

# 解读 HTML5：建议、技巧和技术

---

标签: [HTML](#), [html5](#), 向后兼容

## Design

想总结一下 **HTML5** 的支持列表，写了许久，无果。在众多文章中，这篇文章涉及到了我想到的，以及没想到的。遂做粗略的翻译，因个人英文水平一般，本文 **FYI**。原文请看：[HTML5 Unleashed: Tips, Tricks and Techniques](#)。

=====

现今我们能用 **HTML5** 吗，能用它做些什么呢，是否真的是 **Flash** 杀手？想必你也注意到了这些日渐增长且常被问起的类似问题，被讨论着，甚至被回答过。在我看来，你必须自己回答这些基本的问题。



这篇文章的本意是想帮你通过一些基本指南的学习，以轻松学习代码模板。一旦你熟悉了这些基本，并想更进一步学习，你还将会找到更多提供了建议、技巧和技术的有效资源。

## HTML 5 特性

[官方详细的文档](#)是寻找 **HTML5** 特性的最好地方，当然你还可以轻松通过 **W3Schools** 来学习 **HTML5** 标签。我们将会在文章中涉及到以下的特性：

- 语义化标记
- **Form** 表单增强功能
- 视频 / 音频
- 画布（**Canvas**）
- 可编辑内容
- 拖放
- 稳健的数据存储

## 检测浏览器支持

在你开始尝试 **HTML5** 之前，需要知道各主流浏览器的支持状况。这些有用的资源，将可以帮助你向着正轨走：

- [何时能用](#)
- [网页设计师的浏览器支持列表](#)
- [HTML5 测试](#)
- [布局引擎对比](#)

你还可以运行 **Javascript** (用 **Javascript** 检测浏览器特性) 来检测 **HTML5** 特性的支持。你还应该用 **Modernizr**：一个非常不错的检测 **HTML5/CSS3** 本地支持的 **Javascript** 库。如果你选择用 **Mootools** 可以使用 **MooModernizr** (MooTools port of Modernizr)。

你可能也会想留意不断变化的“[浏览器市场份额分享](#)” — 这些信息对于你决定用何种解决或折衷的方法将会是非常必要的。

## 值得注意的变更点

除了新的特征，你还应该记下这些重要的变更点：

- **简洁的 DOCTYPE** **HTML5** 只需一个简洁的文档类型：`<!DOCTYPE html>`。它有意不使用版本，因此文档将会适用所有版本的 **HTML**。
- **简单易记的语言标签** 你并不需要在 `<html>` 中使用 `xmlns` 或 `xml:lang` 标记。`<html lang="en">` 将对 **HTML5** 有效。
- **简单易记的编码类型** 你现在可以在 `meta` 标签中使用 `"charset"`：`<meta charset="utf-8" />`
- **不需要闭合标签** 在 **HTML5** 中，空标签（如：`br`、`img` 和 `input`）并不需要闭合标签。
- **废弃的标签** 下面这些标签并不被 **HTML5** 支持：`<acronym>`、`<applet>`、`<basefont>`、`<big>`、`<center>`、`<dir>`、`<font>`、`<frame>`、`<frameset>`、`<noframes>`、`<s>`、`<strike>`、`<tt>`、`<u>` 和 `<xmp>`

简单代码示例：

```
<!doctype html>

<html lang="en">

<head>

    <meta charset="utf-8" />

    <title>HTML5 Document</title>

</head>

<body>
```

```
</body>
```

```
</html>
```

你可以使用 [HTML5 Validator](#) 或 [W3C Markup Validation Service](#) 来测试你的 HTML5 文档。

## 语义化标记

HTML5 新增的一些新标签除了不仅仅是更具语义的 `<div>` 标签的替代品，并不提供额外的功能。这些都是新增的标签：`<article>`、`<section>`、`<aside>`、`<hgroup>`、`<header>`、`<footer>`、`<nav>`、`<time>`、`<mark>`、`<figure>` 和 `<figcaption>`。

这些标签被除了 IE 外的所有现代浏览器（Firefox 3+、Safari 3.1+、Chrome 2+、and Opera 9.6+）支持。Javascript 提供了 `document.createElement(tagName)` 的方法，让你可以用来创建新的 HTML5 标签。代替自己创建这些元素，你还可以用 [HTML5 Enabling Script](#) 或 [IE Print Protector](#) — 这些脚本将帮助 IE 正常处理 HTML5 元素的渲染。

你可能会想到添加 CSS Reset 到这些新元素上。这里是一些可以用在你以 HTML5 为基础的项目的 CSS Reset:

- [HTML5 Reset CSS](#)
- [Reset5](#)

