

职业教育国家在线精品课程配套教材
“人工智能+”融媒体一体化教材

HTML5

网页设计任务教程

主编 梁 磊 温 雪 衣文娟



航空工业出版社

内 容 提 要

本教材通过一个完整的网站设计项目，讲解了 HTML5+CSS3 网页设计的基础知识。全书分为 9 个任务，主要讲授如何构建网页的基本结构及使用各种 HTML 标签来实现内容的组织和布局。在此基础上，详细讲解 CSS 样式的引入、为网页添加样式和美化外观，以及表单的创建和验证。同时，教材还介绍了如何处理媒体内容和增加交互性，如添加视频、音频及设计网页动画等；介绍了如何测试网页的性能及如何将网页发布到网络上进行推广。通过本教材的学习，读者将掌握构建一个功能完整、美观实用的网站所需的关键技能。

本教材适用于高职院校计算机应用技术、软件技术、计算机网络技术以及数字媒体技术等专业学习网页设计课程的学生，也可供职业教育培训机构学员、网页设计初学者及对 HTML5 网页开发感兴趣的自学者使用，助力读者掌握 HTML5 网页设计核心技能，满足网页前端开发岗位基础需求。

图书在版编目（CIP）数据

HTML5 网页设计任务教程 / 梁磊，温雪，衣文娟主编.

北京：航空工业出版社，2025. 6. — ISBN 978-7-5165-4214-9

I. TP312.8

中国国家版本馆 CIP 数据核字第 2025X2T706 号

HTML5 网页设计任务教程

HTML5 Wangye Sheji Renwu Jiaocheng

航空工业出版社出版发行

（北京市朝阳区北苑路 58 号楼 20 层 100012）

发行部电话：010-85672666 010-85672683 读者服务热线：010-85672635

中煤（北京）印务有限公司印刷

全国各地新华书店经售

2025 年 6 月第 1 版

2025 年 6 月第 1 次印刷

开本：889×1194 1/16

字数：480 千字

印张：15.5

定价：49.80 元

在线课程学习指南

本书配有国家级在线开放课程“HTML5 网页设计”，读者可在智慧树平台进行学习。

1. 搜索课程

进入智慧树平台官网，在页面上方输入框中输入“HTML5 网页设计”，单击“搜索”按钮后，在搜索结果中选择对应课程（主讲教师：梁磊）。



2. 在线学习

选择对应课程后，进入课程界面，单击“去学习”按钮，登录后即可在线学习该课程。



前言

网页设计是前端开发技术的基础。这本《HTML5 网页设计任务教程》旨在满足广大初学者对 HTML5 网页设计的学习需求。本教材不仅系统地介绍了 HTML5 网页设计的核心技术，还注重培养学生的创新思维与实践能力，并将课程思政与专业知识传授紧密结合，助力培养德智体美劳全面发展的社会主义建设者和接班人。

本教材特色在于将党的二十大精神融入教学内容，立足于落实立德树人根本任务，在传授专业知识与技能的同时，引导学生树立正确的世界观、人生观、价值观，增强爱党报国的情怀，培养敬业奉献、服务人民的精神；通过实践项目的设计与实施，培养学生的团队合作精神和创新能力，为学生未来在网页设计领域的发展奠定基础。

内容方面，本教材涵盖 HTML5 基础、网页结构与标签、CSS 样式、响应式设计、媒体与交互、表单与验证、测试与优化及发布与推广等多个方面，旨在为读者提供一个系统而全面的学习路径。

本教材是国家级在线开放课程“HTML5 网页设计”的配套教材，且配有教案、教学课件等丰富的立体化教学资源，有需要者可致电 13810412048 或发邮件至教学助手 2393867076@qq.com 领取。

本教材是一本校企合作开发的教材，适用于高职院校计算机应用技术、软件技术、计算机网络技术以及数字媒体技术等专业学习网页设计相关课程的学生，也可供网页设计初学者和希望提升网页设计技能的社会人士自学使用。在编写过程中，本教材得到了许多专家学者的支持与帮助，尤其是合作单位青岛海信信息科技股份有限公司的各位工程师，为本教材提供了大量的技术指导，在此表示衷心的感谢。希望本教材能为读者的网页设计学习之旅提供有力的支持与指导。

由于编者水平有限，书中存在的不足之处，敬请各位读者批评指正，万分感谢！

编者

2025 年 1 月

目 录

任务 1

美食文化历史页面设计..... 1

任务导入..... 1

1.1 认识网页 2

1.1.1 网页和网站的关系 2

1.1.2 静态网页与动态网页 2

1.2 理解 Web 工作原理..... 3

1.2.1 网页访问过程 3

1.2.2 浏览器 3

1.2.3 网络访问基本概念 4

1.3 认识及使用开发工具 6

1.3.1 常用开发工具介绍 6

1.3.2 Chrome 浏览器开发者工具 8

1.4 认识网页基本结构 10

1.4.1 标签与页面基本结构 10

1.4.2 标签的类型及属性 12

任务实施..... 14

任务 2

饮食文化页面设计..... 16

任务导入..... 16

2.1 认识 HTML 基本标签 17

2.1.1 文本控制标签 17

2.1.2 图像标签 20

2.1.3 超链接标签 21

2.2 使用样式表 24

2.2.1 样式表的概念及作用 24

2.2.2 定义与引入样式表 24

2.2.3 基本类型选择器 26

2.3 使用文本样式 30

2.3.1 字体样式 30

2.3.2 文本外观样式 32

任务实施..... 35

任务 3

学做美食页面设计..... 39

任务导入..... 39

3.1 使用 HTML 其他元素 40

3.1.1 列表元素 40

3.1.2 结构元素 43

3.1.3 页面交互元素 44

3.2 元素类型及转换 46

3.2.1 块级元素 46

3.2.2 内联元素	46
3.2.3 内联块元素	46
3.3 使用 CSS3 高级选择器	46
3.3.1 属性选择器	47
3.3.2 关系选择器	48
3.3.3 伪选择器	51
3.3.4 结构化伪类选择器	53
任务实施	57

任务 4

大厨风采页面设计..... 64

任务导入	64
4.1 使用 CSS 盒模型	65
4.1.1 盒模型	65
4.1.2 盒模型的宽度和高度	65
4.1.3 边框属性用法	67
4.1.4 边距属性用法	77
4.2 浮动的用法	81
4.2.1 <div> 标签	81
4.2.2 设置元素浮动	82
4.2.3 浮动的清除	84
4.3 使用音视频	88
4.3.1 音视频格式简介	88
4.3.2 音视频在页面中的引入	89
4.3.3 音视频常见属性	91
任务实施	94

任务 5

诗与美食页面设计..... 103

任务导入	103
-------------------	------------

5.1 CSS 背景属性104

5.1.1 背景颜色	104
5.1.2 背景图片	105
5.1.3 背景图片平铺设置	107
5.1.4 背景图片位置设置	108
5.1.5 设置背景复合属性	109

