第2章 HTML5新特性

导言：

互联网技术的发展日新月异，HTML5和CSS3已经成为前端开发的主流技术，很多互联网公司和传统的公司都在使用这些技术开发或者迭代新的web系统。当前的主流浏览器也都全部兼容了这些技术，使得HTML5成为最炙手可热的前端开发技术。本章重点讲解了HTML5技术在实际开发中的应用，后续的课程会继续讲解CSS3技术。

2.1 HTML5 介绍

2.1.1 什么是HTML5

HTML5 技术结合了 HTML4.01 的相关标准并革新，符合现代网络发展要求，在 2008 年正式发布。HTML5 由不同的技术构成，其在互联网中得到了非常广泛的应用，提供更多增强网络应用的标准机。与传统的技术相比，HTML5 的语法特征更加明显，并且结合了 SVG 的内容。这些内容在网页中使用可以更加便捷地处理多媒体内容，而且HTML5中还结合了其他元素，对原有的功能进行调整和修改，进行标准化工作。HTML5 在 2012 年已形成了稳定的版本。（摘自百度百科）

2.1.2 HTML5的主要新功能

1. 语义标签

   在HTML5之前，网页使用div标签来实现页面布局，这些div标签只是提供给浏览器的指令，只是定义一个网页的某些部分，并没有任何实际意义。即便使用CSS样式的id和class形容这块内容也是没有意义的。在HTML5中，将那些之前没“意义”的div标签使用不用的语义关键字来代替，例如：<article>。这样就赋予了该标签相应的含义，这就是“语义标签”。

2. 增强型表单

   表单是实现用户与页面后台交互主要组成部分，HTML5在表单的设计上功能更加强大。input类型和属性的多样性大大地增强了HTML可表达的表单形式，再加上新增加的一些表单标签，使得原本需要JavaScript来实现的控件，可以直接使用HTML5的表单来实现；一些如内容提示、焦点处理、数据验证等功能，也可以通过HTML5的智能表单属性标签来完成。

3. 视频和音频

   HTML5最大特色之一就是支持音频视频，在通过增加了<audio>、<video>两个标签来实现对多媒体中的音频、视频使用的支持，只要在Web网页中嵌入这两个标签，而无需第三方插件（如Flash）就可以实现音视频的播放功能。HTML5对音频、视频文件的支持使得浏览器摆脱了对插件的依赖，加快了页面的加载速度，扩展了互联网多媒体技术的发展空间。

4. Canvas绘图

   HTML5的canvas元素可以实现画布功能，该元素通过自带的API结合使用JavaScript脚本语言在网页上绘制图形和处理，拥有实现绘制线条、弧线以及矩形，用样式和颜色填充区域，书写样式化文本，以及添加图像的方法，且使用JavaScript可以控制其每一个像素。HTML5的canvas元素使得浏览器无需Flash或Silverlight等插件就能直接显示图形或动画图像。

5. SVG绘图

可缩放向量图SVG(Scalable Vector Graphic)，在HTML5标准之前SVG是不能直接使用与网页中的。若是需要使用，只能将SVG编写在独立的XML文档中。在HTML5标准推出后，SVG标签可以直接使用<img>标签在网页中显示。使用<img src="x.svg">嵌入到网页中。

6. 地理定位

   现今移动网络备受青睐，用户对实时定位的应用越来，要求也越来越高。HTML5通过引入Geolocation的API可以通过GPS或网络信息实现用户的定位功能，定位更加准确、灵活。通过HTML5进行定位，除了可以定位自己的位置，还可以在他人对你开放信息的情况下获得他人的定位信息

7. 拖放API

   在HTML5之前，用户是无法把页面中的元素拖放到另一个位置的，而在HTML5当中，可以使用拖放API将一个元素使用“鼠标按下 + 鼠标移动”两个事件将选中的元素从一个位置拖到另一个位置。在 HTML5 中，拖放是标准的一部分，任何元素都能够拖放。

8. WebWorker

   HTML5利用Web Worker将Web应用程序从原来的单线程业界中解放出来，通过创建一个Web Worker对象就可以实现多线程操作。JavaScript创建的Web程序处理事务都是在单线程中执行，响应时间较长，而当JavaScript过于复杂时，还有可能出现死锁的局面。HTML5新增加了一个WebWorkerAPI，用户可以创建多个在后台的线程，将耗费较长时间的处理交给后台面不影响用户界面和响应速度，这些处理不会因用户交互而运行中断。

9. WebStorage

   HTML5较之传统的数据存储有自己的存储方式，允许在客户端实现较大规模的数据存储。为了满足不同的需求，HTML5支持DOM Storage和Web SQL Database 两种存储机制。其中，DOM Storage 适用于具有key/value对的基本本地存储；而WebSQLDatabase是适用于关系型数据库的存储方式，开发者可以使用SQL语法对这些数据进行查询、插入等操作。

10. WebSocket

    WebSocket 是 HTML5 开始提供的一种在单个 TCP 连接上进行全双工通讯的协议。WebSocket 使得客户端和服务器之间的数据交换变得更加简单，允许服务端主动向客户端推送数据。在 WebSocket API 中，浏览器和服务器只需要完成一次握手，两者之间就直接可以创建持久性的连接，并进行双向数据传输。在 WebSocket API 中，浏览器和服务器只需要做一个握手的动作，然后，浏览器和服务器之间就形成了一条快速通道。两者之间就直接可以数据互相传送。

2.1.3 HTML5的优势

从HTML4.0、XHTML到HTML5，从某种意义上讲，这是HTML描述性标记语言的一种更加规范的过程。因此，HTML5并没有给开发者带来多大的冲击。但HTML5增加了很多非常实用的新功能和新特性，下面具体介绍HTML5的一些优势。

1. 解决了跨浏览器问题

   在HTML5之前，各大浏览器厂商为了争夺市场占有率，会在各自的浏览器中增加各种各样的功能，并且不具有统一的标准。使用不同的浏览器，常常看到不同的页面效果。在HTML5中，纳入了所有合理的扩展功能，具备良好的跨平台性能。针对不支持新标签的老式IE浏览器，只需简单地添加JavaScript代码就可以使用新的元素。

2. 新增了多个新特性

   HTML语言从1.0到5.0经历了巨大的变化，从单一的文本显示功能到图文并茂的多媒体显示功能，许多特性经过多年的完善，已经发展成为一种非常重要的标记语言。HTML5新增的特性如下。

- 新的语义标签，比如header、nav、section、article、footer。
- 新的表单元素，比如calendar、date、time、email、url、search。
- 用于绘画的canvas元素。
- 用于媒介回放的video和audio元素。
- 对本地离线存储的更好支持。
- 地理位置、拖曳、摄像头等API。

3. 用户优先的原则

   HTML5标准的制定是以用户优先为原则的，一旦遇到无法解决的冲突时，规范会把用户放在第一位。另外，为了增强HTML5的使用体验，还加强了以下两方面的设计。

   - 安全机制的设计

   为确保HTML5的安全，在设计HTML5时做了很多针对安全的设计。HTML5引入了一种新的基于来源的安全模型，该模型不仅易用，而且对不同的API(Application Programming Interface，应用程序编程接口)都通用。使用这个安全模型，不需要借助于任何不安全的hack就能跨域进行安全对话。

   - 表现和内容分离

   表现和内容分离是HTML5设计中的另一个重要内容。实际上，表现和内容的分离早在HTML4.0中就有设计，但是分离的并不彻底。为了避免可访问性差、代码高复杂度、文件过大等问题，HTML5规范中更细致、清晰地分离了表现和内容。但是考虑到HTML5的兼容性问题，一些陈旧的表现和内容的代码还是可以兼容使用的。

4. 化繁为简的优势

   作为当下流行的通用标记语言，HTML5尽可能地简化，严格遵循了“简单至上”的原则，主要体现在这几个方面：

   - 新的简化的字符集声明;
   - 新的简化的DOCTYPE;
   - 简单而强大的HTML5 API;
   - 以浏览器原生能力替代复杂的JavaScript代码。

2.2 HTML5 网页标准结构

前一章节已经讲过HTML的基础标签，相信大家对HTML的基础标签已经比较熟悉了。随着HTML的不段发展，其经历了“从HTML4的宽松到XHTML的严格再到HTML5宽松”的发展路程。本章前文也介绍了HTML5标签的一些特点和优势，从这些特点和优势可以看出HTML5继承了HTML4的优点并且对其进行了补充和拓展，本章将重点介绍HTML5和HTML4之前的区别，相同部分不再赘述。

2.2.1 结构标签

相比较HTML4和XHTML的结构标签而言，HTML5对结构标签进行了较大简化，精简了声明部分标签的定义过程，使得结构标签更加的简洁。下面先看一下HTML4、XHTML和HTML5结构标签之间的区别。

HTML4：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
        "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <title>Title</title>
    <meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
</head>
<body>
  内容...
</body>
</html>
```

XHML：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
        PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>Title</title>
    <meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
</head>
<body>
  内容...
</body>
</html>
```

HTML5：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
  内容...
</body>
</html>
```

通过对比观察可以发现结构标签中最大的变化在文档声明（DOCTYPE）和字符编码设置（meta）这两个部分。下面详细介绍一下HTML5结构标签各标签的具体功能。

<!DOCTYPE html>

HTML5文档必须以DOCTYPE作为该文档的第一行，这是一个文档类型声明。它用来告诉浏览器或其他分析程序他们所访问的文件类型。

html标签

html标签是HTML5文档的根标签，必须为文档的第二行标签。在HTML5中新增了一个html标签的属性为：manifest。manifest属性的值会定义一个 URL，在这个 URL 上描述了HTML5文档的缓存信息，这些信息会在离线应用的时候使用。

head标签

head标签是所有头部元素的容器。根据前面所学内容，在head标签内部包含了与当前文档相关的声明信息。例如：标题、样式表、脚本、元信息等。

meta标签

meta标签位于head标签内，主要用于定义与当前文档相关联的元信息（meta-information），这些信息用于当前页面的设置工作。例如设置页面编码；设置页面根路径；设置针对搜索引擎的描述和关键字等。

<meta charset="UTF-8">定义了当前文档的字符编码是UTF-8。这个设置是页面必须的设置，否则会出现页面乱码的情况。根据对比大家也会发现这种设置方式较以前的设置更加的简洁。

title标签

title标签位于head标签内，定义了文档的标题。该标签定义的标题会在浏览的工具栏或者标签中显示，同时也是页面收藏时的关键索引，所以非常重要。在以前版本的HTML中title是必须存在的，但是在HTML5中只规定了该标签的定义上限，虽然这意味着不定义title标签页面并不会出现错误，但是因为title标签的重要性，要求大家必须描述此标签。

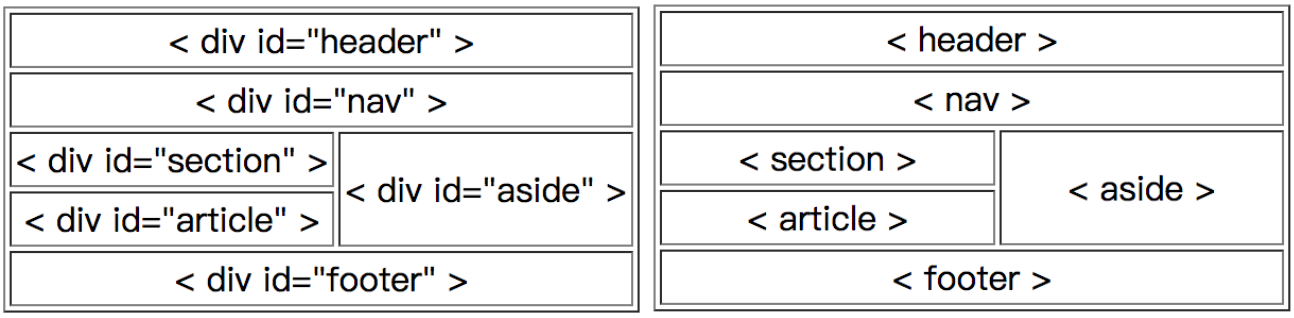
body标签

body标签定义了文档的主体和所有的内容，如文本、图片、超链接、表单、表格等所有的用于显示的标签都必须包含在该标签中。该标签中的内容会在浏览器窗体中被显示出来。

2.2.2 语义化标签

所谓 语义化，是指用恰当的标签去展示恰当的内容，例如 HTML5 中新增的 header、nav、footer 等标签，而非什么内容都用 div 、span这种不具备语义化的标签来布局显示。

不使用语义标签和使用语义标签的对比如下图。



语义标签的优点：

- 在样式丢失的情况下，页面也能呈现出很好地内容结构、代码结构
- 比 标签有更加丰富的含义，方便开发与维护
- 有利于 SEO，提高搜索引擎的有效爬取
- 方便其他设备解析（如移动设备）

下面按照上图描述的顺序介绍一下常见的语义化标签。

2.2.2.1 页眉（header）

header标签是一种具有引导和导航作用的构架元素，通常用来放置整个页面或者页面中一个内容块的标题，若是作为整个页面的标题，应该放在页面的起始位置。示例代码：

