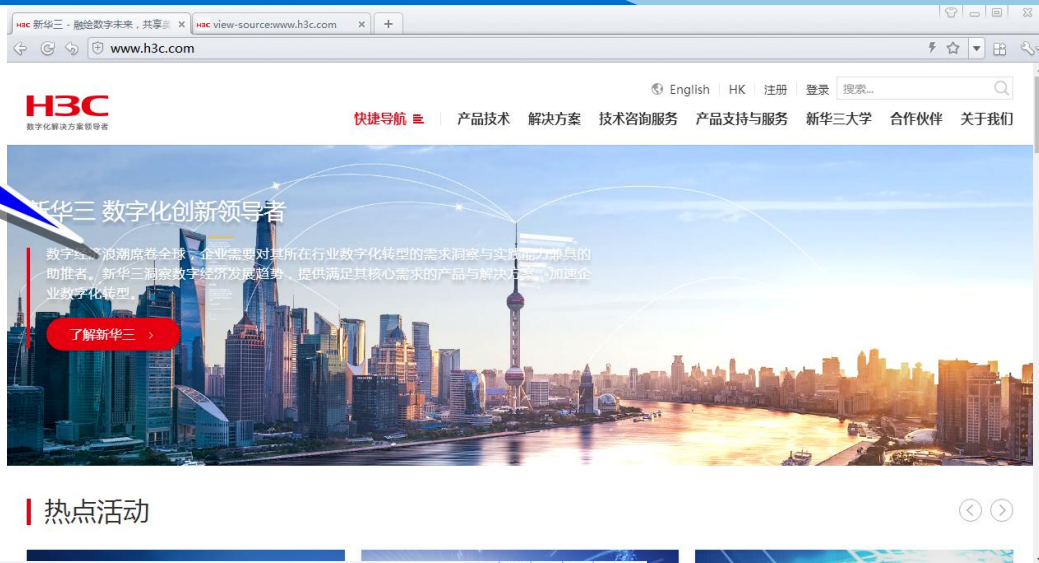


第13章 HTML5 基础与CSS3 应用(6-8课时)

采用HTML5和
CSS3布局



本章学习目标

主要内容：

- 熟悉掌握HTML5 新特性。
- 掌握HTML5 页面结构。
- 学会使用HTML5 新增元素和新增属性。
- 掌握HTML5 新增表单元素及新增属性的设置方法。
- 学会使用HTML5 的Audio 和Video 媒体元素。
- 学会HTML5 本地存储开发简易Web 应用。
- 学会使用CSS3 的转换、过渡和动画等特性设计页面的动态效果。
- 学会设置与应用CSS3 文本效果及多列等属性。

13.1 HTML5 概述

WHATWG (Web Hypertext Application Technology Working Group, Web 超文本应用技术工作组), WHATWG组织专门致力于Web 表单和应用程序, 当时W3C 专注于XHTML 2.0 标准的制定。2006年10 月, W3C 决定与WHATWG 合作共同研制HTML5 相关技术标准。

标记就是用来说明HTML 元素的。一个非空HTML 元素是由开始标记、元素的属性和值、内容和结束标记组成的, 是构成HTML 文件的基本对象。位于起始标记和结束标记之间的文本就是HTML 元素的内容。为HTML 元素提供各种附加信息的就是HTML 属性, 它总是以属性名= “属性值” 这种名值对的形式出现, 而且属性总是在HTML 元素的开始标记中进行定义。

13.1.1 HTML5 的八个特性

1.语义特性 (Semantic) 。HTML5赋予网页更好的意义和结构。

2.离线与存储特性 (Offline & Storage) 。HTML5开发的网页APP，启动时间更短，联网速度更快。由于有HTML5 APP Cache、本地存储功能、Indexed DB和File API说明文档。

3.设备访问特性 (Device Access)。HTML5提供了前所未有的数据与应用接入开放接口。使外部应用可以直接与浏览器内部的数据直接相连，例如视频影音可直接与麦克风及摄像头相联。

4. 多媒体特性(Multimedia)。支持网页端的Audio、Video等多媒体功能，与网站自带的APPS、摄像头、影音功能相得益彰。

13.1.1 HTML5 的八个特性

5. 三维、图形与特效特性 (3D、Graphics & Effects)。基于SVG、Canvas、WebGL及CSS3的3D功能，用户会惊叹于在浏览器中，所呈现的惊人视觉效果。

6. 性能与集成特性 (Performance & Integration)。HTML5会通过Web Workers和XMLHttpRequest2等技术，帮助您的Web应用和网站在多样化的环境中更快速的工作。

7. 连接特性 (Connectivity)。HTML5拥有更有效的服务器推送技术(Server-Sent Event和WebSockets)，能够帮助我们实现服务器将数据“推送”到客户端的功能。

8. CSS3特性(CSS3)。CSS3中提供了更多的风格和更强的效果。

13.1.2 HTML5 的优势

1. **摆脱对平台的依赖。** 打开浏览器，直接就可以访自己的应用。
2. **实时更新。**
3. **离线使用。** 用户可以离线使用，更新下载量及少。
4. **代码更安全。** HTML5可以将Web代码全部加密，本地应用解密后再运行，大大的提供了代码的安全性。
5. **跨平台。** JavaScript的代码可以在许多地方使用，包括移动应用、移动网站、PC网站、各种浏览器插件，甚至可以用WebKit封装作为跨平台的应用程序。
6. **可以充分利用Native。** HTML5可以通过浏览器作为中介充分利用Native的好处(使用GPS、照相机、本地相册、读取本地联系人等)。某些Web无法实现的功能，可以利用Native来实现。

13.1.3 HTML5 新增结构元素及页面元素

1.HTML5中新增加结构元素。例如Header页眉、nav导航、section节、article文章、aside侧栏、footer页脚。

2. HTML5 中新增页面元素。例如video、audio、embed、progress、time、mark、ruby、rt、rp、canvas、command、datalist、output、wbr、source、menu、details等。

13.1.4 HTML5 废除的元素与属性

HTML4.01之前有些标记被不赞成使用，HTML5已经淘汰了，建议使用CSS来替代。还有些标记HTML5已经不再支持，所以也需要淘汰。(1) 纯表现的元素。如font、basefont、center、big、s、u、strike、tt。(2) 对可用性产生负面影响的元素。如frameset、frame、noframes等元素。HTML5只支持浮动框架（内联框架）iframe元素。(3) 易产生混淆的元素。如acronym、applet、isindex、dir等元素。(4) 废除只有部分浏览器支持的元素。如blink、bgsound、marquee等元素。(5) 其它被废除的元素。如废除rb，使用ruby替代；废除listing使用pre替代；废除xmp使用code替代；废除nextid使用guids替代；废除plaintext使用“text/plain” MIME类型替代。HTML5废除属性如表13-3所示。

13.1.5 浏览器支持与选择

一些低版本的浏览器并不支持HTML5，如IE6～IE8浏览器。所有新、旧浏览器，对无法识别的元素均会视作内联(inline)元素来自自动处理。可以通过其它方法让这些浏览器能够处理“未知”的HTML元素。使用<http://html5test.com>来测试浏览器的支持。

html5shiv是针对IE浏览器比较好的解决方案。html5shiv主要解决HTML5提出的不被IE6～IE8识别新的元素，不能作为父节点包裹子元素，不能应用CSS样式。从指定网站上直接下载并保存到本地项目目录中（<https://github.com/aFarkas/html5shiv/>）。

```
<head>
<!--[if lt IE 9]>
    <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"> </script>
<![endif]-->
</head>
```

13.2 HTML5 文档结构

HTML5 文档结构同样是由头部和主体两部分组成，只是新增了一些结构元素，如header、nav、article、section、aside、footer六个结构元素，这些元素都是块级元素。

13.2.1 HTML5 页面结构



图13-2 HTML4.01页面布局

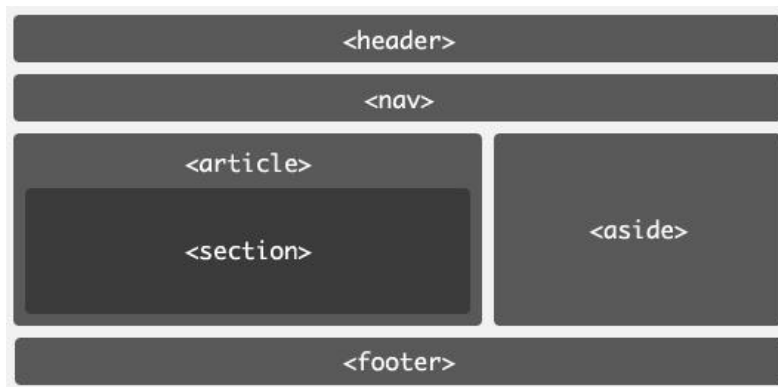


图13-3 HTML5结构元素布局

13.2 HTML5 文档结构

HTML5 页面结构元素语法:

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="Keywords" content="">
    <meta name="Description" content="">
    <title>HTML5文档结构</title>
  </head>
  <body>
    <header>
      <nav>...</nav>
    </header>
    <article>
      <section>...</section>
    </article>
    <aside>...</aside>
    <footer>...</footer>
  </body>
</html>
```

13.2.2 HTML5 新增结构元素

1. header 标记

header 标记定义文档和区域的页眉，通常是一些引导和导航信息。它不局限于写在网页头部，也可以写在网页内容里面。通常 <header> 标记至少包含（但不局限于）一个标题标记（h1 ~ h6），也可以包括hgroup（标题组合） 标记、表格标识、搜索表单、导航等。

【例13-2-1】标题组合标记的应用。

13.2.2 HTML5 新增结构元素

```
<!-- edu_13_2_1.html -->
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>HTML5结构元素header和hgroup标
记的应用</title>
  </head>
  <body>
    <header>
      <hgroup>
        <h1>HTML5 是下一代的 HTML。 </h1>
        <h3>什么是 HTML5? </h3>
        <h5>HTML5 将成为 HTML、XHTML 以及
HTML DOM 的新标准。 </h5>
      </hgroup>
    </header>
  </body>
</html>
```



13.2.2 HTML5 新增结构元素

2. nav 标记

nav 标记代表页面的一个部分，是一个可以作为页面导航的链接组。建议不要在footer元素中使用nav 元素，否则易造成页面显示不正确。配置相应的CSS 代码可以实现水平导航。

【例13-2-2】导航nav标记的应用。

```
<!-- edu_13_2_2.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>HTML5结构元素header和nav标记的应用
</title>
</head>
<body>
<header>
```

```
<nav>
<ul>
<li><a href="#">HTML 参考手册</a></li>
<li><a href="#">HTML 实例</a></li>
<li><a href="#">HTML 测验</a></li>
</ul>
</nav>
</header>
</body>
</html>
```

13.2.2 HTML5 新增结构元素

3.article标记

article标记是一个特殊的section标记，它比section具有更明确的语义，它代表一个独立的、完整的相关内容块，可独立于页面其它内容使用。例如论坛帖子、博客文章、新闻故事、评论等。

一般来说，article会有标题部分，通常包含在header内，有时也会包含footer。article标记可以嵌套，内层的article对外层的article标记有隶属关系。例如1篇博客的文章可以用article显示，然后后续的一些评论可以用article的形式嵌入其中。

【例13-2-3】文章article 标记的应用。

13.2.2 HTML5 新增结构元素

```
<!-- edu_13_2_3.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>HTML5结构元素artical和header标记的应用</title>
</head>
<body>
<article>
<header>
<hgroup>

<h1>HTML 5结构元素的简介</h1>
<h2>HTML 5的诞生</h2>
</hgroup>
<time datetime="2017-04-28">2017-04-28</time>
</header>
<p>HTML5 引入了许多新元素，包括几个用于更好地描述文本结构的元素。在本文中，我们将了解这些
HTML5 引入的新的结构化元素以及如何使用它们将一个文档划分成几个内容块。</p>
</article>
</body>
</html>
```



13.2.2 HTML5 新增结构元素

4. section 标记

section 标记定义文档中的节。例如章节、页眉、页脚或文档中的其他部分。一般用于成节的内容，会在文档流中开始一个新的节。它用来表现普通的文档内容或应用区块，通常由内容及其标题组成。section 元素不是一个普通的容器元素，它表示一段专题性的内容，可以带有标题。如果描述一件具体的事物，建议使用article 来代替section；如果使用section，仍可以使用h1 作为标题，而不用担心它所处的位置。如果一个容器需要定义样式或定义行为，建议用div 元素。

13.2.2 HTML5 新增结构元素