5.2 渐变属性110

5.2.1 线性渐变	110
5.2.2 径向渐变	112
5.2.3 重复渐变	113

任务实施	115
-------------------	------------

任务 6

网站首页设计..... 126

任务导入	126
-------------------	------------

6.1 元素定位127

6.1.1 元素定位属性	127
6.1.2 元素定位方式	127
6.1.3 z-index 属性	136

6.2 @font-face 规则139

6.3 字体图标140

6.3.1 字体图标简介	140
6.3.2 使用字体图标	140

任务实施	143
-------------------	------------

任务 7

注册页面设计..... 164

任务导入	164
-------------------	------------

7.1 创建和使用表格	165	8.3 关键帧动画	216
7.1.1 创建基本表格	165	8.3.1 @keyframes	216
7.1.2 设置表格属性	166	8.3.2 animation 属性	218
7.1.3 设置表格样式	167	任务实施	223
7.1.4 处理表格常见问题	170		
7.2 设计与使用表单	172		
7.2.1 了解表单基本构成	172	任务 9	
7.2.2 了解表单结构与属性	173	网站测试与发布	228
7.2.3 了解常用表单元素	175		
任务实施	187	任务导入	228
		9.1 网站测试	229
任务 8		9.2 空间申请	229
光盘行动页面设计	193	9.3 代码上传	229
		9.4 网站的宣传与推广	229
任务导入	193	9.5 网页维护和更新	230
8.1 过渡	194	9.5.1 静态网页的维护和更新	230
8.1.1 transition-property 属性	194	9.5.2 动态网页的更新	230
8.1.2 transition-duration 属性	194	任务实施	230
8.1.3 transition-timing-function 属性	195		
8.1.4 transition-delay 属性	197		
8.1.5 transition 属性	197		
8.2 元素转换	199	参考文献	236
8.2.1 2D 转换	199		
8.2.2 3D 转换	209		

任务 2

饮食文化页面设计



任务导入

习近平总书记在党的二十大报告中提出“推进文化自信自强，铸就社会主义文化新辉煌”。网页设计是一个综合性的领域，涉及创意、技术和用户体验等多个方面。任务 1 介绍了网站设计的基本流程及一个 HTML 页面的基本结构。本任务将从 HTML 基本标签的用法出发，介绍页面结构的定义，并通过几个不同的案例来学习样式表和选择器的相关知识，完成舌尖文化页面内容及样式的整体制作，并通过数字化手段展现中华饮食文化的独特魅力。

知识目标

- ① 掌握 HTML 基本标签的用法。
- ② 掌握使用 CSS 基础类型选择器定义样式的方法。
- ③ 掌握 CSS 文本样式属性，学会运用相应的属性定义文本样式。
- ④ 理解 CSS 优先级，了解如何区分复合选择器权重的大小。

能力目标

- ① 能够根据页面内容，选择正确的标签。
- ② 能够使用最合理的选择器实现页面效果。
- ③ 能够正确使用选择器，解决因选择器权重问题带来的冲突。

素质目标

- ① 通过学习合理选择和正确使用标签，培养良好的职业习惯。
- ② 强调不使用不规范的、有害的标签或隐藏的标签来“欺骗”搜索引擎或用户，强化诚信意识和道德原则，遵守行业规范，培养职业道德和社会责任感。
- ③ 理解应该适度使用选择器，不应该过度干扰用户浏览体验的理念，培养用户服务意识。

重难点

- ① 选择器的基本用法。
- ② CSS 的优先级、层叠性。

2.1 认识 HTML 基本标签

网页上有各种元素，其中文本数量最多。因此，如何能更为合理地对文本进行组织，是设计者需要考虑的一个重要问题。

2.1.1 文本控制标签

与日常写作一样，在网页上展示文本内容时，为了使结构更加清晰，也需要进行标题和段落的划分。HTML 提供了标题标签、段落标签以及其他文本控制标签来实现这一点。

1. 标题标签

在 HTML 中，标题通过 `<h1>` 到 `<h6>` 这 6 个级别的标签进行划分。其中 `<h1>` 定义最大的标题，`<h6>` 定义最小的标题。在默认情况下，每个标题都会独占一行。

❖【示例 2-1】标题标签的使用。

```
<html>
  <body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3>This is heading 3</h3>
    <h4>This is heading 4</h4>
    <h5>This is heading 5</h5>
    <h6>This is heading 6</h6>
    <p> 请仅把标题标签用于标题文本。
      如果只是为了产生粗体文本而使用它们，
      请使用其他标签或 CSS 代替。</p>
  </body>
</html>
```

标题标签效果如图 2-1 所示。

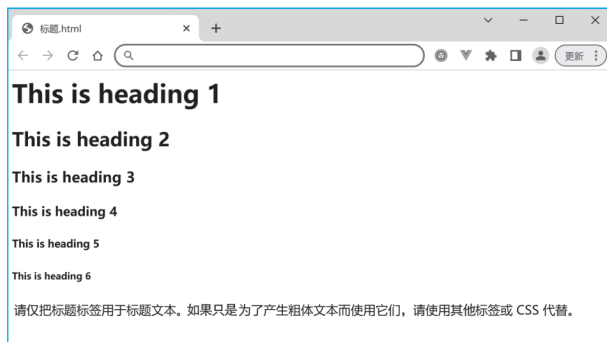


图 2-1 标题标签效果

搜索引擎使用标题为网页的结构和内容编制索引，用户通过标题来快速浏览网页，所以使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），其次是 `h2`（次重要的），再其次是 `h3`，以此类推。

注意

不要仅为了产生粗体或大号的文本而使用标题。

2. 段落标签

段落是通过 `<p>` 标签定义的。与标题类似，默认情况下，每个段落也会独占一行。这种默认情况下独占一行的元素称为块级元素，而按照顺序从左往右排列的元素称为行内元素。

注意

不要只是为了换行而使用段落标签。

❖【示例 2-2】段落标签用法。

```
<html>
  <body>
    <p> 段落默认独占一行，即便只有一个字符。</p>
    <p> 段落内的文本，不管在编辑器里怎么换行，
      在页面上都是到浏览器最右侧才会换行，除非
      设置了样式，或者使用了其他标签。
    </p>
    <p> 段落跟换行不一样，分段必然会换行，但换行未必就是分段。</p>
  </body>
</html>
```

段落标签效果如图 2-2 所示。

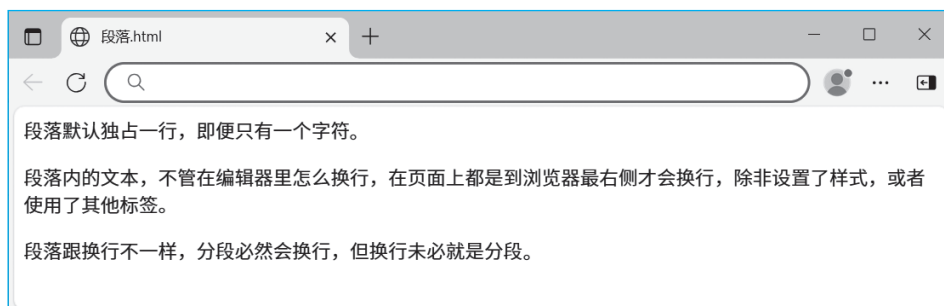


图 2-2 段落标签效果

3. 换行标签

如果在不产生一个新段落的情况下进行换行（新行），可以使用 `
` 标签。`
` 标签是一个单标签，尽管使用 `
` 标签也会产生换行效果，但是使用 `
` 标签更为规范。

❖【示例 2-3】换行标签用法。

```
<body>
  <p> 登鹤雀楼 </p>
  <p>
    白日依山尽，<br />
```

```
    黄河入海流。<br />
    欲穷千里目，<br />
    更上一层楼。<br />
</p>
</body>
```

换行标签效果如图 2-3 所示。

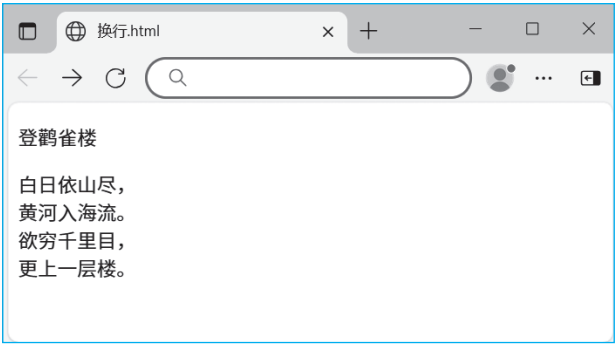


图 2-3 换行标签效果

4. 水平线标签

使用水平线，可以将页面划分为多个部分，使得页面结构更加清晰。水平线可以通过图片的形式插入页面。简单的水平线可以直接使用 <hr /> 标签实现。水平线标签的常用属性如表 2-1 所示。