**简单代码示例:** 兼容 IE 的 HTML5 页面布局

```
<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8" />

  <title>HTML5 Semantic Markup Demo: Cross Browser</title>

  <link rel="stylesheet" href="html5reset.css" type="text/css" />

  <link rel="stylesheet" href="html5semanticmarkup.css" type="text/css" />

  <!--[if lt IE 9]>

    <script src="html5.js"></script>

  <![endif]-->
```

```
</head>

<body>

  <header>

    <hgroup>

      <h1>Page Header</h1>

      <h2>Page Sub Heading</h2>

    </hgroup>

  </header>

  <nav>

    <ul>

      <li><a href="#">Home</a></li>

      <li><a href="#">Projects</a></li>

      <li><a href="#">Portfolio</a></li>

      <li><a href="#">Profile</a></li>

      <li><a href="#">Contact</a></li>

    </ul>

  </nav>

  <article>

    <header>

      <h1>Article Heading</h1>

      <time datetime="2010-05-05" pubdate>May 5th, 2010</time>

    </header>

    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>

    <section>

      <header>

        <h1>Section Heading</h1>
```

```
</header>

<p>Ut sapien enim, porttitor id feugiat non, ultrices non odio.</p>

<footer>

    <p>Section Footer: Pellentesque volutpat, leo nec auctor
eismod</p>

</footer>

</section>

<section>

    <header>

        <h1>Section Heading</h1>

    </header>

    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>

    <figure>

        <figcaption>FigCaption: Club, Heart, Spade and
Diamond</figcaption>

    </figure>

    <p>Ut sapien enim, porttitor id feugiat non, ultrices non odio</p>

    <footer>

        <p>Section Footer: Pellentesque volutpat, leo nec auctor eismod
est.</p>

    </footer>

</section>

<footer>
```

```
        Article Footer

    </footer>

</article>

<aside>

    <header>

        <h1>Siderbar Heading</h1>

    </header>

    <p>Ut sapien enim, porttitor id feugiat non, ultrices non odio.</p>

</aside>

<footer>

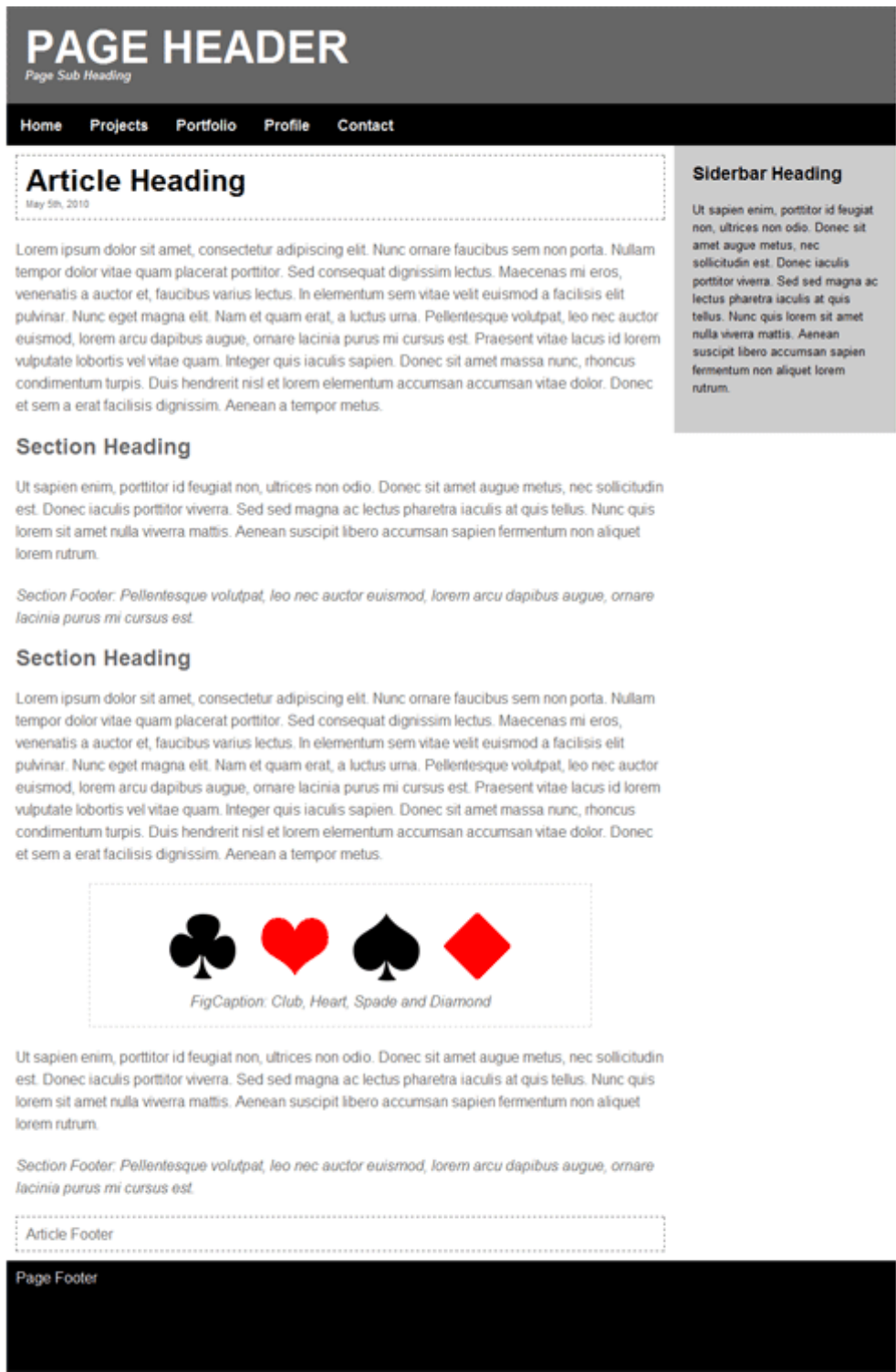
    Page Footer

</footer>

</body>

</html>
```