```
<!-- edu_13_2_4.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>HTML5结构元素article和section标记的应用
  </title>
</head>
<body>
<section>
<h1>section标记</h1>
<p>用来定义文档中的节(section、区段)。比如章节、
  页眉、页脚或文档中的其他部分。</p>
</section>
<section>
<h1>article标记</h1>
<p>article标记标识了Web页面中的主要内容。以博客
  为例，每篇帖子都构成一个重要内容。</p>
</section>
</body>
</html>
```



13.2.2 HTML5 新增结构元素

5. aside 标记

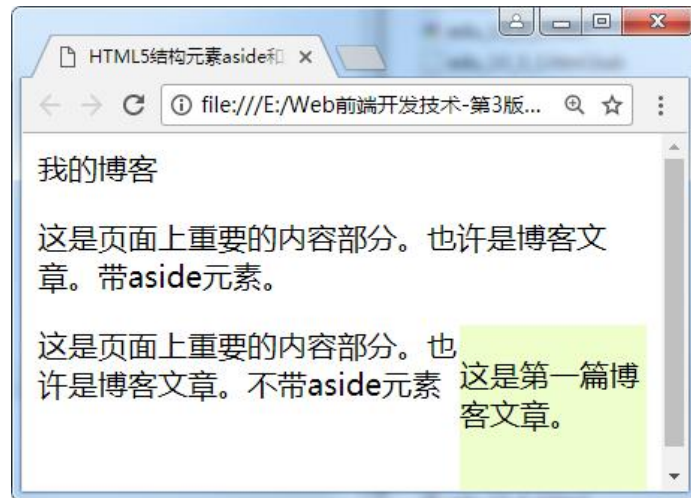
aside（侧栏，也称为旁注）标记用来说明其所包含的内容与页面主要内容相关，但不是该页面的一部分，类似于使用括号对正文进行注释（就像这样）。括号中的内容提供关于该元素的一些附加信息，例如广告、成组的链接、侧栏等。

【例13-2-5】aside 标记的应用。

```
<!-- edu_13_2_5.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>HTML5结构元素aside和artical标记的应用</title>
</head>
```

13.2.2 HTML5 新增结构元素

```
<body>
<header>我的博客</header>
<section>
<article>
<p>
这是页面上重要的内容部分。也许是博客文章。带aside元素。
</p>
<aside
  style="float:right;width:100px;height:100px;background
  :#EEffcc;">
  <p>这是第一篇博客文章。 </p>
</aside>
</article>
<article>
<p>这是页面上重要的内容部分。也许是博客文章。不带aside
元素</p>
</article>
</section>
</body>
</html>
```

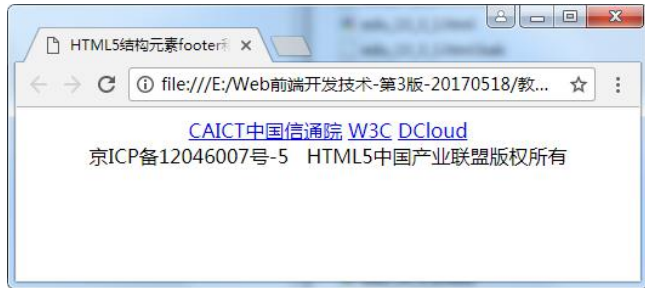


13.2.2 HTML5 新增结构元素

6. footer 标记

footer 标记定义section 或文档的页脚，包含了与页面、文章或部分内容有关的信息，例如文章的作者或者日期。作为页面的页脚时，一般包含了版权、相关文件和链接。它与页眉header 标记用法相同，在一个页面中可以多次使用，若在一个区段的最后使用footer标记，那么它就相当于该区段的页脚。

【例13-2-6】 footer 标记的应用



13.2.2 HTML5 新增结构元素

```
<!-- edu_13_2_6.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>HTML5结构元素footer和section标记的应用</title>
</head>
<body>
<footer>
<div style="text-align:center;">
<section>
<a href="http://www.caict.ac.cn/" target="_blank">CAICT中国信通院</a>
<a href="//www.w3.org/" target="_blank">W3C</a>
<a href="//www.dcloud.io/" target="_blank">DCloud</a>
</section>
<span style="padding:2px 5px;">京ICP备12046007号-5</span>
<span style="padding:2px 5px;">HTML5中国产业联盟版权所有</span>
</div>
</footer>
</body>
</html>
```

13.3 HTML5 新增页面元素

HTML5除了新增的结构元素header、nav、article、aside、section、footer外，还增加了新的内联元素(time、meter及progress等)、新的内嵌元素(video和audio)、新的交互元素（details、datagrid和command等）及其它页面元素。

13.3.1 hgroup 标记

标题组合hgroup 标记是对网页或区段section 的标题元素（h1~h6）进行组合。例如，在某一区段中需要连续设置多个标题标记，可以使用hgroup 标记来组合。

基础语法：

```
<hgroup>  
  <h1>标题1</h1>  
  <h2>标题2</h2>.....  
</hgroup>
```

【例13-3-1】hgroup 标记的应用

```
<!-- edu_13_3_1.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
  <title>HTML5页面元素hgroup标记的应用
</title>
</head>
<body>
  <hgroup>
    <h1>JSDoc+规范</h1>
    <h2 style="color:red;">介绍</h2>
  </hgroup>
  <p style="text-indent:2em;">编写JSDoc
  是为了增强代码的可读性, 以及方便导出API
  文档。它的规范可参考JSDoc 3对于代码规范
  要求高的工程师和JS框架的开发者, 熟悉
  JSDoc是必须的技能。</p>
</body>
</html>
```



13.3.2 figure 标记与figcaption 标记

figure标记用于对元素进行组合，常用于图像与图像描述组合。figcaption（图题）标记用于定义figure元素的标题（caption），可以给一组图像标记定义标题，但figcaption 标记不是必需的。如果包含了figcaption元素，那么它必须放置在figure元素的第一个或最后一个子元素的位置上。

基础语法：

```
<figure>  
  <p>图像文件说明内容。</p>  
  <img src="" width="" alt="" title="" />  
  <figcaption >为图添加标题)</figcaption>  
</figure>
```

【例13-3-2】 figure 与figcaption 标记的应用

```
<!-- edu_13_3_2.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>HTML5页面元素figure与figcaption标记
的应用</title>
</head>
<body>
```

```
<figure>
  <p>HTML5具有语义、离线与存储、设备访问等八个新特性，其对应的logo如下图所示： </p>
  
  
  
  <figcaption>HTML5新logo(图题)</figcaption>
</figure>
</body>
</html>
```



13.3.3 mark 标记与time 标记

记号mark 标记用来定义带有记号的文本。在需要突出显示文本时可以使用mark 标记。此标记对关键字做高亮处理（黄底色标注），突出显示，标注重点，在搜索方面可以应用。

时间time 标记用来定义公历的时间（24 小时制）或日期，时间和时区偏移是可选的。该标记能够以机器可读的方式对日期和时间进行编码。该标记不会在任何浏览器中呈现任何特殊效果。

1. 基本语法

`<mark>重点标注的内容</mark>`

`<time>9:00</time> <!-- 定义时间 -->`

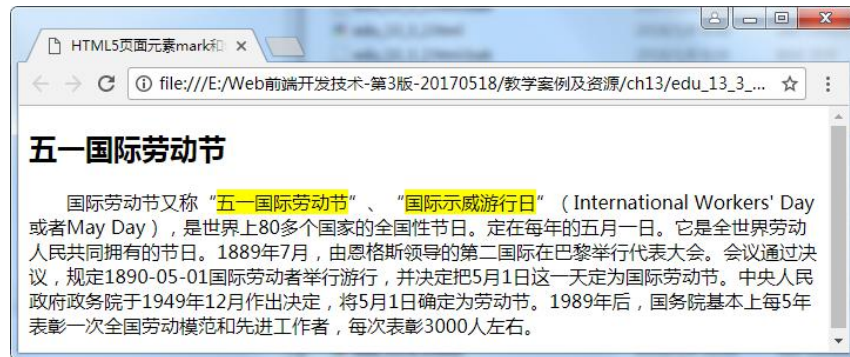
`<time datetime="2017-05-01" pubdate="pubdate">国际劳动节</time>`

`<!--定义日期 -->`

time标记的pubdate属性：指示该标记中的日期/时间是文档（或最近的article标记）的发布日期。time标记的datetime属性：规定日期/时间。

【例13-3-3】 mark 和time 标记的应用

```
<!-- edu_13_3_3.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>HTML5页面元素mark和time标记的应用
</title>
</head>
<body>
<article>
<header>
<h1>五一国际劳动节</h1>
</header>
<p style="text-indent:2em;">国际劳动节又称“<mark>五一国际劳动节</mark>”、“<mark>
国际示威游行日</mark>”(International Workers' Day或者May Day)，是世界上80多个国家的全
国性节日。定在每年的五月一日。它是全世界劳动人民共同拥有的节日。1889年7月，由恩格斯领导的第
二国际在巴黎举行代表大会。会议通过决议，规定<time datetime="1890-05-01">1890-05-
01</time>国际劳动者举行游行，并决定把5月1日这一天定为国际劳动节。中央人民政府政务院于1949
年12月作出决定，将5月1日确定为劳动节。1989年后，国务院基本上每5年表彰一次全国劳动模范和先进
工作者，每次表彰3000人左右。</p></article></body>
</html>
```



13.3.4 details 标记与summary 标记

细节details标记是一个开关式、交互式控件，用来定义用户可见的或者隐藏的需求补充细节，任何形式的内容都能被放在该标记中。该元素的内容对用户是不可见的，除非设置了open属性。

摘要summary 标记配合使用可以为details定义标题，summary元素应该是details元素的第一个子元素。标题是可见的，用户单击标题时，会显示出details。只有Chrome、Safari 6 以上支持summary标记。

●基本语法

`<details open>`

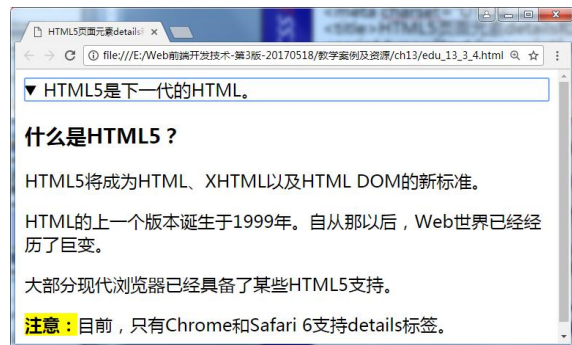
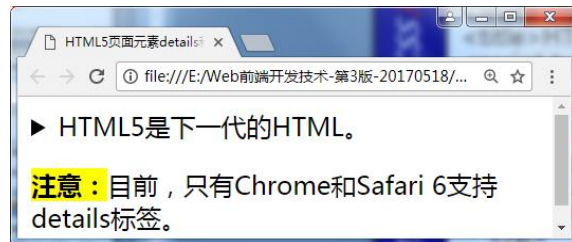
`<summary> details的标题</summary>`

details的详细内容

`</details>`

【例13-3-4】 details 和summary 标记的应用

```
<!-- edu_13_3_4.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>HTML5页面元素details和summary标记的应用</title>
<script type="text/javascript" src="html5shiv.js"> </script>
</head>
<body>
<details>
<summary>HTML5是下一代的HTML。</summary>
<h3>什么是HTML5? </h3>
<p>HTML5将成为HTML、XHTML以及HTML DOM的新标准。
</p>
<p>HTML的上一个版本诞生于1999年。自从那以后，Web世界已经
经历了巨变。</p>
<p>大部分现代浏览器已经具备了某些HTML5支持。</p>
</details>
<p><strong><mark>注意：</mark></strong>目前，只有
Chrome和Safari 6支持details标签。</p>
</body>
</html>
```



13.3.5 progress 标记与meter 标记

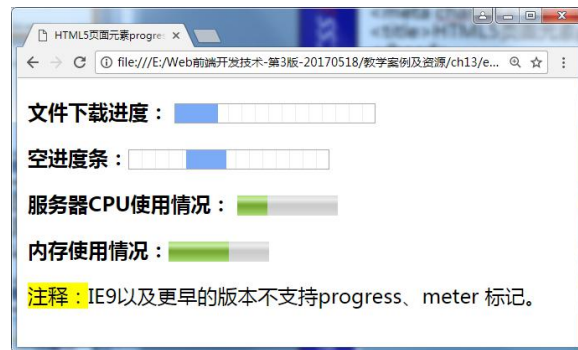
- 进度Progress标记用来定义运行中的任务进度（进程）。该标记有两个属性：max表示规定需要完成的值；value规定进程的当前值。度量meter 标记定义已知范围或分数值内的标量测量，也被称为gauge(尺度)。如磁盘用量、CPU使用率等等。meter标记属性如表13-4所示。

表13-4 meter标记的属性、值及描述表

| 属性名 | 值 | 描述 |
|---------|---------|-----------------|
| form | form_id | 规定meter元素所属的表单。 |
| high | number | 规定被界定为高值的范围。 |
| low | number | 规定被界定为低值的范围。 |
| max | number | 规定范围的最大值。 |
| min | number | 规定范围的最小值。 |
| optimum | number | 规定度量的最优值。 |
| value | number | 必需。规定度量的当前值。 |

【例13-3-5】 progress 和meter 标记的应用