表 2-1 水平线标签的常用属性

属性名	含义	属性值
align	设置水平线的对齐方式	可选择 left、right、center3 种值，默认为 center，居中对齐
size	设置水平线的粗细	以像素为单位，默认为 2 像素
color	设置水平线的颜色	可用颜色名称、十六进制 #RGB、rgb(r,g,b)
width	设置水平线的宽度	可以是确定的像素值，也可以是浏览器窗口的百分比，默认为 100%

在示例 2-3 代码的第 2 行后加入如下代码，即可实现水平线的插入。

```
<hr align="left" size="3" color="red" width="500">
```

水平线标签效果如图 2-4 所示。

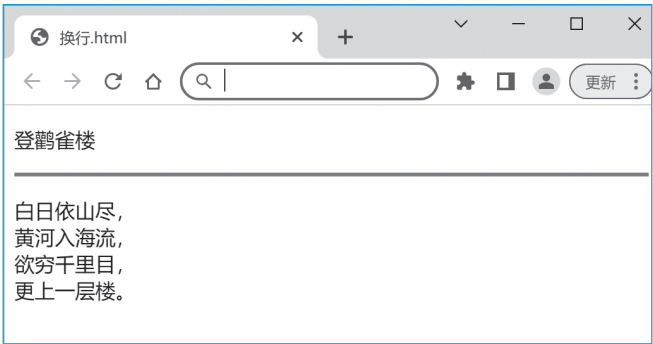


图 2-4 水平线标签效果

尽管可以通过属性设置对水平线进行修饰，但这样会使结构样式混在一起，建议使用 CSS 样式来完成效果

修饰。

5. 文本格式化标签

页面上经常会看到有些文字以加粗或斜体的形式出现，还有的文字带有下划线^①、删除线等效果。如果只是少数几处用到的话，这些效果可以通过表 2-2 中的文本格式化标签来实现。但更推荐的做法是使用专门处理文本的标签 结合 CSS 样式来实现这些效果。

表 2-2 文本格式化标签

标签	显示效果
 和 	文字以粗体方式显示（ 定义文本粗体， 定义强调文本）
<i></i> 和 	文字以斜体方式显示（<i> 定义斜体字， 定义强调文本）
<s></s> 和 	文字以加删除线方式显示（HTML5 不赞成使用 <s>）
<u></u> 和 <ins></ins>	文字以加下划线方式显示（HTML5 不赞成使用 <u>）

6. 特殊字符标记

如果需要在页面上显示人民币符号、版权符号等特殊标记的话，可以使用 HTML 提供的特殊字符标记，具体如表 2-3 所示。

表 2-3 特殊字符标记

HTML 代码	显示结果	描述
<	<	小于号或显示标记
>	>	大于号或显示标记
&	&	可用于显示其他特殊字符
"	“	引号
®	®	已注册
©	©	版权
™	™	商标
 		空格

2.1.2 图像标签

如果页面上只有文字，那么页面会比较单调，对访问者来说也缺乏足够的吸引力。适度地在页面上安排图片，可以使页面变得生动，也更有吸引力。

在 HTML 中，图像由 标签定义。 是单标签，它使用一条斜线表示标签的闭合。从技术上讲，图像并不会被插入到 HTML 页面中，而是被链接到 HTML 页面上。 标签的作用是为被引用的图像创建占位符。

1. 图像格式

图像有不同的类型，最常见的 3 种类型是 JPG、PNG 和 GIF。

- （1）JPG 格式是一种常用于摄影作品或写实作品的有损压缩图像格式。
- （2）PNG 格式是一种清晰、支持透明、无损压缩的位图图像格式。
- （3）GIF 格式是一种色彩效果差、体积小、支持透明和动画的图像文件格式。

^① “下划线” 应为 “下画线”，为与计算机专业方面使用习惯一致，本书采用 “下划线”。

在格式的选择上, 色彩丰富的图片建议首选 JPG, 如果图像需要支持透明格式, 建议首选 PNG, 尽管 GIF 也支持透明, 但它不能实现渐变透明。

2. 图像属性

`` 标签有两个必需的属性: `src` 和 `alt`。

要在页面上显示图像, 需要使用源属性 `src`, 它的值是图像的 URL 地址。定义图像的语法如下:

```

```

在图像加载的时候, 有时候会因为网络较差或者浏览器不兼容, 造成图像无法正常显示。这时候, 可以通过 `alt` 属性给图像加上一个替换文本, 告诉用户图像的信息。`alt` 属性除了可用于文本替换, 还可以用于网页搜索引擎优化, 做好 `alt` 属性的优化可以增加图片在 Google Image 的曝光量, 并引流到网站, 提高网站流量。其用法如下:

```

```

另外, 也可以通过 `title` 属性对图像进行更进一步的描述。当鼠标指针放置在定义了 `title` 属性的图像元素上方的时候, 页面会显示出 `title` 属性的值。

❖【示例 2-4】在页面上显示一幅图, 设置其 `alt` 属性及 `title` 属性的值为“红船精神”。

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

图像标签效果如图 2-5 所示。



图 2-5 图像标签效果

除了上述几个属性之外, 图像元素还有很多属性, 如 `width` (宽度)、`height` (高度)、`align` (对齐方式)、`border` (边框) 等。但是, 在开发过程中, 为了保证结构与样式的分离, 通常使用 CSS 样式表而不是元素属性来对图像进行样式设置, 所以在这里就不再对这部分内容进行详细讲解。

2.1.3 超链接标签

访问一个站点时, 通常会通过站点首页的 URL 地址进入该页面, 但每个网站都是由多个页面组成的, 要想进入其他页面, 就要通过超链接来进行页面间的跳转。

所谓的超链接是指从一个网页指向一个目标的连接关系, 这个目标可以是另一个网页, 也可以是相同网页

上的不同位置,还可以是图片、电子邮件地址、文件,甚至是一个应用程序。而在一个网页中用来设置超链接的对象,可以是一段文本或一个图片。当把鼠标指针移动到网页中的某个链接上时,箭头会变为一只小手。单击已经链接的文字或图片,链接目标将显示在浏览器上,并且浏览器会根据目标的类型来打开或运行。

1. 创建超链接

在 HTML 中,可以使用 <a> 标签来创建链接。<a> 标签对中的内容是链接对象。链接对象可以是文本,也可以是图像。超链接有两个基本属性。

(1) href: 表示链接地址。可以是网络上的地址,也可以是站内地址或当前页面中的某个位置,如果跳转目标位置未定,可以直接填入“#”。

(2) target: 表示链接页面打开的方式,其属性值可以为 _blank 或者 _self。_blank 表示链接页面在一个新的页签中打开,_self 表示链接页面在当前页签中打开。