**注意：**没有一个 `div` 标签，`id` 或 `class` 属性被使用到 — 简洁、小巧和更语义化的标记 ([View Demo](#))。你仍可以用 [HTML5 Validator](#) 或 [W3C Markup Validation Service](#) 来检测你的 HTML5 文档。



**注意：**这个示例在 IE6 下并未正常显示。这只是因为我使用 CSS child combinators 来避免使用额外的 class。你可以在 IE6 下用自己的 CSS，像其他浏览器一样处理 HTML5 标记。

**其他资源**

- [HTML5 的 "articles" 和 "sections": 有什么区别?](#)
- [HTML5 中的结构标签](#)

- **Figure** 和 **caption** 元素
- 解读 **HTML5: <nav>**
- **<section>** 并不仅仅是 "语义化的 **<div>**"
- 用 **jQuery** 提供 **HTML5 <details>** 的完美折衷方法
- 不用脚本实现 **IE** 对 **HTML5** 样式控制

## 表单增强

**HTML5** 为表单提供了几个新的属性、**input** 类型和标签。到目前为止，只有 **Opera** 对 **HTML5** 有比较好的支持。你因此应该下载 **Opera** 来查看大部分新特性如何工作。

- 新的 **INPUT** 类型  
**color, email, date, month, week, time, datetime, datetime-local, number, range, search, tel,** 和 **url**
- 新属性  
**required, autofocus, pattern, list, autocomplete** 和 **placeholder**
- 新元素  
**<keygen>, <datalist>, <output>, <meter>** 和 **<progress>**

值得高兴的是尽管支持有限，使用这些特性也是个不错的选择。因为新的 **INPUT** 类型会漂亮地降级为 **TEXT** 类 **INPUT**（译注：**INPUT** 的默认 **type** 为 **text**）。且记住现今你仍可以折衷使用 **Javascript** 控制表单（这个技巧是指首先检测浏览器自身支持，然后才是使用折衷方法）。

**简单代码示例：**列出一些今天你可以拿来测试的新特性

```
<form>

  <fieldset>

    <legend>New Attributes</legend>

    <p>

      <label>Required:</label>

      <input type="text" name="html5required" required="true">

      <small>Works in Opera & Chrome</small>

    </p>

    <p>

      <label>AutoFocus:</label>

      <input type="text" name="html5autofocus" autofocus="true">

      <small>Works in Opera, Chrome & Safari</small>

    </p>

  </fieldset>

</form>
```



```
</p>
```

```
<p>
```

```
  <label>Placeholder:</label>
```

```
  <input type="text" name="html5placeholder" placeholder="This Will  
Show in WebKit">
```

```
  <small>Works in Chrome & Safari</small>
```

```
</p>
```

```
<p>
```

```
  <label>Input Pattern:</label>
```

```
  <input type="text" pattern="[0-9][A-Z]{3}" name="html5pattern"  
required title="Enter a digit followed by three uppercase letters"/>
```

```
  <small>Works in Opera & Chrome</small>
```

```
</p>
```

```
<p>
```

```
  <label>Multiple Files:</label>
```

```
  <input type="file" name="html5multiplefileupload" multiple>
```

```
  <small>Works in Chrome, Safari & Firefox</small>
```

```
</p>
```

```
<p>
```

```
  <label>List:</label>
```

```
  <input type="text" name="html5textwithdatalist" list="colors">
```

```
  <datalist id="colors">
```

```
    <option value="Red">
```

```
    <option value="Green">
```

```
    <option value="Blue">
```

```
  </datalist>
```

```
        <small>Works in Opera</small>

    </p>

</fieldset>

<fieldset>

    <legend>New Input Types</legend>

    <p>

        <label>Email:</label>

        <input type="email" name="html5email">

        <small>Works in Opera</small>

    </p>

    <p>

        <label>URL:</label>

        <input type="url" name="html5url">

        <small>Works in Opera</small>

    </p>

    <p>

        <label>Number:</label>

        <input type="number" name="html5number" min="1" max="10" step="1"
value="1">

        <small>Works in Opera</small>

    </p>

    <p>

        <label>Range:</label>

        <input type="range" name="html5range" min="-100" max="100" value="0"
step="10">
```

```
<small>Works in Opera, Chrome & Safari</small>

</p>

<p>

  <label>Time:</label>

  <input type="time" step="900" name="html5time">

  <small>Works in Opera</small>

</p>

<p>

  <label>Date:</label>

  <input type="date" name="html5date">

  <small>Works in Opera</small>

</p>

<p>

  <label>Month:</label>

  <input type="month" name="html5month">

  <small>Works in Opera</small>

</p>

<p>

  <label>Week:</label>

  <input type="week" name="html5week">

  <small>Works in Opera</small>

</p>

<p>

  <label>DateTime:</label>

  <input type="datetime" name="html5datetime">

  <small>Works in Opera</small>
```

```
</p>
```

```
</fieldset>
```

```
<div><button>Submit</button></div>
```

```
</form>
```

### New Attributes

Required:

*Works in Opera & Chrome*

AutoFocus:

*Works in Opera, Chrome & Safari*

Placeholder:

*Works in Chrome & Safari*

Input Pattern:

*Works in Opera & Chrome*

Multiple Files:

Choose...