```
<!-- edu_13_3_5.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>HTML5页面元素progress和meter标记的应用</title>
</head>
<body>
<p><strong>文件下载进度：</strong>
<progress value="22" max="100">设置属性</progress></p>
<p><strong>空进度条：</strong><progress>未设置属性
</progress></p>
<p><strong>服务器CPU使用情况：</strong>
<meter value="0.3" high="0.9" low="0.1"
optimum="0.5">3/10</meter></p>
<p><strong>内存使用情况：</strong><meter value="0.6"
max="1" min="0" optimum=".75">60%</meter></p>
<p><mark>注释：</mark>IE9以及更早的版本不支持progress
meter 标记。</p>
</body>
</html>
```



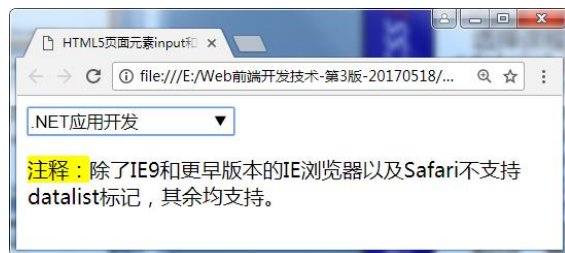
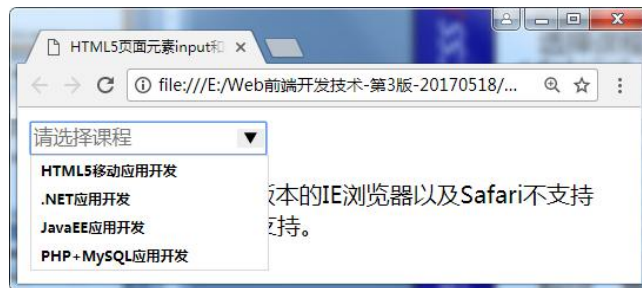
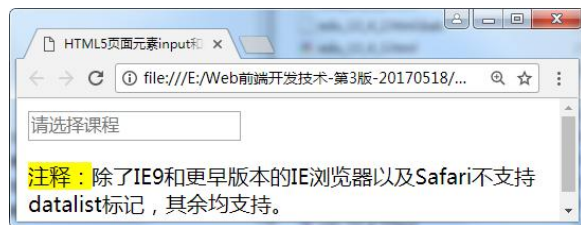
13.3.6 input 标记与datalist 标记

- input 标记用于搜集用户信息。详细介绍请参见第12章，此处仅介绍通过input 标记的list 属性与datalist 标记的id 属性进行关联，即将此两个属性的值设置为相同的值，通过datalist 标记列出所有合法的输入值列表。
- 选项列表datalist 标记用来定义input 标记可能的选项列表。一般与input 标记配合使用，主要用来定义input 可能的值，提供“自动完成”的功能，方便用户输入。datalist 标记及其选项不会被显示出来，只有当用户鼠标盘旋在input 标记域时，才能看到“▼”，然后单击“▼”弹出一个下拉列表，提供用户选择作为用户的输入数据。
- 【例13-3-6】input和datalist标记的应用。

```
<!-- edu_13_3_6.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
```

【例13-3-6】input和datalist标记的应用

```
<title>HTML5页面元素input和datalist标记的应  
用</title>  
</head>  
<body>  
<input list="courese" placeholder="请选择  
课程" />  
<datalist id="courese">  
<option value="HTML5移动应用开发">  
<option value=".NET应用开发">  
<option value="JavaEE应用开发">  
<option value="PHP+MySQL应用开发">  
</datalist>  
<p><mark>注释: </mark>除了IE9和更早版本  
的IE浏览器以及Safari不支持datalist标记, 其余  
均支持。</p>  
</body>  
</html>
```



13.4 HTML5 表单

表单是HTML 中获取用户输入的手段，HTML5 对表单系统做了彻底的改造，以适应当前的应用。

在HTML5 中增加了从用户收集特定类型数据的新方法和在浏览器中检查数据的能力，但在使用有些新增特性前最好先检查一下浏览器的支持情况。

下面从表单新增属性、表单新增元素及表单新增类型等方面分别进行介绍。

13.4.1 HTML5 新增的表单属性

HTML5表单新增一些新属性。这些属性是autocomplete、novalidate。

- form标记的新属性

- 1) autocomplete属性

autocomplete: on|off。属性规定form标记或类型为text、search、url、telephone、email、password、date pickers、range、color的input标记是否具有自动完成的功能。当表单元素设置了自动完成功能后，会记录用户输入过的内容，双击表单元素会显示历史输入。

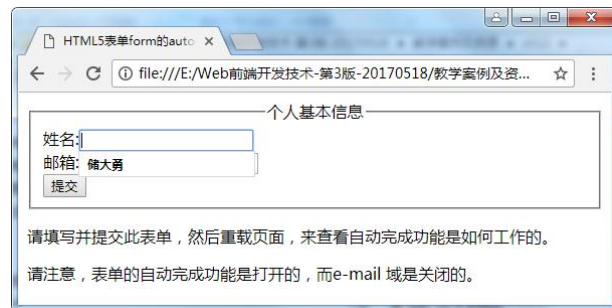
- 2) novalidate属性

novalidate: true|false。属性规定在提交表单时不进行验证form或类型为text、search、url、telephone、email、password、date pickers、range、color的input标记。

【例13-4-1】表单属性autocomplete和novalidate的应用

```
<!-- edu_17_4_1.html -->
<!doctype html>
<html lang="en">
<head><meta charset="UTF-8">
<title>HTML5表单form的autocomplete和novalidate属性的应用</title>
<script type="text/javascript"
src="html5shiv.min.js"></script>
</head>
```

```
<body>
<form action="" method="get" novalidate="novalidate" autocomplete="on">
<fieldset>
<legend align="center">个人基本信息</legend>
姓名:<input type="text" name="name" /><br/>
邮箱:<input type="email" name="email" autocomplete="off" /><br/>
<input type="submit" value="提交" />
</fieldset></form>
<p>请填写并提交此表单，然后重载页面，来查看自动完成功能是如何工作的。</p>
<p>请注意，表单的自动完成功能是打开的，而e-mail 域是关闭的。</p>
</body></html>
```



13.4.1 HTML5 新增的表单属性

2. input 标记的新属性

- height 和 width 属性。height 和 width 属性规定只适用于 image 类型的 input 标记的图像高度和宽度。
- form 属性。form 属性规定输入域所属的一个或多个表单。form 属性必须引用所属表单的 id。
- list 属性。list 属性规定输入域的 datalist。datalist 标记是输入域的选项列表。list 属性适用于类型为 text、search、url、telephone、email、password、date pickers、range、color 的 input 标记。
- placeholder 属性。

placeholder 属性提供一种提示，描述输入域所期待的值。当用户将

13.4.1 HTML5 新增的表单属性

鼠标盘旋在该域上时，单击“▼”会弹出下拉列表选项，简短的提示在用户输入值前会显示在输入域上。方便用户快速选择输入。该属性支持类型为text、search、url、telephone、email、password的input标记。

- autofocus属性。autofocus属性规定在页面加载时，该域自动地获得焦点。该属性适用于所有input标记的类型。

【例13-4-2】 input 标记的新增部分属性的应用。

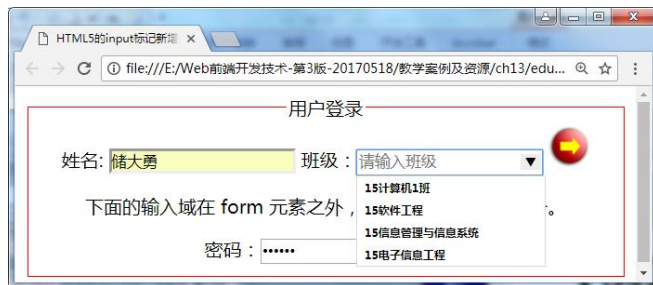
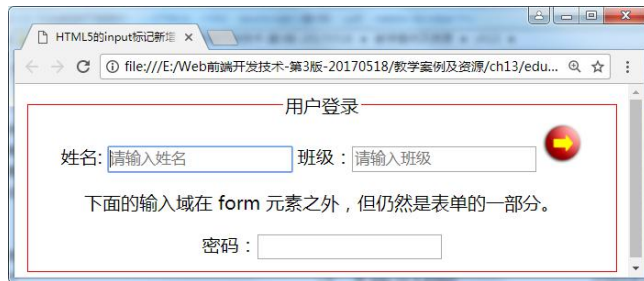
```
<!-- edu_13_4_2.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>HTML5的input标记新增部分属性的应用</title>
</head> <body>
<fieldset style="text-align:center;border:1px solid red;">
```

【例13-4-2】input 标记的新增部分属性的应用

```

<legend align="center">用户登录</legend>
<form name="myform" action="" method="get">
姓名: <input type="text" name="name" placeholder="请输入姓名" autofocus="autofocus" />
班级: <input type="text" name="class" placeholder="请输入班级" list="class_list"/>
<datalist id="class_list">
<option value="15计算机1班">
<option value="15软件工程">
<option value="15信息管理与信息系统">
<option value="15电子信息工程">
</datalist>
<input type="image" src="eg_submit.jpg" width="35" height="35"/>
</form>
<p>下面的输入域在 form 元素之外，但仍然是表单的一部分。
</p>
密码: <input type="password" name="user_key" form="myform">
</fieldset>
</body> </html>

```



13.4.1 HTML5 新增的表单属性

- required 属性。required 属性规定必须在提交之前填写输入域（不能为空）。required 属性适用于类型为text、search、url、telephone、email、password、date pickers、number、checkbox、radio、file 的input 标记。属性设置方法为required= "required" 或直接使用该属性required。
- min、max 和step 属性。min、max 和step 属性用于为包含数字或日期的input 类型规定限定（约束）。其中max属性规定输入域所允许的最大值。min 属性规定输入域所允许的最小值。step 属性为输入域规定合法的数字间隔，例如step= "5" ，则合法的数是 -5、0、5、10 等。该组属性适用类型为 date pickers、number、range 的input 标记。

13.4.1 HTML5 新增的表单属性

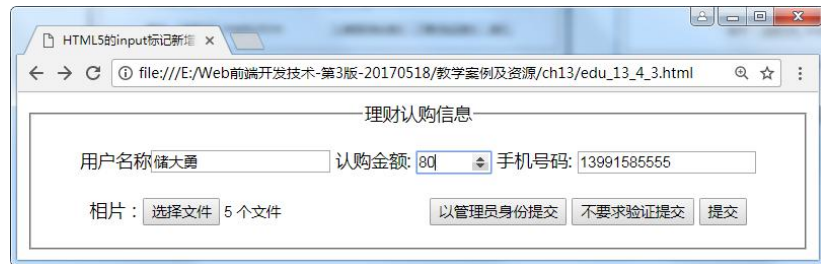
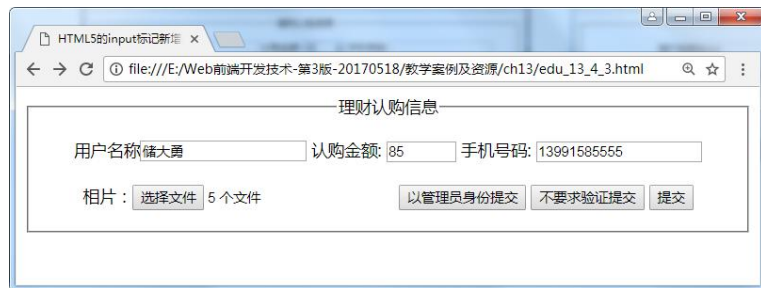
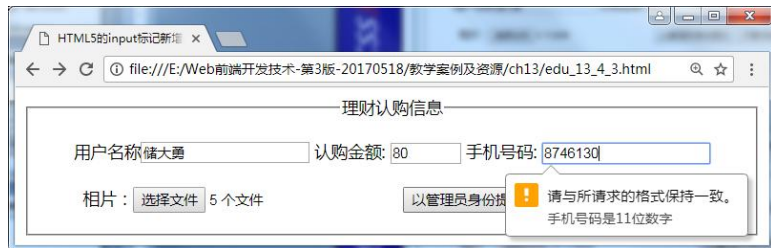
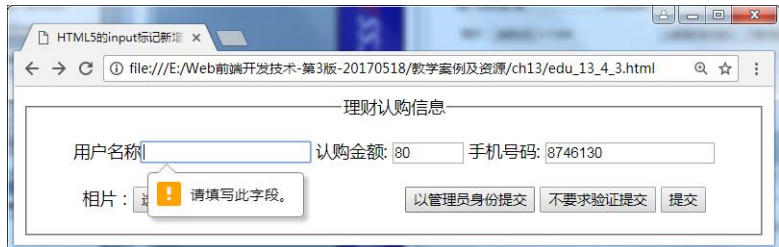
- multiple 属性。规定输入域中可选择多个值。适用于类型为email、file 的input 标记。
- form overrides 表单重写属性 (form override attributes) 允许重写form 元素的某些属性设定。这些重写属性分别是重写表单的action 属性formaction、重写表单的enctype 属性formenctype、重写表单的method 属性formmethod、重写表单的novalidate 属性formnovalidate、重写表单的target 属性formtarget。表单重写属性适用于类型为submit 和image 的input 标记。
- pattern属性（一般为正则表达式）。规定用于验证input域的模式。适用于text、search、url、tel、email、password等类型的input标记。

【例13-4-3】input 标记的新增其他属性的应用