```
<body>
    <header>
        <h1>我是整个页面的标题</h1>
    </header>
    <article>
        <header>我是文章的标题</header>
        <div>文章的正文...</div>
    </article>
</body>
```

header中也可以包含table、form、nav等标签，只要应该在头部区域显示的内容都可以放到header中。

2.2.2.2 导航（nav）

nav标签是一个可以用作页面导航的链接组。其中的导航元素可以将页面导航至其他关联页。需要注意的是并不是需要将一个页面的所有超链接都放到nav标签中，只放具有导航意义的超链接即可。

例如网站首页都会有整个网站的功能导航部分，此部分就需要使用nav标签来实现。示例代码：

```
<body>
    <nav>
        <ul>
            <li><a>首页</a></li>
            <li><a>手机</a></li>
            <li><a>电脑</a></li>
            <li><a>pad</a></li>
            <li><a>其他</a></li>
        </ul>
    </nav>
</body>
```

> 注意：在HTML5中最好不要使用menu标签来替代nav标签实现导航功能，这种操作是错误的，因为menu标签主要用于交互命令的菜单上。

2.2.2.3 分区（section）

section标签用于对页面上的内容进行分区，可以通过其将页面的内容分成多个部分。每个部分之间又有一定的联系。一个section标签通常由标题和内容组成。一般不会出现没有标题的section。示例代码：

```
<div>
    <h1>排行榜</h1>
    <section>
        <h3>最热排行</h3>
        <ol>
```

```
            <li>歌曲1</li>
            <li>歌曲2</li>
            <li>...</li>
        </ol>
    </section>
    <section>
        <h3>新歌排行</h3>
        <ol>
            <li>歌曲1</li>
            <li>歌曲2</li>
            <li>...</li>
        </ol>
    </section>
</div>
```

> 注意：div标签关注结构独立性，section关注内容独立性。

2.2.2.4 文章（article）

article标签用来表示页面中的独立的、完整的、可以独自被外部引用的内容。它可以是一篇文章、一段评论、或者一段描述。一个article通常都包含一个header标签作为该文章的标题。示例代码：

```
<body>
    <article>
        <header><h1>搭建“云实验室” 产教融合助力大学生就业</h1></header>
        <p>疫情期间，大学生在家“云上课”，去机房做实验难以实现，急需实操训练的计算机...</p>
    </article>
</body>
```

article标签可以嵌套使用，但是要求内层的标签内容必须和外层的标签内容有关。例如外层标签是介绍手机性能的，那么内层标签可以介绍手机规格等内容。

```
<body>
<article>
    <header><h1>选择华为手机的六大理由</h1></header>
    <p>1、麒麟985 5G芯片....</p>
    <p>2、....</p>
    <article>
        <header><h3>包装清单</h3></header>
        <p>手机 * 1</p>
        ....
    </article>
</article>
</body>
```

除此之外，article标签还可以用来表示一个插件。使用它可以使插件看起来像内嵌入页面中一样。示例代码如下：

```
<article>
    <header>插件</header>
    <object>
        <embed type="video/webm" src="video.mp4" width="400" height="300">
    </object>
</article>
```

article和section的区别

1. article代表文档或页面中独立完整的，可被外界引用的内容。例如一篇文章或者一个帖子。而section则通常作为单纯的分区来使用，例如一篇文章的多个自然段。
2. article元素通常会包含header或者footer标签来实现独立内容的完整性。而section虽然也要求有标题元素，单会不使用header而通常会使用<hn>来实现定义标题的功能。
3. 从某种意义上来看，article可以看成是特殊的section。article更强调独立性、完整性，而section更强调相关性。
4. 如何article、aside或者nav标签更符合使用条件，那么就不要使用section。

2.2.2.5 附注栏（aside）

aside标签用来描述当前页面内容的附注信息，它可以是当前页面内容相关的引用、广告、侧边栏等。这些内容要区别于当前页面的主体内容。aside有两种使用方式：

- 作为主要内容的附属信息，例如当前文章的参考资料等。

```
<article>
    <h1>HTML5</h1>
    <p>HTML5是当前主流的...</p>
    <h3>参考资料</h3>
    <aside>
        <ul>
            <li>《HTML5从入门到精通》</li>
            <li>《HTML5开发指南》</li>
        </ul>
    </aside>
</article>
```

- 作为和页面相关的附属信息，例如侧边栏广告。

```
<aside>
    <nav>
        <h3>友情链接</h3>
        <ul>
            <li>链接1</li>
            <li>链接2</li>
            <li>...</li>
        </ul>
    </nav>
</aside>
```

2.2.2.6 页脚（footer）

footer标签可以作为内容的注脚，例如在网页中添加版权信息等。示例代码：

```
<footer>
    <ul>
        <li>关于</li>
        <li>导航</li>
        <li>联系</li>
    </ul>
</footer>
```

2.2.2.7 其他语义标签

标题分组（hgroup）

hgroup标签可以为标题或者子标题进行分组，标题通常使用<hn>组合。一块内容中的标题和子标题可以通过hgroup进行分组，若是只有一个标题则不用分组。示例代码：

```
<header>
    <hgroup>
        <h1>主标题</h1>
        <h3>副标题标题</h3>
    </hgroup>
</header>
```

时间（time）

time标签可以对日期进行管理操作。它代表某个日期或者是24小时中的某个时刻。当表示时刻的时候，可以使用时区进行显示。例如：

```
<time datetime="2020-6-9">2020年6月9日</time>
<time datetime="2020-6-9T20:00">2020年6月9日晚8点</time>
<time datetime="2020-6-9T20:00Z">2020年6月9日晚8点</time>
```

搜索引擎会读取datetime属性的值，而标签中的文字是在页面中显示的文字。其中"T"代表的是日期和时间的间隔。"z"代表的是时间使用的是UTC标准时间。

time标签还有一个特殊的属性为：pubdate。该属性是布尔类型，通常用在article标签中代表该文章的发布日期。示例代码：

```
<article>
    <header>
        <h1>文章标题</h1>
        <p>发布日期<time datetime="2020-6-9" pubdate>2020年6月9日</time></p>
    </header>
</article>
```

2.2.2.8 总结

通过对语义化标签的学习可以发现很多语义化同样具有划分区域的功能，例如article、section等。那么是否可以用这些标签来取代div实现页面布局呢？答案是否定的。首先article、section等标签是HTML5新定义的页面元素，从浏览器兼容的角度考虑若是碰到不兼容HTML5的浏览器，那么页面就会出现混乱。当然现在主流的浏览器都已经支持HTML5。

另外从使用的角度考虑，在HTML4的时候div就已经作为布局元素在使用，开发者已经习惯使用div作为容器进行开发。虽然HTML5添加了新元素，但这些新标签仅是作为div标签的补充，为div标签添加了一些新的含义，其最终的目的也是为了使div的布局工作更纯正。

2.3 HTML5 多媒体

随着技术的发展，web多媒体应用经历了多次重大的变革。早期的web仅支持图片和文字，无法支持多媒体。后期开始支持简单的MIDI和GIF动画。到现在可以支持各种音频和视频的播放。HTML4的多媒体播放需要各种插件来实现，例如：Window Media Play 、 Flash Player等。这些插件种类繁多，各自支持的视频格式也不一致，这给网站开发和用户的使用都带来了极大的不便。HTML5中多媒体功能的出现解决了这些问题，用户不需要安装任何插件就能播放页面中的多媒体音频和视频。HTML5提供了两个重要的多媒体标签audio和video。并提供了通用的、完整的、可脚本控制的API。通过对这两个标签的设置，开发人员就可以实现对多媒体的完全控制。下面就让我们来学习一下这两个标签的用法。

2.3.1 audio标签

在HTML5中，使用audio标签来播放声音文件或者音频流，该标签支持Ogg、MP3、WAV等音频格式。语法代码如下：

```
<audio src="samplemovie.mp4" controls="controls">您的浏览器不支持video标签</audio>
```

其中src属性用于指定要播放的声音文件，controls属性用于提供播放、暂停和音量控件。标签中的文本为提示信息，如果当前浏览器不支持audio元素，则可以显示这些信息。

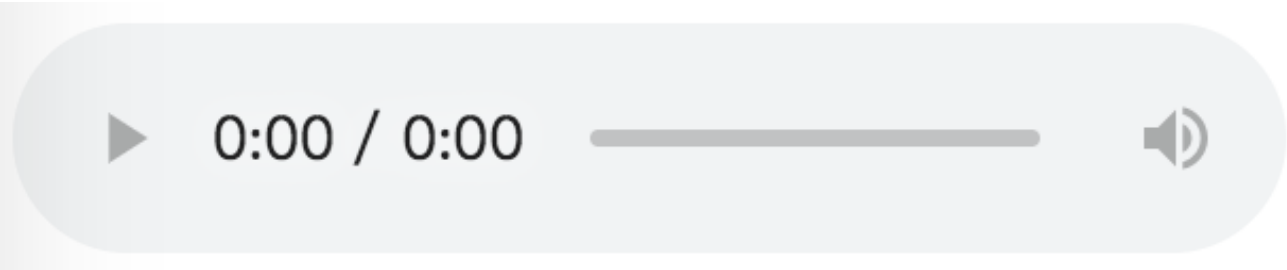
另外audio标签还可以通过子标签<source>来进行多数据源的设置。一个audio可以包含多个source，当播放器无法识别当前格式的播放源时会调用下一个source播放源进行播放。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<audio controls>
    <source src="test.mp3" type="audio/mpeg">
    <source src="test.mp3" type="audio/ogg">
    您的浏览器不支持audio标签
</audio>
</body>
```

```
</html>
```