*Works in Chrome, Safari & Firefox*

List:

*Works in Opera*

### New Input Types

Email:

*Works in Opera*

URL:

*Works in Opera*

Number:

*Works in Opera*

Range:

*Works in Opera, Chrome & Safari*

Time:

*Works in Opera*

Date:

Month:

Week	Mon	Tue	Wed	Thu	Fri	Sat	Sun
17	26	27	28	29	30	1	2
18	3	4	5	6	7	8	9
19	10	11	12	13	14	15	16
20	17	18	19	20	21	22	23
21	24	25	26	27	28	29	30
22	31	1	2	3	4	5	6

Today None

Week:

Date Time:

Submit

在我们的 DEMO 中出现的特性，都顺利地运行在 Opera 上，但你仍需要使用 Chrome 或 Safari 来看其他小部分功能的实际运行状态（[View Demo](#)）。

了解更多：

- [疯狂的表单](#)
- [互联网的未来：如何创建一个 HTML5 表单](#)
- [HTML5 Input 类型](#)
- [忙于 HTML5 表单的一天](#)
- [对 HTML5 表单的再次思考](#)
- [HTML5 表单来吧](#)

工具 & 资源列表：

- [利用 jQuery 和 Modernizr 实现跨浏览器 HTML5 Placeholder](#)
- [使用 jQuery Tool 实现跨浏览器 Date/Range/Validationns 类型 INPUT](#)
- [jQuery HTML5 文件上传：异步上传多文件](#)
- [有 jQuery 插件启动 Web Forms 2.0](#)
- [跨浏览器 Web Forms 2.0](#)

## 不需要插件的视频和音频

HTML5 对视频和音频特性规范文档的制定是被讨论最多的。除浏览器自带支持的明显好处外，评论点集中在浏览器提供商对音频/视频格式的不同选择。如果你准备使用 HTML5 的 `<audio>` 和 `<video>`，使你熟悉下面这些视频/音频的解码器和浏览器支持是非常重要的：

- **音频：**[ogg \(ogg, oga\)](#), [mp3](#), [wav](#), [AAC](#)
- **视频：**[ogg \(ogv\)](#), [H.264 \(mp4\)](#)

另外你还需要留意一下 Google 的 **VP8** 视频解码，这个将被作为一个开源格式来结束（格式选择的）纷争。HTML5 提供的一个解决方案是，让你可以指定多个不同格式的源文件，以便于用户浏览器选择它认识的文件。对于 `<IE9` 和旧浏览器，你将需要一个折衷的解决方案。

当你第一次尝试 HTML5 的音频/视频，你可能会想知道这些可能对你有帮助的东东：

- 你的服务器必须支持你打算提供的音频/视频的 **MIME** 格式。你将需要检查一下其在本地服务器上是否被支持。
- 如果你使用 Safari 来检测 HTML5 的音频/视频支持，需要安装 QuickTime Player，没有 QuickTime Player 的 Safari 什么都不能播放。

简单代码示例：音频标记 ([View Demo](#))

```
<audio controls>

  <source src="demo-audio.ogg" />

  <source src="demo-audio.mp3" />
```

```
</audio>
```



### 简单代码示例：视频标记 ([View Demo](#))

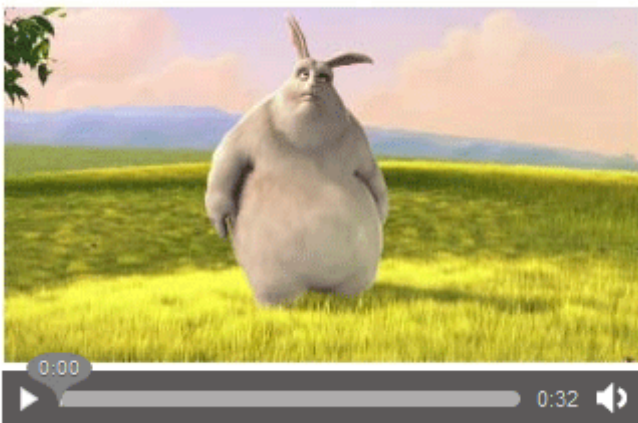
```
<video width="320" height="240" controls preload="none"
poster="videoframe.jpg">

  <source src="demo-video.mp4" type="video/mp4" />

  <source src="demo-video.ogv" type="video/ogg" />

</video>
```

译注：记得加上 **type**，不然，即使格式对了，也可能播放不了



尽管相对于要依赖第三方插件已经迈了一大步，但上面提供的示例，只是一个开始。由于各个浏览都提供了不同外观的控制栏，你可能会希望改变他们以适应你的设计。看看下面例子如何利用 **DOM API** 来创建一个属于你自己的自定义控制栏。

### 简单代码示例：拥有自定义控制栏的视频 ([View Demo](#))

```
<video width="320" height="240" preload="none" poster="videoframe.jpg">

  <source src="demo-video.mp4" type="video/mp4" />

  <source src="demo-video.ogv" type="video/ogg" />

</video>
```

```
</video>

<script>

    var video = document.getElementsByTagName('video')[0];

    function toggleMute() {

        video.muted = !video.muted;

    }

</script>

<p>

    <a href="JavaScript:video.play();">Play</a> |

    <a href="JavaScript:video.pause();">Pause</a> |

    <a href="JavaScript:toggleMute();">Sound On/Off</a>

</p>
```



[Play](#) | [Pause](#) | [Sound On/Off](#)

