

# 第 1 章

## HTML5 结构

### 本章技能目标

- 了解 HTML5 的应用
- 了解 HTML5 及其发展
- 掌握 HTML5 的网页结构和新增元素
- 会用高级选择器美化网页

### 本章简介

从 2010 年起，HTML5 和 CSS3 就已经成为了互联网技术一直关注和讨论的话题，在 1999 年 HTML4 就已经停止开发了，直到 2008 年 1 月 22 日才公布了 HTML5 的第一份正式草案。2010 年，HTML5 开始用于解决实际问题。这时，各大浏览器厂商开始升级自己的产品以支持 HTML5 的新功能。尽管有时候浏览器对其支持得不是十分友好，但大部分现代浏览器已经基本支持了 HTML5。现在 HTML5 已经广泛应用于网站制作、游戏开发、移动应用开发等各种行业。



目前 HTML5 技术已经日趋成熟，支持 HTML5 的浏览器包括 Firefox（火狐浏览器）、IE 9 及其更高版本、Chrome（谷歌浏览器）、Safari、Opera 等；国内的傲游浏览器（Maxthon），以及基于 IE 或 Chromium（Chrome 的工程版或称实验版）所推出的 360 浏览器、搜狗浏览器、QQ 浏览器、猎豹浏览器等国产浏览器同样具备支持 HTML5 的功能。随着谷歌、Twitch、YouTube 等大型企业将视线投向 HTML5，更加确认了 HTML5 在互联网时代的发展前景。在不久的将来，HTML5 将会与我们的生活息息相关，HTML5 不仅在 PC 端，更是在移动端上有着广泛的应用和发展。

## 1 什么是 HTML5

HTML5 是用于取代 1999 年所制定的 HTML 4.01 和 XHTML 1.0 标准的 HTML 标准版本。HTML5 首先强化了 Web 网页的表现性能，其次追加了本地数据库等 Web 应用的功能。HTML5 实际指的是包括 HTML、CSS 和 JavaScript 在内的一套技术组合，它希望能够减少浏览器对于需要插件的丰富性网络应用服务（Rich Internet Application, RIA），如 Adobe Flash、Microsoft Silverlight，与 Oracle JavaFX 的需求，并且提供更多能有效增强网络应用的标准。

HTML5 具有如下特性：

- 跨平台

HTML5 可以运行在 PC 端、iOS 或 Android 移动设备端，和服务器以及设备无关，只要有一个支持 HTML5 的浏览器即可运行。

- 兼容性

如果读者以前做过 Web 前端开发，就会了解 Web 兼容性（尤其是 IE6）是一件多么让人崩溃的事，几乎要为每一个浏览器做兼容处理。但是 HTML5 趋于成熟，只要浏览器支持 HTML5 就能实现各种效果，开发人员不需要再写浏览器判断之类的代码。

- 各大浏览器厂商的支持

下面使用 WebStorm 工具创建两个文件，一个是 HTML4，另一个是 HTML5，如图 1.1 所示。

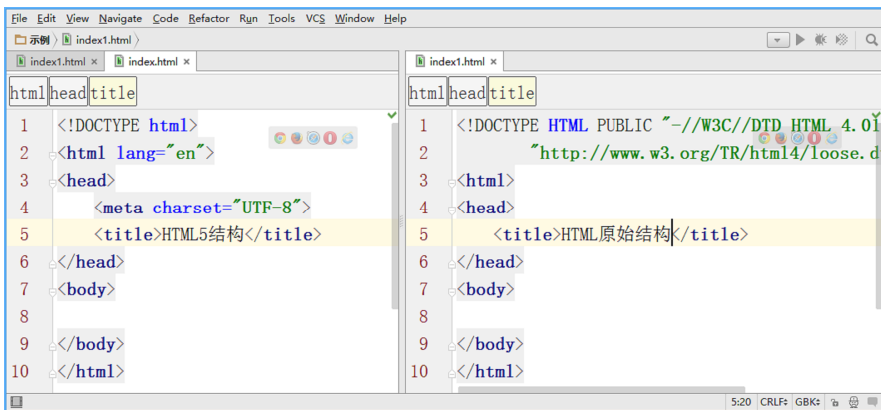


图 1.1 HTML4 和 HTML5 的对比

通过图 1.1 中的对比，可以看到 HTML5 的结构要简洁得多。其实其他的方面 HTML5 也有很多变化，这在后续的学习中会一一讲解到。

## 2 HTML5 新增元素

### 2.1 HTML5 结构元素

研究过 HTML 的人都知道，一个页面是由许多元素按一定的顺序组合实现的，但是在早先的 HTML 版本中并没有有实际意义的元素。一般是通过 div、table 等元素来确定布局和功能，并通过一些如 class、id 等选择器来表示其实际意义。这样对于页面结构的理解以及在搜索引擎优化上都有或多或少的缺陷，为了解决这些问题，HTML5 在 HTML4 的基础上进行了大量的修改，下面我们就学习一下 HTML5 新增的功能。