显示效果：



可以看到页面出现一个简单的播放器，包括播放控制、时间显示、进度条显示和音量调节等控件。需要注意的是src属性指定的播放文件的格式一定要和type属性指定的类型保持一致，因为一个播放器可以播放各种类型的文件，type会告知播放器按那种类型的文件解码，如果播放的文件和类型的编码格式不匹配，那么浏览器就会以为该文件无法播放，所以一个比较简单的方法就是不写type属性，由浏览器自行检测编码方式。

很多网站的页面都带背景音乐，这个需求也可以通过audio标签来实现。示例代码如下：

```
<audio autoplay loop>
    <source src="test.mp3">
</audio>
```

其中的autoplay属性为设置自动播放。loop属性为循环播放。

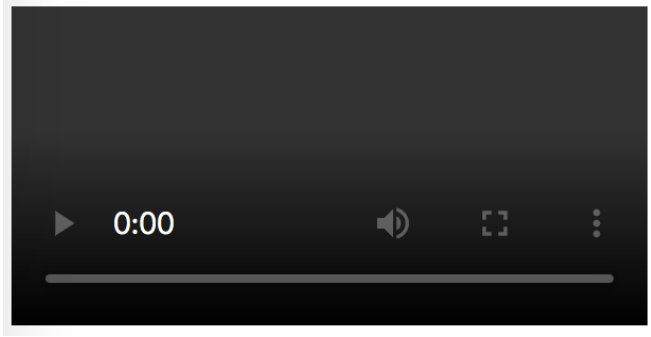
2.3.2 video标签

HTML5视频播放标签video的使用方式和audio的使用方式类似，都是可以通过video标签来对播放器进行设置，通过source标签来对播放源进行设置，当遇到无法播放的文件时也会显示标签内的内容。通过controls属性设置控制栏的显示，通过autoplay设置是否自动播放。

示例代码：

```
<video controls autoplay>
    <source src="test.mp4">
</video>
```

显示效果：



在某些时候可能会需要不使用控制栏来对视频进行控制，这时也可以使用一些内置的JavaScript函数和属性来控制播放器。功能如下表：

函数	描述
load()	用于加载音频或视频文件，为播放做准备，通常不需要使用。
play()	用于开始播放音频或视频文件，若该文件已经暂停，则继续播放。
pause()	用于暂停处于播放状态的音频或视频文件。
canPlayType(type)	用于检测video元素是否支持给定的MIME类型的文件。

示例代码：

```
<video onmousemove="this.play()" onmouseout="this.pause()">
    <source src="movie.mp4">
</video>
```

显示效果：

2.3.3 浏览器兼容控制

HTML5提供的audio和video标签已经可以被当前主流浏览器所支持，但是也是有部分浏览器不支持该播放器。如何解决浏览器多媒体兼容问题已经成为开发者在开发时必须考虑的问题。最终的设计目标确保视频文件在所有浏览器中（Internet Explorer, Chrome, Firefox, Safari, Opera）和所有硬件上（PC, Mac , iPad, iPhone）都能够正常播放。浏览器兼容问题的解决方案有很多，本节重点介绍两种常见的解决方案以供参考。

<video>+<object>标签方案

HTML5支持使用video标签进行多媒体播放，并且按照前面章节所讲内容，当video标签无法支持播放时，会进入到标签内的执行提示信息操作。因此可以在外层部署video标签，等video标签无法正常执行时，使用object标签执行。object标签是HTML4可以识别并执行的标签。这样就保证了在浏览器不支持HTML5的情况下以HTML4标准来进行解析。

示例代码：

```
<video width="320" height="240" controls="controls">
  <source src="movie.mp4" type="video/mp4" />
  <object data="movie.mp4" width="320" height="240">
    <embed src="movie.swf" width="320" height="240" />
  </object>
</video>
```

JavaScript控制方案

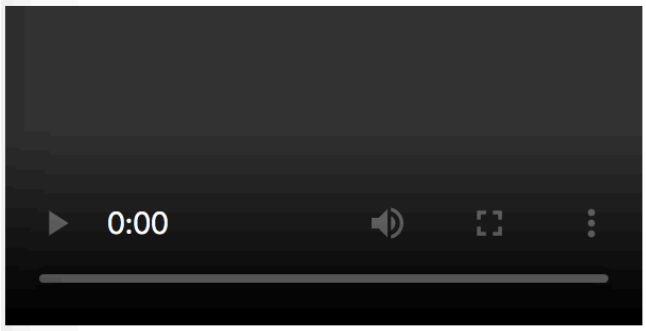
此方案从本质上来说还是先判断浏览器是否支持HTML5，不支持的情况下使用HTML4来替换多媒体标签实现播放的操作。和第一种方案的区别在于浏览器判断和标签的使用操作是由JavaScript脚本来实现控制。

检测视频是否能正常播放可以使用canPlayType(type)函数来实现。该函数只通过一行代码即可验证当前浏览器是否支持传入的视频格式。

示例代码：

```
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <script>
        function videoCheck() {
            var testVideo = document.getElementById("#video");
            if ("" == testVideo.canPlayType('video/ogg; codecs="theora, vorbis"') && ""
== testVideo.canPlayType('video/mp4; codecs="avc1.42E01E, mp4a.40.2"')) {
                document.getElementById("showMsg").innerText = '使用HTML4的flash播放！';
                testVideo.outerHTML = '<object data="movie.mp4" width="320"
height="240">\n' +
                    '    <embed src="movie.swf" width="320" height="240" />\n' +
                    '  </object>';
            } else {
                document.getElementById("showMsg").innerText = '使用HTML5的video播放！';
            }
        }

        window.onload = function () {
            videoCheck();
        }
    </script>
</head>
<body>
<div id="showMsg"></div>
<video id="#video" controls>
    <source src="test.mp4">
</video>
</body>
```

使用HTML5的video播放！



使用HTML4的flash播放！

2.4 HTML5 新表单元素和属性

表单是HTML中的一个重要组成部分，是用户和网站进行信息交互的一个重要途径。通过表单用户可以将需要进行处理的信息例如搜索的关键字，提交到网站服务器。服务器对关键字进行处理后将处理结果返回到浏览器。用户可以在网站页面看到处理后的结果。表单在整个流程中起到信息传递的功能。

HTML4支持的表单元素已经有很多，但是随着用户提交信息的多样化和数据验证的严谨性要求，原有的表单已经无法满足新的要求。HTML5中增加了更多的表单元素，并对表单标签进行了优化。这些都有利于前端设计工程师可以更加省力和高效的制作出更标准的web表单。

2.4.1 新的 input 元素

在HTML5之前，表单支持的input输入类型有十个，如下表所示：

输入类型	关键字	描述
文本域	text	输入单行文本，默认20个字宽度。
密码域	password	输入单行文本，内容会被“*”隐藏。
单选按钮	radio	在一组选择项中只能选择一项。
复选按钮	checkbox	在一组选择项中可以选多项。
提交按钮	submit	提交表单到服务器，以按钮的形式显示。
单击按钮	button	无任何功能，以按钮的形式显示。
图像按钮	image	提交表单到服务器，以图片的形式显示。
重置按钮	reset	重置表单元素内容。例如清空文本。
隐藏域	hidden	无法在页面显示，隐藏的文本域。
文件域	file	可以用于上传用户电脑文件。

在HTML5中新增了八个input输入类型。

- email
- url
- number

- range
- Date pickers (date, month, week, time, datetime, datetime-local)
- search
- tel
- color

下面将逐一介绍这些新增的输入类型。

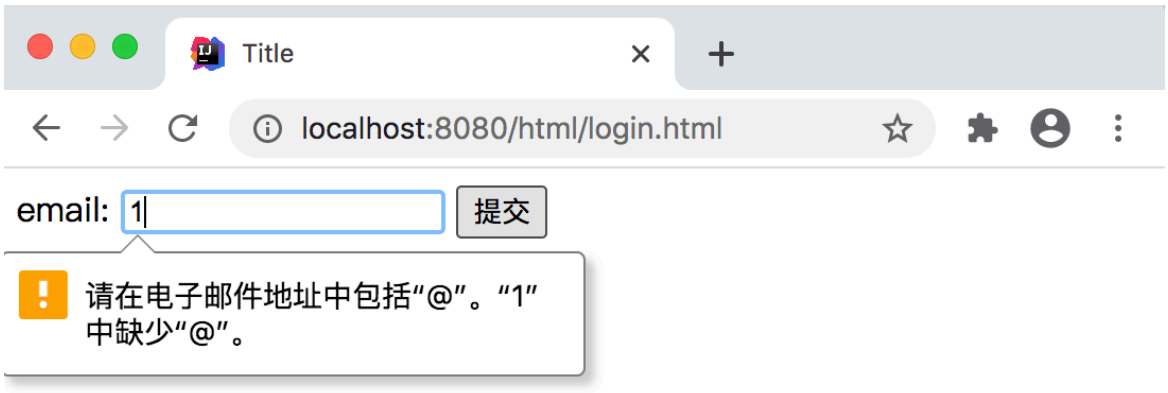
2.4.1.1 email类型

email类型是input标签中专门用来输入电子邮件的文本输入框，在提交表单的时候会对文本框中的内容进行自动校验，如果该文本框中的内容不是一个email格式的字符串，则不允许表单被提交。早期的的版本中采用“text文本域+JavaScript”的方式来实现此功能。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    email: <input type="email">
    <input type="submit" value="提交">
</form>
</body>
</html>
```

显示效果：



HTML5的email类型在浏览器的表现形式上同普通的text类型没有区别。但是在手机浏览器中这两种类型额input元素是有区别的。在手机中text元素会用正常的手机输入键盘来显示输入信息，但当该元素是email类型时，手机的输入键盘会通过改变输入键盘来配合该输入框。这种改变无疑提供了更好的用户体验。

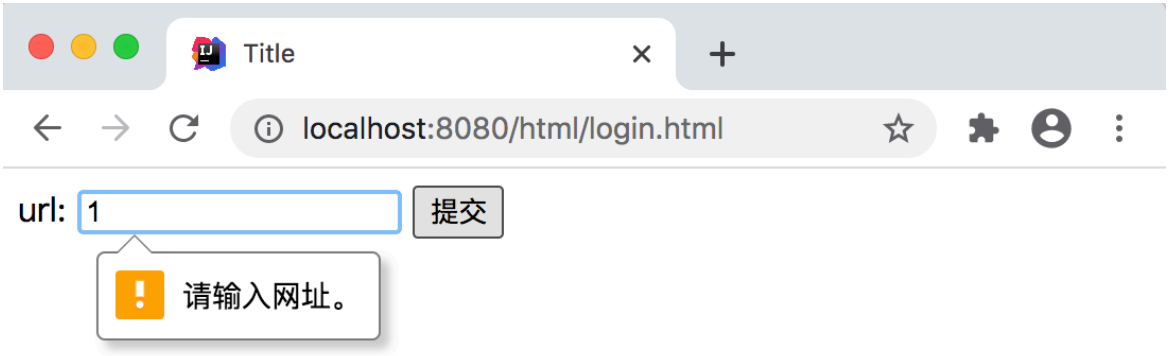
2.4.1.2 url类型

url类型是input标签中专门用来输入url地址的文本输入框，当表单提交时，如果输入的文本内容不是url格式，则不允许提交。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    url: <input type="url">
    <input type="submit" value="提交">
</form>
</body>
</html>
```