你还可以利用 **DOM API** 来为音频/视频做更多。而且如果你乐意添加一些其他的東西，将会把这个带向一个完全不同的级别。例如，你可以把视频放到了 **HTML5 canvas** 元素中。这将会允许你读取一个视频的像素数据、控制、和帧，并做一些你想做的好玩的东东。（自己看看：[Blowing up HTML5 Video](#), [Ambilight for HTML5 Video](#)）

了解更多：



- [关于 HTML5 音频/视频的一切](#)
- [在 HTML5 页面播放音频/视频](#)
- [浏览器自带 HTML5 支持](#)
- [网页中的视频 \(不需要额外支持\)](#)
- [HTML5Video.org](#)
- [视频类型的参数](#)

**跨浏览器解决方案:**

- [SublimeVideo](#)
- [Kaltura HTML5 Media Library](#)
- [Video for Everybody](#)
- [jPlayer: jQuery Audio Player](#)
- [Projekktor](#)
- [YUI HTML5 Player](#)
- [OIPlayer jQuery Plugin](#)
- [Degradable HTML5 audio and video Plugin](#)
- [Xiph QuickTime Component to make Ogg work in Safari](#)

**JavaScript 交互:**

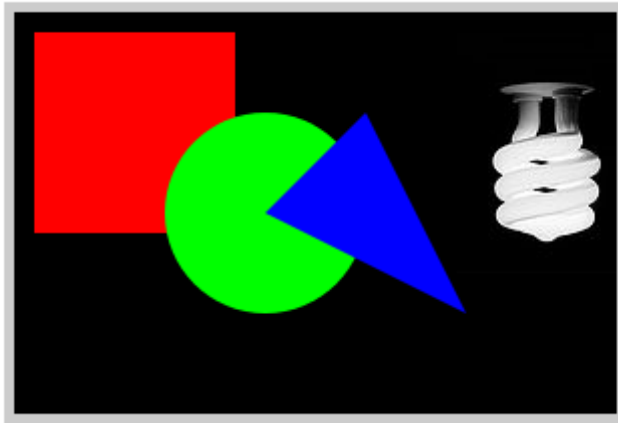
- [处理 HTML5 视频的 Mootools](#)

**编码、转换工具:**

- [Miro Video Converter](#)
- [SUPER](#)
- [MiniCoder](#)
- [HandBrake](#)
- [ffmpeg2theora](#)
- [Firefogg](#)

## 基于 JavaScript 的 2D 绘画

HTML5 中最让人兴奋的特性是 `<canvas>` — 那个用来作画的东东。在你的页面中插入 canvas 就像插入其他标记一样平常，但你将需要一些编程的经验来绘制形状、图表、动画、游戏、图片作品等。



在除 IE 外的所有现代浏览器（Firefox 3, Safari 3.1, Chrome 2, and Opera 9.6）都支持 Canvas。你可以使用 [ExplorerCanvas](#) 这个 Javascript 解决方案来为 IE 添加这个新特性。

**简单示例代码：**基于指令的 HTML5 Canvas 2D 绘画 ([View Demo](#))

```
<!doctype html>

<html lang="en">

<head>

    <meta charset="utf-8" />

    <title>HTML5 Canvas Demo</title>

    <link rel="stylesheet" href="html5reset.css" type="text/css" />

    <link rel="stylesheet" href="html5simple.css" type="text/css" />

    <!--[if lt IE 9]>

        <script src="html5.js"></script>

        <script src="excanvas.js"></script>

    <![endif]-->

    <script type="text/javascript">

        function draw(){

            var canvas = document.getElementById('mycanvas');

            if (canvas.getContext){

                var ctx = canvas.getContext('2d');
```

```
// Draw Rectangle

ctx.fillStyle = "rgb(255,0,0)";

ctx.fillRect (10, 10, 100, 100);


// Draw Circle

ctx.fillStyle = "rgb(0,255,0)";

ctx.beginPath();

ctx.arc(125,100,50,0,Math.PI*2,true);

ctx.fill();

ctx.closePath();


// Draw Custom Shape With Lines

ctx.fillStyle = "rgb(0,0,255)";

ctx.beginPath();

ctx.moveTo(125,100);

ctx.lineTo(175,50);

ctx.lineTo(225,150);

ctx.fill();

ctx.closePath();


// Draw Image From External File

var myImage = new Image();

myImage.onload = function(){

    ctx.drawImage(myImage, 220, 10);

}
```

```
        myImage.src = "sample.jpg";

    }

}

</script>

<style type="text/css">

    canvas {

        border: 5px solid #ccc;

        background: #000;

    }

</style>

</head>

<body onload="draw();">

    <header>

        <h1>HTML5 Canvas Demo</h1>

    </header>

    <figure>

        <canvas id="mycanvas" width="300" height="200">Fallback content, in case
the browser does not support Canvas.</canvas>

        <figcaption>Works in Firefox 3+, Safari 3.1+, Chrome 2+ and Opera
9.6+</figcaption>

    </figure>

</body>

</html>
```

**了解更多:**

- [HTML5 Canvas – Web 绘画的未来](#)
- [Canvas 教程 at Mozilla Developer Center](#)
- [如何使用 HTML 5 Canvas](#)
- [HTML5 Canvas 基础 \(Opera 开发者社区\)](#)
- [让我们称之为画层](#)

