前端的发展如此之迅猛，一不留神，大侠你可能就会被远远地甩在后面了。如果你不想被HTML5 的改变/更新搅得不知所措的话，可以把本文的内容作为必须了解的热身课程。

一、新的 Doctype

//zxx:”doctype”中文意思指“文档类型”

仍在使用麻烦的，不可能记得住的 XHTML 文档类型？

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.html5cn.org/">如果是，为什么还在用呢？使用新的 HTML5 文档类型代替吧。你会活得更久的——正如 Douglas Quaid 说的

<!DOCTYPE html>我就琢磨着，为了 HTML5 搞个这厮代码，您可能会对这段代码究竟靠不靠谱表示怀疑。不用担心，如今这是可行的，只有老的浏览器需要一个特定的 doctype（文档类型）。浏览器如果不知道 doctype，就会很简单的以标准模式对包含的标签进行渲染。所以，妹妹你大胆的向前冲，把小心谨慎都抛到九霄云外，去拥抱新的 HTML5 文档类型吧。

二、图形元素(The Figure Element)

看看下面给图片添加的标示：

 Image of Mars. 文字裹在 p 标签里 ,与 img 标签各行其道 ,很难让人联想到这就是标题。HTML5 通过采用<figure>元素对此进行了改正。当合<figcaption>元素组合使用时，我们就可以语义化地联想到这就是图片相对应的标题

<figure> <figcaption> This is an image of something interesting. </figcaption> </figure>

三、<small>重新定义

还在不久前，<small>元素被用来创建靠近 logo 且相关的副标题。这是个很有用的表现元素，但是，现在，这种用法可能就不正确了。<small>元素已经被重新定义了，指小字，因而更具可用性。试想下你网站底部的版权状态，根据对此元素新的 HTML5 定义，<small>可以正确地包裹这些信息。

small 元素专指“小字”。

四、脚本(scripts)和链接(links)无需 type

您可能现在仍在给 link 和 script 标签增加 type 属性。

<link rel="stylesheet" href="path/to/stylesheet.css" type="text/css" /> <script type="text/javascript" src="path/to/script.js"></script>这已经是老黄花菜，非必需品了。这意味着，这些标签都各自指向样式表和脚本。因此，我们可以把 type 属性一起干掉。

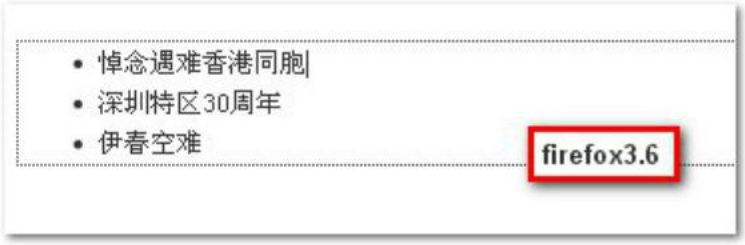
<link rel="stylesheet" href="path/to/stylesheet.css" /> <script src="path/to/script.js"></script>

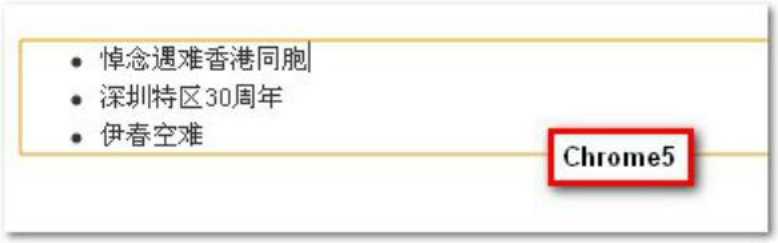
五、引号还是不要引号

…这确实是个问题。记住，HTML5 不是 XHTML，要是你不愿意，你没有必要非得用引号标记包裹你的属性，没有必要非得闭合元素。换句话说，只要你自己觉得舒服，就没有什么对错之分。对于我自己来说就是如此。

