项目 3

实现网页布局

知识目标

- 了解 CSS 的基本知识。
- 了解网页布局基本知识。
- 理解盒模型原理。
- 理解元素定位的方法。
- 理解浮动的概念。
- 了解流式布局和弹性布局。
- 了解 CSS 框架的原理。

技能目标

- 会选择合适的 CSS 选择器。
- 掌握网页元素的定位方法。
- 掌握使用 CSS 进行网页基本布局的方法。
- 能够制作兼容不同浏览器窗口尺寸的布局。
- 会使用 CSS 框架。
- 能够使用 960 CSS 框架快速布局网页。

素养目标

- 理解网页元素布局的思想。
- 探索自定义 CSS 框架。
- 具有设计页面布局的能力。

任务　进行网页布局

▶ 学习情境

小黄通过项目 2 的学习，掌握了 HTML 5 基本知识，开始着手建设网站。首先需要设

计网页的基本结构，也就是布局。

本任务主要介绍 CSS 的基本概念，网页布局的知识和基本方法，理解盒模型、定位以及浮动的概念，通过使用 CSS 进行网页布局，并且实现固定宽度、流式和弹性布局，最后介绍使用 CSS 框架进行网页布局的方法。

▶ 任务描述

根据实际需求进行网页布局。公司 Logo、导航列表和搜索框在页面上方，这里称为头部；中间是主内容区域，分为两列，左侧是产品图片列表，右侧是新闻列表；下面放置公司地址、版权等信息。布局如图 3-1 所示。

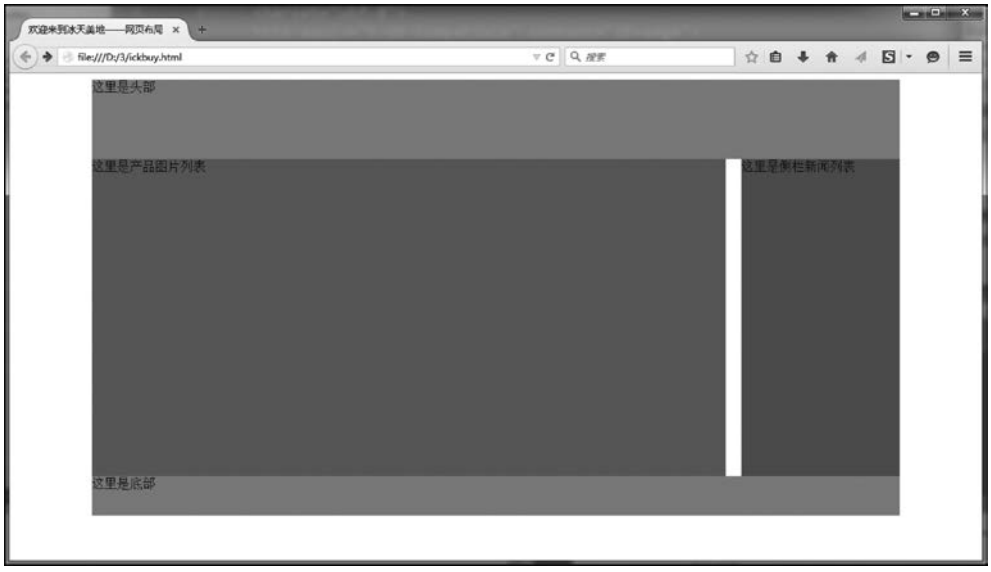

图 3-1　布局效果图

本任务的主要内容如下。

(1) 设计网页固定宽度的布局。

(2) 根据浏览器窗口尺寸自适应的布局。

(3) 利用 CSS 框架快速布局。

问题引导：

(1) 什么是 CSS?

(2) 什么是网页布局?

(3) 常见的网页布局有哪几种?

(4) 如何实现网页布局?