#### 工具 & 资源列表:

- [HTML5 Canvas 帮助手册](#)
- [jQuery Visualize Plugin: 基于 Table 的图表](#)
- [Processing.js: Port of the Processing Visualization Language](#)
- [Cartagen: Framework For Rendering Maps in HTML5](#)
- [RGraph: HTML5 Canvas Graph Library](#)

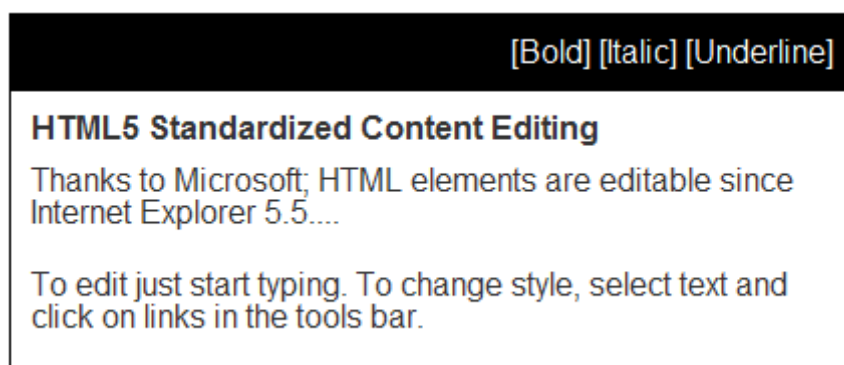
#### 示例 & 应用:

- [Chrome Experiments](#)
- [Harmony: Procedural Drawing Tool](#)
- [Sketchpad: Online Paint/Drawing Application](#)
- [CanvasPaint: Copy of Microsoft Paint](#)

### 让内容可编辑

如果你想让页面的某个地方，允许用户编辑页面，所有的你需要做的事就是添加 **contenteditable** 属性到父容器中。你可能曾在 **WYSIWYG** 编辑器中看过。这个属性是 HTML5 的一部分，且它几乎被所有主流浏览器支持（Internet Explorer 5.5+, Firefox 3+, Safari 3.1+, Chrome 2+, and Opera 9.6+）：

值得注意的是设置 **contenteditable="true"** 只是允许用户编辑、删除、插入内容，并不会自动提供其他类似于 **WYSIWYG** 编辑器的保存或应用样式的功能。你将需要自己用 **Javascript** 来添加这些功能。



**简单示例代码:** 有加粗、倾斜和下划线功能的基本编辑器 ([View Demo](#))

```
<!doctype html>

<html lang="en">
```

```
<head>

  <meta charset="utf-8" />

  <title>HTML5 ContentEditable Demo</title>

  <link rel="stylesheet" href="html5reset.css" type="text/css" />

  <link rel="stylesheet" href="html5simple.css" type="text/css" />

  <!--[if lt IE 9]>

    <script src="html5.js"></script>

  <![endif]-->

  <script language="javascript">

    function editStyle(cmd) {

      document.execCommand(cmd, null, null);

    }

  </script>

</head>

<body onload="draw();">

  <header>

    <h1>HTML5 ContentEditable Demo</h1>

  </header>

  <div id="democontainer">

    <div id="editingcontrols">

      <a href="#" onclick="editStyle('bold');">[Bold]</a>

      <a href="#" onclick="editStyle('italic');">[Italic]</a>

      <a href="#" onclick="editStyle('underline');">[Underline]</a>

    </div>

  </div>

</body>

</html>
```

```
<div id="editor" contenteditable="true">

    <h2>HTML5 Standardized Content Editing</h2>

    <p>Thanks to Microsoft; HTML elements are editable since Internet Explorer 5.5....</p>

    <p>To edit just start typing. To change style, select text and click on links in the tools bar.</p>

</div>

</div>

</body>

</html>
```

了解更多:

- [通向 HTML5 之路: contentEditable](#)
- [如何利用 webkit 和 HTML 创建一个桌面版的 WYSIWYG 编辑器](#)

工具 & 资源列表:

- [QuirksMode: execCommand 兼容列表](#)
- [Internet Explorer commands](#)
- [Mozilla Firefox commands](#)
- [基于网页的免费 HTML 编辑器](#)

## 拖放

HTML5 的拖放将会把与用户交互带向别一个等级,并将会对你如何设计用户交互产生重大影响。现今, **Firefox 3.5+** 已经对此特性提供了最大化的支持,其他浏览还只是保守地支持了一小部分(**Opera** 完全不支持)。不幸的是根据现阶段浏览器提供商的执行状况,你将需要依赖大量的 Javascript 和 CSS 来让做跨浏览器支持。

注意:

- 图片和超链接默认是可拖放的。对于所有的元素,HTML5 引进了一个新的属性"draggable",这将用来设置元素是否接受拖放;
- 下列事件对应 HTML5 的拖放:  
**dragstart, drag, dragenter, dragover, dragleave, drop** 和 **dragend**。当你要 fire 掉事件的时候,将需要写 function (事件处理器) 来处理你需要的,也可能需要给元素绑定事件或做事件指派来简化你的代码;

- 拖放事件还让你可以通过被拖放元素来传递数据，使用事件属性 "dataTransfer" 的 `getData` 和 `setData` 方法。
- 你可以在不同的浏览器或程序中进行拖放。