 Start the reactor.对此取舍你还得自己拿主意。如果你更倾向于结构化的文档，就算天塌下来，也要把引号牢牢拽在怀里。

六、内容可编辑





最新的浏览器有个很赞的新属性可以应用到元素上，叫做 contenteditable。顾名思意，就是允许用户编辑元素内容包含的任意文本，包括子元素。类似的用途还有很多，像是简单的待办事项清单应用程序，可大大利用其本地存储的优势。

<ul contenteditable="true"> 悼念遇难香港同胞 </li& 深圳特区 30 周年

伊春空难 或者，根据前面所学到的一些技巧，我们可以把它写成：

<ul contenteditable=true<

七、Email 输入(Inputs)

如果我们给表单输入框应用名为”email”的 type 属性，我们可以命令浏览器只允许符合有效的电子邮件地址结构的字符串。没错，内置表单验证即将到来，由于一些显而易见的原因，我们还不能 100%依赖内置验证，较旧的浏览器不认识这个”email”型，它们会简单地退回到普通文本框。

```
<form action="" method="get">
    <label for="email">邮箱:</label><input id="email" name="email"
type="email" />
    <button type="submit">确定</button>
</form>
```

//zxx:经我小测了下，貌似仅在 Chrome 浏览器下有效果(xp 系统)，当输入内容不是合法邮箱格式，点击“确定”按钮是没有反应的；当输入为合法邮箱，点击“确定”按钮才会提交刷新页面。

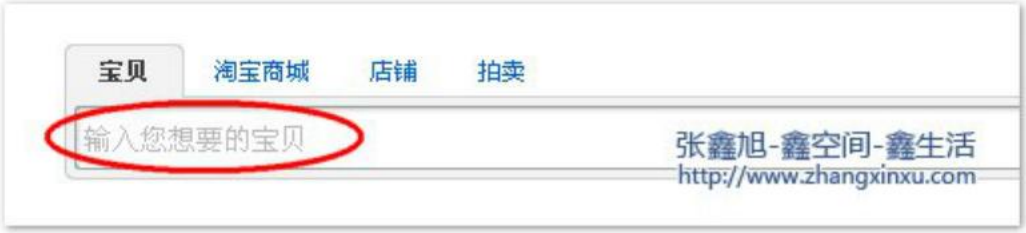
目前而言，我们不能依赖浏览器验证，客户端/服务器验证还是必须的。

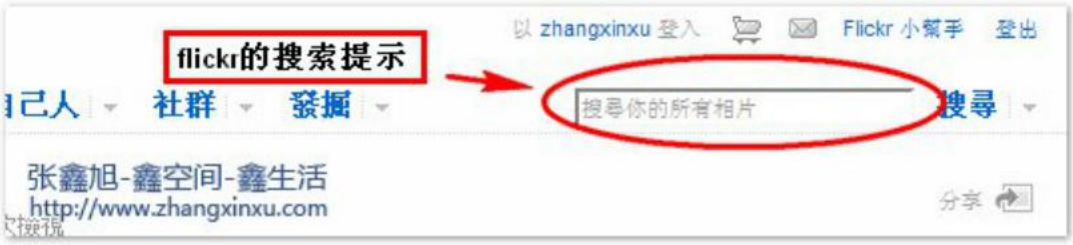
还应当指出，当谈到哪些元素和属性支持和不支持时，当前所有的浏览器都有点靠不住的。例如，Opera 似乎支持电子邮件验证，但仅在 name 属性被指定的时候。而且，它不支持占位符属性，这个我们将会在后面学到。底线是不依赖于这种形式的验证...但你仍然可以使用它！

八、占位符(Placeholders)