在具体学习之前先分析一下当当图书分类页面结构，如图 1.2 所示。

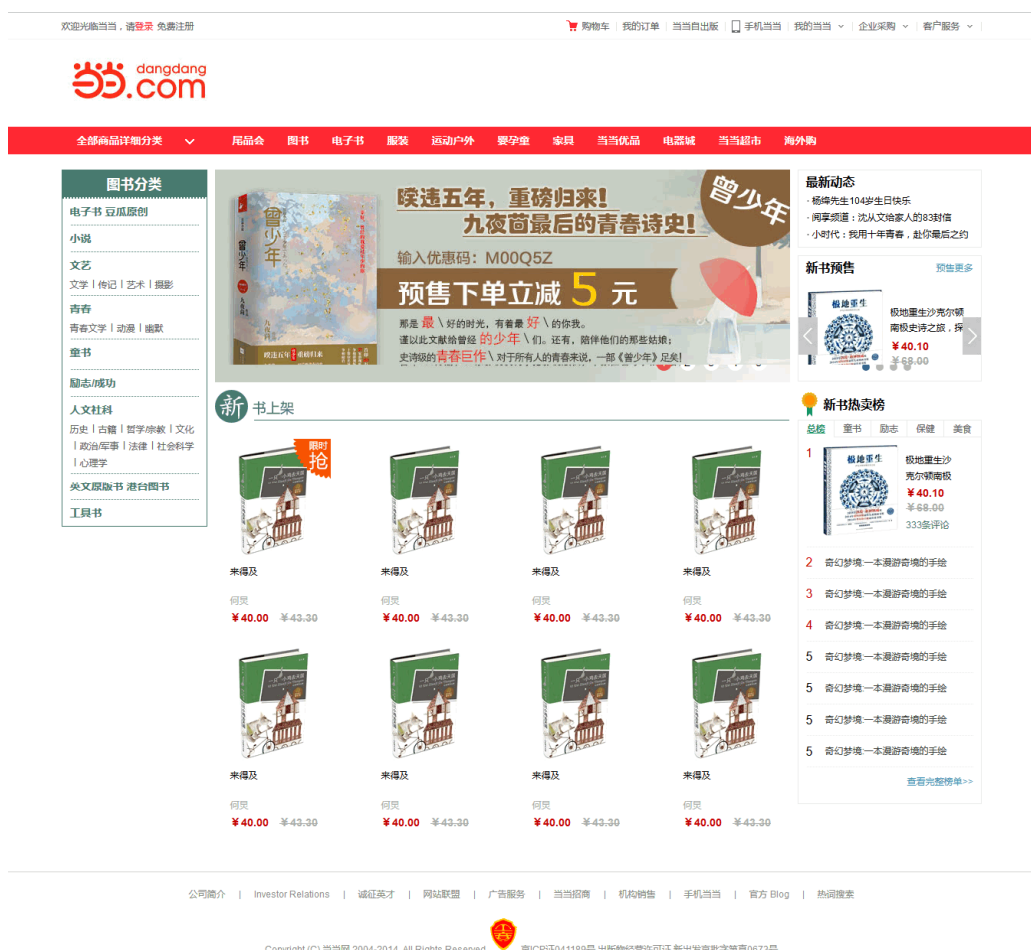


图 1.2 当当图书分类页面

由图 1.2 可以看出，该页面是由上中下三部分组成，其中中间部分由左中右的结构组成。如果用 HTML4 实现，需要使用如示例 1 所示的代码。

## ➔ 示例 1

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>HTML5 页面结构</title>
</head>
<body>
<!-- 页面容器 -->
  <div id="head">网页头部</div>
  <div id="container">
    <div>左边栏</div>
    <div>中间主题部分</div>
    <div>右边栏</div>
  </div>
  <div id="footer">网页底部</div>
</body>
</html>
```

上述代码没有任何错误，在任何浏览器和环境都能正确运行，并且能得到想要的结果，而且绝大部分的开发人员也是这样设计的。但是对于浏览器来讲，所知道的只有一系列的 `div`，具体做什么，在哪里显示，都是通过 ID 来获取的。如果开发者不同，ID 的命名也不同，这就会导致 HTML 代码的可读性很不好。

使用 HTML5 新增的结构元素可以很好地定位标记，明确某标记在页面中的位置和作用，如示例 2 所示。

## ➔ 示例 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>HTML 新增元素</title>
</head>
<body>
<!-- 页面容器 -->
  <header>网页头部</header>
  <article>
    <aside>左边栏</aside>
    <section>中间主体部分</section>
    <aside>右边栏</aside>
  </article>
  <footer>网页底部</footer>
</body>
</html>
```

通过示例 1 和示例 2 代码可以看到，两个示例代码虽然不一样，但是在浏览器里显示的效果是一样的，可以看出使用 HTML5 新增元素创建的页面代码更加简洁和高效，而且更容易被搜索引擎搜索到。

一个普通的页面，会有头部、导航、文章内容，还有附着的左右边栏，以及底部等模块，可通过 id 或 class 进行区分，并通过不同的 CSS 样式来处理。但相对于 HTML 来说，id 和 class 不是通用的标准的规范，搜索引擎只能去猜测某部分的功能。