**简单代码示例：**将图片拖放到另一个窗口 ([View Demo](#))

```
<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8" />

  <title>HTML5 Drag & Drop Demo</title>

  <link rel="stylesheet" href="html5reset.css" type="text/css" />

  <link rel="stylesheet" href="html5simple.css" type="text/css" />

  <!--[if lt IE 9]>

    <script src="html5.js"></script>

  <![endif]-->

  <script>

    function DragHandler(target, e) {

      e.dataTransfer.setData('Text', target.id);

    }

    function DropHandler(target, e) {

      var id = e.dataTransfer.getData('Text');

      target.appendChild(document.getElementById(id));

      e.preventDefault();

    }

  </script>

</head>
```



```
<body>

  <header>

    <h1>HTML5 Drag & Drop Demo</h1>

  </header>

  <div id="dndcontainer">

    <div ondrop="DropHandler(this, event)" ondragenter="return false"
ondragover="return false" class="dndbox">

    </div>

    <div ondrop="DropHandler(this, event)" ondragenter="return false"
ondragover="return false" class="dndbox"></div>

    <div id="demonotes">Drag and drop images from one container to another. Works
in all major browsers except Opera.</div>

  </div>
```

### 了解更多:

- [跨浏览器 HTML5 拖放](#)
- [Firefox 3.5 中的 HTML5 拖放](#)
- [MDC: 拖放](#)
- [自带拖放支持](#)
- [HTML5 拖放 + Microformats = 没有什么不可能](#)

- [HTML5 拖放之灾](#)

#### JavaScript 解决方案:

- [HTML5 自带拖放 jQuery 插件](#)
- [基于 Simple UI Kit 的拖放](#)

### 稳健的数据存储

**Web 存储**规范提供了与 HTTP session cookies 相似的稳健的 Web 存储属性。他们是 "sessionStorage" 和 "localStorage"。

- **sessionStorage** 用以存储浏览最顶层环境生存周期内的数据。例如浏览器 Tab 的或窗口持续打开周期内的数据。
- **localStorage** 用以存储周期较长、多页面，以及多浏览器 session 的内数据。甚至持续到你重起浏览器或者电脑。

本地和 session 存储特性被如下浏览器所支持: IE8+, Firefox 3.5+, Safari 4.0+, Chrome 2.0+ and Opera 10.5+。

**简单代码示例:** 使用 localStorage 来创建页面计数器 ([View Demo](#))

```
<p>You Have Viewed This Page <b>

<script>

    if (!localStorage.pageCounter)

        localStorage.setItem('pageCounter',0);

        localStorage.setItem('pageCounter',parseInt(localStorage.pageCounter)+1);

        document.write(localStorage.pageCounter);

</script>

</b> Time(s).</p>

<p><input value="Clear localStorage" type="button"
onClick="localStorage.clear();location.reload(true);" />

<input value="Reload Page" type="button" onClick="location.reload(true);"
/></p>
```

你也可以把上面的代码中所有的 localStorage 的实例替换成 sessionStorage 来测试 sessionStorage。你的计数器将会在你关闭 Tab 或窗口前正常工作。如果你重起浏览器，计数器将会重新计数 ([View Demo](#)) 。

了解更多:

- [学习 HTML5 Local Storage](#)

**JavaScript 解决方案:**

- [realstorage: JavaScript compatibility wrapper for localStorage – provides fallback to Gears](#)
- [sessionstorage: 跨浏览器 HTML5 sessionStorage](#)
- [jQuery localStorage 插件](#)
- [jStorage: 在支持的地方使用 HTML5 local storage 在旧版的 IE 上使用 userData behavior](#)

更多\_\_\_\_\_

除了已经讨论的内容外, 还有更多你会想跟进了解的特性和相关的技术。这些是会帮助你研究的摘要列表:

### 网页版离线程序

HTML5 规范还定义了当用户的网络被断开后如何让它们继续与网页程序和文档进行交互。这个特性现在被 Firefox 3.5+, Chrome 4.0+, Safari 4.0+ and Mobile Safari 3.1+ 所支持。

你可以通过提供一个 manifest 文件来定义哪些文件需要被缓存, 哪些需要在离线的时候有折衷方案替代。当用户访问这个页面, 支持的浏览器将会猎取一个 manifest 版本。它将下载并缓存所有的涉及到的文件, 并且当 manifest 相对于用户上次的浏览的版本有变化, 它将会再次下载并缓存所有的文件。

- [Safari 客户端存储及离线应用程序编程指南](#)
- [使用离线应用程序提供离线支持](#)
- [Mozilla Hacks: 离线网页应用程序](#)
- [如何在 iPhone 上创建离线应用程序](#)

### 跨文档通讯

新的规范还提供了"postMessage" 让我们可以向不同域的页面发送消息。这个特性在所有现代浏览器中被支持: IE8+, Firefox3+, Safari 4+, Chrome 2+ and Safari 9.6+。

- [Safari Reference Library: 跨文档通信](#)
- [Mozilla Developer Center: postMessage](#)

### 用 className 访问 DOM 元素

我们在 Javascript 中最常见的事是选择 DOM 元素并动态地做一些事。我们大部分人在原生 Javascript 中使用 "getElementById" 来做这事。HTML5 规则已经添加了 "getElementsByClassName" 方法, 当我们要用 className 来访问元素的时候, 这将会大大提升我们脚本的性能。 这个特性在除 IE 外的其他现代浏览器都支持, 有 Firefox3+, Safari 3.2+, Chrome 2+ 和 Safari 9.6+。