❖【示例 2-5】创建一个超链接,单击文字跳转到学习强国网站。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>
    <a href="https://www.xuexi.cn/" target="_blank"> 学习强国 </a>
  </body>
</html>
```

2. 链接类型

网页上的超链接一般分为 3 种。

(1) 外部链接: 可以链接到网络上的一个站点或网页。

(2) 内部链接: 可以链接到同一个网站的其他网页。

(3) 锚点链接: 可以链接到同一个页面内的不同位置。

在一个内容较多的页面上,需要经常上下翻动页面来找到自己想要浏览的内容的情况下,使用锚点链接可以在页面上实现快速定位。

❖【示例 2-6】在页面中通过锚标记实现页面快速定位。

首先,创建一个锚标记,锚标记的名称可以使用元素的 id 值。其次,将超链接的 href 属性赋值为锚标记的值,锚标记前记得要加“#”,如“ 点我跳转到锚标记 ”。这样,当单击链接对象时,页面就会跳转到页面中 id 值为 top 的位置。

```
<body>
  <p id="top">
    <a href="#bottom"> 跳转到底部 </a>
  </p>
  <!-- 可将下面一行代码复制 50 次检验跳转效果 -->
  <p> 验证锚点链接 </p>
  <p id="bottom">
```

```
<a href="#top"> 跳转到顶部 </a>
</p>
</body>
```

在第2行和第7行代码处，分别定义了一个锚链接。定义锚链接时，id属性的值不加“#”。在第3行和第8行代码处，通过将链接的href属性赋值为锚标记，即可实现页面内部跳转的效果。需要注意的是，给href属性赋值的时候，需要加上“#”。

不管哪一种形式的超链接，都需要使用href属性定义跳转的目标路径。路径的描述方式有两种，分别是绝对路径和相对路径。

1) 绝对路径

绝对路径是指跳转的目标位置在网络上或在硬盘上的位置。例如，单击了某个链接对象后，要跳转到京东商城站点首页，那么用绝对路径描述该地址就是“http://www.jd.com”。

如果要跳转到当前站点中的某个位置，那么路径的描述方式就会类似于“D:\酒大美食网\dashifengcai.html”。

但需要说明的是，如果跳转到站点内的页面，最好用相对路径的方式。因为设计页面时在本地使用的路径和将页面发布到服务器上的路径很可能是不一样的。

2) 相对路径

相对路径是指从当前页面出发，到目标页面所经过的一系列目录形成的一条路径。以图2-6所示的文件夹结构为例，可以明显看出有3种相对路径描述方式。

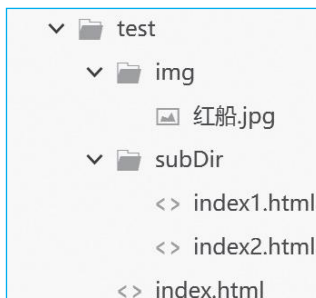


图 2-6 文件夹结构

(1) 目标位置与当前位置在同一个目录中。在 index1.html 文件中，如果希望通过超链接跳转到同一文件夹下的 index2.html 文件，可在 index1.html 文件的超链接内，直接将文件名赋值给 href 属性，代码如下：

```
<a href="index2.html">
```

(2) 目标位置在当前位置的下级目录中。如果跳转的目标文件在链接所在文件位置的下一级目录中，这时候需要从当前目录出发，然后进入下级目录，再列出目标文件。其中“./”表示当前目录。例如，在 index.html 文件内，希望通过超链接跳转到 subDir 文件夹下的 index1.html 文件，那么可以按照如下代码设置链接的 href 属性。

```
<a href="./subDir/index1.html">
```

(3) 目标位置在当前位置的上级目录中。如果链接跳转的目标位置在链接所在文件位置的上级文件夹中，则需要先跳出当前目录，然后再逐步进入目标位置。其中，跳出当前目录使用“../”的形式。例如，在 index1.html 中，需要使用 img 文件夹下的图像作为超链接，跳转到 index2.html 中那么 img 元素的 src 属性应当描述为“../img/红船.jpg”，代码如下：

```
<body>
  <p>
    <a href="../index2.html">
      
    </a>
```



```
</p>
</body>
```

2.2 使用样式表

CSS 简介



网页设计过程中,设计者除了使用 HTML 来定义页面上要显示出来的各种元素之外,还要考虑这些元素将要以一种什么样的形式展示给浏览者,如文字字体、字号大小、文字颜色、图片大小、位置等。

尽管通过 HTML 元素的属性设置,可以对页面效果进行修饰,但是这种方式会将结构和样式掺杂在一起,显得页面非常混乱,维护起来也很不方便。

使用样式表的对比如图 2-7 所示。图 2-7 (a) 使用了元素属性设置段落文字居中效果,而图 2-7 (b) 使用了样式表。如果要将段落文字对齐效果由居中对齐改为向左对齐,那么图 2-7 (a) 中需要对 3 个段落都进行修改,而图 2-7 (b) 只需要修改样式表里的文本对齐属性即可。很明显,使用样式表可以让工作效率变得更高。

<pre><head> <meta charset="utf-8"> <title></title> </head> <body> <p align="center">这是段落1</p> <p align="center">这是段落2</p> <p align="center">这是段落3</p> </body></pre>	<pre><style> p{ text-align: center; } </style> </head> <body> <p>这是段落1</p> <p>这是段落2</p> <p>这是段落3</p> </body></pre>
(a)	(b)

图 2-7 使用样式表的对比

因此,建议采用样式表来进行页面的修饰。使用样式表可以实现结构与样式的分离,也就是说,使用 HTML 来编写页面内容,使用样式表进行页面布局及元素样式修饰,这也符合页面开发的标准。

2.2.1 样式表的概念及作用

串联样式表 (Cascading Style Sheets, CSS),通常简称为样式表,可以使用它对网页中的元素进行像素级的控制。样式表最早于 1994 年提出,经过不断地迭代发展,形成了现在的 CSS3 版本。样式表的使用为网页设计者提供了巨大的便利,尤其是在设计页面数量较多的站点时,这种优势体现得尤为明显。

2.2.2 定义与引入样式表

正确使用样式表的前提是要弄清楚样式表的结构和定义方法。通常情况下,样式表可以分为以下 3 种不同的形式。

1. 行内样式表

行内样式表位于元素标签内,它通过元素的 style 属性进行样式的定义,格式如下:

```
< 标签 style=" 属性名: 属性值; ... "></ 标签 >
```

行内样式表使用元素属性 style 进行样式控制,它没有遵循结构与样式分离的原则,因此这种方式并不常用。如果只是对个别元素进行样式设置,可以使用这种方式,但不推荐广泛使用行内样式表进行设置。

2. 嵌入式样式表

嵌入式样式表一般位于页面的 head 部分,它使用一对 <style></style> 标签将样式包含起来,通过设置标签

的 type 属性值为“text/css”，通知浏览器这个标记包含的是 CSS 代码，格式如下：

```
<style type="text/css">
    选择器 {
        属性名:属性值;
        :
    }
</style>
```

嵌入式样式表除了可以放在页面的 head 部分内，也可以放到页面其他位置。但由于浏览器解析页面的顺序是自上而下的，因此不建议将嵌入式样式表放到页面 head 部分之外的位置。

注意

设计页面的时候，一般遵循先结构后样式的顺序。也就是说，要先把页面内容定义出来，然后再去进行样式设置。尽管出于浏览器解析页面的顺序考虑，把样式表放到了页面内容的前面，但是在设计页面的过程中，不能把顺序颠倒，因为这不符合页面设计的基本流程。

3. 外部样式表

外部样式表一般定义为一个后缀为 .css 的文件，然后在页面的 head 部分使用 <link> 标签进行引用，格式如下：

