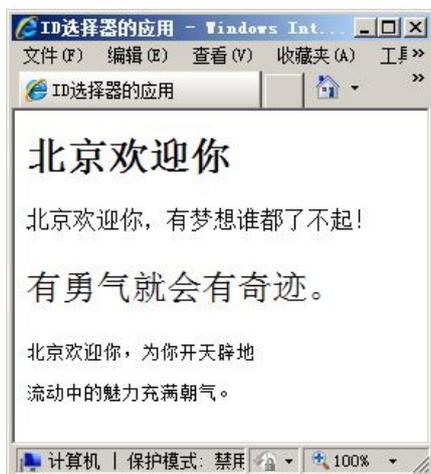


图 1.30 ID 选择器的效果图

ID 选择器与类选择器不同，同一个 id 属性在同一个页面中只能使用一次。虽然这样，但是它在网页中也是经常用到的。例如，在布局网页时，页头、页面主体、页尾或者页面中的菜单和列表等通常都使用 id 属性，这样看到 id 名称就知道此部分的内容，使页面代码具有非常高的可读性。

操作案例 4：选择器的使用

需求描述

完成使用不同的选择器控制不同的字体样式。

- 第一行字体大小为 20，颜色红色，使用标签选择器实现。

- 第二行字体大小为 24，颜色绿色，使用类选择器实现。
- 第三行字体大小为 28，颜色黑色，使用 ID 选择器实现。

完成效果

打开网页 index.html，显示页面不同的字体样式，如图 1.31 所示。



图 1.31 选择器的使用

技能要点

- 使用标签选择器。
- 使用类选择器。
- 使用 ID 选择器。

关键代码

```
p{font-size:20px;color:red;}
.second{font-size:24px;color:#096;}
#third{font-size:28px;color:black;}
```

2.4 网页中引用 CSS 样式

在前面的几个例子中，所有的 CSS 样式都是通过<style>标签放在 HTML 页面的<head>标签中，但是在实际制作网页时，这种方式并不是唯一的，还有其他两种方式可应用 CSS 样式。在 HTML 中引入 CSS 样式的方法有 3 种，分别是：

- 行内样式。
- 内部样式表。
- 外部样式表。

下面依次学习各种应用方式的优缺点及应用场景。

1. 行内样式

行内样式就是在 HTML 标签中直接使用 style 属性设置 CSS 样式。style 属性提供了一种改变所有 HTML 元素样式的通用方法，其用法如下。

```
<h1 style="color:red;">style 属性的应用</h1>
<p style="font-size:14px; color:green;">直接在 HTML 标签中设置的样式</p>
```

这种使用 style 属性设置 CSS 样式的方法仅对当前的 HTML 标签起作用，并且要写在 HTML 标签中。

**注意**

行内样式法不能使内容与表现相分离，本质上没有体现出 CSS 的优势，因此不推荐使用。

2. 内部样式表

正如前面的所有示例一样，把 CSS 代码写在<head>的<style>标签中，与 HTML 内容位于同一个 HTML 文件，这就是内部样式表。

这种方式方便在同页面中修改样式，但不利于在多页面间共享复用代码及维护，对内容与样式的分离也不够彻底。实际开发时，会在页面开发结束后，将这些样式代码保存到单独的 CSS 文件中，将样式和内容彻底分离开，即下面介绍的外部样式表。

3. 外部样式表

外部样式表是把 CSS 代码保存为一个单独的样式表文件，文件扩展名为.css，在页面中引用外部样式表即可。HTML 文件引用外部样式表有两种方式，分别是链接式和导入式。

● 链接外部样式表

链接外部样式表就是在 HTML 页面中使用<link/>标签链接外部样式表，这个<link/>标签必须放到页面的<head>标签内，语法如下。

```
<head lang="en">
.....
<link href="style.css" rel="stylesheet" type="text/css" />
.....
</head>
```

其中，rel="stylesheet"是指在页面中使用这个外部样式表；type="text/css"是指文件的类型为样式表文本；href="style.css"是文件所在的位置，style.css 是 CSS 样式表文件。

● 导入外部样式表

导入外部样式表就是在 HTML 网页中使用@import 导入外部样式表，导入样式表的语句必须放在<style>标签中，而<style>标签必须放到页面的<head>标签内，语法如下。

```
<head lang="en">
.....
<style type="text/css">
<!--
@import url("style.css");
-->
</style>
</head>
```

其中@import 表示导入文件，前面必须有一个@符号，url("style.css")表示样式表文件位置。

两种引用外部样式表的方式的本质相同，都是将一个独立的 CSS 样式表引用到 HTML 页面中，但是它们还是有一些差别的，现在看一下两者的不同之处。

(1) <link/>标签属于 XHTML 范畴，而@import 是 CSS 2.1 中特有的。

(2) 使用<link/>链接的 CSS 是客户端浏览网页时先将外部 CSS 文件加载到网页当中，再进行编译显示，所以这种情况下显示出来的网页与用户预期的效果一样，即使网速再慢也是一样的效果。

(3) 使用@import 导入 CSS 文件，客户端在浏览网页时先将 HTML 结构呈现出来，再把

外部 CSS 文件加载到网页当中，当然最终的效果也与使用<link/>链接文件的效果一样，只是当网速较慢时会先显示没有 CSS 统一布局的 HTML 网页，这样就会给用户很不好的感觉。这也是目前大多数网站采用链接外部样式表的主要原因。