HTML5 新定义了一组语义化的元素，虽然这些元素可以用传统的 HTML 元素（如：div、p、span 等）来代替，但是它们可以简化 HTML 页面的设计，无需再大量使用 id 或 class 选择器，而且在搜索引擎搜索的时候也会用到这些元素，在目前的主流浏览器中已经可以使用这些元素了。新增加的结构元素如表 1.1 所示。

表 1.1 HTML5 新增加的结构元素

元素	说明
header	页面或页面中某一区域的页眉，通常是一些引导和导航信息
nav	代表页面的一部分，可以作为页面导航的链接组
section	页面中的一个内容区块，通常由内容及其标题组成
article	代表一个独立的、完整的相关内容块，可独立于页面其他内容使用
aside	非正文的内容与页面的主要内容是分开的，被删除而不会影响到网页的内容
footer	页面或页面某一区块的脚注



HTML5 的设计是以效率优先为原则，要求样式和内容分离，因此在 HTML5 的实际开发中，必须使用 CSS 来定义样式。

下面就一一介绍 HTML5 新增的结构元素。如图 1.3 所示，该网站是典型的 HTML5 结构的网站。该网站的网址是：<http://html5games.com/>。

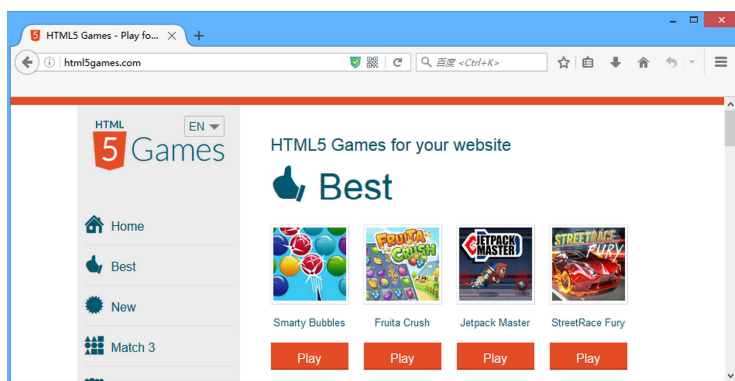


图 1.3 典型 HTML5 网站

下面根据这个网站分别讲解一下主要的几个结构元素。

### 1. header 元素

在 HTML5 之前习惯使用 div 元素布局网页，HTML5 在 div 元素基础上新增 header 元素，

也叫<header>头部元素。以前开发人员在设计 HTML 布局时常常把网页大致分为头部、内容、底部,使用<div>里加 id 进行布局。头部一般使用<div id="header"></div>或<div class="header"></div>进行布局。

在此基础上 HTML5 进行了修改,并没有减少元素,而是新增了元素。在大家公认的 HTML 布局中以 header 为常用命名,所以在 HTML5 中新增了 header 元素。除了直接使用 header 元素外,也可以对 header 设置 class 或 id 属性。

```
<header>
  <h1>网站标题</h1>
  <h1>网站副标题</h1>
</header>
```

因为 header 元素是 HTML5 新增元素,所以旧版本浏览器均不支持,需要 IE9+、最新 Chrome 等浏览器支持。

图 1.4 展示了 header 中添加的内容以及显示的位置。

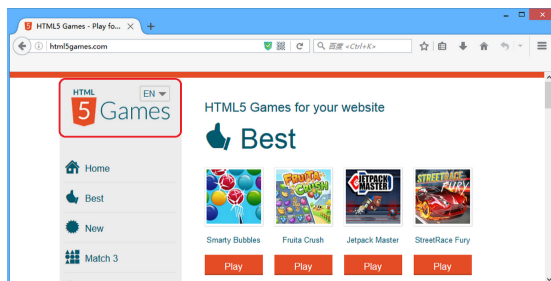


图 1.4 header 效果

## 2. nav 元素

nav 元素代表页面的一部分,是一个可以作为页面导航的链接组。其中的导航元素链接到其他页面或者当前页面的其他部分,使 HTML 代码在语义化方面更加精确,同时对于屏幕阅读器等设备的支持也更好。

```
<nav>
  <ul>
    <li>博客首页</li>
    <li>心情日志</li>
    <li>我的邮箱</li>
  </ul>
</nav>
```

nav 元素是与导航相关的,所以一般用于网站导航布局,完全就像使用 div 元素、span 元素一样来使用。而 nav 元素与 div 元素又有不同之处,此元素一般只用于导航相关地方,所以在在一个 HTML 网页布局中 nav 元素可能就使用在导航条处,或与导航条相关的地方,它通常与 ul、li 元素配合使用。图 1.5 显示了 nav 元素在页面中的应用效果。

## 3. section 元素

section 元素不只是一个普通的容器元素,主要用于表示一段

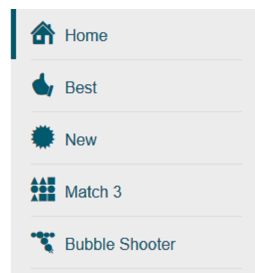


图 1.5 nav 效果

专题性的内容，通常用于带有标题和内容的区域，如文章的章节、元素对话框中的元素页，或者论文中有编号的部分。

一般来说，当一个元素只是为了样式化或者方便脚本使用时，应该使用 `div` 元素，当元素内容明确地出现在文档大纲中时，应该使用 `section` 元素。

```
<section>
  <h1>中华人民共和国</h1>
  <p>中华人民共和国建立于 1949 年……</p>
</section>
```