```
<link href="css/index.css" rel="stylesheet" type="text/css">
```

<link> 标签内有 3 个属性需要定义，其功能说明如表 2-4 所示。

表 2-4 <link> 标签属性的功能说明

属性	作用
href	外部样式表文件路径，可以是相对路径或绝对路径
rel	链接文档与当前文档的关系，设置为 stylesheet
type	链接文档类型，链接样式表需要设置为 text/css

使用外部样式表，可以实现结构与样式的完全分离。因此在实际开发过程中，外部样式表的使用是最多的。通常情况下，开发工程师会单独建立一个文件夹（一般以 css 定义该文件夹）用于存放多个外部样式表，如图 2-8 所示。然后可以在页面中根据需要使用 <link> 标签引入多个外部样式表文件，在外部样式表文件内可以直接定义样式，不需要使用 <style> 标签，如图 2-9 所示。

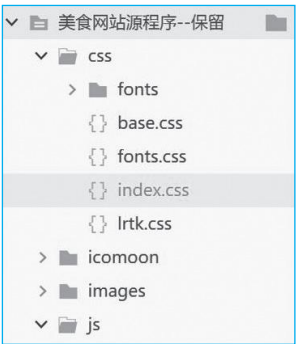


图 2-8 外部样式表位置



图 2-9 外部样式表的定义



上述 3 种类型的样式表，仅作了解即可，后续会根据任务需要选用不同类型的样式表。

2.2.3 基本类型选择器

样式表里定义的样式最终需要渲染到页面的元素上，这需要通过选择器来完成样式与元素之间的关联。选择器负责选中页面上的元素，然后浏览器会将样式渲染到选择器选中的元素上。基本类型选择器主要有标签选择器、类选择器、ID 选择器、标签指定选择器、后代选择器、通配符选择器、并集选择器等。选择器后面紧跟着样式的定义，选择器用法规则如图 2-10 所示。

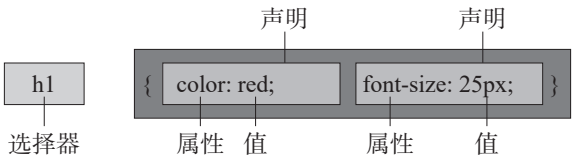


图 2-10 选择器用法规则

在上面的样式规则中，选择器用于指定样式作用的 HTML 对象。其中，花括号内是对该对象设置的具体样式；属性和属性值以“键值对”的形式出现；属性是对指定的对象设置的样式，如字体大小、文本颜色等；属性和属性值之间用英文“:”连接；多个“键值对”之间用英文“;”进行区分。

下面将具体讲述不同的选择器在用法上的区别。

1. 标签选择器

标签选择器又称标记选择器，是最常见的一种选择器类型。它使用元素的标签名作为选择器的名称，页面中凡是使用了该标签的元素，都会被这个选择器选中。

❖【示例 2-7】通过标签选择器将页面中的标题元素设置为红色，将所有段落文字大小设为 18 px。样式部分的参考代码如下：

```
<style>
  h2 {
    color: red;
  }
  p {
    font-size: 18px;
  }
</style>
```

使用标签选择器的效果如图 2-11 所示。

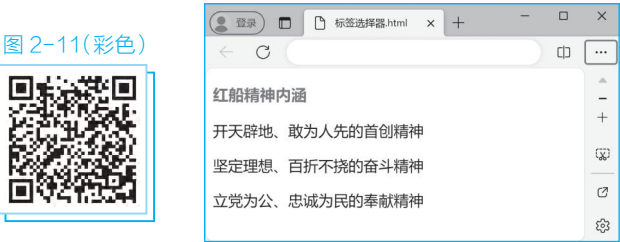


图 2-11 使用标签选择器的效果

2. 类选择器

类选择器选择有特定 class 属性的 HTML 元素，它使用“.”进行标识，后面紧跟类名。与



标签选择器不同，类选择器可以用于不同的元素上，凡是被类选择器选中的元素，都会呈现出样式中定义的特征。

注意

类名不能以数字开头，且严格区分大小写。

❖【示例 2-8】请将示例 2-7 中第二个段落的文字颜色设置为红色。

这里如果继续使用标签选择器，所有的段落都会变为红色。因此，可以使用类选择器，将第二个段落选中，实现单独设置一类元素中某个具体元素的显示效果。

在样式表中添加类选择器，代码如下：

```
.red { color: red; }
```

在页面中第二个段落标签内，使用 class 属性引用该选择器（或让 .red 选择器选中第二个段落），即可实现要求的效果。

```
<p class="red"> 坚定理想信念、百折不挠的奋斗精神 </p>
```

使用类选择器的效果如图 2-12 所示。

此外，还可以为元素添加多个类选择器的样式。例如，将示例 2-7 中的第一个段落设置为加粗效果，第二个段落设置为红色文字效果，第三个段落设置为红色加粗效果，可以添加一个类选择器用于设置加粗效果，代码如下：

```
.bold { font-weight: bold; }
```

然后将结构部分调整为如下代码，即可实现要求的效果。

```
<body>
  <h2> 红船精神内涵 </h2>
  <p class="bold"> 开天辟地、敢为人先的首创精神 </p>
  <p class="red"> 坚定理想信念、百折不挠的奋斗精神 </p>
  <p class="red bold"> 立党为公、忠诚为民的奉献精神 </p>
</body>
```

使用多个类选择器的效果如图 2-13 所示。

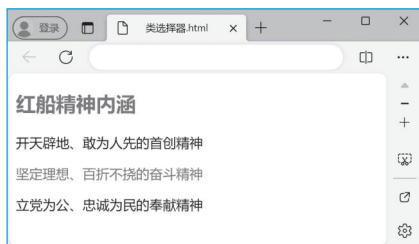


图 2-12 使用类选择器的效果

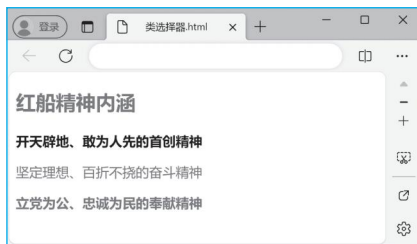


图 2-13 使用多个类选择器的效果

图 2-12(彩色)



图 2-13(彩色)

**提示**

在为元素添加多个选择器的时候，注意选择器名称之间要用空格隔开。

ID 选择器



3. ID 选择器

ID 选择器使用“#”开头，后跟该元素的 id。ID 选择器与类选择器最大的区别在于，类选择器可以在同一个页面中使用多次，但 ID 选择器只能用一次。

❖ 【示例 2-9】使用 ID 选择器将示例 2-8 中所有段落的文字颜色都改为红色。

分析：根据 ID 选择器的使用特点，如果要使用 ID 选择器完成本例，需要使用 3 个选择器分别对每个段落进行设置。这里可以尝试一下，如果像使用类选择器一样，给 3 个段落都设置了相同的 ID 选择器，结果会怎样呢？

观察发现，页面上所有的段落都被设置成了红色。尽管从页面执行效果来看没有问题，但不表示这么做是被允许的。由于在 JavaScript 中会使用 document.getElementById() 方法通过元素的 id 获取元素，且获取到的元素只能是唯一的，因此，在进行页面设计的时候，应当合理使用选择器，进而做到高效、准确地编写代码。

标签指定选择器



4. 标签指定选择器

标签指定选择器由两个选择器构成，第一个必须为标签选择器，第二个可以是类选择器或 ID 选择器中的任意一个，需要特别注意的是，这两个选择器之间不能有空格，具体格式为“h2.center”或“p#id”。

在示例 2-8 的基础之上，如果想给标题设置下划线，可以添加一个新的类选择器设置下划线效果，也可以使用标签指定选择器来完成设置。

(1) 添加新的类选择器，代码如下：