▶ 任务知识

表格布局方式使代码越来越难以理解和维护，而 CSS(层叠样式表)不仅可以控制页面

的外观，并且将文档的表现部分与内容分隔开。目前主流网站都是采用 CSS 来控制样式。

1. 认识 CSS

1）CSS 简介

层叠样式表 CSS(cascading style sheets)是一种指定文档该如何呈现给用户的语言。它给我们带来很多好处，如避免重复、更容易维护，以及为不同目的，使用不同的样式而内容相同。

样式可以被定义在外部文件(外链样式，如 style.css)，也可以在页面的头部定义(内联样式)，或者定义在特定的元素上(行内样式)。

CSS 规则主要由两个部分构成：选择器与一条或多条声明，结构如下。

```
selector {declaration1; declaration2; ... declarationN }
```

例如：

```
p {font-size: 20px; color: #F00; }
```

选择器通常用于需要改变样式的 HTML 元素。

每条声明由一个属性和一个值组成，属性是希望设置的样式属性，每个属性有一个值，属性和值被冒号分开，结构如下。

```
selector {property: value}
```

CSS 注释以“/*”开始，以“*/”结束。养成良好的注释习惯会给将来的阅读和维护带来很大的方便。

2）使用 CSS 的方法

(1) 行内样式

行内样式(inline style)也称为内联样式，即在 HTML 的标签中使用 style 属性来设置 CSS 样式。

例如：

```
<p style="color:#F00; font-size:20px;">我喜爱的运动：</p>
```

其作用范围是本标签。

(2) 内联样式

内联样式是指将样式放在本页面的<head>标签中，使用<style type="text/css"> ... CSS 样式代码 </style> 进行声明。

例如：

```
<head>
<meta charset="utf-8">
```

```
<title>有序列表</title>
<style type = "text/css">
    p {
        font - size: 20px;
        color: #F00;
    }
</style>
</head>
```

其作用范围是当前整个页面。

(3) 外部样式表

外部样式表分为链入外部样式表和导入外部样式表。

① 链入外部样式表。链入外部样式表也称为链接外部样式表,是指将样式表保存为外部样式表文件,如 main.css,再在页面的<head>标签中用<link>标签链接到这个样式表文件。

例如:

```
<head>
    <meta charset = "utf - 8">
    <title>有序列表</title>
    <link href = "main.css" rel = "stylesheet" type = "text/css">
</head>
```

样式表存放在 main.css 外部样式表文件中,代码如下。

```
@charset "utf - 8";
p {
    font - size: 20px;
    color: #F00;
}
```

其作用范围是引用该样式表文件的页面。

② 导入外部样式表。导入外部样式表是指使用 import 指令在<style>标签中导入 CSS 文件。

```
<head>
    <meta charset = "utf - 8">
        <title>有序列表</title>
    <style type = "text/css">
            @import "main.css";
    </style>
</head>
```

其作用范围是导入该样式表文件的页面。

链入外部样式表和导入外部样式表的区别如下。

使用链入外部样式表时，会在加载页面主体部分之前加载 CSS 文件，这样显示出来的网页从一开始就带有样式效果。

使用导入外部样式表时，在整个页面加载完成后再加载 CSS 文件。对于有的浏览器来说，在一些情况下，如果网页文件比较大，就会先显示无样式页面，在闪烁一下之后才出现设置样式后的效果。用户体验不佳。