(4) 由于@import 是 CSS 2.1 中特有的，因此对于不兼容 CSS 2.1 的浏览器来说就是无效的。

综合以上几个方面的因素，大家可以发现，现在大多数网站还是比较喜欢使用链接外部样式表的方式引用外部 CSS 文件的。

外部样式表实现了样式和结构的彻底分离，一个外部样式表文件可以应用于多个页面。当改变这个样式表文件时，所有页面的样式都会随之改变。这在制作大量相同样式页面的网站时非常有用，不仅减少了重复的工作量，利于保持网站的统一样式和网站维护，同时用户在浏览网页时也减少了重复下载代码的次数，提高了网站的速度。

4. 样式优先级

前面一开始就提到 CSS 的全称为层叠样式表，因此对于页面中的某个元素，它允许同时应用多个样式（即叠加），页面元素最终的样式即为多个样式的叠加效果。但这存在一个问题——当同时应用上述的 3 类样式时，页面元素将同时继承这些样式，但样式之间如有冲突，应继承哪种样式？这就存在样式优先级的问题。同理，从选择器角度，当某个元素同时应用标签选择器、ID 选择器、类选择器定义的样式时，也存在样式优先级的问题。CSS 中规定的优先级规则如下。

- 行内样式>内部样式表>外部样式表。
- ID 选择器>类选择器>标签选择器。

行内样式>内部样式表>外部样式表，即“就近原则”。如果同一个选择器中样式声明层叠，那么后写的会覆盖先写的样式，即后写的样式优先于先写的样式。

操作案例 5：外部样式表的使用

需求描述

使用链接外部样式表的方法重新完成操作案例 4，实现页面效果。

- 第一行字体大小为 20，颜色红色，使用标签选择器实现。
- 第二行字体大小为 24，颜色绿色，使用类选择器实现。
- 第三行字体大小为 28，颜色黑色，使用 ID 选择器实现。

完成效果

打开网页 index.html，显示页面不同的字体样式，如图 1.31 所示。

技能要点

- 使用标签选择器。
- 使用类选择器。
- 使用 ID 选择器。
- 链接外部样式表。

实现步骤

- 新建 HTML 文件 index.html、样式表文件 style.css 文件。
- 设置标签选择器样式、类选择器样式、ID 选择器样式。

- 在 index.html 中链接外部样式表 style.css。

本章总结

- HTML 文件的基本结构包括页面声明、页面基本信息、页面头部和页面主体等。
- 网页基本标签包括标题标签<h1>~<h6>、段落标签<p>、水平线标签<hr/>、换行标签
等。
- 插入图片时使用标签, 要求 src 和 alt 属性必选。
- 超链接<a>标签用于建立页面间的导航链接, 链接可分为页面间链接、锚链接、功能性链接。
- CSS 语法规则, 使用<style>标签引入 CSS 样式。
- CSS 选择器分为标签选择器、类选择器和 ID 选择器。
- 在 HTML 中引入 CSS 样式的 3 种方式分别是行内样式、内部样式表和外部样式表, 其中外部样式表使用<link/>标签链接外部 CSS 文件。CSS 样式的优先级依据就近原则。

本章作业

1. 请写出网页的基本标签、作用和语法。
2. 超链接有哪些类型? 它们的区别是什么?
3. 制作聚美优品常见问题页面, 页面标题和问题使用标题标签完成, 问题答案使用段落标签完成, 客服温馨提示部分与问题列表之间使用水平线分隔, 完成效果如图 1.32 所示。

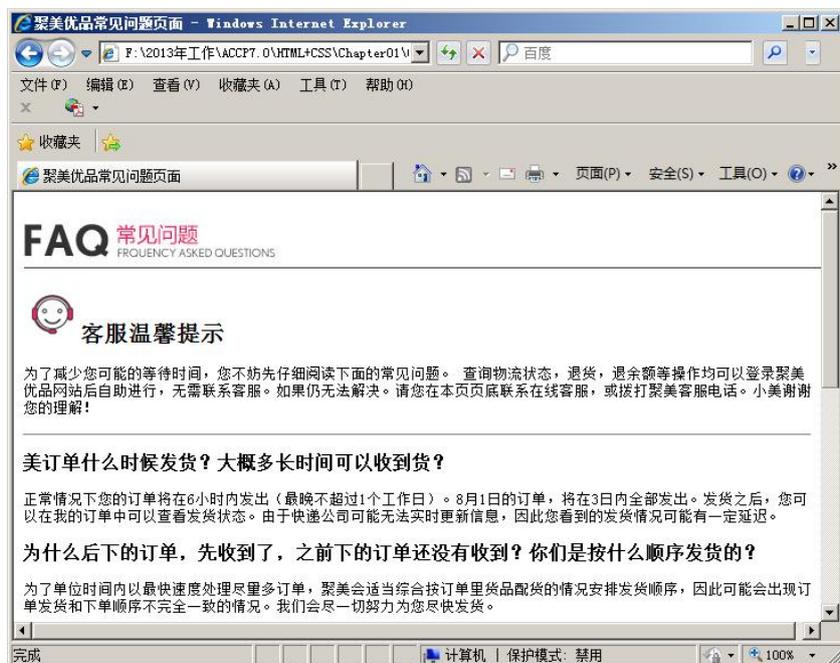


图 1.32 聚美优品常见问题页面

4. 使用 CSS 制作网页有哪些优势?
5. 使用<style>标签和 style 属性引入 CSS 样式有哪些相同点和不同点?
6. 请登录课工场，按要求完成预习作业。