显示效果：



url类型同样在手机中会有特殊的键盘配合输入。

2.4.1.3 number类型

number类型是input标签专门用于输入数值的文本框，再使用该类型的时候，input标签还提供属性对所输入的数字进行限制。包括允许输入的最大值和最小值、合法的数字间隔和默认值。如果输入的数字不在规定的范围内，同样会提示错误信息。另外该类型的input标签只允许输入数字，其他字符无法输入。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    number: <input type="number" max="10">
    <input type="submit" value="提交">
</form>
</body>
</html>
```

显示效果：

number类型可配置属性

属性	描述
max	输入框允许的最大值
min	输入框允许的最小值
step	合法的数字间隔，例如step=2，则合法为：2、4、6、8
value	默认值

number类型同样在手机中会有特殊的键盘配合输入。效果如下：

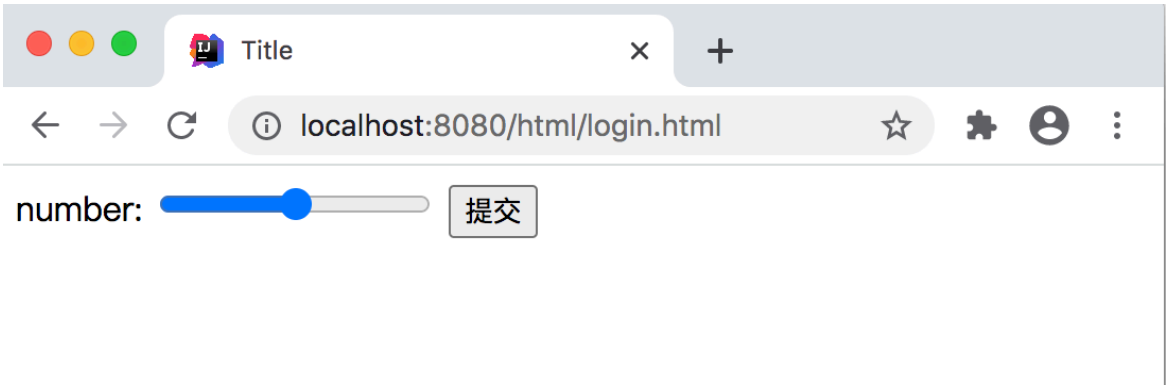
2.4.1.4 range类型

range类型是input元素提供输入在指定范围内数字值的文本框，在页面以滑动条的形式显示。使用range类型的标签时，必须指定最大值、最小值。否则该表单元素只能显示和操作，但无法正常使用。

示例代码：

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    number: <input type="range">
    <input type="submit" value="提交">
</form>
</body>
</html>
```

显示效果：



rang类型的属性和number类型的属性是一致的，功能也是类似的，区别在于外观显示不同。另外，支持range类型的浏览器会将其显示成滑块形式，不支持的浏览器会将其显示成text形式的纯文本输入框。

2.4.1.5 Date pickers类型

日期是网页中经常用到的一种输入类型。日期类型的输入数据若格式不对，在后期的程序开发过程中就会遇到很大的麻烦。为了规范输入日期的格式，同时也为了方便用户操作，在HTML5之前的版本开发过程中通常使用第三方的日期控件来实现日期的选择输入功能，这样操作既方便了用户的输入也可以对日期格式进行有效的控制。

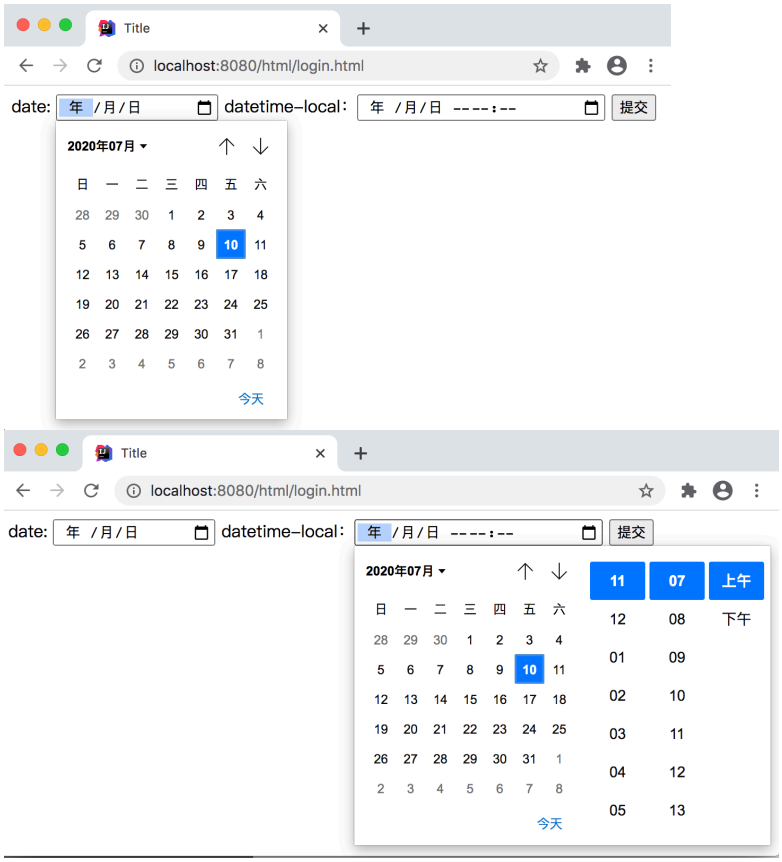
在HTML5中默认提供了六种日期选择控件，统称为Date pickers。通过对这些日期选择控件的使用可以灵活的对各种日期输入数据进行选择操作。这六种选择控件如下表所示：

类型	描述
date	选取年、月、日
month	选取年、月
week	选取年、周
time	选取时间小时、分钟
datetime	选取年、月、日和UTC时间
datetime-local	选取年、月、日和本地时间

这里只演示date和datetime-local两个日期选择控件。示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    date: <input type="date"> datetime-local: <input type="datetime-local">
    <input type="submit" value="提交">
</form>
</body>
</html>
```

显示效果：



这里需要注意datetime和datetime-local选择项虽然是HTML5标准中定义的日期类型选择项，但是在很多的浏览器中并不支持这个插件。例如最新的chrome支持datetime-local但是不支持datetime。所以使用的时候还是要注意选择。

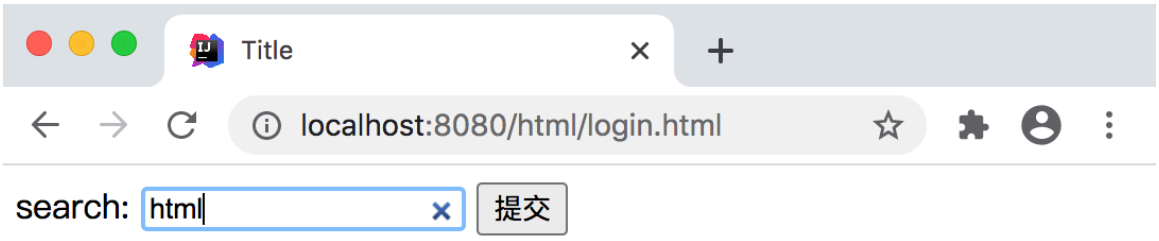
2.4.1.6 search类型

search类型的input标签提供用于输入搜索关键字的的文本框，从外观看search和text是一样的，功能也是相近的可以输入任意的字符串，实际操作过程可以看到当文本框中存在文字时，文本框的右侧会出现一个“×”按钮，单击此按钮可以情况文本框中的内容。这是search和text最大的区别。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    search: <input type="search">
    <input type="submit" value="提交">
</form>
</body>
</html>
```