为了便于维护，一般应如下选择使用方式。

如果仅需要引入少量 CSS 文件，则使用链接方式。

如果需要引入多个 CSS 文件，则首先用链接方式引入一个 CSS 文件，在这个 CSS 文件中再使用导入方式引入其他 CSS 文件。

练习 3-1：用上述四种 CSS 使用方式实现如图 3-2 所示效果的网页，要求文本颜色为红色，字体大小为 20px。

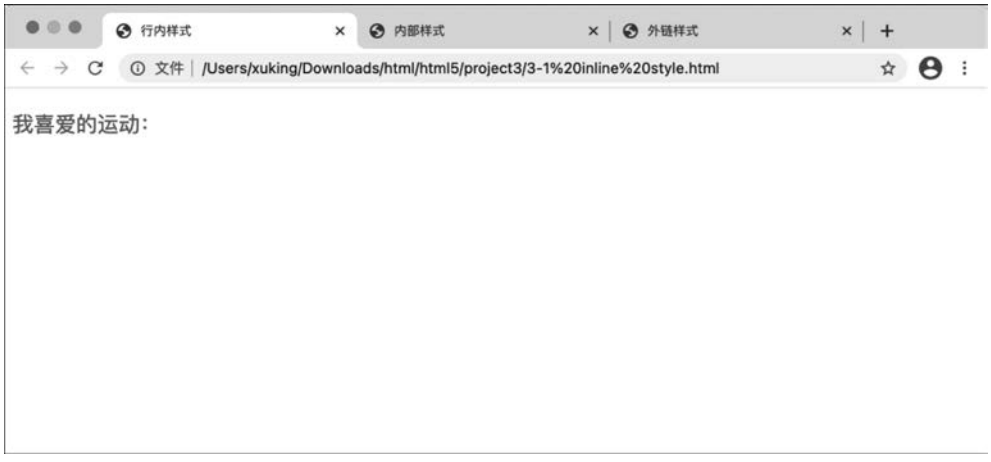

图 3-2　练习 3-1 效果图

2. 选择器

1）基本选择器

浏览器在展示一个文档的时候，必须要把文档内容和相应的样式信息结合起来展示，结合的依据来自 CSS 选择器。CSS 选择器有多种类型，如元素选择器、类选择器、ID 选择器、通用选择器等。掌握这些选择器，才能灵活地设置样式效果。

（1）元素选择器。文档的元素是最基本、最常见的选择器。元素选择器又称为类型选择器。如 h2 {color:blue;}，该代码将 h2 级别的标题设置为蓝色。设置的元素可以是 HTML 文档中所有的标签元素，甚至是<html>标签本身，如 html{color:red;}使文本呈现红色。

（2）类选择器。除了元素名称，还可以在选择器中使用属性值，这样可以更具体地描述规则。其中，class 和 id 两个属性比较重要。通过设置元素的 class 属性，可以为元素指定类

名，类名由开发者自己指定。文档中的多个元素可以拥有同一个类名。在编写样式表时，类选择器是以英文句号“.”开头的。

以下代码中，<p>元素具有 class 属性。

```
<p class = "key">
```

在一个 CSS 样式表中，以下代码中的规则会使所有 class 属性为 key 的元素文字的颜色显示为绿色。

```
.key {
  color: green;
}
```

(3) ID 选择器。通过设置元素的 id 属性为该元素指定 ID，ID 由开发者定义。需要注意的是：每个 ID 在文档中必须是唯一的。在编写样式表时，ID 选择器以“#”开头。

例如，以下代码中的规则会设置 id 属性为 principal 的元素字体大小为 20px。

```
#principal{
    font-size:20px;
}
```

类选择器与 ID 选择器都可以应用于任何元素。两者的区别如下。

① ID 选择器在文档中只能使用一次，而类选择器可以使用多次。

例如，以下代码是正确的。

```
<p class = "key"> ID 选择器在文档中只使用一次</p>
<p class = "key">而类选择器可以使用多次</p>
```

而以下代码是不正确的。

```
<p id = "key"> ID 选择器在文档中只使用一次</p>
<p id = "key">而类选择器可以使用多次</p>
```

② 可以为同一个元素使用多个类选择器的方式实现其多个样式，但是 ID 选择器不可以。

例如，以下代码是正确的。

```
<p class = "key principal">可以为同一个元素使用多个类选择器的方式实现其多个样式，但是 ID 选择器不可以.</p>
```

而以下代码是不正确的。

```
<p id = "key principal">可以为同一个元素使用多个类选择器的方式实现其多个样式，但是 ID 选择器不可以.</p>
```

（4）通用选择器。通用选择器用一个“*”表示，它的作用是匹配所有的可用元素。例如，以下代码实现了去除页面中所有元素的内边距和外边距的功能。

```
*{
    padding:0;
    margin:0;
}
```