```
.underline{ text-decoration: underline; }
```

为标题添加选择器，代码如下：

```
<h2 class="center underline"> 红船精神内涵 </h2>
```

(2) 可以使用标签指定选择器，代码如下：

```
h2.center{ text-decoration: underline; }
```

使用标签指定选择器的效果如图 2-14 所示。

图 2-14(彩色)



图 2-14 使用标签指定选择器的效果

这种做法不需要修改页面元素部分的代码。

```
<h2 class="center"> 红船精神内涵 </h2>
```

可以看到，标题中 class 属性值没有变化，但是由于使用了标签指定选择器，标题元素会同时被 3 个选择器选中，分别是标签选择器、类选择器和标签指定选择器。完整样式代码如下：

```
<style>
  h2{
    color: red;
```

```

    }
    h2.center{
        text-decoration: underline;
    }
    .bold{
        font-weight: bold;
    }
    .red{
        color: red;
    }
    .center{
        text-align: center;
    }
</style>

```

5. 后代选择器

后代选择器匹配属于指定元素后代的所有元素，其写法就是把外层标签写在前面，内层标签写在后面，中间用空格进行分隔。当需要在一对标签内定义另外的标签，即发生标签的嵌套时，内层标签就成了外层标签的后代。

注意

标签的嵌套，一定是完整的嵌套，以下第二种形式不属于标签嵌套。

```

<body>
    <!-- 正确包含 -->
    <p><a href="https://www.xuexi.cn/"> 学习强国网站 </a></p>
    <!-- 错误包含 -->
    <p><a href="https://www.xuexi.cn/"> 学习强国网站 </p></a>
</body>

```

❖ 【示例 2-10】阅读代码，判断超链接的文字颜色。

```

<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <style>
            p a {
                color: red;
            }
        </style>
    </head>

```



```

<body>
  <p><a href="https://www.xuexi.cn/"> 学习强国网站 </a></p>
  <a href="https://www.xuexi.cn/"> 学习强国网站 </a>
</body>
</html>

```

分析：通过第 6 行至第 8 行代码可以得知，段落 p 的后代元素 a 的文字颜色应该是红色，且第 12 行代码处的超链接符合要求，因此它会被后代选择器选中，呈现红色文字效果。第 13 行代码处的第二个超链接元素，由于它不是段落 p 的后代，因此不会被选择器选中，呈现默认链接文本颜色。使用后代选择器的效果如图 2-15 所示。

6. 通配符选择器

通配符选择器使用“*”作为选择器名称，它可以选中页面上所有元素，一般用于清除元素的默认样式，具体用法在后面介绍盒模型的时候会进行讲解。

7. 并集选择器

并集选择器通过逗号将多个选择器连接起来，任何形式的选择器都可作为并集选择器的一部分。一般来讲，若多个元素具有相同的样式，就可以使用并集选择器来设置，以达到减少冗余代码的效果。

例如，使用并集选择器将示例 2-7 中所有页面元素的文字改为红色，对齐方式设置为居中对齐，代码如下：

图 2-15(彩色)



图 2-16(彩色)



```

h2, p {
  color: red;
  text-align: center;
}

```

使用并集选择器的效果如图 2-16 所示。



图 2-15 使用后代选择器的效果



图 2-16 使用并集选择器的效果

上述 7 种类型的选择器是最基本的选择器。在 CSS3 中，还有很多不同类型的选择器，如兄弟选择器、属性选择器等，这些都会在后续任务中进行介绍。在页面设计时，没有规定必须使用哪一种类型的选择器，只要根据实际情况，选择最便于进行设计的选择器类型即可。

2.3 使用文本样式

在浏览页面的时候，会看到页面上的文字有不同的字体和字号，也会看到有些文本是加粗或倾斜的，这些都可以通过文本类的样式属性进行设置。

2.3.1 字体样式

字体设置会对网页浏览者的观感产生巨大影响，使用合适的字体可以突出品牌形象、提升页面美感。例如，应当考虑网站的风格和定位选择字体，如新闻网站和儿童玩具网站的字体选择就应当有差别。不管选用什么字

体、字体大小是多少，都需要先进行一个整体的规范，统一网页字体的颜色、大小、粗细、间隔等。

一般来讲，网页字体宜选择宋体等比较常见的字体，尽量不要使用生僻字体，防止用户因没安装对应字体而造成显示错误。尽管各种浏览器都支持 @font-face 规则以供用户下载特殊字体，但考虑到字体下载、加载等因素造成的影响，还是建议谨慎使用特殊字体。

CSS 字体样式属性主要包括以下几个。

1. font-size 属性

font-size 属性用于定义字体的大小，其值可以使用相对长度单位，也可以为绝对长度单位。其中，相对长度单位比较常用，而绝对长度单位使用较少。推荐使用像素单位 px，具体如表 2-5 所示。

表 2-5 长度单位表

相对长度单位	说明	绝对长度单位	说明
em	相对于当前对象内文本的字体尺寸	in	英寸
		cm	厘米
px	像素单位，最常用，推荐使用	mm	毫米
		pt	点

2. font-family 属性

网页中常用的字体有宋体、微软雅黑、黑体等。font-family 属性可以用来定义字体。例如，将网页中所有段落文本的字体设置为微软雅黑，使用的 CSS 样式代码如下：

```
p { font-family: "微软雅黑", "宋体", "隶书"; }
```

注意

(1) 可以同时指定多种字体，中间以英文半角的逗号隔开。浏览器如果不能显示第一种字体，便会尝试使用下一种字体；如果所有字体都不能显示，便会按照默认字体显示。

(2) 中文字体需要加英文状态下的引号，英文字体一般不需要加引号。但如果英文字体名称中有空格或其他符号，也需要使用英文半角的单引号或双引号将字体名称包含起来。

3. font-weight 属性

font-weight 属性定义字体的粗细，该属性的可用值有 normal、bold、bolder、lighter、100~900（100 的整数倍）。其中，数字 400 等价于 normal，数字 700 等价于 bold。

提示

标签 和 也可以实现文本加粗的效果。

4. font-style 属性

font-style 属性定义字体风格。当需要对文本设置斜体、倾斜或正常字体效果时，可使用该属性，其常用的值如下。

(1) italic：显示斜体的字体样式（常用）。

(2) oblique：显示倾斜的字体样式（与 italic 效果相同，不常用）。

(3) normal：默认值，浏览器会显示标准的字体样式。

5. font 属性

font 属性是一个综合属性，在进行字体样式属性设置的时候，可以单独设置每个属性，也可以通过 font 属

性一次完成所有字体属性的定义。使用 font 综合属性时, 注意要严格按照 “font-style font-variant font-weight font-size/line-height font-family” 的顺序。

如果有属性不需要设置, 可将其省略, 但是 font-size 和 font-family 属性必须保留。

2.3.2 文本外观样式

1. color 属性

color 属性用于定义文本颜色, 其取值有以下 3 种方式。

- (1) 预定义的颜色值, 如 red、green、blue 等。
- (2) 十六进制, 如 #FF0000、#29D794 等。
- (3) RGB 值, 如红色可以表示为 rgb(255,0,0) 或 rgb(100%,0%,0%)。

注意

(1) 使用十六进制值定义颜色, 前两位表示红色的取值, 中间两位表示绿色的取值, 最后两位表示蓝色的取值。若某个颜色取值的两位数字相同, 则可将六位颜色值简写为三位的形式, 如 FFFFFFFF 可简写为 FFF、FFCC00 可以简写为 FC0。

(2) 如果使用百分比定义颜色, 那么 0% 的百分号不能省略。

2. letter-spacing 属性

letter-spacing 属性用于定义字符之间的距离, 其属性值可以是不同单位的数值, 允许使用负值, 默认为 normal。

3. word-spacing 属性

word-spacing 属性用于定义英文单词之间的距离, 其属性值可为不同单位的数值, 允许使用负值, 默认为 normal。

word-spacing 属性与 letter-spacing 属性都可以对英文进行设置, 但 letter-spacing 对中文也起作用, 而 word-spacing 对中文无作用。页面部分代码如下:

```
<body>
  <p class="l_spacing">letter-spacing 对英文和中文都起作用 </p>
  <h3 class="w_spacing">word-spacing 属性对中文不起作用 </h3>
  <h3 class="fuzhi">可以使用负值 </h3>
  <!-- letter-spacing 和 word-spacing 对英文都起作用 -->
  <h3 class="l_spacing w_spacing">this is an example.</h3>
</body>
```

样式部分代码如下:

```
<style>
  .l_spacing { letter-spacing: 15px; }
  .w_spacing { word-spacing: 30px; }
  .fuzhi { letter-spacing: -5px; }
</style>
```

效果如图 2-17 所示。



图 2-17 中英文字符间距效果

4. line-height 属性

在 CSS 中，定义了顶线、中线、基线和底线 4 条线。其中，顶线位于中文汉字的上端沿，中线位于英文字母“X”的中间交叉点的位置，基线位于英文字母“X”的下端沿，而不是汉字文字的下端沿，底线位于中文汉字的下端沿。

line-height 属性用于定义行高，它表示的是两行的基线之间的距离。而行距（见图 2-18）表示的是上一行的底线到下一行的顶线之间的距离。二者并不相同，但通常可以通过调整 line-height 属性的值来调整两行之间的距离，即行距。

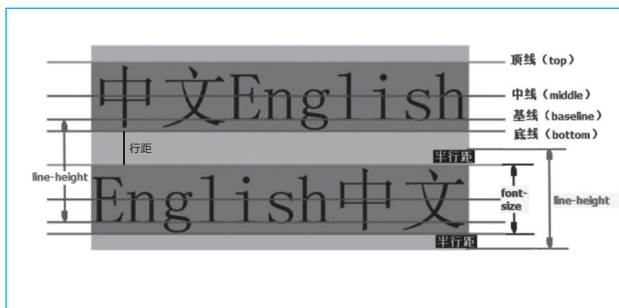


图 2-18 行距示意图

line-height 属性值常用的单位有 3 种，分别为像素 px、相对值 em 和百分比 %。在实际工作中，使用最多的是像素 px。

5. text-transform 属性

text-transform 属性用于控制英文字符的大小写，其可用属性值如下。

- (1) none: 不转换（默认值）。
- (2) capitalize: 首字母大写。
- (3) uppercase: 全部字符转换为大写。
- (4) lowercase: 全部字符转换为小写。

6. text-decoration 属性

text-decoration 属性用于设置文本的下划线、上划线、删除线等装饰效果，其属性值如下。

- (1) none: 没有装饰（正常文本默认值）。
- (2) underline: 下划线。
- (3) overline: 上划线。
- (4) line-through: 删除线。

text-decoration 的值可以有多个，用于给文本添加多种效果。

7. text-align 属性

text-align 属性用于设置文本对齐方式，其属性值如下。

- (1) left: 靠左对齐 (默认值)。
- (2) center: 居中对齐。
- (3) right: 靠右对齐。

8. text-indent 属性

text-indent 属性用来设置文本的缩进。一般而言，text-indent 属性是只用在块级元素上面的，而不用在内联元素上面。text-indent 的数值可以为正，也可以为负 (em 或 px)，还可以设置为百分比。但需要注意的是，百分比是根据其父元素宽度的大小进行计算取值的。页面内容代码如下：

```
<body>
  <!-- 缩进 2 em，会按照当前段落元素字体大小的 2 倍进行缩进。 -->
  <p class="p1">text-indent 属性用来设置文本的缩进。一般而言，text-indent 属性是只用在块级元素上
  面的，而不用在内联元素上面。text-indent 的数值可以是正，也可以是负 (em 或者 px)，还可以设置 %。</p>
  <p class="p2">使用像素值作为单位进行缩进时，不管当前段落文字大小是多少，都会按照设定的缩
  进大小进行缩进。</p>
  <p class="p3">使用百分比进行缩进的时候，是根据其父元素宽度大小进行计算取值的。</p>
  <p class="p4">如果缩进的值是负数，那从效果上来看就类似于悬挂缩进的效果了。</p>
</body>
```

样式部分代码如下：

```
<style>
  p {
    width: 500px;
    margin: 0 auto; /* 此行代码实现段落居中，后面会讲到。 */
  }
  .p1 { text-indent: 2em; }
  .p2 { text-indent: 50px; }
  .p3 { text-indent: 10%; }
  .p4 { text-indent: -2em; }
</style>
```

文本缩进属性效果如图 2-19 所示。

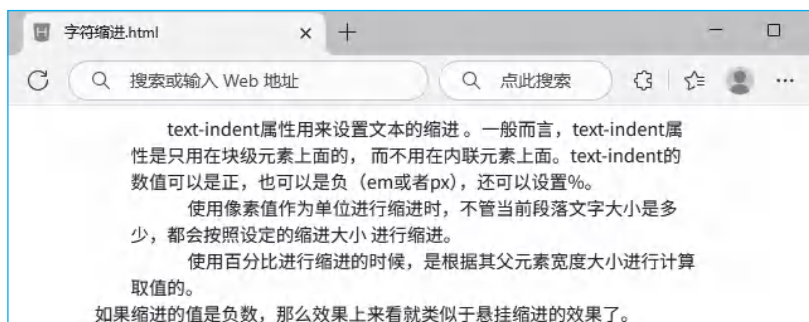


图 2-19 文本缩进属性效果

任务实施

本任务主要完成舌尖文化页面的设计。该页面设计主要融合了文本、图像、超链接等元素及各种基本类型选择器的用法。首先需要完成页面结构的布局。从效果图来看，页面可以简单分为头部、主体、底部3个部分，每个部分又可以再次划分为不同的部分。页面整体效果如图2-20所示。

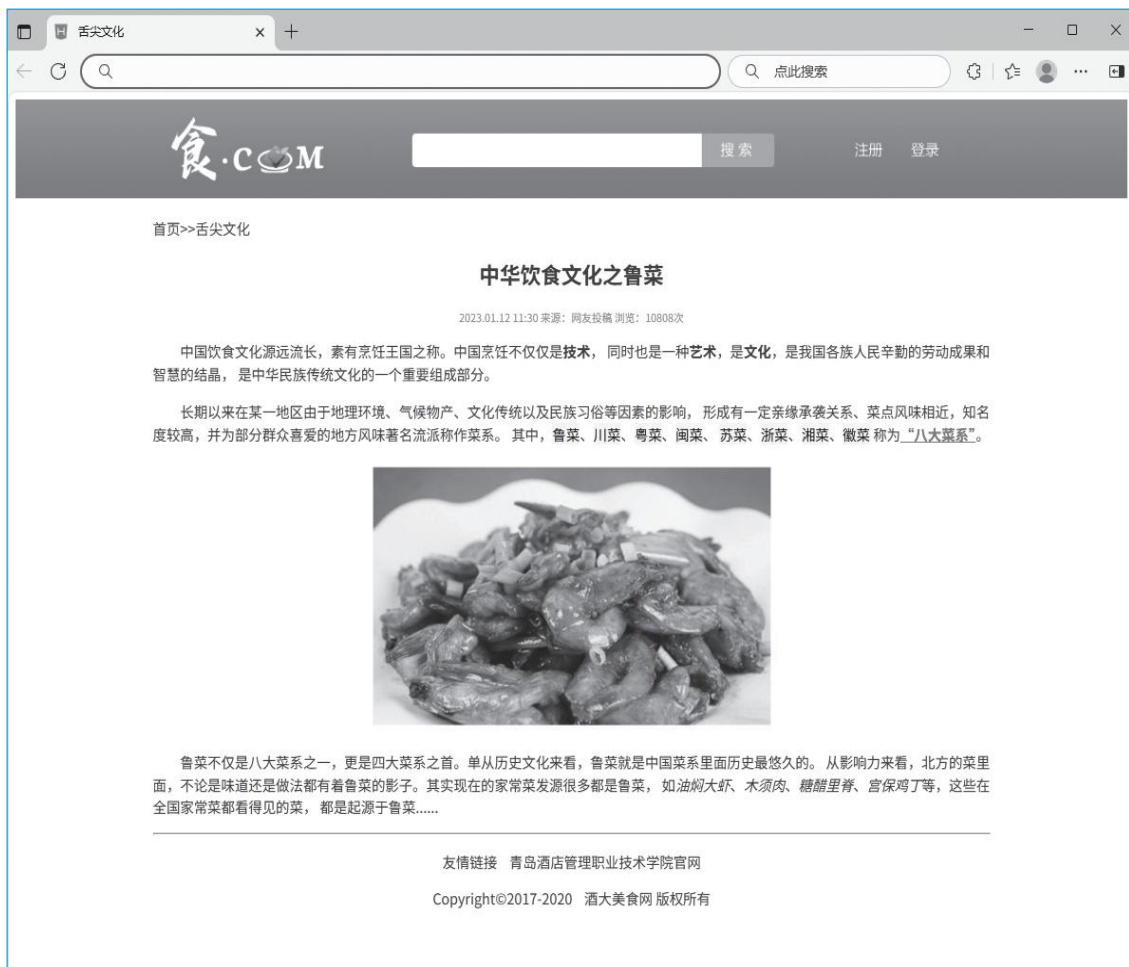


图2-20 页面整体效果

页面主体主要包含1个标题、3个段落及1张图。这些都可以通过使用简单的段落、标题和图像标签进行定义。页面底部主要是版权和链接，可以通过超链接元素来实现。

1. 定义页面整体结构

HTML5添加了一些语义布局标签，通过这些标签可以清楚地了解某一块内容代表哪一部分。例如，header元素表示整个区域或其中一个区域的头部区域，通常用来放置整个页面或页面内一个内容区块的标题，但也可以包含其他的内容，如在header里面放置logo图片、搜索表单等。而footer元素表示页面的底部，以前在布局网页底部版权时，习惯使用“id=“footer””或“class=“footer””表示页面的底部，而在HTML5中可以直接使用<footer>标签。

注意

一个页面内并没有限制header元素的出现次数。也就是说在同一页面内，可以为不同的内容区块分别加上一个header元素。

2. 头部设计

因为头部的设计需要使用到盒模型的知识，所以这里暂时用一张图片替代，后续任务中再进行详细讲解，参考代码如下：