图 1.6 演示了 `section` 元素在页面中的应用效果。

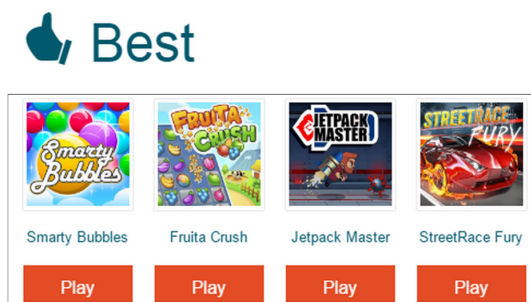


图 1.6 section 效果

#### 4. article 元素

`article` 元素是一个特殊的 `section` 元素，它比 `section` 元素具有更明确的语义。它代表一个独立的、完整的相关内容块。通常 `article` 元素会有标题部分（包含在 `header` 内），有时也会包含 `footer` 元素。虽然 `section` 元素也是带有主题性的一块内容，但是无论从结构上还是内容上来说，其独立性和完整性都没有 `article` 强。

```
<article>
  <h1>Internet Explorer 9</h1>
  <p>Windows Internet Explorer 9（简称 IE9）于 2011 年 3 月 14 日发布……</p>
</article>
```

#### 5. aside 元素

`aside` 元素在网站制作中主要有以下两种使用方法：

(1) 被包含在 `article` 元素中作为主要内容的附属信息部分，其中的内容可以是与当前文章有关的相关资料、名词解释等。

```
<article>
  <h1>...</h1>
  <p>...</p>
  <aside>...</aside>
</article>
```

(2) 在 `article` 元素之外使用，作为页面或站点全局的附属信息部分。最典型的是侧边栏。

```
<aside>
  <h2>...</h2>
  <ul>
    <li>...</li>
```

```

        <li>...</li>
    </ul>
    <h2>...</h2>
    <ul>
        <li>...</li>
        <li>...</li>
    </ul>
</aside>

```

## 6. footer 元素

footer 元素一般用于页面或区域的底部，通常包含文档的作者、版权信息、使用条款链接等。如图 1.7 所示，页面底部可以使用 footer 元素来布局。

```

<footer class="yanshi">
    脚注信息
</footer>

```

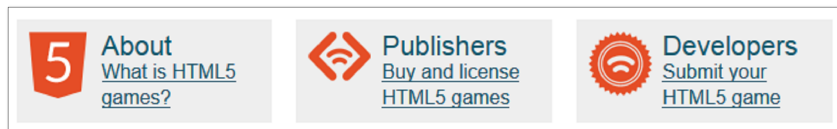


图 1.7 页面底部位置的 footer 元素



**注意**

HTML5 新增的结构元素都是块元素，独占一行，使用的时候要引起注意。

## 操作案例 1：HTML5 结构元素的应用

### 需求描述

利用 HTML5 的结构元素，完成图 1.8 所示的效果。

### 完成效果

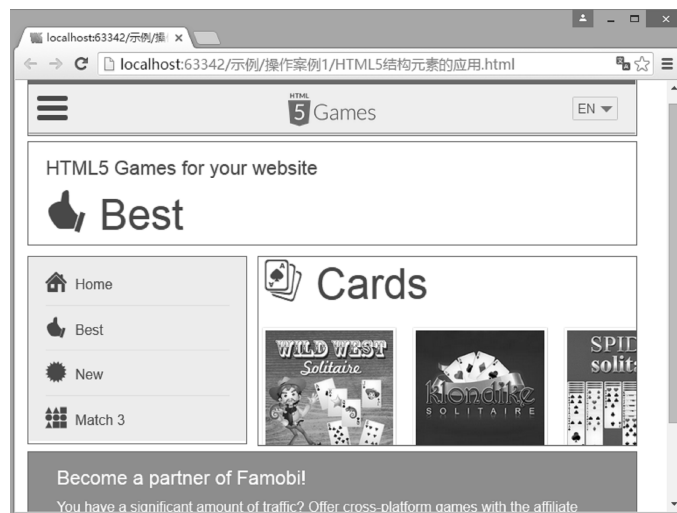


图 1.8 完成效果图



### 技能要点

header、nav、section、article、aside、footer 等元素的使用。

### 关键代码

```
<div id="container">
  <header></header>
  <nav></nav>
  <section>
    <nav></nav>
    <article></article>
  </section>
  <footer>
    
  </footer>
</div>
```

## 2.2 HTML5 网页元素

上一节所讲的是 HTML5 结构元素，结构元素只用于设计网页的结构，编写框架。一个完整的网页不能只有框架，还必须有具体的网页内容，这里就需要网页元素对 HTML 的支持了，HTML5 新增加了如表 1.2 所示的网页元素。

表 1.2 网页元素

元素	说明
video	定义视频，如电影片段或其他视频流
audio	定义音频，如音乐或其他音频流
canvas	定义图形
datalist	定义可选数据的列表
time	元素定义日期或时间
mark	在视觉上向用户呈现哪些需要突出的文字
progress	运行中的进度（进程）