//zxx:此处内容非直译，有删改

Placeholders 什么意思呢，就是文本框/文本域空间默认会有个文字提示，获得焦点时，此提示文字消失；失去焦点时如果内容为空，提示文字又出现。如下图所示：





这些表单控件里面显示的些提示性的文字就是占位符。按照以往的做法，我们需要使用一点 JavaScript 代码实现占位符效果，例如我之前的“文本框/域文字提示自动显示隐藏 jQuery 小插件”一文所展示的。当然，你需要设定一个初始的默认的 value 值，然后根据输入内容进行判断，从而决定文本框值的改变与否。如果您使用占位符(placeholders)属性，一切就轻松了。

<label for="email">邮箱:</label> <input id="email" type="email" placeholder="zhangxinxu@zhangxinxu.com" size="26" />根据我的测试，目前仅 webkit 核心的浏览器支持 placeholders 属性，像是 Chrome5，Safari4，结果如下所示：

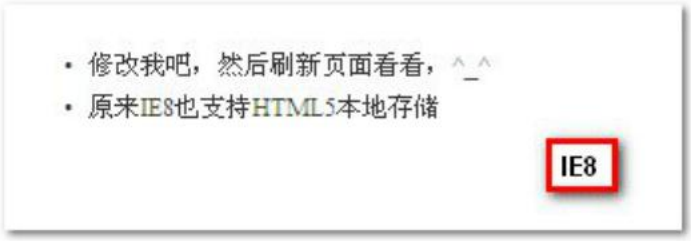
九、本地存储(Local Storage)

多亏了本地存储(非正式的 HTML5，本着方便归纳的目的)，我们可以让高级浏览器记住我们的编辑后的内容，即使浏览器被关掉或是页面刷新。

//zxx:原视频默认展示的是 YouTube 视频，不翻墙看不了，所以，这里展示来自另外一个网站的 video。建议全屏观看，以看清其中的 HTML 与 JavaScript 代码

//zxx:根据视频内容，我自己做了个 demo，关于本地存储的。

IE8 浏览器已经支持了本地存储，如下截图所示：



尽管显然不支持所有的浏览器，我们可以在 Internet Explorer8 时，Safari 4 和 Firefox 3.5 下期待此工作方式。请注意，为了弥补旧的浏览器将无法识别本地存储，你应该先测试，以确定 window.localStorage 是否存在。

	MAC				WIN							
	CHROME	FIREFOX	OPERA	SAFARI	CHROME	FIREFOX		OPERA		SAFARI	IE	
	5	3.6	10.1	4	4	3.6	3	10	10.5	4	6	7	8
Local Storage	✔	✔	✘	✔	✔	✔	✘	✘	✔	✔	✘	✘	✔

十、语义的 Header 和 Footer

那些过往的日子：

...

...

div 嘛，很自然的，没有语义化的结构——即使在应用了 id 后。现在，通过 HTML5，我们可以使用<header>和<footer>元素。以上的代码可以替换成：

<header>

...

</header>

<footer>

...

</footer>它完全适合您有多个页眉和页脚的项目。

尽量不要混淆”header”和”footer”这些元素。他们只是指他们的容器。因此，将博客底部的，例如，元信息放在 footer 元素内部是说得通的。这同样也适用于 header。

十一、更多 HTML5 表单特征(More HTML5 Form Features)

通过下面视频学习更多有用的 HTML5 表单特征：//zxx:TouTuBe 视频，需要翻墙

十二、IE 和 HTML5(Internet Explorer and HTML5)

不幸的是，讨厌的 IE 浏览器需要动点小手术才能理解新的 HTML5 元素。

所有元素，默认的，都有个 inline 的 display

为了确保所有新的 HTML5 元素能以 block 水平的元素正确地渲染 ,有必要对其做如下定义：

header, footer, article, section, nav, menu, hgroup { display: block; }不幸的是，IE 仍旧忽略这些样式，因为它不知道这些标签从哪里来的，好比是 header 元素。幸运的是，有一个简单的解决办法：

document.createElement("article"); document.createElement("footer");

document.createElement("header"); document.createElement("hgroup");

document.createElement("nav"); document.createElement("menu");奇怪的是，这段代码似乎触发 IE 浏览器。为了更简单将此应用到每个新的应用过程中，雷米夏普(Remy Sharp)创建了一个脚本，通常称为 HTML5 shiv。该脚本同样修复了些显示问题。