2）伪类选择器

CSS 伪类是加在选择器后面的用来指定元素状态的关键字。比较常见的链接状态就是使用伪类选择器来实现的。以下代码可以实现未访问链接、已访问链接、鼠标悬停和选定的链接效果。

```
a:link {color: #FF0000}          /* 未访问的链接 */
a:visited {color: #00FF00}       /* 已访问的链接 */
a:hover {color: #FF00FF}         /* 鼠标移动到链接上 */
a:active {color: #0000FF}        /* 选定的链接 */
```

注意：在 CSS 定义中，a:hover 必须被置于 a:link 和 a:visited 之后，才是有效的。a:active 必须被置于 a:hover 之后，才是有效的。

:first-child 用于向元素的第一个子元素添加样式。例如：

```
<div>
  <span>这里是绿色显示。</span>
  <span>这里不是绿色。</span>
</div>
```

对以上代码设置如下样式后，效果如图 3-3 所示。

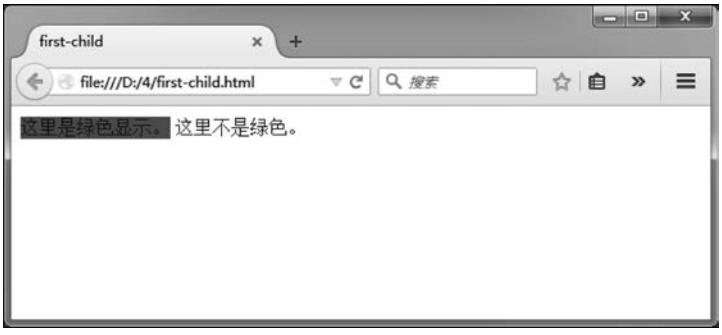

图 3-3 :first-child 样式实例

```
span:first-child {
    background-color: green;
}
```

表 3-1 所示为常见伪类名称及其功能描述。

表 3-1　常见伪类名称及其功能描述

伪 类 名 称	功 能 描 述
:link	向未访问链接添加样式
:visited	向访问过的链接添加样式
:active	向选定的链接添加样式
:hover	光标悬停在元素上方时,向元素添加样式
:focus	向拥有键盘输入焦点的元素添加样式
:first-child	向元素的第一个子元素添加样式
:lang	允许创作者来定义制定的元素中使用的语言

3）伪元素选择器

CSS 伪类用于向某些选择器添加特殊的效果,而伪元素用于将特殊的效果添加到某些选择器。

两者看起来很相似,都是与选择器相关,而且都是添加特殊效果,但是它们是有区别的,例如:

```
<style>
    p>i:first-child {color: red}
</style>
<p>
    <i>第一个 i</i>
    <i>第二个 i</i>
</p>
```

以上代码采用了伪类:first-child,其效果如图 3-4 所示。

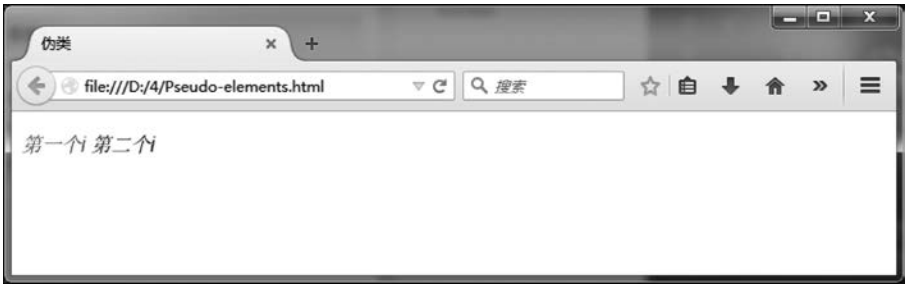

图 3-4　伪类实例

如果希望不用伪类而达到上面的效果,就要为 i 元素添加一个类,代码如下。

```
<style>
    .first-child {color: red}
</style>
```

```
<p>
    <i class = "first - child">第一个 i</i>
    <i>第二个 i</i>
</p>
```

再看伪元素 first-letter，HTML 和 CSS 代码如下，效果如图 3-5 所示。

```
<style>
    p:first - letter{color: red;}
</style>
<p>
    这里第一个字为红色。
</p>
```