```
<!-- edu_13_4_3.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>HTML5的input标记新增部分属性的应用</title>
</head>
<body>
<form action="" method="get">
<fieldset style="text-align:center;padding:20px;">
<legend align="center">理财认购信息</legend>
用户名称<input type="text" name="username" required><!-- 不能为空 -->
认购金额: <input type="number" name="money" min="5" max="100" step="5"/>
手机号码: <input type="text" name="phone" pattern="^1[3|4|5|8][0-9]{8}$" title="
手机号码是11位数字" required /><br/><br/><!-- 不能为空且必须为11位数字 -->
<label>相片: </label><input type="file" multiple="multiple"/><!-- 支持多选 -->
<input type="submit" formaction="admin.asp" value="以管理员身份提交" /><!-- 重写action -
->
<input type="submit" formnovalidate="true" value="不要求验证提交" /><!-- 重写novalidate -
->
<input type="submit" value="提交" />
```

【例13-4-3】input 标记的新增其他属性的应用

```
</fieldset>
</form>
</body>
</html>
```



13.4.2 HTML5 新增的表单元素

HTML5 新增output、keygen、datalist 等表单元素，其功能描述如表13-5 所示。

表13-5 HTML5 新增表单元素

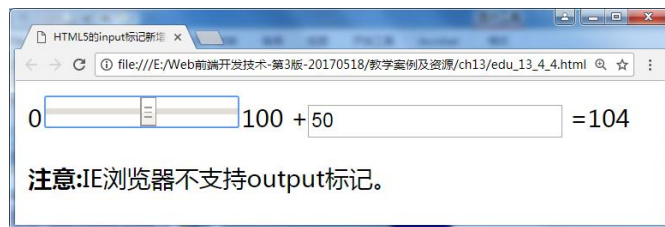
| 标记名称 | 标记功能描述 |
|------------------------|--------------------------------------|
| <output> </output> | 定义不同类型的输出，比如脚本的输出。 |
| <keygen> </keygen> | 规定用于表单的密钥对生成器字段。 |
| <datalist> </datalist> | 定义选项列表。与input元素配合使用该元素，来定义input可能的值。 |

1. output 标记

output 标记定义不同类型的输出。该标记有for、form、name 三个属性。for 属性用于描述计算中使用的元素与计算结果之间的关系，其值为每一元素的id，多个id 之间用空格分隔。form 属性用于定义输入字段所属的一个或多个表单。name 属性用于定义对象的唯一名称（表单提交时使用）。

【例13-4-4】新增output 标记的应用

```
<!-- edu_13_4_4.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>HTML5的input标记新增部分属性的应用</title>
<script type="text/javascript"
src="html5shiv.js"></script>
</head>
<body>
<form
oninput="sum.value=parseInt(num1.value)+parseInt(n
um2.value)">
0<input type="range" id="num1" value="50" min="0"
max="100">100
+<input type="number" id="num2" value="50">
=<output name="sum" for="num1 num2"></output>
</form>
<p><strong>注意:</strong>IE浏览器不支持output标记。
</p>
</body>
</html>
```



13.4.2 HTML5 新增的表单元素

2. keygen 标记

- keygen 标记用来提供一种验证用户的可靠方法。keygen 元素是密钥对生成器 (key-pairgenerator) 。当提交表单时, 会生成两个键: 一个是私钥 (private key) , 一个公钥 (public key) 。
- 私钥存储于客户端, 公钥则被发送到服务器。公钥可用于之后验证用户的客户端证书 (clientcertificate) 。目前, 浏览器对此元素的糟糕的支持度不足以使其成为一种有用的安全标准。

3. datalist 标记

- datalist 标记规定了input 标记可能的选项列表。datalist 标记被用来为input 标记提供“自动完成”的特性。使用input 标记的list 属性来绑定datalist 元素 (list 属性=datalist 的id 属性值) 。

13.4.3 HTML5 新增的input 类型

HTML5 增加很多新的表单输入类型，分别为color、date pickers（日期选择器，包括date、month、week、datetime、time、datetime-local）、email、number、range、search、tel、url。目前所有的主流浏览器一般都支持新的input 类型，即使不被支持，仍可以显示为常规的文本域。

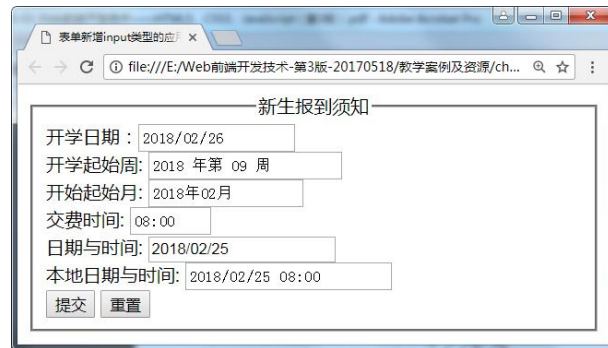
● Input 类型——Date Pickers（日期选择器）。

HTML5 提供多个可供选取日期和时间的新输入类型：

- (1) date——选取日、月、年。
- (2) month——选取月、年。
- (3) week——选取周和年。
- (4) time——选取时间（小时和分钟）。
- (5) datetime——选取时间、日、月、年（UTC 时间）。
- (6) datetime-local——选取时间、日、月、年（本地时间）。

【例13-4-6】表单日期选择器的应用

```
<!-- edu_13_4_6.html -->
<!DOCTYPE html>
<html><head>
<meta charset="UTF-8">
<title>表单新增input类型的应用</title>
</head>
<body><fieldset>
<legend align="center">新生报到须知</legend>
开学日期: <input type="date" /> <br />
开学起始周: <input type="week" name="user_date"
/> <br />
开始起始月: <input type="month"
name="user_date"/> <br />
交费时间: <input type="time" name="user_date"/> <br />
日期与时间: <input type="datetime"
name="user_date"/> <br />
本地日期与时间: <input type="datetime-local"
name="user_date" /> <br />
<input type="submit" value="提交" />
<input type="reset" />
</fieldset></body>
</html>
```



13.4.3 HTML5 新增的input 类型

- input 类型：color。
`<input type="color" name="favcolor"> <!-- 从取色器拾取颜色 -->`
- input 类型：tel。定义输入电话号码字段。
`<input type="tel" name="usrtel">`
- input 类型：email。email 类型用于包含e-mail 地址的输入域。在提交表单时，会自动验证email 域的值是否合法有效。
`<input type="email" name="useremail"> <!-- 自动验证邮箱格式 -->`
- input 类型：number。number 类型用于包含数值的输入域。此类型的input 标记常用属性如表13-6 所示。
`<input type="number" name="mynumber" min="0" max="100">`

表13-6 number 类型input 标记的属性及说明

13.4.3 HTML5 新增的input 类型

表13-6 number 类型input 标记的属性及说明

| 属性 | 描述 |
|-----------|-------------------|
| disabled | 规定输入字段是禁用的。 |
| max | 规定允许的最大值。 |
| maxlength | 规定输入字段的最大字符长度。 |
| min | 规定允许的最小值。 |
| pattern | 规定用于验证输入字段的模式。 |
| readonly | 规定输入字段的值无法修改(只读)。 |
| required | 规定输入字段的值是必需的。 |
| size | 规定输入字段中的可见字符数。 |
| step | 规定输入字段的合法数字间隔。 |
| value | 规定输入字段的默认值。 |

13.4.3 HTML5 新增的input 类型

- input 类型：range。range 类型用于包含一定范围内数字值的输入域。range 类型显示为滑动条。
`<input type="range" name="money" min="1" max="1000" step="5">`
- input 类型：search。search 类型用于搜索域，例如站点搜索或 Google 搜索。
`<input type="search" name="websidesearch">`
- input 类型：url。url 类型用于包含URL 地址的输入域。在提交表单时，会自动验证url 域的值。
`<input type="url" name="homepage">`

【例13-4-7】其他新增input 类型综合应用

```
<!-- edu 13 4 7.html -->
<!DOCTYPE html>
<html>
<head><meta charset="UTF-8">
<title>表单新增input类型的应用</title>
</head><body>
<fieldset style="width:500px;height:200px;padding:20px 50px;">
<legend align="center">新增其它input类型</legend>
<form method="post" action="">
设置颜色: <input type="text" name="color1" id="color1" readonly>
<input type="color" name="color2" oninput="color1.value=color2.value"> <br>
输入邮箱: <input type="email" name="useremail"> <br>
站内搜索: <input type="search" name="insidesearch"> <br>
电话号码: <input type="tel" name="usrtel"> <br>
个人主页: <input type="url" name="homepage"> <br>
年龄: <input type="range" name="age" min="1" max="120"
oninput="age_num.value=age.value "><output name="age_num" for="age"></output> <br>
期望薪酬: <input type="number" name="quantity" min="2500" max="10000" step="100"
value="2500"> <br><input type="submit" value="提交" />
<input type="reset" />
</form></fieldset></body></html>
```

【例13-4-7】其他新增input 类型综合应用

新增其它input类型

设置颜色：

输入邮箱：

站内搜索：

电话号码：

个人主页：

年龄： 85

期望薪酬：

13.5 HTML5 视频与音频

大多数商业网站都喜欢采用视频来宣传自己的公司或推销自己的产品或服务。然而在HTML4.01 版本基础之前，只能通过相关插件（比如 Flash）来播放，而且所有浏览器不一定都有同样的插件，还需要安装其他插件才能实现。HTML5 提供了video 标记和audio 标记，很好地解决这一问题。

13.5.1 video 标记及属性

HTML5 规定了一种通过video 元素来包含视频的标准方法。Video 标记支持三种视频格式，分别为MP4、WebM、Ogg。

- Ogg：带有Theora 视频编码和Vorbis 音频编码的Ogg 文件。
- MPEG4：带有H.264 视频编码和AAC 音频编码的MPEG 4 文件。
- WebM：带有VP8 视频编码和Vorbis 音频编码的WebM 文件。

13.5.1 video 标记及属性

1. 基本语法