显示效果：



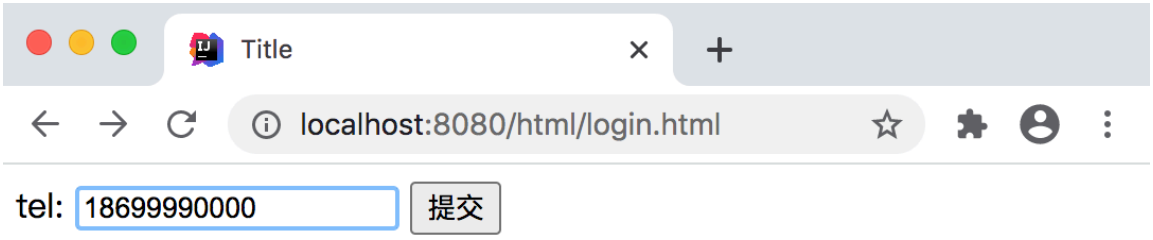
2.4.1.7 tel类型

tel类型是input标签专门用来输入电话号码的文本框，它并不仅局限输入数字，还可以输入一些其他的字的字符例如：-+ （）等。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    tel: <input type="tel">
<input type="submit" value="提交">
</form>
</body>
</html>
```

显示效果：



在浏览器中效果并不明显，但是在手机浏览器输入数据时，手机会默认使用数字键盘来进行输入操作。

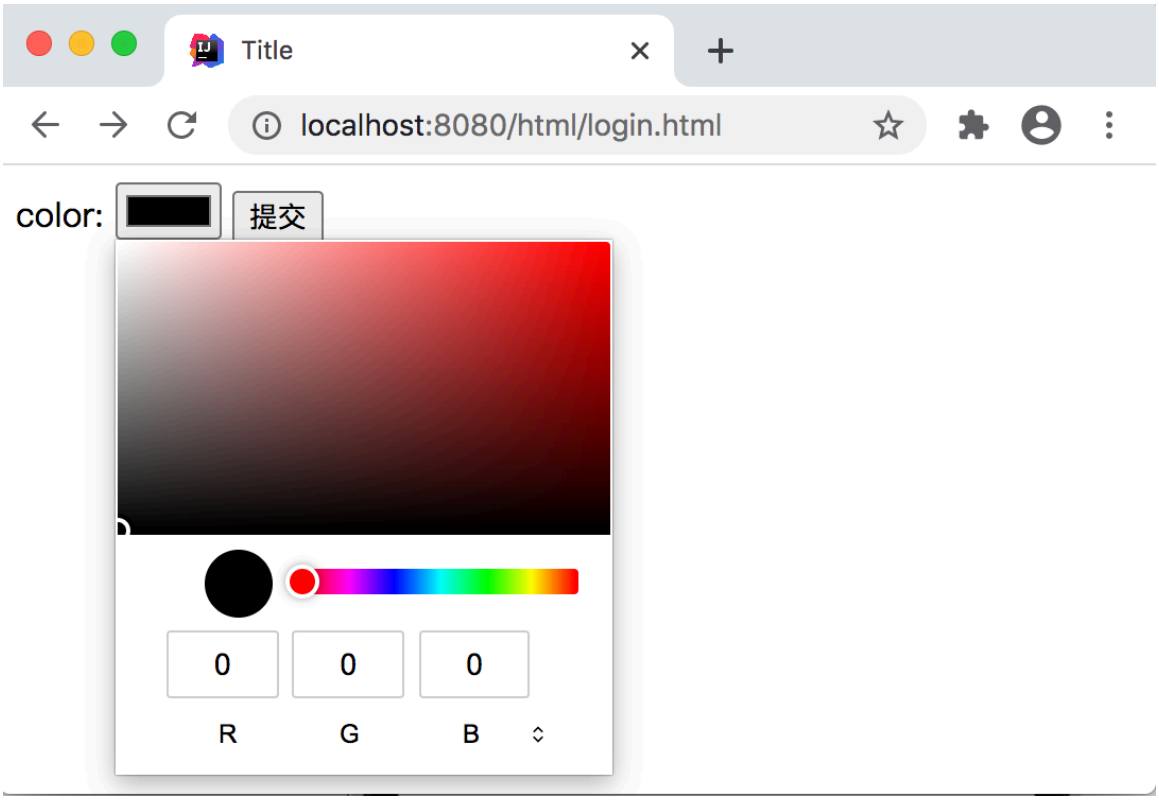
2.4.1.8 color类型

color类型是input标签提供的专门用于设置颜色的文本框。通过单击文本框可以打开一个调色板，用户可以在面板中选择需要的颜色。不同的操作系统打开的实测面板也不相同。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    color: <input type="color">
<input type="submit" value="提交">
</form>
</body>
</html>
```

显示效果：



2.4.2 新的 input 属性

input标签不仅添加了新的输入类型，还新增了一些input属性。这些属性可以指定输入类型的行为和限制。接下来逐一介绍一下这些属性的用法。

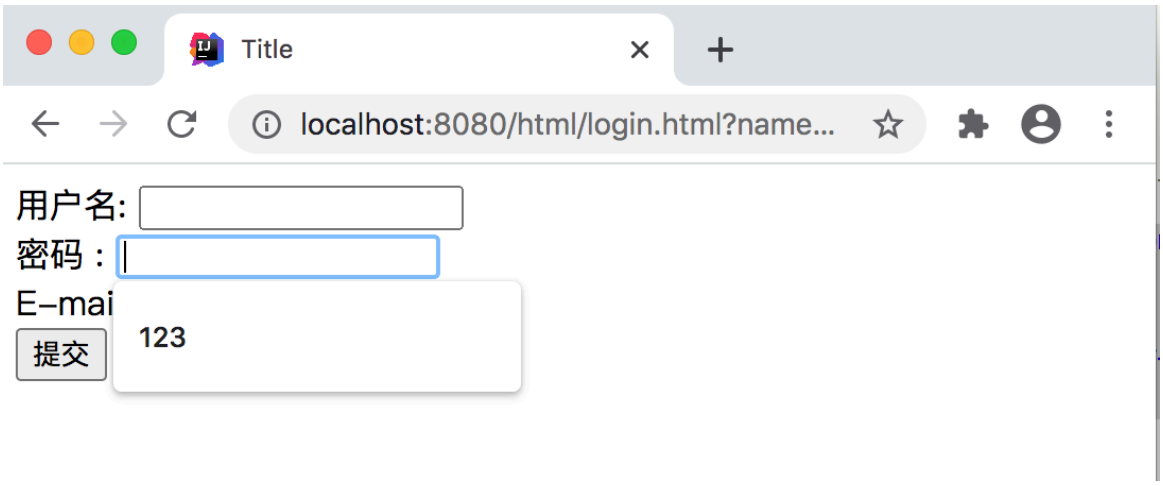
2.4.2.1 autocomplete 属性

当前主流浏览器都自带自动补全功能。该功能可以让用户在输入过一次数据后，再次输入相同数据时自动完成内容的补全。表单的autocomplete属性也可以完成这样的操作。支持的input类型包括几乎所有的可输入的文本框。autocomplete属性包括三种值：on、off、空值。form表单页包括该属性，作用是让该表单的所有表单元素都能自动补全。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form autocomplete="on">
    用户名: <input type="text" name="name" /><br />
    密码 : <input type="text" name="pass" /><br />
    E-mail: <input type="email" name="email" autocomplete="off" /><br />
    <input type="submit" />
</form>
</body>
</html>
```

显示效果：



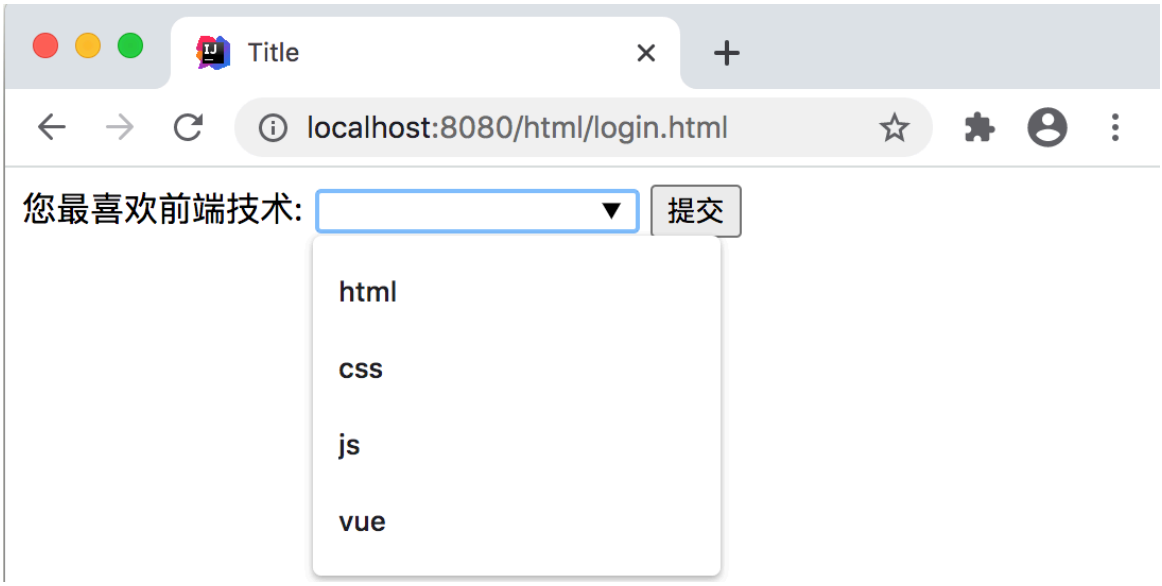
通过操作可以发现，用户名和密码都是可以提示的，email则不能。若是浏览器关闭了自动提示功能，此功能则无法实现。请设置浏览器的自动提示功能再实现此代码。

autocomplete属性也可以通过HTML5提供的datalist标签实现预设自动补全值的功能。用户可以选择预设值来实现自动填充功能。示例代码如下：

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

```
    <title>Title</title>
</head>
<body>
<form autocomplete="on">
    您最喜欢前端技术: <input type="text" list="selectList" />
    <datalist id="selectList">
        <option>html</option>
        <option>css</option>
        <option>js</option>
        <option>vue</option>
    </datalist>
    <input type="submit" />
</form>
</body>
</html>
```

显示效果：



> 注意：要将input标签和datalist标签关联在一起，需要input标签的list属性和datalist的id属性相同。

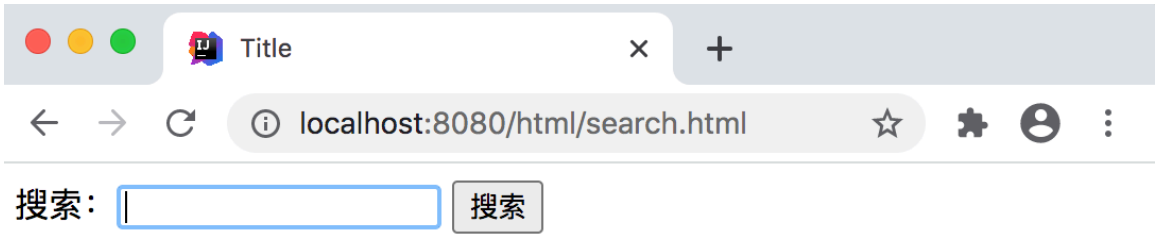
2.4.2.2 autofocus 属性

当打开一个页面时，当某个文本框需要获得光标焦点的时候。可以使用 autofocus 属性来实现。例如百度搜索页面，用户打开的时候搜索栏会自动获得光标焦点，用户直接输如需要搜索的内容即可，这种设置可以很好地提高用户体检。这个功能再HTML5前的版本是通过JavaScript脚本来实现。现在则可以使用属性实现这个功能。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    搜索: <input type="text" autofocus> <input type="submit" value="搜索">
</form>
</body>
</html>
```

显示效果:



当然有部分的浏览器不支持autofocus属性所以无法达到预期的效果，这时也可以使用一段JavaScript脚本来解决这个问题，首先判断一下浏览器是否支持该属性，不支持则通过JavaScript控制来获得。代码如下:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    搜索: <input id="search" type="text" autofocus> <input type="submit" value="搜索">
</form>
<script>
    if (!("autofocus" in document.createElement("input"))) {
        document.getElementById("search").focus();
    }
</script>
</body>
</html>
```

2.4.2.3 form 属性

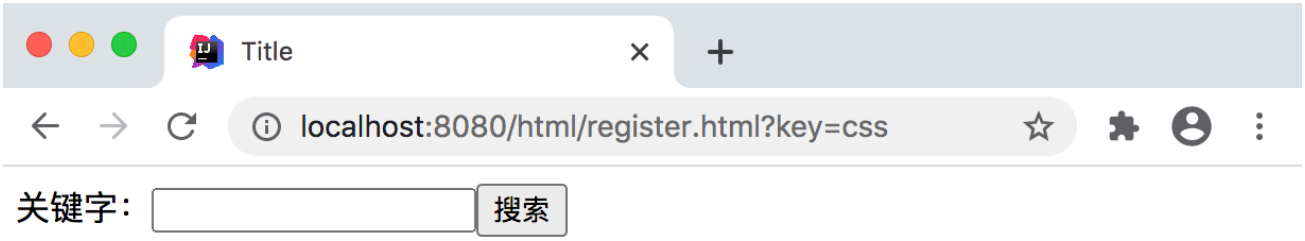
在HTML5之前的版本，当需要使用form表单的时候，所有的表单元素必须包含着<form>标签当中，若某个表单元素出现在<form>标签之外，那么当表单提交时此元素的值不会被发送到服务端。若需要提交<form>标签外的表单元素，除非使用JavaScript脚本控制单独提交或者使用Ajax提交才可以。这样的操作给开发带来了不变。

HTML5中每个表单元素都新增了一个form属性，通过该属性可以将表单元素绑定到指定的<form>标签上，这样就可以灵活进行布局，同时一个表单元素可以从属于多个表单，这就让表单和表单元素的组合变得更加灵活。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
    <form id="myForm"></form>
    关键字: <input type="text" name="key" form="myForm">
          <input type="submit" value="搜索" form="myForm">
</body>
</html>
```