```
<header>
    
</header>
```

3. 主体部分设计

进行主体部分设计时，首先需要定义一个宽度，然后再设置居中显示的效果，这部分内容也需要对盒模型有所了解，暂时先不讲，但可以先按照参考代码完成样式设置，在后续任务中再作详细解释。样式代码如下：

```
.wrapper, footer {
    width: 980px;
    margin: 0 auto;
}
```

可以看到，主体和底部的宽度都设置为 980 px，并通过将外边距 margin 属性的值设置为 0 auto 来实现主体水平居中。

页面主体左上方是一个面包屑导航，这部分可以用段落和超链接来实现，代码如下：

```
<div class="wrapper">
    <p class="nav_text"><a href="index.html"> 首页 </a>><a href="#"> 舌尖文化 </a></p>
    <h2 class="center"> 中华饮食文化之鲁菜 </h2>
    <p class="info center"><time>2023.01.12 11:30 </time> 来源：网友投稿 浏览：10808 次 </p>
    <p class="content"> 中国饮食文化源远流长，素有烹饪王国之称。中国烹饪不仅仅是 <b>技术 </b>，
    同时也是一种 <b>艺术 </b>，是 <b>文化 </b>，是我国各族人民辛勤的劳动成果和智慧的结晶，是中华民族
    传统文化的一个重要组成部分。
    </p>
    <p class="content"> 长期以来在某一地区由于地理环境、气候物产、文化传统以及民族习俗等因素
    的影响，形成有一定亲缘承袭关系、菜点风味相近，知名度较高，并为部分群众喜爱的地方风味著名流派称
    作菜系。其中，<mark>鲁菜 </mark>、<mark>川菜 </mark>、<mark>粤菜 </mark>、<mark>闽菜 </mark>、
    <mark>苏菜 </mark>、<mark>浙菜 </mark>、<mark>湘菜 </mark>、<mark>徽菜 </mark> 称为 <strong>“八
    大菜系” </strong>。
    </p>
    <p class="center"></p>
    <p class="content"> 鲁菜不仅是八大菜系之一，更是四大菜系之首。单从历史文化来看，鲁菜就是中
    国菜系里面历史最悠久的。从影响力来看，北方的菜里面，不论是味道还是做法都有着鲁菜的影子。其实现在
    的家常菜发源很多都是鲁菜，如 <em>油焖大虾 </em>、<em>木须肉 </em>、<em>糖醋里脊 </em>、<em>
    宫保鸡丁 </em> 等，这些在全国家常菜都看得见的菜，都是起源于鲁菜 .....
    </p>
    <hr>
</div>
```


如果不对超链接进行样式处理,超链接默认会带有下划线,但大部分超链接都不显示下划线,因此需要通过样式表进行设置来取消下划线效果,代码如下:

```
body { color: #333333; }
a { text-decoration: none; color: #333; }
```

可以通过将 text-indent 属性的值设为 0 em,实现段落中的文本靠左对齐,代码如下:

```
.wrapper p.nav_text { text-indent: 0em; }
```

页面的标题使用 <h2> 标签,其文字居中显示,代码如下:

```
.center { text-align: center; }
```

标题下的信息与其他文字相比颜色较浅,可以单独设置样式,代码如下:

```
.wrapper p.info {
    text-indent: 0em;
    font-size: 12px;
    color: #999999;
}
```

需要注意的是,“class=“info center””使用了两个类选择器,中间需要用空格分开。另外,页面使用了 HTML5 新增的 <time> 标签,该标签用于定义日期和时间,且以机器可读的方式对日期和时间进行编码,可以帮助搜索引擎生成更智能的搜索结果。

后面几个段落的结构基本类似,差别在于文字的修饰。带有加粗效果的文字可以使用 标签或 标签实现;斜体效果可以使用 标签实现;黄色底色的文字可以使用 <mark> 标签来实现,这个标签也是 HTML5 中新增的一个标签,它包含的文字会显示出黄色的底色。

4. 底部设计

底部设计比较简单,可以通过段落超链接实现,这里不再详细讲解。样式代码整体如下:

```
<style type="text/css">
    body {
        color: #333333;
    }
    a {
        text-decoration: none;
        color: #333;
    }
    .center {
        text-align: center;
    }
    header {
        text-align: center;
        background: url('img/bg.png') repeat-x;
    }
}
```



```
.wrapper, footer {  
    width: 980px;  
    margin: 0 auto;  
}  
.wrapper p.content {  
    text-indent: 2em;  
    line-height: 1.5;  
}  
.wrapper p.info {  
    text-indent: 0em;  
    font-size: 12px;  
    color: #999999;  
}  
.wrapper p.nav_text {  
    text-indent: 0em;  
}  
.wrapper p strong {  
    text-decoration: underline;  
    color: #ff0000;  
}  
</style>
```