```
<video src="movie.ogg" width="320" height="240" controls="controls">  
    您的浏览器不支持 video 标记。  
</video>
```

2. 属性说明

width 和 height 属性：控制视频的尺寸。使用时需要设置视频的高度和宽度，便于视频播放。如果不设置宽度和高度，页面就会根据原始视频的大小而改变。**src** 属性：规定要播放的视频的url。**loop**：设置该属性，则当媒体文件完成播放后再次开始播放。**preload**：设置该属性，则视频在页面加载时进行加载，并预备播放。如果使用autoplay，则忽略该属性。该属性有三种值：auto（一旦页面加载，则开始加载音频/视频）、metadata（当页面加载后

13.5.1 video 标记及属性

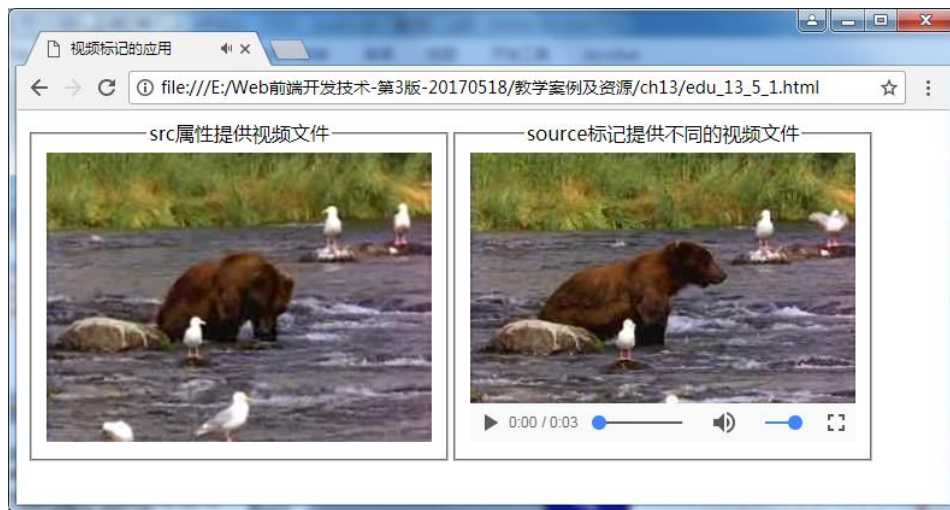
仅加载音频/视频的元数据)、none (页面加载后不应加载音频/视频)。格式如下:

`<video preload="auto|metadata|none">`

poster 属性: 用于在视频下载时显示的图像 (海报图片), 或者在用户点击播放按钮前显示的图像。如果未设置该属性, 则使用视频的第一帧来代替。赋值方法: `poster="url"`。如果浏览器不支持`<video>`标记, 就在`<video>`与`</video>`标记之间插入相关提示信息。**video 标记支持多个source 标记**。可以使用source 标记为video 标记和audio 标记提供多个不同的音频、视频文件, 以解决浏览器支持。如果浏览器支持将使用第一个可识别的格式。IE8 或者更早的IE 版本不支持video 标记。

13.5.1 video 标记及属性

```
<video width="320" height="240" controls="controls">  
  <source src="movie.ogg" type="video/ogg">  
  <source src="movie.mp4" type="video/mp4">  
  您的浏览器不支持 video 标记。  
</video>
```



【例13-5-1】 video 标记的应用

```
<!--edu 13 5 1.html-->
<!DOCTYPE html>
<html>
<head><meta charset="UTF-8">
<title>视频标记的应用</title>
</head>
<body>
<fieldset style="text-align:center;float:left;">
<legend>src属性提供视频文件</legend>
<video src="movie.ogg" poster="url" loop autoplay width="320" height="240"
controls="controls">您的浏览器不支持 video标记。</video>
</fieldset>
<fieldset style="text-align:center;float:left;">
<legend>source标记提供不同的视频文件</legend>
<video width="320" height="240" controls="controls">
<source src="movie.ogg" type="video/ogg">
<source src="movie.mp4" type="video/mp4">
您的浏览器不支持 video标记。
</video>
</fieldset>
</body>
</html>
```

13.5.2 audio 标记及属性

HTML5 使用audio 标记能够播放声音文件或者音频流。同样可以使用source 标记给audio 标记提供不同格式的音频文件，浏览器将使用第一个支持的音频文件。

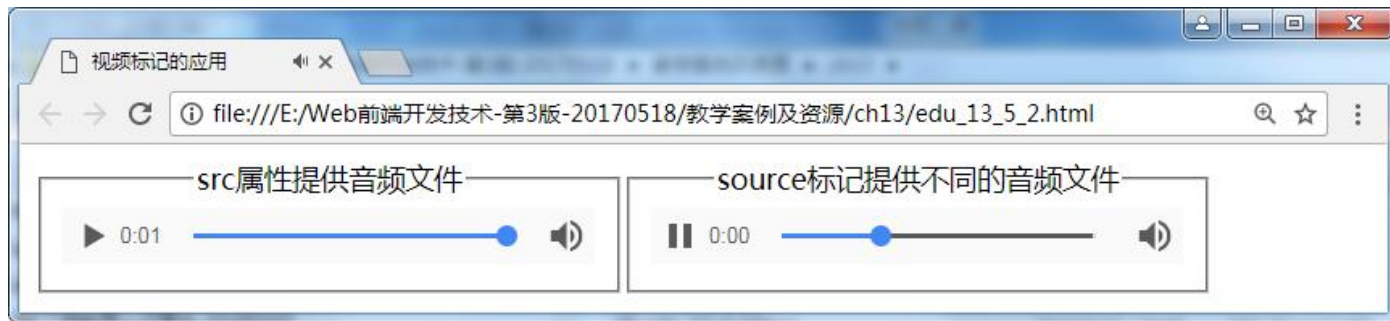
```
<audio width="320" height="240" controls="controls">
```

```
<source src="horse.ogg" type="audio/ogg">
```

```
<source src="horse.mp3" type="audio/mpeg">
```

您的浏览器不支持 video 标记。

```
</audio >
```



【例13-5-2】 audio 标记的应用

```
<!--edu_13_5_2.html-->
<!DOCTYPE html>
<html>
<head><meta charset="UTF-8">
<title>视频标记的应用</title>
</head>
<body>
<fieldset style="text-align:center;float:left;">
<legend>src属性提供音频文件</legend>
<audio src="horse.ogg" controls="controls">
您的浏览器不支持 audio标记（元素）。
</audio>
</fieldset>
<fieldset style="text-align:center;float:left;">
<legend>source标记提供不同的音频文件</legend>
<audio controls="controls">
<source src="horse.ogg" type="audio/ogg">
<source src="horse.mp3" type="audio/mpeg">
您的浏览器不支持 audio标记（元素）。
</audio>
</fieldset></body>
</html>
```

13.6 CSS3 基础应用

13.6.1 CSS3 新特性

- 为了满足Web UI 的开发需求，它提供了一系列强大的功能，如许多新的CSS 属性（文字、布局、颜色等）、各种CSS 特效、CSS 动画、元素的变换等。
- CSS3 被细分为许多“模块”。CSS2 中已经拆分成小块，又新增加了一些最重要的CSS3模块，分别为选择器、盒模型、背景和边框、文字特效、2D/3D 转换、动画、多列布局、用户界面。许多新的CSS3 属性已在目前主流的浏览器中得到应用。

13.6.2 CSS3 浏览器兼容性

1. 常用的浏览器属性前缀

为了让CSS样式能够满足不同浏览器版本的需要，需要在样式属性前面增加一些区分不同浏览器的前缀。

-webkit-：适用于webkit 核心浏览器，包含Safari、Chrome 等。

-moz-：适用于Firefox 浏览器等。

-ms-：适用于IE 浏览器。

-o-：适用于Opera 浏览器。

为了满足不同浏览器对CSS3新特性的支持，需要如下声明：

```
Div{-moz-animation: myfirst 5s;      /* Firefox */
  -webkit-animation: myfirst 5s;     /* Safari 和 Chrome */
  -o-animation: myfirst 5s;         /* Opera */
  -ms-animation: myfirst 5s;        /* IE */
  animation: myfirst 5s;            /* 标准属性写在最后，*/
}
```

13.6.2 CSS3 浏览器兼容性

2. CSS3 前缀解决方案

为了简化开发过程和相关的代码冗余问题，在页面中引入了-prefix-free 这个类库，可以自动帮助在CSS 中添加相关的浏览器特有的前缀属性。

-prefix-free 是一个JavaScript 工具库，用户再也不需要编写带有浏览器前缀的CSS代码，在需要的时候，-prefix-free 会自动帮助添加当前浏览器需要的前缀。引用方式如下：

```
<script  
src="http://cdn.gbtags.com/prefixfree/1.0.7/prefixfree.min.js"></script>  
<script  
src="http://leaverou.github.com/prefixfree/prefixfree.min.js"></script>  
http://leaverou.github.com/prefixfree/上直接下载到本地引用。如下：  
<script src="js/prefixfree.min.js"></script>
```


13.6.2 CSS3 浏览器兼容性

3. CSS 样式重置方案

Normalize.css 是一个很小的CSS 文件，但它在默认的HTML 元素样式上提供了跨浏览器的高度一致性。相比于传统的CSS reset，Normalize.css 是一种现代的、为HTML5 准备的优质替代方案。

用户可以从Github 下载Normalize.css，然后引用到页面中就可以。也可以在Normalize.css 源码的基础上重新编写，在必要的时候用自己写的CSS 覆盖默认值。

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

13.6.3 CSS3 边框

CSS3具有3个边框属性，如表13-7所示。

表13-7 CSS3边框属性及描述表

| 属性 | 描述 |
|---------------|-------------------------------|
| border-image | 设置所有border-image-*属性的复合属性。 |
| border-radius | 设置所有四个border-*-radius属性的复合属性。 |
| box-shadow | 向矩形方框添加一个或多个阴影。 |

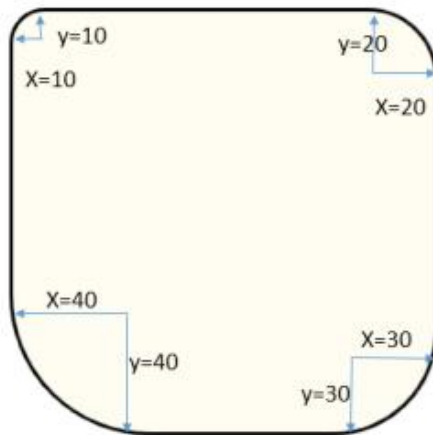
1.border-radius圆角边框

语法: **border-radius:** 水平半径(1-4个值)px|%/垂直半径(1-4个值)px|%;
border-radius:2em; /* 等同于下列4行定义 */
border-top-left-radius:2em; /* 定义左上角半径 */
border-top-right-radius:2em; /* 定义右上角半径 */
border-bottom-right-radius:2em; /* 定义右下角半径 */
border-bottom-left-radius:2em; /* 定义左下角半径 */

1.border-radius圆角边框

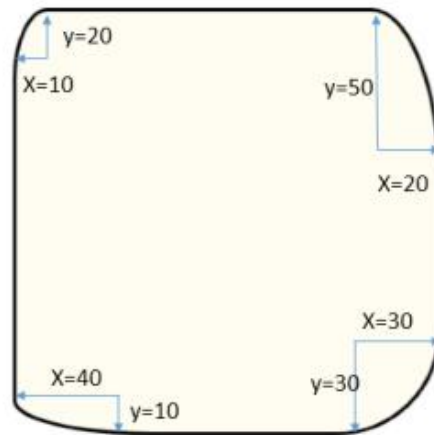


图13-36 圆角的半径表示图



border-radius:10px 20px
30px 40px;

图13-37 不同圆角的不同半径表示图



border-radius:10px 20px 30px
40px/20px 50px 30px 10px;

【例13-6-1】CSS3 圆角边框的应用

```
<!-- edu_13_6_1.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>CSS3边框的应用</title>
<style type="text/css">
div{float:left;width:120px;height:120px;margin:5
    0px 80px;background:#dadada;border:6px solid
    #00cc66;padding:10px;}
#div1{border-radius:25px;}
#div2{border-radius:25px 50px;}
#div3{border-radius:80px 100px 60px
    120px/50px 60px 70px 70px;}
</style>
</head>
<body>
<h3>CSS3圆角边框</h3> <hr>
<div id="div1" class="">
<p>半径均相同</p>
</div>
```

```
<div id="div2" class="">
<p>左、右对角的半径相同</p>
</div>
<div id="div3" class="">
<p>每个角水平与垂直半径不同</p>
</div>
</body>
</html>
```



2. box-shadow 边框阴影

box-shadow 边框阴影是复合属性，含有六个属性。

表13-8 box-shadow 属性值及说明

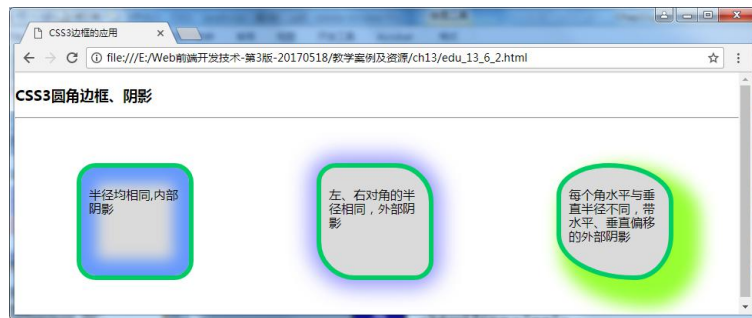
| 值 | 描述 |
|----------|---------------------------|
| h-shadow | 必需。水平阴影的位置。允许负值。 |
| v-shadow | 必需。垂直阴影的位置。允许负值。 |
| blur | 可选。模糊距离。 |
| spread | 可选。阴影的尺寸。 |
| color | 可选。阴影的颜色。请参阅 CSS 颜色值。 |
| inset | 可选。将外部阴影 (outset) 改为内部阴影。 |

语法: box-shadow: h-shadow v-shadow blur spread color inset;
box-shadow: 0 0 30px 20px #6699ff inset; /*内部阴影*/
box-shadow: 0px 0px 45px 10px #9999ff; /*外部阴影*/
box-shadow: 20px 20px 35px 15px #99ff33; /*外部阴影*/

【例13-6-2】 CSS3 边框阴影的应用