显示效果：



通过URL地址可以看到表单可以正常工作。“key=css”会被提交到服务器。

2.4.2.4 form overrides 属性

HTML5新增了几个表单重写属性，用于重写表单的某些属性。但是这些属性只适用于submit和image输入类型。详情见下表：

属性	描述
formaction	重写表单的action属性
formenctype	重写表单的enctype属性
formmethod	重写表单的method属性
formnovalidate	重写表单的novalidate属性
formtarget	重写表单的target属性

例如现在有一个表单可以提交用户名和密码，用户可以选择不同的提交地址来实现不同身份的登录。示例代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form action="1.jsp" id="myForm">
    用户名：<input type="text" name="name"><br>
    密码：<input type="password" name="pass"><br>
    <input type="submit" value="用户登录">
    <input type="submit" value="员工登录" formaction="2.jsp">
</form>
</body>
</html>
```

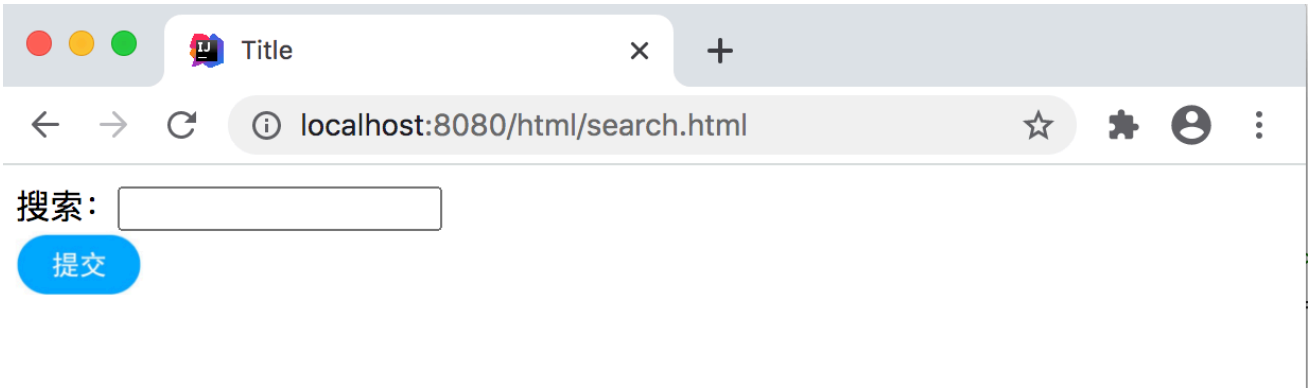
2.4.2.5 height 和 width 属性

height 和 width 属性用于规定image输入类型图片的高度和宽度。这两个属性只适用于image输入类型，其他输入类型无法使用。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    搜索：<input id="search" type="text" autofocus><br>
    <input type="image" src="images/submit.png" width="60" height="30">
</form>
</body>
</html>
```

显示效果：



2.4.2.6 list 属性

HTML5中添加了一个新标签datalist，该标签可以实现提供数据列表的功能。在自动补全练习里演示过该标签的使用方法。而list属性则是为了将该标签和指定文本框进行关联所设计的。示例代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form autocomplete="on">
    您最喜欢前端技术：<input type="text" list="selectList" />
    <datalist id="selectList">
        <option>html</option>
        <option>css</option>
        <option>js</option>
        <option>vue</option>
    </datalist>
    <input type="submit" />
</form>
</body>
```

```
</html>
```

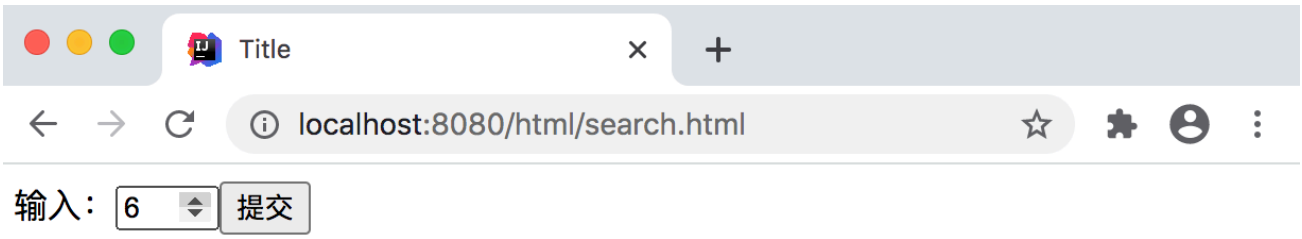
> 注意：list属性的值为某个datalist标签的id属性。

2.4.2.7 min, max 和 step 属性

HTML5新增了min, max 和 step 属性用于为数字或日期输入类型的input提供数值限制。可以使用该属性的标签有number、range、Date pickers。示例代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    输入：<input type="number" min="0" max="100" step="2"><input type="submit" value="提交">
</form>
</body>
</html>
```

显示效果：



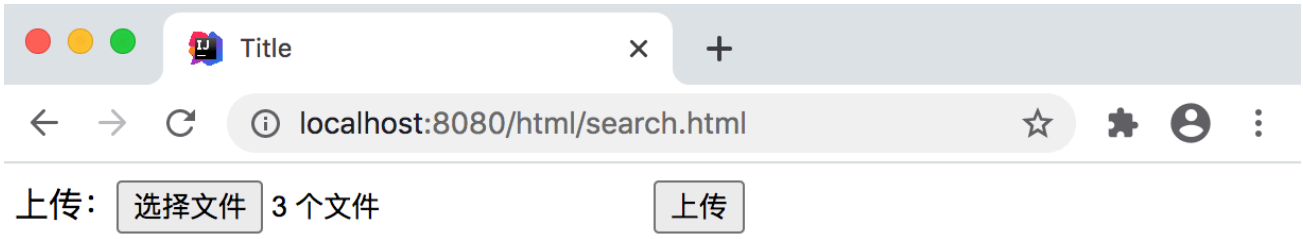
2.4.2.8 multiple 属性

HTML5提供了multiple 属性可以允许一次选择多个文件并上传，这在HTML5之前的版本中是不支持的。该属性主要适用于文件上传功能，其他例如email也可以使用该属性。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    上传：<input type="file" multiple><input type="submit" value="上传">
</form>
</body>
</html>
```

显示效果：



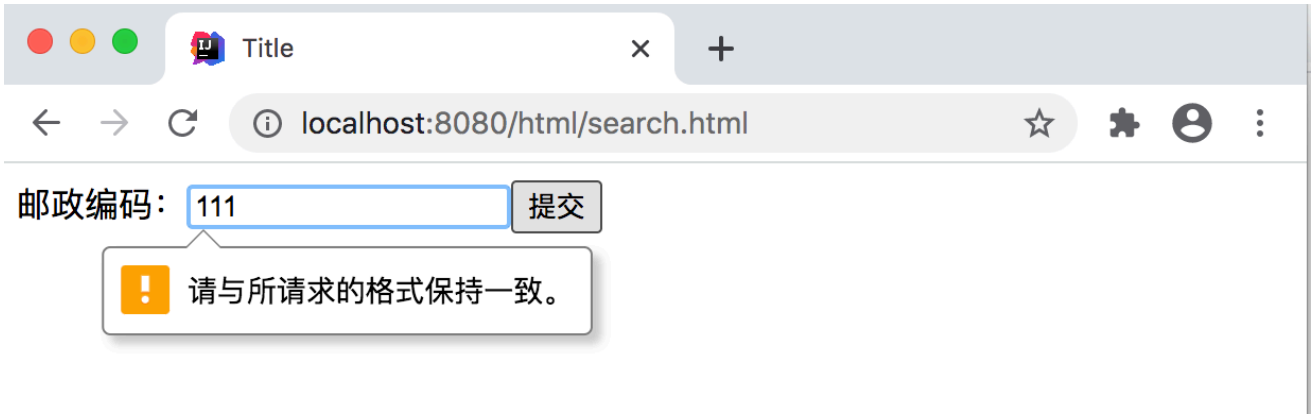
2.4.2.9 pattern (regexp) 属性

HTML5新增pattern属性用于对用户输入信息进行正则表达式校验的功能。该属性适用于大多输的input类型。例如text、pass、search等。而许多的输入类型其实已经定义好正则表达式来进行验证了。例如email、url等。使用正则校验过的数据才能确保数据的准确性。HTML5之前的版本一般都使用JavaScript脚本来实现数据校验的功能。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    邮政编码：<input type="text" pattern="[0-9]{6}"><input type="submit" value="提交">
</form>
</body>
</html>
```

显示效果：



patten属性的值为一个正则表达式，正则表达式的用法和数据校验的详细用法在下一节讲解。

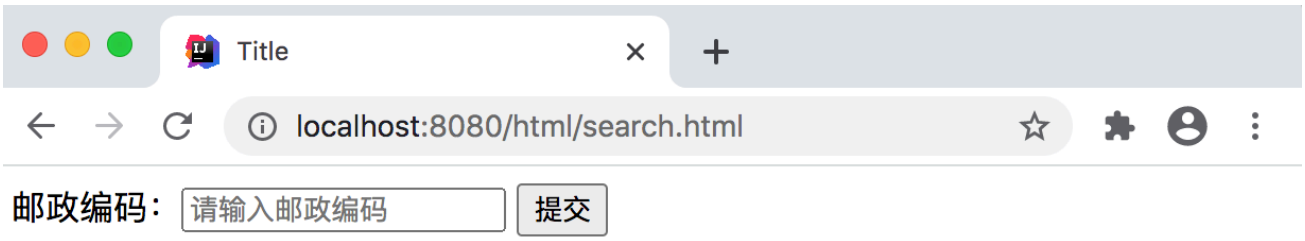
2.4.2.10 placeholder 属性

HTML5新增placeholder属性用于为文本框提供提示说明功能，提示说明该文本框希望用户输入何种数据。当用光标焦点聚焦到该文本框时，提示说明文字消失。不输入任何信息，提示说明文字不会被当做用户输入数据提交。该功能再HTML5之前的版本使用JavaScript来实现。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    邮政编码：<input type="text" pattern="[0-9]{6}" placeholder="请输入邮政编码">
    <input type="submit" value="提交">
</form>
</body>
</html>
```

显示效果：



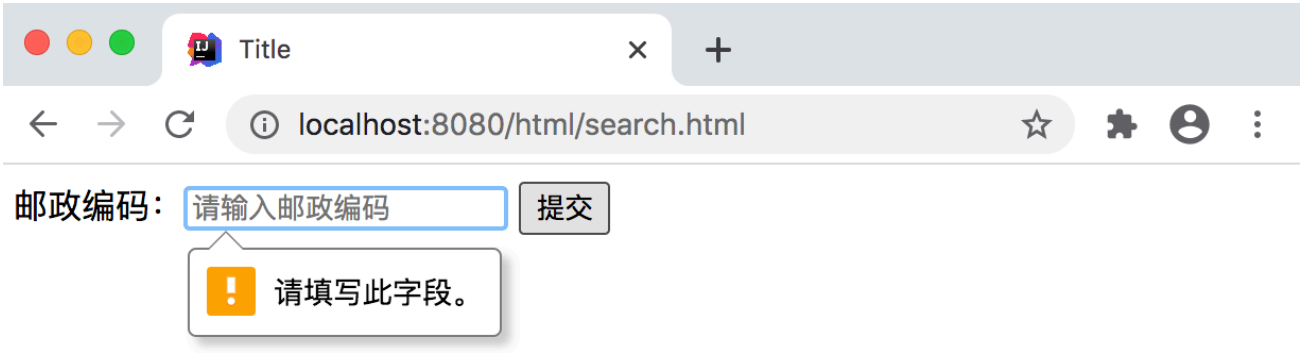
2.4.2.11 required 属性

HTML5新增的required 属性用于限定文本框内容不能为空。当文本框内容为空时无法提交表单。该属性适用于text、password等大多数文本框。HTML5前版本通过JavaScript实现该功能。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    邮政编码：<input name="key" type="text" placeholder="请输入邮政编码" required>
    <input type="submit" value="提交">
</form>
</body>
</html>
```