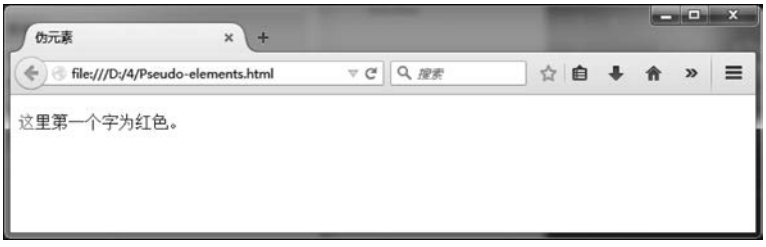

图 3-5　伪元素实例

同样，如果还是不希望用伪元素达到图 3-5 所示的效果，就要为第一个字添加一个＜span＞，然后给＜span＞添加样式，代码如下。

```
<style>
    .first - letter{color: red;}
</style>
<p>
    <span class = "first - letter">这</span>里第一个字为红色。
</p>
```

从这个例子中可以看出：伪类的效果可以通过添加一个实际的类来达到，而伪元素的效果需要通过添加一个实际的元素才能达到。

注意：:first-letter 伪元素只能用于块级元素。表 3-2 是常见伪元素的属性及含义。

表 3-2　常见伪元素的属性及含义

属　　性	含　　义
:first-letter	向文本的第一个字母添加特殊样式
:first-line	向文本的首行添加特殊样式
:before	在元素之前添加内容
:after	在元素之后添加内容

4）属性选择器

属性选择器可以根据元素的属性及属性值来选择元素。

例如，将所有包含标题的元素变成红色，可以用以下代码。

```
<style>
    *[title]{ color: red; }
</style>
<p>这里没有 title 属性</p>
<a href="#" title="title1">这里是具有 title 属性的链接。</a>
```

实现的效果如图 3-6 所示。

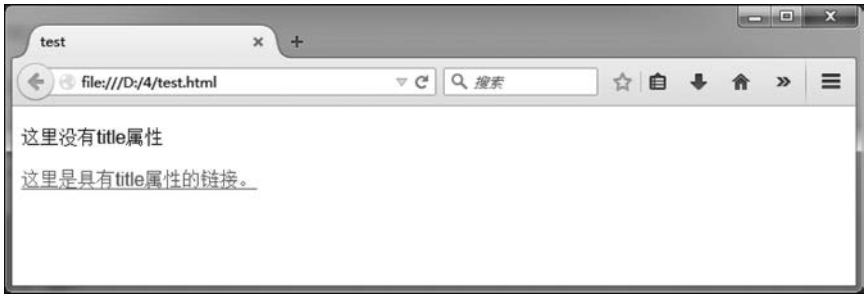

图 3-6　属性选择器实例

还可以根据多个属性进行选择，只须将属性选择器链接在一起即可。例如，a[href][title]{color:red;}将同时具有 href 和 title 属性的 HTML 超链接的文本设置为红色。也可以根据具体属性值来缩小选择范围，如 a[href="http://www.whit.ah.cn"]{color:red;}使只有指向 Web 服务器上某个指定地址的超链接变成红色。

表 3-3 是属性选择器类型及其功能描述。根据某个属性是否存在或属性的值来寻找元素，能够实现很强大的效果。

表 3-3　属性选择器类型及其功能描述

选　择　器	功 能 描 述
[attribute]	用于选取带有指定属性的元素
[attribute=value]	用于选取带有指定属性和值的元素
[attribute~=value]	用于选取属性值中包含指定词汇的元素
[attribute|=value]	用于选取带有以指定值开头的属性值的元素，该值必须是整个单词
[attribute^=value]	匹配属性值以指定值开头的每个元素
[attribute$=value]	匹配属性值以指定值结尾的每个元素
[attribute*=value]	匹配属性值中包含指定值的每个元素

3. CSS 属性

CSS 属性可以分为字体属性、颜色及背景属性、文本属性、方框属性等几类。表 3-4 为