```
<!-- edu_13_6_2.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>CSS3边框的应用</title>
<style type="text/css">
div{float:left;width:120px;height:120px;margin:50
px 80px;background:#dadada;border:6px solid
#00cc66;padding:10px; }
#div1{border-radius:25px;box-shadow: 0 0 30px
20px #6699ff inset;}
#div2{border-radius:25px 50px;box-shadow:0px
0px 45px 10px #9999ff;}
#div3{border-radius:80px 100px 60px
120px/50px 60px 70px 70px;
box-shadow: 20px 20px 35px 15px #99ff33;}
</style>
</head>
<body>
<h3>CSS3圆角边框、阴影</h3><hr>
```

```
<div id="div1" class=""><p>半径均相同,内
部阴影</p></div>
<div id="div2" class=""><p>左、右对角的
半径相同,外部阴影</p></div>
<div id="div3" class=""><p>每个角水平与
垂直半径不同,带水平、垂直偏移的外部阴影
</p></div>
</body>
</html>
```



3. border-image 边框图像

border-image 属性创建带有图像的边框，三个参数分别是图像、剪裁位置、重复性。该属性有五个子属性。语法如下：

```
border-image: border-image-source  
border-image-slice/border-image-width/ border-image-outset border-  
image-repeat;  
border-image: url( "border.png" ) 27 27 27 27 fill/27 27 27 27/27px 27px  
27px 27px repeat; /* 剪裁和宽度不需要单位，偏移量需要单位。fill表示可选项  
，指定中间第九块为非透明块。*/
```

表13-9 border-image 属性值及说明

| 值 | 描述 |
|---------------------|----------------------------|
| border-image-source | 规定边框中图像的路径。 |
| border-image-slice | 规定图像边框向内偏移，可以是数字或百分比。 |
| border-image-width | 规定图像边框的宽度。 |
| border-image-outset | 规定边框图像区域超出边框的量。 |
| border-image-repeat | 规定图像边框是否应平铺(复制)、铺满(环绕)或拉伸。 |

3. border-image 边框图像

- border-image-source 属性（边框图像）。

默认无边框图像，如果设置边框图像，则使用绝对或相对url 地址指定边框图像。

border-image-source: none |url(图像文件);

border-image-source:url("border.png");

- border-image-slice 属性（图像切片/剪裁）。

该属性规定图像边框向内偏移，可以是数字或百分比。可以1 ~ 4 个值，类似于padding属性的设置方法。语法如下：

border-image-slice: number |% |fill;

border-image-slice:27 27 27 27; /* 边框图像切9块，每个角为27px*27px*/

- border-image-repeat 属性（边框图像重复）。

该属性用于设置边框图像的重复方式。语法如下：

3. border-image 边框图像

border-image-repeat: stretch | round | repeat

该属性值有三种取值，分别为stretch（拉伸）、round（环绕）、repeat（复制）。默认值为stretch。stretch表示拉伸图像来填充区域；repeat表示直接用图像来填充区域，填充时图像可能有残缺；round与repeat效果类似，如果无法完整平铺所有图像，则对图像进行缩放以适应区域。

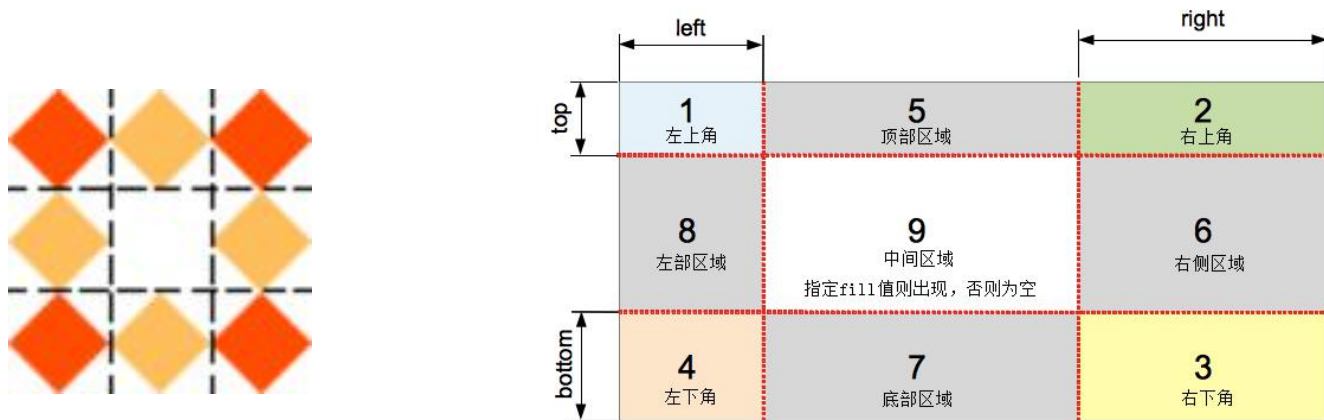


图13-40 W3C指定的81px×81px位图及九宫格分割法

3. border-image 边框图像

- border-image-width 属性（边框图像宽度）。

border-image-width: number | %; /* 可以有1~4个值*/

border-image-width: 27px 1 10% 27px;

/* 边框图像宽度设置为top:27px,right:1倍, bottom:10%,left:27px */

- border-image-outset 属性（图像外凸）

border-image-outset: length | number | percentage | auto; /* 1~4个值 */

例如，设置div 的类样式如下，边框图像不向外凸出。

.box{width: 200px; height: 50px;

border: 54px solid red; /* 边框宽度54px */

border-image: url("border.png") 27/27px round; /*边框图像高度与宽度均为27px */

/*边框图像向外凸出10px*/

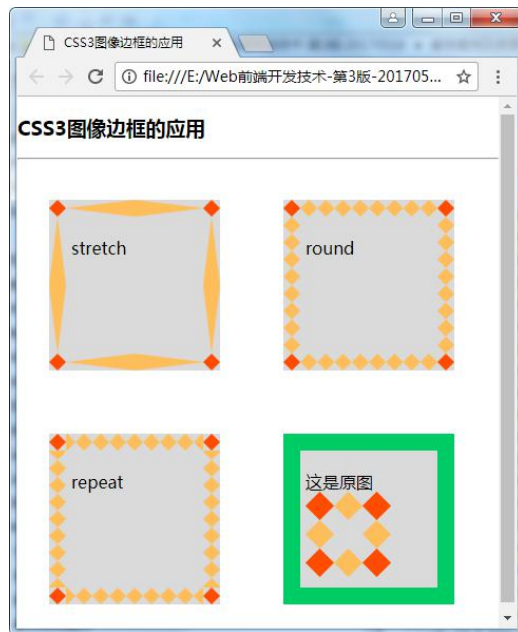
border-image: url("border.png") 27/15px/10px round; /*指定边框背景图像宽度为15px、偏移量为10px */

}

【例13-6-3】 CSS3 图像边框的应用

```
<!-- edu_13_6_3.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>CSS3图像边框的应用</title>
<style type="text/css">
div{float:left;width:120px;height:120px;margin:30px
  30px;background:#dadada;border:1em solid
  #00cc66;padding:5px;  }
#div1{border-image:url("border.png") 27 27 stretch;}
#div2{border-image:url("border.png") 27 27 round;}
#div3{border-image:url("border.png") 27 27 repeat;}
</style>
</head>
<body>
<h3>CSS3图像边框的应用</h3><hr>
<div id="div1" class=""><p>stretch</p></div>
<div id="div2" class=""><p> round</p></div>
<div id="div3" class=""><p>repeat</p></div>
<div id="div4" class="">
```

```
<p>这是原图</p>
</div>
</body></html>
```



13.6.4 CSS3 转换transform 属性

1. CSS3 2D 转换

CSS3 2D 转换常用方法有translate()、rotate()、scale()、skew()、matrix()。

- 位移translate(x,y)

translate(x,y)方法的作用是将元素从当前位置根据给定的x 轴坐标和y 轴坐标进行移动。x 表示left，父元素的左边界；y 表示top，父元素的上边界。translate()方法还提供根据单一轴移动的方法，分别是translateX()和translateY()。使用方法如下：

```
transform:translate(50px,50px); /* 向右移动50px， 向下移动50px */  
transform:translate(50px,0); /* 向右移动50px */  
transform:translateX(50px); /* 向右移动50px */  
transform:translate(0,50px); /* 向下移动50px */  
transform:translateY(50px); /* 向下移动50px */
```

13.6.4 CSS3 转换transform 属性

● 旋转rotate(deg)。

可以对元素旋转给定的角度，正值为顺时针，负值为逆时针。

```
transform:rotate(deg);      /* 基本语法 */  
transform:rotate(10deg);    /* 旋转10° */  
transform:rotate(120deg);   /* 旋转120° */
```

【例13-6-4】CSS3 位移与旋转的应用。



```
<!-- edu_13_6_4.html -->  
<!doctype html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<title>CSS3 2D转换-位移与旋转</title>  
<script type="text/javascript"  
src="html5shiv.js"></script>  
<link rel="stylesheet" href="css/normalize.css"  
type="text/css">
```

【例13-6-4】CSS3 位移与旋转的应用

```
<style type="text/css">
div{width:180px;height:50px;background:#dadada;border:
  1px solid #00cc66;}
#div1{transform:translate(50px,50px);} /* 位移 */
#div2{transform:rotate(30deg);} /* 旋转 */
#div3{transform:rotate(120deg);} /* 旋转 */
td{text-align:left;vertical-align:top;}
</style>
</head>
<body>
<h3>CSS3 2D转换-位移与旋转</h3> <hr>
<table border="1px" bordercolor="red" width="750px"
  height="200px">
<tr>
<td>
<div id="" class=""><p>这是原div</p></div>
<div id="div1" class=""><p>这个div向右移动50px, 向下
  移动50px</p></div>
</td>
<td>
<div id="" class=""><p>这是原div</p></div>
```

```
<div id="div2" class="">
<p>这个div旋转30度</p>
</div>
</td>
<td>
<div id="" class=""><p>这是原
div</p></div>
<div id="div3" class=""><p>这个div旋
转120度</p></div>
</td>
</tr>
</table>
</body>
</html>
```

13.6.4 CSS3 转换transform 属性

- 缩放scale(x, y)。

scale(x,y)方法的作用是缩放指定的元素，参数x 表示元素宽度的缩放倍数，参数y 表示元素高度的缩放倍数。scale 方法也可以接受负值，当参数x 为负值时，元素内容会横向倒置；当参数y 为负值时，元素内容会纵向倒置。

```
transform:scale(x,y);
```

```
transform:scale(1,4);
```

```
transform:scale(2,2);
```

- 扭曲skew(deg, deg)。

skew(x,y)方法的作用是将元素翻转（扭曲）给定的角度，参数x、y 分别表示围绕x轴翻转给定的角度、围绕y 轴翻转给定的角度。

13.6.4 CSS3 转换transform 属性

```
transform: skew(deg, deg);
```

```
transform: skew(30deg, 30deg); /*围绕x轴翻转30°, 围绕y轴翻转30° */
```

```
transform: skew(15deg, 65deg); /*围绕x轴翻转15°, 围绕y轴翻转65° */
```

- 综合转换matrix(n,n,n,n,n,n)。

matrix()方法和2D 变换方法合并成一个。matrix()方法是一个综合性的方法，它综合了上述的移动、旋转、缩放等功能。matrix()方法有六个参数，包含旋转、缩放、移动（平移）和倾斜功能。语法如下，参数的作用如下：

```
transform:matrix(scaleX, skewX, skewY, scaleY, translateX, translateY);/* 基本语法*/
```

```
transform:matrix(0.866,0.5,-0.5,0.866,20,20); /* x轴、y轴缩放0.866;x轴、y轴扭曲0.5和-0.5;x轴、y轴位移20px */
```


【例13-6-5】CSS3 缩放、扭曲、矩阵综合应用