显示效果：



2.4.3 新的 form 元素

HTML5新添加了几个表单元素，分别是datalist、keygen和output。这些元素可以方便表单的使用。其中keygen在最新的HTML5的标准中已经被废弃，所以这里只介绍剩下两个元素。

2.4.3.1 datalist 元素

前面的章节已经演示了datalist的作用，datalist标签主要用于为输入框提供一个可选择的数据列表。在具有自动提示功能的文本框中可以使用这些选择项来自动补全文本框。该标签由<datalist>和<option>标签组合应用。一个datalist标签中可以包含多个option标签。他们的关系和表单元素<select>和<option>的关系一样。前面已经演示了datalist的使用，这里就不在赘述。

> 注意：option标签的value属性为提交值，标签中的为显示值。若不设置value属性，提交值即为显示值。

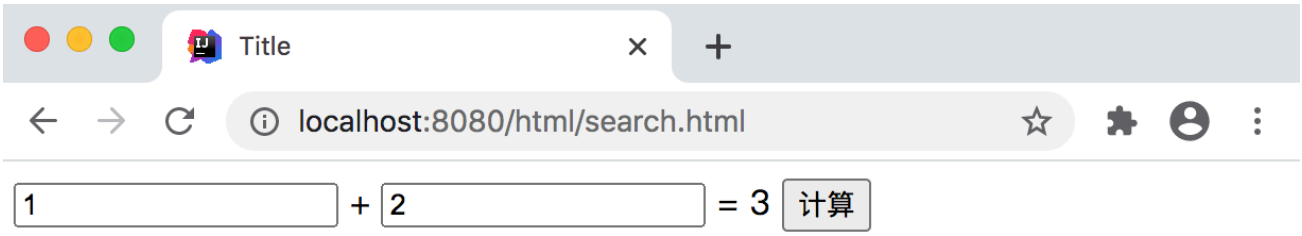
2.4.3.2 output 元素

HTML5新增的output元素用于在浏览器中显示计算结果或者脚本输出。输出的额结果在标签之间显示。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <script>
        function calc() {
            document.getElementById("result").value =
            parseInt(document.getElementById("num1").value) +
            parseInt(document.getElementById("num2").value);
        }
    </script>
</head>
<body>
<form>
    <input id="num1"/> +
    <input id="num2"/> =
    <output id="result"></output>
    <input type="button" onclick="calc()" value="计算">
</form>
</body>
</html>
```

显示效果：



2.4.4 新的 form 属性

HTML5新增了两个form属性：autocomplete 属性和novalidate 属性。下面分别介绍一下这两个属性。

2.4.4.1 autocomplete 属性

form的autocomplete 属性主要是用于设置表单元素的自动补全功能。该属性在前面的章节已经演示过，这里就不在赘述，该属性的作用是让对应表单中的所有元素都有自动补全功能。使用时可以不设置form的该属性而按需求每个表单元素单独设置。也可以设置form的该属性，在不需要的表单元素上设置关闭该属性。具体如何应用根据个人的使用习惯。

2.4.4.2 novalidate 属性

form的novalidate 属性用于取消表单的自动验证功能，例如前面已经讲过的email元素，当用户输入数据格式不符合email的格式要求时，系统会自动提示格式不正确，这个功能就是自动验证功能。若是用户不希望系统提示错误信息而是由自己写代码来实现验证提示功能的话，可以使用novalidate 属性关闭该功能。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form novalidate>
    email: <input type="email" name="email"><input type="submit" value="提交"><br>
</form>
</body>
</html>
```

2.4.5 正则表达式与表单验证

通过前面章节的学习，大家可以发现HTML5新增了多个input输入类型，这些输入类型自带验证规则，同时HTML5也新增了pattern属性，通过该属性可以设置正则表达式格式的验证规则。本节详细介绍一下HTML5自带的验证方式。

2.4.5.1 正则表达式

正则表达式又称为规则表达式，它描述了一种字符串匹配的模式，可以用来检查一个串是否含有某种子串、将匹配的子串替换或者从某个串中取出符合某个条件的子串等。

正则表达式可以在以下场景中使用：

- 表单验证
- 字符串查找
- 字符串替换

学习正则表达式主要从这几个方面学起：字符类型、数量限定、位置限定和特殊符号。

字符类型

字符类型包括所有大写和小写字母、所有数字、所有标点符号和一些其他符号。正则表达式还定义了一些特殊字符来代替一些字符组合，也称为字符类型。常见的字符类型如下表：

字符	描述
.点	匹配除“\n”和"\r"之外的任何单个字符。要匹配包括“\n”和"\r"在内的任何字符，请使用像“[\s\S]”的模式。
x|y	匹配x或y。例如，“z|food”能匹配“z”或“food”(此处请谨慎)。“[z|f]ood”则匹配“zood”或“food”。
[xyz]	字符集合。匹配所包含的任意一个字符。例如，“[abc]”可以匹配“plain”中的“a”。
[^xyz]	负值字符集合。匹配未包含的任意字符。例如，“abc”可以匹配“plain”中的“plin”任一字符。
[a-z]	字符范围。匹配指定范围内的任意字符。例如，“[a-z]”可以匹配“a”到“z”范围内的任意小写字母字符。注意:只有连字符在字符组内部时,并且出现在两个字符之间时,才能表示字符的范围; 如果出字符组的开头,则只能表示连字符本身.
\d	匹配一个数字字符。等价于[0-9]。
\D	匹配一个非数字字符。等价于"^0-9"。
\w	匹配包括下划线的任何单词字符。类似但不等价于“[A-Za-z0-9_]”，这里的"单词"字符使用Unicode字符集。
\W	匹配任何非单词字符。等价于“^A-Za-z0-9_”。

数量限定

正则表达式中可以对字符出现的次数进行限定。用来描述正则表达式的一个给定组件必须要出现多少次才能满足匹配常见的限定符号如下：

字符	描述
*	匹配前面的子表达式零次或多次。例如，zo* 能匹配 "z" 以及 "zoo"。* 等价于{0,}。
+	匹配前面的子表达式一次或多次。例如，'zo+' 能匹配 "zo" 以及 "zoo"，但不能匹配 "z"。+ 等价于 {1,}。
?	匹配前面的子表达式零次或一次。例如，"do(es)?" 可以匹配 "do" 、 "does" 中的 "does" 、 "doxy" 中的 "do" 。? 等价于 {0,1}。
{n}	n 是一个非负整数。匹配确定的 n 次。例如，'o{2}' 不能匹配 "Bob" 中的 'o'，但是能匹配 "food" 中的两个 o。
{n,}	n 是一个非负整数。至少匹配n 次。例如，'o{2,}' 不能匹配 "Bob" 中的 'o'，但能匹配 "foooood" 中的所有 o。'o{1,}' 等价于 'o+'。'o{0,}' 则等价于 'o*'。
{n,m}	m 和 n 均为非负整数，其中n <= m。最少匹配 n 次且最多匹配 m 次。例如，"o{1,3}" 将匹配 "fooooood" 中的前三个 o。'o{0,1}' 等价于 'o?'。请注意在逗号和两个数之间不能有空格。

位置限定

位置限定又称为定位符，是用来描述字符串或单词的边界，**^** 和 **$** 分别指字符串的开始与结束，**\b** 描述单词的前或后边界，**\B** 表示非单词边界。

字符	描述
^	匹配输入字符串开始的位置。如果设置了 RegExp 对象的 Multiline 属性，^ 还会与 \n 或 \r 之后的位置匹配。
$	匹配输入字符串结尾的位置。如果设置了 RegExp 对象的 Multiline 属性，$ 还会与 \n 或 \r 之前的位置匹配。
\b	匹配一个单词边界，即字与空格间的位置。
\B	非单词边界匹配。

特殊符号

正则表达式的特殊字符就是一些有特殊含义的字符，这些字符是正则表达式的保留字符，在使用的时候需要通过转义符转义才可以使用。

字符	描述
$	匹配输入字符串的结尾位置。如果设置了 RegExp 对象的 Multiline 属性，则 $ 也匹配 '\n' 或 '\r'。要匹配 $ 字符本身，请使用 $。
( )	标记一个子表达式的开始和结束位置。子表达式可以获取供以后使用。要匹配这些字符，请使用 (和)。
*	匹配前面的子表达式零次或多次。要匹配 * 字符，请使用 *。
+	匹配前面的子表达式一次或多次。要匹配 + 字符，请使用 +。
.	匹配除换行符 \n 之外的任何单字符。要匹配 . ，请使用 . 。
[	标记一个中括号表达式的开始。要匹配 [，请使用 [。
?	匹配前面的子表达式零次或一次，或指明一个非贪婪限定符。要匹配 ? 字符，请使用 \?。
\	将下一个字符标记为或特殊字符、或原义字符、或向后引用、或八进制转义符。例如， 'n' 匹配字符 'n'。'\n' 匹配换行符。序列 '\' 匹配 "\"，而 '(' 则匹配 "("。
^	匹配输入字符串的开始位置，除非在方括号表达式中使用，当该符号在方括号表达式中使用时，表示不接受该方括号表达式中的字符集合。要匹配 ^ 字符本身，请使用 \^。
{	标记限定符表达式的开始。要匹配 {，请使用 {。
|	指明两项之间的一个选择。要匹配 |，请使用 |。