- [getElementsByClassName 速度对比](#)
- [对 HTML5's getElementsByClassName 的一些思考](#)

### 更多的特性

官方规范文档:

- [Microdata](#)
- [Microdata Vocabularies](#)
- [Channel Messaging](#)

## 相关技术

官方规范文档:

- [Web Workers](#)
- [WebSocket API](#)
- [WebSocket Protocol](#)
- [Server Sent Events](#)
- [Web SQL Database](#)
- [Geolocation API](#)
- [SVG](#)
- [MathML](#)
- [XMLHttpRequest](#)

## 额外的资源

如果你还想要涉猎更多的知识，应该看看这些资源:

### 必要的书签

- [HTML5 Draft](#)  
WHATWG 的草案
- [The WHATWG Blog](#)  
WHATWG 官方博客 – 负责 HTML5 规范的团队
- [WHATWG Wiki](#)  
WHATWG 贡献者提交和编纂他们对 WHATWG 规范建议的地方
- [Dive Into HTML5](#)  
Mark Pilgrim 在 HTML5 规范中精心挑选的特性。现在正在进行中，最终将会由 O'Reilly 制作为纸质读物
- [HTML5 Validator](#)  
在线 HTML5 文档检测工具
- [Planet HTML5](#)
- [HTML5 Revision Tracker](#)  
提供在线的规范选择和版本对比
- [HTML5 Doctor](#)  
注重于帮助你在今天实施 HTML5 的在线社区

### 示例 / 展廊

- [HTML5 Demos and Examples](#)  
一系列由 Remy Sharp 制作的 HTML5 实验
- [HTML5 Presentation](#)  
由 Marcin Wichary 开发，Ernest Delgado 修改。这个报告展示了桌面和移动浏览器上的 HTML5 特性
- [Information and Samples for HTML5 and related APIs](#)  
一系列由 Robert Nyman 制作的 HTML5 演示和例子

- [HTML5 Gallery](#)  
收集使用 HTML5 标记网站的 Gallery

#### 帮助手册

- [HTML5 Quick Reference Guide](#)
- [HTML5 Cheat Sheet](#)
- [HTML5 Visual Cheat Sheet](#)
- [HTML5 Security Cheatsheet](#)

#### 框架

- [Less](#)  
一个为多种屏幕宽度提供灵活多栏布局，由驱动于 HTML5 的 CSS 框架
- [52Framework](#)  
一个以让你在今天就能在项目上应用 HTML5 和 CSS3 的页面开发框架
- [SproutCore](#)  
一个由 Ruby gem 写的 JavaScript HTML5 应用程序框架。
- [Frame](#)  
Frame 是一个支持 Layout, Typography, Forms, Code, Table, Reset, 和 Print 标签样式化，并提供 HTML5 默认样式和元素支持的 CSS 框架

#### 综述性文章 & 教程

- [用 HTML5 制作手机 web 程序](#)  
By Michael Galpin (IBM developerWorks – May 6th, 2010)
- [Touch The Future: 用 HTML5 和 CSS3 制作更优雅的网站](#)  
By Piervincenzo Madeo (PV.M Garage – April 23rd, 2010)
- [现在如何在你的客户需求上使用 HTML5](#)  
By Richard Clark (HTML5 Doctor – March 30th, 2010)
- [使用 HTML5 制作 web 应用程序](#)  
By Michael Galpin (IBM developerWorks – March 30th, 2010)
- [如何制作一个 HTML5 iPhone App](#)  
By Alex Kessinger (Six Revisions – March 26th, 2010)
- [为什么 HTML5 值得你花时间](#)  
By Mac Slocum (O'Reilly – March 15th, 2010)
- [HTML5 让我震惊了](#)  
By Shane Jeffers (Three Styles – March 11th, 2010)
- [使用 HTML5 和 CSS3 创建时尚网站](#)  
By Joe Lennon (IBM developerWorks – March 2nd, 2010)
- [用 HTML5 和 CSS3 制作一个向后兼容的单页作品集页面](#)  
By Tom Kenny (Inspect Element – January 25th, 2010)
- [HTML5 将会如何改变你使用网站的习惯](#)  
By Kevin Purdy (Lifehacker – December 1st, 2009)
- [Coding A HTML 5 Layout From Scratch](#)  
By Enrique Ramirez (Smashing Magazine – August 4th, 2009)
- [HTML5 和 CSS3 的威力所在](#)  
By Jeff Starr (Perishable Press – July 19th, 2009)
- [是的，今天就可以使用 HTML5 了!](#)  
By Bruce Lawson (SitePoint – July 1st, 2009)

- [HTML5 中的新元素](#)

By Elliotte Rusty Harold (IBM developerWorks – August 7th, 2007)

## 示例 & 下载

你可以下载打包了所有 Demo 的 ZIP 文件。注意：视频/音频并没有包含在下载中。

[查看 Demo](#) | [下载演示文件](#)

=====

文章后面的那句话就不翻啦，套话。对于个人来说，目前会逐渐接触更多的 HTML5 和 CSS3 相关的新技术，希望能在后续的学习和研究中有所收获，并能将其应用到产品中，以提升产品在功能、速度和视觉上的体验。