这几个元素主要完成 Web 页面具体内容的引用和表述，是丰富内容展示的基础。下面分别讲解一下这几个新增元素。

### 1. datalist 元素

datalist 元素用于为文本框提供一个可选数据的列表，用户可以直接选择列表中预先添加的某一项，从而免去输入的麻烦。如果用户不需要选择或者选项里不存在用户需要的内容，也可以自行输入其他内容。在实际应用中，如果把 datalist 提供的列表绑定到某文本框，则需要使用文本框的 list 属性来引用 datalist 元素的 id 属性。

datalist 元素是由一个或多个 option 元素组成的，而且每一个 option 元素都必须设置 value 属性。

### ➔ 示例 3

```
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>datalist 元素的用法</title>
  </head>
  <body>
    <input type="text" list="list1"/>
    <datalist id="list1">
      <option value="苹果">苹果</option>
      <option value="香蕉">香蕉</option>
      <option value="菠萝">菠萝</option>
    </datalist>
  </body>
</html>
```

页面效果如图 1.9 所示。

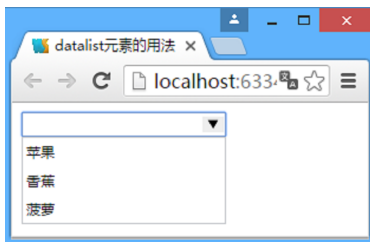


图 1.9 datalist 页面效果

#### 2. time 元素

在网页中使用 time 元素和不使用 time 元素效果没有什么区别，但是使用 time 元素容易被搜索引擎搜索到。

```
<p>我们在每天早上 <time>9:00</time> 开始营业。</p>
```

```
<p>我在 <time datetime="2008-02-14">情人节</time> 有个约会。</p>
```

#### 3. mark 元素

当把一行文字包含在 mark 元素之内时，页面上显示当前文字有背景，用于重点突出。

```
<mark>天气渐渐变暖了</mark>
```

#### 4. progress 元素

progress 元素在页面上显示为一个进度条。value 属性表示当前已完成的进度，max 属性表示总进度。

```
<progress value="20" max="100"></progress>
```

其他的几个元素如 video、audio、canvas 功能比较多，操作起来比较复杂。这几个元素将在后面的章节详细讲述，在此不做赘述。

### 操作案例 2: HTML5 网页元素的应用

#### 需求描述

利用 HTML5 新增加的网页元素，完成图 1.10 所示的效果。

## 完成效果

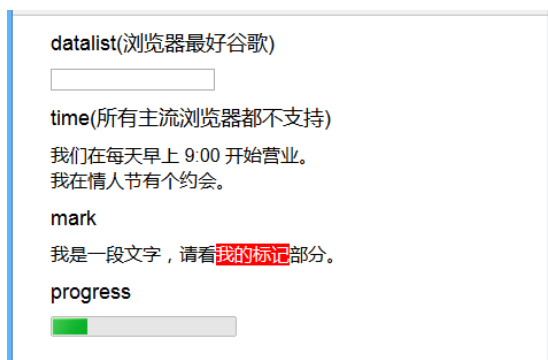


图 1.10 完成效果图

## 技能要点

HTML5 新增的网页元素的使用。

## 素材准备

登录课工场 <http://www.kgc.cn/>，下载素材。

## 2.3 HTML5 废除的元素

HTML5 增加元素的同时还废除了一些元素。由于浏览器支持不好以及被代替等一些原因，这些被废除的元素平时基本上用得很少。

废除的元素主要有：

- 能用 CSS 替代的元素：big、center、font、strike 等。
- frame 框架。
- 只有部分浏览器支持的元素：applet、blink、marquee 等。
- 其他被废除的元素：rb、dir、isindex、listing 等。

## 2.4 HTML5 全局属性

HTML5 新增了全局属性的概念，所谓全局属性是指可以对任何元素都使用的属性。新增的全局属性见表 1.3。

表 1.3 新增的全局属性

属性	说明
contentEditable	规定是否允许用户编辑内容
designMode	规定整个页面是否可编辑
hidden	规定对元素进行隐藏
spellcheck	规定是否必须对元素进行拼写或语法检查
tabindex	规定元素的 Tab 键移动顺序

下面依次对这几个属性进行讲解。

### 1. contentEditable

`contentEditable` 属性的主要功能是允许用户在线编辑元素中的内容。`contentEditable` 属性可以设定两个值：`true` 和 `false`。当设置值为 `true` 的时候，页面元素允许被编辑，如果为 `false`，页面元素不能被编辑，如果未指定 `true` 或者 `false`，该元素的编辑状态由父元素来决定。下面的代码段中 `ul` 元素在页面上是可以被编辑的。

```
<ul contentEditable="true">
  <li>列表一</li>
  <li>列表二</li>
</ul>
```

在编辑完元素内容后，如果想要保存这些内容，只能把该元素的 `innerHTML` 发送到服务器端进行保存，因为改变元素内容后该元素的 `innerHTML` 内容也会随之改变。