```
<!-- edu_13_6_5.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>CSS3 2D转换-扭曲、缩放</title>
<script type="text/javascript" src="html5shiv.js"></script>
<link rel="stylesheet" href="css/normalize.css" type="text/css">
<script type="text/javascript" src="js/prefixfree.min.js"></script>
<style type="text/css">
    div{width:100px;height:50px;background:#dadada;border:1px solid #00cc66;        }
    #div1{transform:scale(1.5,1.5);margin:10px auto;}
    #div2{transform:skew(30deg,30deg);margin:10px auto;}
    #div3{transform:matrix(0.866,0.5,-0.5,0.866,20,20);
        /* x轴、y轴缩放0.866;x轴、y轴扭曲0.5和-0.5;x轴、y轴位移20px*/}
    td{text-align:left;vertical-align:top;}
</style>
</head>
<body>
<h3>CSS3 2D转换-缩放、扭曲、矩阵</h3><hr>
<table border="1px" bordercolor="red" width="750px" height="200px">
```

【例13-6-5】CSS3 缩放、扭曲、矩阵综合应用

```
<tr>
<td>
<div id="" class=""><p>这是原div</p></div>
<div id="div1" class=""><p>这个div缩放1.5倍
  </p></div>
</td>
<td>
<div id="" class=""><p>这是原div</p></div>
<div id="div2" class=""><p>这个div扭曲方法
  </p></div>
</td>
<td>
<div id="" class=""><p>这是原div</p></div>
<div id="div3" class=""><p>这是div采用matrix
  方法</p></div>
</td>
</tr>
</table>
</body>
</html>
```



13.6.4 CSS3 转换transform 属性

2. CSS3 3D 转换

CSS3 可以使用3D 转换来对元素进行格式化。常用的3D 转换方法有rotateX()、rotateY()。

- 旋转rotateX()方法。

通过rotateX()方法，元素围绕其X 轴以给定的度数进行旋转。

- 旋转rotateY()方法。

通过rotateY()方法，元素围绕其Y 轴以给定的度数进行旋转。

```
transform: rotateX(angle); /* X轴方向旋转一定角度 */
```

```
transform: rotateY(angle); /* Y轴方向旋转一定角度 */
```

```
#div1{transform:rotateX(120deg);}
```

```
#div2{transform:rotateY(120deg);margin:10px auto;}
```

【例13-6-6】 CSS3 3D 旋转的应用

```
<!-- edu_13_6_6.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>CSS3 3D转换</title>
<script type="text/javascript" src="html5shiv.js"></script>
<link rel="stylesheet" href="css/normalize.css" type="text/css">
<script type="text/javascript" src="js/prefixfree.min.js"></script>
<style type="text/css">
    div{width:150px;height:80px;background:#dadada;border:1px solid #00cc66;    }
    #div1{transform:rotateX(120deg);}
    #div2{transform:rotateY(120deg);margin:10px auto;}
    td{text-align:left;vertical-align:top;}
</style>
</head>
<body>
<table border="1px" align="center" width="450px" height="200px">
<caption><h3>CSS3 3D转换</h3></caption>
<tr>
<td>
```

【例13-6-6】CSS3 3D 旋转的应用

```
<div id="" class="">
  <p>这是原div</p>
</div>
<div id="div1" class="">
  <p>沿X轴旋转这个div</p>
</div>
</td>
<td>
  <div id="" class="">
    <p>这是原div</p>
  </div>
  <div id="div2" class="">
    <p>沿Y轴旋转这个div</p>
  </div>
</td>
</tr>
</table>
</body>
</html>
```



13.6.5 CSS3 过渡transition 属性

- transition 属性是一个复合属性，它有四个过渡属性。语法如下：

transition: property duration timing-function delay;
transition: width 2s; /* 宽度上过渡2s */

Transition
属性的值
及描述表

| 值 | 描述 |
|----------------------------|--------------------|
| transition-property | 规定设置过渡效果的CSS属性的名称。 |
| transition-duration | 规定完成过渡效果需要多少秒或毫秒。 |
| transition-timing-function | 规定速度效果的速度曲线。 |
| transition-delay | 定义过渡效果何时开始。 |

Transition
-timing-
Function
的值及描述
表

| 值 | 描述 |
|-----------------------|--|
| linear | 规定以相同速度从开始至结束的过渡效果(cubic-bezier(0,0,1,1))。 |
| ease | 规定以慢速开始、变快、慢速结束的过渡效果。 类似于cubic-bezier(0.25,0.1,0.25,1)。 |
| ease-in | 规定以慢速开始的过渡效果(cubic-bezier(0.42,0,1,1))。 |
| ease-out | 规定以慢速结束的过渡效果(cubic-bezier(0,0,0.58,1))。 |
| ease-in-out | 规定以慢速开始和结束的过渡效果(cubic-bezier(0.42,0,0.58,1))。 |
| cubic-bezier(n,n,n,n) | 在cubic-bezier函数中定义自己的值。可能的值是0~1 之间。 |

Transition子属性设置语法

```
transition-property: none|all| property;  
transition-property: width; /* width属性上转场 */  
transition-duration: time;  
transition-duration: 3s;  
transition-timing-function: linear|ease|ease-in|ease-out|ease-in-out|  
cubicbezier(n,n,n,n);  
transition-timing-function: ease-in-out;  
transition-delay: time;  
transition-delay: 2s;
```

【例13-6-7】 CSS3 过渡与转换综合的应用

```
<!-- edu_13_6_7.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>CSS3 过渡</title>
<script type="text/javascript" src="html5shiv.js"></script>
<link rel="stylesheet" href="css/normalize.css" type="text/css">
<script type="text/javascript" src="js/prefixfree.min.js"></script>
<style>
div{width:100px;height:50px;background:#009999;color:white;font-weight:bold;
transition:width 2s,height 2s,transform 2s;/* 3个属性过渡 */}
#div1 {transition-timing-function: linear;}
#div2 {transition-timing-function: ease;}
#div3 {transition-timing-function: ease-in;}
#div4 {transition-timing-function: ease-out;}
#div5 {transition-timing-function: ease-in-out;}
div:hover{width:200px; height:100px;transform:rotate(60deg);
/* 盘旋时过渡+旋转 */}
</style>
</head>
```


【例13-6-7】 CSS3 过渡与转换综合的应用

```
<body>
<h3>CSS3 过渡transition与transform综合应用</h3><hr
  color="red">
<div id="div1" style="top:100px">linear</div>
<div id="div2" style="top:150px">ease</div>
<div id="div3" style="top:200px">ease-in</div>
<div id="div4" style="top:250px">ease-out</div>
<div id="div5" style="top:300px">ease-in-out</div>
<p>请把鼠标指针移动到红色的div元素上，就可以看到<mark>过渡和转换</mark>的效果。</p>
</body>
</html>
```



13.6.6 CSS3 动画animation

CSS3 动画是指元素从一种样式逐渐变化为另一种样式的效果。通过CSS3的@keyframes（关键帧）规则，可以创建动画，从而取代动画图片、Flash 动画以及JavaScript编写的动画。在@keyframes 中规定某项CSS 样式，就能创建由当前样式逐渐改为新样式的动画效果。

1. CSS3 动画animation 基本语法

animation 是一个复合属性，语法如下：

```
animation:animation-name|animation-duration|animation-timing-  
function|animation-delay | animation-iteration-count| animation-direction
```

CSS3 动画属性及描述表

| 属性 | 描述 |
|---------------------------|---|
| @keyframes | 规定动画。 |
| animation | 所有动画属性的复合属性，除了animation-play-state属性。 |
| animation-name | 规定@keyframes动画的名称。 |
| animation-duration | 规定动画完成一个周期所花费的秒或毫秒，默认是0。 |
| animation-timing-function | 规定动画的速度曲线，默认是ease，其它与transition-timing-function属性值相同。 |
| animation-delay | 规定动画何时开始，默认是0。 |
| animation-iteration-count | 规定动画被播放的次数n(值为1(默认)、infinite)。 |
| animation-direction | 规定动画是否在下一周期逆向地播放(值为normal(默认)、alternate)。 |
| animation-play-state | 规定动画是否正在运行或暂停，其值为running(默认)、paused。 |
| animation-fill-mode | 规定对象动画时间之外的状态（其值为None、forwards、Backwards、both）。 |

13.6.6 CSS3 动画animation

2. @keyframes 规则定义

- 采用@keyframes 规则创建动画，需要将它绑定到一个CSS 的选择器，否则动画不会有任何效果。定义至少以下两项CSS3 动画属性，即可将动画绑定到选择器：
规定动画的名称、规定动画的时长。

@keyframes 基本语法：

```
@keyframes myAnimation {  
    from {Properties:Properties value; }  
    Percentage {Properties:Properties value; }  
    to {Properties:Properties value; }  
}  
  
@keyframes myAnimation {  
    0% {Properties:Properties value; }  
    Percentage {Properties:Properties value; }  
    100% {Properties:Properties value; }  
}
```

13.6.6 CSS3 动画animation

3. @keyframes 规则的绑定

绑定动画名称（例如myAnimation）到某个元素(div)的样式上，并指定时长。格式如下：

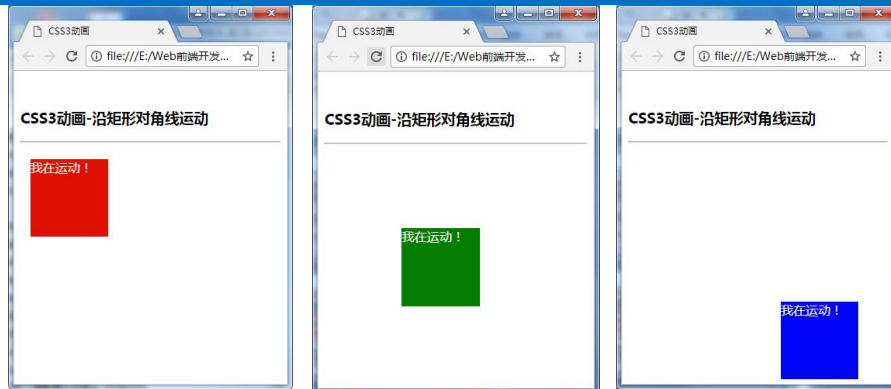
```
div{  
    /* 设置图层基本样式 */  
    width:100px;height:100px;background:red;position:relative;  
    /* 设置标准动画子属性 */  
    animation: myAnimation 8s;  
    -moz-animation: myAnimation 8s;      /* Firefox */  
    -webkit-animation: myAnimation 8s;   /* Safari 和 Chrome */  
    -o-animation: myAnimation 8s;       /* Opera */  
}
```

CSS3 动画animation子属性设置

```
div{
/* 设置图层基本样式 */
width:100px;height:100px;background:red;position:
relative;
/* 设置标准动画子属性 */
animation-name:myMOve;
animation-duration:5s;
animation-timing-function:linear;
animation-delay:2s;
animation-iteration-count:infinite;
animation-direction:alternate;
animation-play-state:running;
/* 仅以Safari and Chrome浏览器为例，其余类似。 */
-webkit-animation-name: myMOve;
-webkit-animation-duration:5s;
-webkit-animation-timing-function:linear;
-webkit-animation-delay:2s;
-webkit-animation-iteration-count:infinite;
-webkit-animation-direction:alternate;
-webkit-animation-play-state:running;
}
```

```
@keyframes myMOve{
/* 定义不同关键帧的样式 */
0% {background:red; left:0px;
top:0px;}
25% {background:yellow; left:200px;
top:0px;}
50% {background:blue; left:200px;
top:200px;}
75% {background:green; left:0px;
top:200px;}
100% {background:red; left:0px;
top:0px;}
}
```

【例13-6-8】CSS3 动画的应用



```
<!-- edu_13_6_8.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>CSS3动画</title>
<style>
div{width:100px;height:100px;background:red;position:relative; color:white;
animation:mymove 5s ;
```

```
-moz-animation:mymove 5s infinite;
-webkit-animation:mymove 5s infinite;
-o-animation:mymove 5s infinite;
}
@keyframes mymove{
from,0%
{left:0px;background:red;top:0px;}
50%
{left:100px;background:green;top:100px;}
to,100%{left:200px;background:blue;top:
200px;}
}
@-webkit-keyframes mymove /* Safari
与 Chrome */
{
from,0%
{left:0px;background:red;top:0px;}
50%
{left:100px;background:green;top:100px;}
```

【例13-6-8】 CSS3 动画的应用