常用正则表达式：

```
Email地址：^\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*$
域名：[a-zA-Z0-9][-a-zA-Z0-9]{0,62}(/.[a-zA-Z0-9][-a-zA-Z0-9]{0,62})+/.?
InternetURL：^[a-zA-z]+://[^\s]* 或 ^http://([\w-]+\.)+[\w-]+(/[\w-./?%&=]*)?$
手机号码：^(13[0-9]|14[5|7]|15[0|1|2|3|5|6|7|8|9]|18[0|1|2|3|5|6|7|8|9])\d{8}$
国内电话号码(0511-4405222、021-87888822)：\d{3}-\d{8}|\d{4}-\d{7}
身份证号(15位、18位数字)：^\d{15}|\d{18}$
帐号是否合法(字母开头，允许5-16字节，允许字母数字下划线)：^[a-zA-Z][a-zA-Z0-9_]{4,15}$
密码(以字母开头，长度在6~18之间，只能包含字母、数字和下划线)：^[a-zA-Z]\w{5,17}$
强密码(必须包含大小写字母和数字的组合，不能使用特殊字符，长度在8-10之间)：^(?=.*\d)(?=.*[a-z])(?
=.*[A-Z]).{8,10}$
日期格式：^\d{4}-\d{1,2}-\d{1,2}
一年的12个月(01～09和1～12)：^(0[1-9]|1[0-2])$
一个月的31天(01～09和1～31)：^((0[1-9])|((1|2)[0-9])|30|31)$
```

2.4.5.2 表单验证

使用HTML5实现表单验证需要一下几个前提条件：

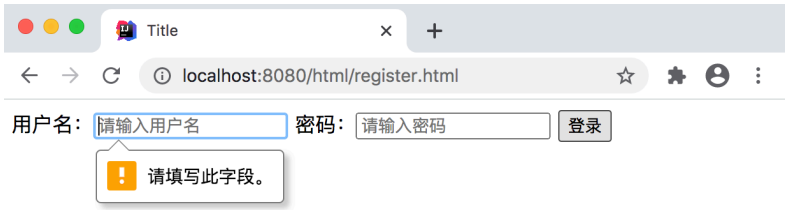
- form标签不能有novalidate属性
- input标签一定要有required 和pattern 属性
- 必须要有form标签和提交按钮
- 然后再脚本部分监听input的oninput 和 oninvalid事件 ， 使用validity 对象的相关属性

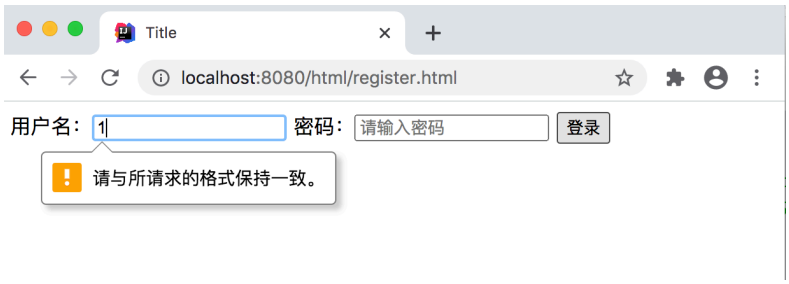
表单验证需要做两点，一个是保证required属性，另一个是要确定pattern属性值。例如一个用户登录页面，用户名必填，要求格式为字母数字和下划线组合，长度在6到20直接。密码为数字，长度也是6到20之间。

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form>
    用户名：<input type="text" id="userName" placeholder="请输入用户名" required
pattern="\w{6,20}">
    密码：<input type="password" id="password" placeholder="请输入密码" required
pattern="\d{6}">
    <button>登录</button>
</form>
</body>
</html>
```

显示效果：



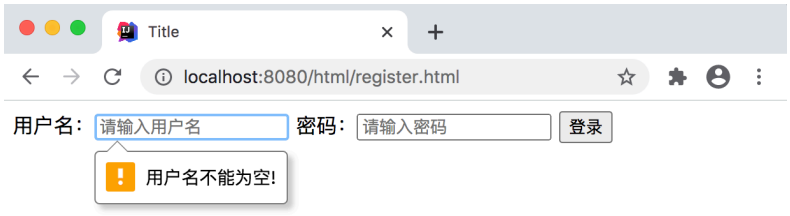


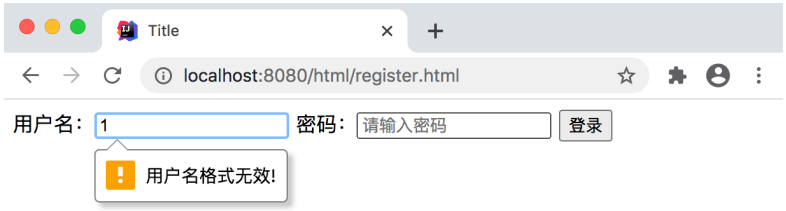
可以看到现在既可以判断输入信息为空的情况，也可以判断格式不符合的情况。但是这里有一个问题，假如需要更详细的提示该怎么办呢，这时就需要修改提示信息。可以使用JavaScript来控制。完整代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <script>
        window.onload = function () {
            var name = document.querySelector('#userName');
            var pass = document.querySelector('#password');

            name.oninvalid = checkUserName;
            name.oninput = checkUserName;

            function checkUserName() {
                if (this.validity.valueMissing) { //validity对象相关属性
                    this.setCustomValidity('用户名不能为空!');
                } else if (this.validity.patternMismatch) {
                    this.setCustomValidity('用户名格式无效!');
                } else {
                    this.setCustomValidity('');
                }
            }

            pass.oninvalid = checkPassword;
            pass.oninput = checkPassword;

            function checkPassword() {
                if (this.validity.valueMissing) {
```

```
                    this.setCustomValidity('密码不能为空!');
                } else if (this.validity.patternMismatch) {
                    this.setCustomValidity('密码必须为6-20位数字!');
                } else {
                    this.setCustomValidity('');
                }
            }
        }
    </script>
</head>
<body>
<form>
    用户名: <input type="text" id="userName" placeholder="请输入用户名" required
pattern="\w{6,15}">
    密码: <input type="password" id="password" placeholder="请输入密码" required
pattern="\d{6}">
    <button>登录</button>
</form>
</body>
</html>
```

显示效果:





2.5 HTML5 代码规范

任何代码都是需要有一定的规范进行约束，特别是当前工厂开发模式下的软件开发环境对代码的规范要求更加严格，使用规范化的代码能够方便开发者和调用者阅读和调用代码。

早期的HTML规范要求并不严格导致Web开发人员并没有一个统一的规范约束开发。在2000年至2010年，许多Web开发人员从 HTML 转换到 XHTML的开发，XHTML严格的约束条件使得Web开发人员逐渐养成了比较好的HTML 编写规范。虽然HTML5又放松了开发的规范要求，但是一个优秀的软件工程师一定要具备好的代码开发规范。下面就让我们系统的学习一下HTML5的代码规范。

1、使用正确的文档类型

文档类型声明位于HTML文档的第一行:

```
<!DOCTYPE html>
```

如果你想跟其他标签一样使用小写，可以使用以下代码，但不建议使用。

```
<!doctype html>
```

2、使用小写元素名

HTML5 元素名可以使用大写和小写字母。但是推荐使用小写字母：

- 混合了大小写的风格是非常糟糕的。
- 开发人员通常使用小写 (类似 XHTML)。
- 小写风格看起来更加清爽。
- 小写字母容易编写。

不推荐：

```
<SECTION>
  <p>这是一个段落。</p>
</SECTION>
```

非常糟糕：

```
<Section>
  <p>这是一个段落。</p>
</SECTION>
```

推荐：

```
<section>
  <p>这是一个段落。</p>
</section>
```

3、关闭所有 HTML 元素

在 HTML5 中, 你不一定要关闭所有元素 (例如

 元素)，但我们建议每个元素都要添加关闭标签。

不推荐：

```
<section>
  <p>这是一个段落。
  <p>这是一个段落。
</section>
```

推荐：

```
<section>
  <p>这是一个段落。</p>
  <p>这是一个段落。</p>
</section>
```

4、关闭空的 HTML 元素

在 HTML5 中, 空的 HTML 元素也不一定要关闭。

不推荐：

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

推荐：

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

5、使用小写属性名

HTML5 属性名允许使用大写和小写字母。但是推荐使用小写字母属性名。

- 同时使用大小写是非常不好的习惯。
- 开发人员通常使用小写 (类似 XHTML)。
- 小写风格看起来更加清爽。
- 小写字母容易编写。

不推荐：

```
<div CLASS="menu">
```

推荐：

```
<div class="menu">
```

6、属性值

HTML5 属性值可以不用引号，属性值我们推荐使用引号。

- 如果属性值含有空格需要使用引号。
- 混合风格不推荐的，建议统一风格。
- 属性值使用引号易于阅读。

不推荐：

```
<table class=table striped>
```

推荐：

```
<table class="table striped">
```

7、图片属性

图片通常使用 alt 属性。 在图片不能显示时，它能替代图片显示。

```
<img src="html5.gif" alt="HTML5" style="width:128px;height:128px">
```

8、空格和等号

等号前后可以使用空格。但是推荐不加空格。

不推荐：

```
<link rel = "stylesheet" href = "styles.css">
```

推荐：

```
<link rel="stylesheet" href="styles.css">
```

9、避免一行代码过长

使用 HTML 编辑器，左右滚动代码是不方便的。所以每行代码尽量少于 80 个字符。

10、空行和缩进

- 不要无缘无故添加空行。
- 为每个逻辑功能块添加空行，这样更易于阅读。
- 缩进使用两个空格，不建议使用 TAB。
- 比较短的代码间不要使用不必要的空行和缩进。

不推荐：

```
<body>

  <h1>蓝桥软件学院</h1>

  <h2>HTML5</h2>

  <p>
    蓝桥软件学院。
      蓝桥软件学院。
  蓝桥软件学院。
    蓝桥软件学院。
  </p>

</body>
```

推荐：

```
<body>
  <h1>蓝桥软件学院</h1>
  <h2>HTML5</h2>
  <p>
    蓝桥软件学院。
    蓝桥软件学院。
    蓝桥软件学院。
    蓝桥软件学院。
  </p>
</body>
```

11、省略标签

在HTML5中可以省略<html>、<head>、<body>、<title>这些结构标签，页面也能正常显示，但是不建议省略使用。

12、HTML 注释

注释可以写在 ，较长的注释可以换行。

```
<!-- 这是注释 -->
<!--
这是较长的注释
这是较长的注释
-->
```

13、使用小写文件名

大多 Web 服务器 (Apache, Unix) 对大小写敏感： london.jpg 不能通过 London.jpg 访问。其他 Web 服务器 (Microsoft, IIS) 对大小写不敏感： london.jpg 可以通过 London.jpg 或 london.jpg 访问。为了保持统一的风格建议统一使用小写的文件名。

14、文件扩展名

HTML 文件后缀可以是 .html (或 .htm)。建议使用.html来命名。

CSS 文件后缀是 .css 。

JavaScript 文件后缀是 .js 。

总结：

本章讲解了HTML5的新特性，包括网页的标准、代码的规范、HTML5的多媒体标签、HTML5的表单标签等。这些新特性重点偏重于Web开发过程中的网站开发，而对于HTML5新特性所支持的动画、绘图、缓存等知识点并没有深入的讲解，有兴趣的同学可以课后自学一下这些部分的内容。

本章练习：