### 2. designMode

`designMode` 属性用来指定整个页面是否可编辑，当页面可编辑的时候，页面中任何支持 `contentEditable` 属性的元素都变成了可编辑状态。`designMode` 属性只能在 JavaScript 脚本里被修改，该属性有两个值：`on` 和 `off`。当值为 `on` 的时候页面可编辑，为 `off` 的时候页面不可编辑，使用 JavaScript 来指定属性的用法如下所示。

```
<script>
  document.designMode=on;
</script>
```

通常整个页面是不能被修改的，所以该属性使用的并不是十分广泛，此处不再示例。

### 3. hidden

在 HTML5 中所有的元素都允许使用一个 `hidden` 属性，该属性类似于 `input` 元素中的 `hidden`，功能是使该元素处于不可见状态，`hidden` 属性也是一个 `bool` 值，设为 `false` 元素可见，设为 `true` 元素不可见。

### 4. spellcheck

`spellcheck` 属性是 HTML5 中针对单行文本框和多行文本框的，它的功能是对用户输入的文本内容进行拼写和语法检查，该属性也是一个 `bool` 类型，设为 `true` 进行语法检查，否则不检查，但是如果元素的 `readOnly` 属性和 `disabled` 属性生效的话，`spellcheck` 属性失效。

### 5. tabindex

`tabindex` 是开发中的一个基本概念，当不断按 `Tab` 键让窗口或页面中的控件获取焦点，对窗口或页面中的所有控件进行遍历的时候，每一个控件的 `tabindex` 属性表示该控件是第几个被访问到的。

## 2.5 HTML5 废除的属性

对于 HTML5 废除了一些属性，读者可作为了解部分。

- `table` 部分属性：`align`、`bgcolor`、`border`、`cellpadding` 等。
- `html` 的 `version` 属性。
- `a` 元素或 `link` 元素的 `charset` 属性。
- `br` 的 `clear` 属性，`img` 的 `align` 属性。

### 3 CSS3 高级选择器

CSS 是网页设计师可用的最强大的工具之一。使用它开发人员可以在几分钟内改变一个网站的页面，而不用改变页面的元素。但绝大部分程序员使用的 CSS 选择器远未发挥它们的潜力，有的时候还趋向于使用过多的和无用的 class、id、div、span 等，把 HTML 写得很凌乱。避免写这些凌乱的属性和元素且保持代码简洁和语义化的最佳方式，就是使用更复杂的 CSS 选择器，它们可以定位于指定的元素而不用使用额外的 class 或 id，而且通过这种方式也可以让我们的代码和样式更加灵活。

下面先简单讲解几个 CSS3 的高级选择器，如表 1.4 所示。

表 1.4 CSS3 高级选择器

选择器	说明
:first-of-type	p:first-of-type: 选择属于其父元素的首个 p 元素的每个 p 元素
:last-of-type	p:last-of-type: 选择属于其父元素的最后 p 元素的每个 p 元素
:last-child	p:last-child: 选择属于其父元素的最后一个子元素的每个 p 元素
:nth-child(n)	p:nth-child(n): 选择属于其父元素的第 n 个子元素的每个 p 元素
:before	p:before: 在每个 p 元素的内容之前插入内容
:after	p:after: 在每个 p 元素的内容之后插入内容

为了更好的演示效果，编写统一的结构代码如下：

#### ➔ 示例 4

```
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>CSS 高级选择器应用</title>
    <style></style>
  </head>
  <body>
    <p>外部 p 元素</p>
    <header>
      <p>header 里面的第一个 p 元素</p>
      <p>header 里面的第二个 p 元素</p>
      <p>header 里面的第三个 p 元素</p>
    </header>
    <div>
      <p>div 里面的第一个 p 元素</p>
      <p>div 里面的第二个 p 元素</p>
      <p>div 里面的第三个 p 元素</p>
    </div>
  </body>
</html>
```

上述代码仅仅是本小节涉及的结构，其他结构读者可以自行添加。

### 1. :first-of-type

:first-of-type 表示获取某元素内部指定类型的第一个子元素。通常:first-of-type 需要其他的一些限制，如以下代码所示：

```
p: first-of-type {
    background-color: #cccccc;
}
```

运行效果如图 1.11 所示。

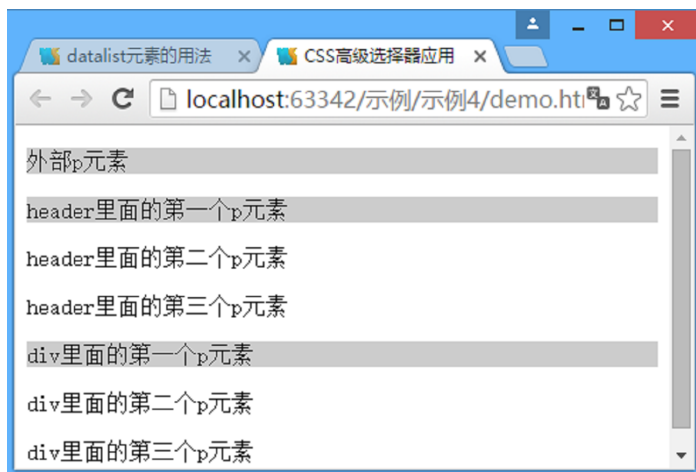


图 1.11 :first-of-type 无限制使用

通过示例 4 代码和图 1.11 可以看出，`p:first-of-type` 只是获取第一个子元素是 `p` 元素的元素，在该示例中能够获取 `body` 的 `p` 元素，也能获取 `div` 的第一个子元素 `p`，同时还能获取 `header` 中的第一个 `p` 元素。如果只想获取 `header` 中的第一个子元素可以将 CSS 代码改成如下代码：

```
header p:first-of-type {
    background-color: #cccccc;
}
```