```
    to,100%{left:200px;background:blue;top:200px;}
}
@-moz-keyframes mymove  /* Firefox */
{
    from,0% {left:0px;background:red;top:0px;}
    50%    {left:100px;background:green;top:100px;}
    to,100%{left:200px;background:blue;top:200px;}
}
@-o-keyframes mymove  /* Opera */
{
    from,0% {left:0px;background:red;top:0px;}
    50%    {left:100px;background:green;top:100px;}
    to,100%{left:200px;background:blue;top:200px;}
}
</style>
</head>
<body>
    <h3>CSS3动画-沿矩形对角线运动</h3><hr>
    <div>我在运动! </div>
</body>
</html>
```


13.6.7 CSS3 多列属性

使用CSS3 多列属性可以创建多个列来对文本进行布局，如同编辑报纸和杂志一样。常用的CSS3 多列属性主要有column-count、column-gap、column-rule 等。

表13-13 CSS3 多列属性值及说明

| 属性 | 描述 |
|-------------------|-------------------------------------|
| columns | 规定设置column-width和column-count的复合属性。 |
| column-count | 规定元素应该被分隔的列数。 |
| column-width | 规定列的宽度。 |
| column-fill | 规定如何填充列。 |
| column-gap | 规定列之间的间隔。 |
| column-rule | 设置所有column-rule-*属性的复合属性。 |
| column-rule-width | 规定列之间规则的宽度。 |
| column-rule-style | 规定列之间规则的样式。 |
| column-rule-color | 规定列之间规则的颜色。 |
| column-span | 规定元素应该横跨的列数。 |

13.6.7 CSS3 多列属性

基本语法

```
columns: column-width column-count; /* 复合属性*/  
column-count: number|auto  
column-width: auto|length;  
column-rule: column-rule-width column-rule-style column-rule-color;  
/* 复合属性*/  
column-rule-width: thin|medium|thick|length;  
column-rule-style: none|hidden|dotted|dashed|solid|double|groove|ridge|  
inset|outset;  
column-rule-color: color;  
column-gap: length|normal;  
column-fill: balance|auto; /* balance列长短平衡; auto列顺序填充*/
```

【例13-6-9】CSS3多列属性的应用

```
<!-- edu_13_6_9.html -->
<!doctype html>
<html lang="en">
  <head><meta charset="UTF-8">
  <title>CSS3多列属性的应用</title>
  <style>
    p{text-indent:2em;
    column-count:3; /* 设置列数 */
    column-gap:50px; /* 设置列间隙 */
    column-rule:4px outset #ff0000; /* 设置列宽度、线型、颜色 */
    }
    h2{
    column-span:all; /* 设置标题跨所有列 */
    text-align:center;background:#99ff99;
    height:40px;font-size:28px;padding:6px
    auto;}
  </style>
</head>
<body>
<h2>HTML 5 简介</h2>
```



【例13-6-9】CSS3多列属性的应用

<p>HTML标准自1999年12月发布的HTML4.01后，后继的HTML5和其它标准被束之高阁，为了推动Web标准化运动的发展，一些公司联合起来，成立了一个叫做 Web Hypertext Application Technology Working Group (Web超文本应用技术工作组 -WHATWG) 的组织。WHATWG 致力于 Web 表单和应用程序，而W3C (World Wide Web Consortium, 万维网联盟) 专注于 XHTML2.0。在 2006 年，双方决定进行合作，来创建一个新版本的 HTML。
HTML5草案的前身名为 Web Applications 1.0，于2004年被WHATWG提出，于2007年被W3C接纳，并成立了新的 HTML 工作团队。
HTML 5 的第一份正式草案已于2008年1月22日公布。HTML5 仍处于完善之中。然而，大部分现代浏览器已经具备了某些 HTML5 支持。
2012年12月17日，万维网联盟 (W3C) 正式宣布凝结了大量网络工作者心血的HTML5规范已经正式定稿。根据W3C的发言稿称：“HTML5是开放的Web网络平台的奠基石。”
2013年5月6日，HTML 5.1 正式草案公布。该规范定义了第五次重大版本，第一次要修订万维网的核心语言：超文本标记语言 (HTML)。在这个版本中，新功能不断推出，以帮助Web应用程序的作者，努力提高新元素互操作性。
本次草案的发布，从2012年12月27日至今，进行了多达近百项的修改，包括HTML和XHTML的标签，相关的API、Canvas等，同时HTML5的图像img标签及svg也进行了改进，性能得到进一步提升。
支持Html5的浏览器包括Firefox (火狐浏览器)，IE9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的能力。</p></body></html>

13.6.8 CSS3 文本效果

1. 文本阴影text-shadow 属性

1) 基本语法

text-shadow: h-shadow v-shadow blur color; //语法

text-shadow:2px 2px 8px #FF0000;//示例

2) 语法说明

text-shadow 属性向文本添加一个或多个阴影。该属性是空格分隔的阴影列表，其中h-shadow 定义水平阴影，允许负值，必需；v-shadow 定义垂直阴影，允许负值，必需；

blur 可选。模糊的距离。color 可选。阴影的颜色。省略的长度是0。

2. 文本换行text-wrap 属性

13.6.8 CSS3 文本效果

1) 基本语法

`text-wrap: normal|none|unrestricted|suppress;`

2) 语法说明

text-wrap: 文本换行规则。所有浏览器目前均不支持此属性。

3. 控制换行word-wrap 属性

1) 基本语法

`word-wrap: normal|break-word;`

2) 语法说明

word-wrap 自动换行属性允许强制文本进行换行，会对单词进行拆分。该属性有两个值：normal、break-word。其中normal表示只在允许的断字点换行（浏览器保持默认处理）；break-word表示在长单词或 URL 地址内部进行换行。

13.6.8 CSS3 文本效果

4. 文本溢出text-overflow 属性

1) 基本语法

`text-overflow: clip|ellipsis|string;`

2) 语法说明

text-overflow: 属性规定当文本溢出包含元素时发生的事情。该属性有三个属性值，分别为clip、ellipsis、string。其中clip 表示修剪文本；ellipsis 表示显示省略符号来代表被修剪的文本；string 表示使用给定的字符串来代表被修剪的文本。

【例13-6-10】CSS3 文本效果属性的应用



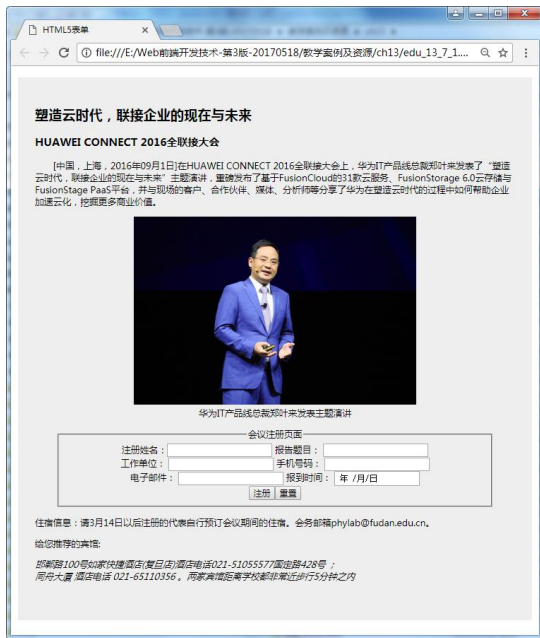
```
<!-- edu_13_6_10.html -->
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>CSS3文本效果</title>
<style>
h2{text-
align:center;background:#99ccff;padding:5px
auto;}
h1 {text-shadow:2px 2px 8px #FF0000;/* 设置文
本阴影 */}
p.test{width:15em; border:1px solid #000000;
word-wrap:break-word;/* 设置自动换行*/}
div.test{white-space:nowrap;/* 规定文本不进行换行
*/ width:12em;overflow:hidden; border:1px solid
#000000;}
</style>
</head>
<body>
<h2>CSS3文本效果</h2>
```


【例13-6-10】CSS3 文本效果属性的应用

```
<h1>具有模糊效果的文本阴影</h1>
<p>【未设置换行和宽度的段落】This paragraph contains a very long word:
thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next
line.</p>
<p class="test">【设置强制换行和宽度的段落】This paragraph contains a very long word:
thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next
line.</p>
<h3>下列div 包含长文本，都能正常显示</h3>
<div id="" class="">HTML 5 的第一份正式草案已于2008年1月22日公布。HTML5 仍处于完善之中
。然而，大部分现代浏览器已经具备了某些 HTML5 支持。</div>
<h3>下面两个div包含无法在框中容纳的长文本。不能完全显示，文本被修剪了。</h3><hr>
<p>下列div使用 "text-overflow:ellipsis" : </p>
<div class="test" style="text-overflow:ellipsis;">HTML 5 的第一份正式草案已于2008年1月22日
公布。HTML5 仍处于完善之中。然而，大部分现代浏览器已经具备了某些 HTML5 支持。</div>
<h3>下列div使用 "text-overflow:clip": </h3>
<div class="test" style="text-overflow:clip;">HTML 5 的第一份正式草案已于2008年1月22日公
布。HTML5 仍处于完善之中。然而，大部分现代浏览器已经具备了某些 HTML5 支持。</div>
</body>
</html>
```

13.7 综合实例

以“HUAWEI CONNECT 2016 全联接大会”的会议注册页面为例，采用HTML5 构建页面。



(1) 整个页面采用article 标记构架。使用header、footer、hgroup、figure、figcaption、form、fieldset 等标记来进行页面布局。

(2) 会议注册页面。采用fieldset、legend 进行表单元素分组。在其中分别采用input类型为text、email、tel、date、submit、reset 等来布局页面，注册姓名、报告题目、工作单位等文本输入域不能为空。

(3) footer 部分中宾馆信息采用address 标记进行布局。

详见案例：edu_13_7_1.html

综合实例代码

```
<!-- edu_13_7_1.html -->
<!doctype html>
<html lang="en">
<head><meta charset="UTF-8">
<title>HTML5表单</title>
<style type="text/css">
img:hover{transform:rotate(30deg);}
</style>
</head>
<body>
<article style="margin:20px
    auto;width:850px;height:900px;background:#eeeeee;padding:30px;">
<header>
<hgroup>
<h1>塑造云时代，联接企业的现在与未来</h1>
<h3>HUAWEI CONNECT 2016全联接大会</h3>
<p style="text-indent:2em;">[中国，上海，<time datetime="2016-09-01">2016年09月1日
    </time>]在HUAWEI CONNECT 2016全联接大会上，华为IT产品线总裁郑叶来发表了“塑造云时代
    ，联接企业的现在与未来”主题演讲，重磅发布了基于FusionCloud的31款云服务、FusionStorage
    6.0云存储与FusionStage PaaS平台，并与现场的客户、合作伙伴、媒体、分析师等分享了华为在塑造
    云时代的过程中如何帮助企业加速云化，挖掘更多商业价值。</p>
```

综合实例代码

```
<figure style="text-align:center;">
<br>
<figcaption>华为IT产品线总裁郑叶来发表主题演讲</figcaption>
</figure>
</hgroup>
</header>
<form method="post" action="">
<fieldset style="text-align:center;margin:10px 40px;">
<legend>会议注册页面</legend>
注册姓名: <input type="text" name="" required>
报告题目: <input type="text" name="" required><br>
工作单位: <input type="text" name="" required>
手机号码: <input type="tel"><br>
电子邮件: <input type="email" name="">
报到时间: <input type="date" /><br>
<input type="submit" value="注册"><input type="reset">
</fieldset>
</form>
<footer>
<p>住宿信息: 请3月14日以后注册的代表自行预订会议期间的住宿。会
务邮箱phylab@fudan.edu.cn。</p>
```

```
<p>给您推荐的宾
馆:<address>邯郸路100
号如家快捷酒店(复旦店)酒
店电话021-51055577国
定路428号; <br> 同舟
大厦 酒店电话 021-
65110356。两家宾馆距
离学校都非常近步行5分钟
之内</address></p>
</footer>
</article>
</body>
</html>
```

本章小结

本章介绍了HTML5 新特性和一些基础的HTML5 的应用。重点讲述HTML5 的新增属性、新增表单属性、新增表单的input 类型、媒体元素（视频、音频）等方面的知识和程序设计技巧。

HTML5 新增了header、nav、article、section、aside、footer 等结构元素,使用这些语义的标记构建网页更为方便、快捷。HTML5 新增的其他页面元素也极大地丰富了页面内容与表现，结合JavaScript 脚本能够设计具有更好的用户体验的网站。HTML5 技术在移动互联网时代会具有更加杰出的表现。

运用CSS3 新增转换、过渡和动画特性可以增强页面的表现效果。运用CSS3 多列属性、文本效果属性可以美化页面排版效果。