:last-of-type 的用法和:first-of-type 相同，只不过检索的是最后一个子元素。



### 注意

在进行子元素查找的时候，要注意元素的包含关系，即子元素的子元素，如果其类型满足要求也能被检索到。

### 2. :first-child

:first-child 表示获取属于其父元素的第一个子元素，效果类似于:first-of-type。:last-child 的用法和:first-child 的用法相同，:last-child 表示获取最后一个元素。

```
p:first-child {
    background-color: #cccccc;
}
```

:first-child 和:first-of-type 的区别如下所示。

:first-child 选择器是 CSS2 中定义的选择器，就是指获取第一个子元素，如以下代码所示：

## ➔ 示例 5

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>:first-child 和:first-of-type 的区别</title>
  <style>
    p:first-child, h1:first-child, span:first-child {
      background-color: yellow;
    }
    p:first-of-type, h1:first-of-type, span:first-of-type {
      font-style: italic;
    }
  </style>
</head>
<body>
<div>
  <p>div 里面的第一个元素</p>
  <h1>div 里面的第二个元素</h1>
  <span>div 里面的第三个 p 元素</span>
  <span>div 里面的第四个 p 元素</span>
</div>
</body>
</html>
```

通过示例 5 可以看出：

- **p:first-child**：匹配到的是 p 元素，因为 p 元素是 div 的第一个子元素。
- **h1:first-child**：匹配不到任何元素，因为在这里 h1 是 div 的第二个子元素，而不是第一个。
- **span:first-child**：匹配不到任何元素，因为在这里两个 span 元素都不是 div 的第一个子元素。

再使用 CSS3 中定义的 **:first-of-type** 选择器，还是同样的代码。

- **p:first-of-type**：匹配到的是 p 元素，因为 p 是 div 的所有类型为 p 的子元素中的第一个。
- **h1:first-of-type**：匹配到的是 h1 元素，因为 h1 是 div 的所有类型为 h1 的子元素中的第一个。
- **span:first-of-type**：匹配到的是第三个子元素 span，这里 div 有两个为 span 的子元素，匹配到的是它们中的第一个。

示例 5 的运行效果如图 1.12 所示。

通过以上内容可以得出结论：**:first-child** 匹配的是某父元素的第一个子元素，可以说是结构上的第一个子元素；**:first-of-type** 匹配的是某父元素下相同类型的子元素中的第一个，比如 **p:first-of-type** 就是指所有类型为 p 的子元素中的第一个。这里不再限制是第一个子元素了，只要是该类型元素的第一个就行了。



图 1.12 样式区别显示效果

同样类型的选择器:`:last-child`和`:last-of-type`、`:nth-child(n)`和`:nth-of-type(n)`也可以这样去理解。

### 3. `:nth-child(n)`

`:nth-child(n)` 选择器匹配属于其父元素的第  $n$  个子元素，不限制元素的类型。 $n$  可以是数字、关键词或公式。可以通过`:nth-child(n)`对下面的代码片段操作：

```
<div>
  <p>div 里面的第一个 p 元素</p>
  <p>div 里面的第二个 p 元素</p>
  <p>div 里面的第三个 p 元素</p>
</div>
```

当  $n$  是数字时，获取 `div` 中的第  $n$  位置的元素（注意  $n$  从 1 开始）。

```
div p:nth-child(1) {
  background-color: #cccccc;
}
```

该代码段的作用是获取 `div` 中第一个 `p` 元素，给这个 `p` 元素设置背景色。 $n$  除了是数字之外，还可以是表达式，如  $an+b$ ，此时  $n$  从零开始。如下代码表示给 `div` 中所有奇数位置处的 `p` 元素加背景色：

```
div p:nth-child(2n+1) {
  background-color: #cccccc;
}
```

另外还可以使用 `odd` 和 `even` 关键字，`odd` 和 `even` 是可用于匹配下标是奇数或偶数的子元素的关键词（第一个子元素的下标是 1）。下面的 CSS 代码分别为 `div` 中奇数 `p` 元素和偶数 `p` 元素的背景指定两种不同的背景色：

```
div p:nth-child(odd) {
  background-color: #cccccc;
}
div p:nth-child(even) {
  background-color: #46face;
}
```

### 4. `:before`

`:before` 选择器在被选元素的内容前面通过使用 `content` 属性来插入内容。

```
div p:before {
  content: "新增内容";
}
```



此段代码表示将 `div` 中所有的 `p` 元素的内容前面都添加文本“新增内容”，`:after` 和 `:before` 的用法相同，只不过表示添加在内容之后。

示例 6 演示了 `:nth-child(n)` 和 `:before` 的用法。

### ➔ 示例 6

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS 高级选择器</title>
  <style>
    p:nth-child(1) {
      color: white;
    }
    p:nth-child(2n+1) {
      background-color: pink;
    }
    p:nth-child(odd) {
      font-size: 2em;
    }
    p:first-child:before {
      content: "新增内容";
    }
  </style>
</head>
<body>
  <p>div 里面的第一个 p 元素</p>
  <p>div 里面的第二个 p 元素</p>
  <p>div 里面的第三个 p 元素</p>
  <p>div 里面的第四个 p 元素</p>
</body>
</html>
```

示例 6 的效果如图 1.13 所示。



图 1.13 CSS 高级选择器

CSS 的高级选择器对浏览器的支持有一定要求，Internet Explorer 9.0 以下都不支持。Firefox、Safari 和 Opera 目前通用的版本都支持这些选择器。如果说网页的用户大部分用的还是 IE8 以下的版本，建议还是使用原始的 class 和 id 选择器。但不可否认，CSS3 的高级选择器越来越被浏览器支持，随着新版本的浏览器越来越普及，这些高级选择器在开发中所发挥的作用会越来越大。

### 操作案例 3: CSS3 高级选择器

#### 需求描述

利用 CSS3 高级选择器，完善 QQ 会员页面的登录部分和底部导航部分。将“登录”按钮和“开通超级会员”按钮通过 CSS 编写成如图 1.14 所示的效果，当鼠标移到“登录”按钮上时“登录”按钮背景变成黄色，当鼠标移出时背景还原。

鼠标移入底部导航如“官方 QQ 账号”“官方微信账号”等链接上时，背景发生变化，变成黄色，鼠标移出时，背景还原，如图 1.15 所示。

#### 完成效果



图 1.14 QQ 会员登录



图 1.15 QQ 会员底部导航

#### 技能要点

使用 HTML5 结构元素布局页面。

使用 CSS3 高级选择器美化页面。

- :first-of-type
- :last-of-type
- :last-child
- :nth-child(n)

## 关键代码

“登录”按钮和“开通超级会员”按钮的结构如下：

```
<div class="top-right">
  <a href="" >登录</a>
  <a href="" >开通超级会员</a>
</div>
```

首先，将两个 a 元素变为圆角，同时还需要设置宽和高，因此需要将 a 元素设置为块元素，鼠标移出背景发生变化，部分代码如下：

```
.top-right a {
  display:inline-block;/*设置为行内块元素*/
  font-size:16px;
  text-align:center;
  margin-top:25px;
  border-radius:35px;/*设置为圆角*/
}
/*当鼠标移入时第一个 a 元素样式改变*/
.top-right a:first-of-type:hover {
  color:#986b0d;
  background:#fad65c;
}
```

其次，页面底部的导航背景是通过改变背景图片来实现的。

```
footer li span:first-of-type {
  width:100px;
  height:85px;
  /*获取大背景图片*/
  background:url(..img/sprites_footer.png) no-repeat;
}
footer li:nth-child(1) span:first-of-type {
  background-position:0 0; /*根据大背景图片的位置显示小背景*/
}
footer li:nth-child(1) span:first-of-type:hover {
  background-position:-100px 0; /*移动大背景图片的位置改变小背景*/
}
```

这里有几点要注意：

- a 元素本身是行内元素，不能设置宽和高，需要通过 display:inline-block 属性将其变为可放置在一行内的块元素。
- 在 CSS 中鼠标移入移出改变元素的样式，可以通过: hover 伪类来实现。
- 当改变元素背景图片的时候可以用多张图片来实现，但这样做会影响效率，通常的做法是用一张大图片，里面有很多小背景图片，通过定位大图片来改变元素的背景。

## 实现步骤

- (1) 打开素材网页，分析页面结构。
- (2) 通过 CSS3 高级选择器查找需要处理的元素，不要添加 class 和 id。
- (3) 按照需求编写 CSS 代码。

## 素材准备

登录课工场 <http://www.kgc.cn/>，下载素材。

## 本章总结

- HTML5 实现了兼容性和跨平台性，目前各大浏览器厂商都支持。
- HTML5 新增了用于语义化结构的结构元素：`header`、`footer`、`section`、`article`、`aside` 等。
- HTML5 新增加网页元素 `video`、`audio`、`canvas`、`datalist`、`time`、`mark`、`progress` 用于丰富和优化 HTML 页面。
- 可以通过 HTML5 的全局属性对 HTML 元素进行进一步处理。
- CSS3 的高级选择器能够进一步优化页面，减少 `class` 选择器和 `id` 选择器的使用，使表现和结构进一步分离。

## 本章作业

1. 根据你的理解，简述 HTML5 的优势和特点。
2. 请说一下 `article` 元素和 `section` 元素的区别。
3. 结合 HTML5 结构元素和 CSS3 实现以下功能。
  - 使用 HTML5 的结构元素设计页面。
  - 使用 CSS3 高级选择器处理页面效果。
  - 博文的主体部分可编辑。
  - 效果如图 1.16 所示。



图 1.16 个人博客

4. 仔细分析作业 3，你认为作业 3 缺少什么，有什么需要修改的地方，该怎么实现？
5. 请登录课工场，按要求完成预习作业。