<!--[if IE]> <script src="http://www.html5cn.org/"> <![endif]-->

十三、文档某一部分的信息(hgroup)

想象一下，在我的网站的标题，我有我的站点的名称，随后立即由一个副标题。虽然我们可以使用一个<h1>和 lt;h2>标签，为其分别创造标记，但是依旧没有（因为 HTML4）一个简单的方法来语义上说明了两者之间的关系。此外，一个 h2 标记的使用提出了更多的问题，在层次结构上，当涉及到其他网页上显示的标题时。通过使用不影响文档的大纲流 hgroup 元素，我们可以将这些标题组合在一起。

&header& &hgroup& &h1& Recall Fan Page &/h1& &h2& Only for people who want the memory of a lifetime. &/h2& &/hgroup& &/header&

十四、必要的属性(Required Attribute)

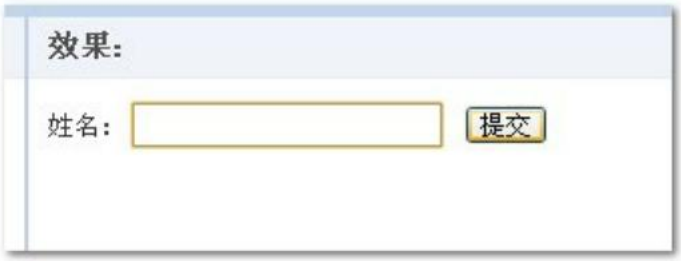
表单允许新的必要属性，用来指定是否需要特殊的 input。这取决于你的代码偏好，你可以以下面两种方式之一申明此属性。

<input type="text" name="someInput" required>或者，使用更结构化的方法：

<input type="text" name="someInput" required="required">两种方法都行。有了这个代码，并且浏览器支持此属性，如果“someInput”文本框是空白，则表单不会被提交。下面是一个简单的例子，我们还将添加占位符属性，因为没有理由不这样做。

<form action="" method="get"> <label for="name">姓名:</label> <input id="name" name="name" type="text" placeholder="zhangxinxu" required="required" /> <button type="submit">提交</button> </form>

如果 input 里面内容是空白，则表单提交的时候，文本框会高亮显示。//zxx:貌似仅在 Chrome 浏览器下有点小效果



十五、Autofocus 属性

同样 ,HTML5 的解决方案消除了对 JavaScript 的需要。如果一个特定的输入应该是“选择”，或有重点的，默认情况下，我们现在可以利用自动获取焦点属性。

<input type="text" name="someInput" placeholder="zhangxinxu" required autofocus>有趣的是，虽然我个人更倾向于喜欢 XHTML 的方法（用引号，等等），写作“autofocus=autofocus”让人感到有点怪。因此，我们将坚持使用单一关键字的方法。

十六、Audio 支持

我们无需再依赖第三方插件区渲染音频。HTML5 提供了&audio&元素，嗯，至少，最终，我们将不必担心这些插件。就目前，只有最近期的的浏览器提供 HTML5 音频支持。在这个时候，它仍然是一个很好的做法提供一些向后兼容的形式。

<audio autoplay="autoplay" controls="controls"> <source src="file.ogg" /> <source src="file.mp3" /> Download this file. </audio>Mozilla 和 WebKit 的还没有完全相处，当涉及到音频格式， Firefox 会希望看到一个.ogg 文件，而 WebKit 的浏览器支持.mp3 扩展。这意味着，至少在现在，你应该创建两个版本的音频。

当 Safari 加载页面时，它不会承认.ogg 格式，会跳过它并移动到的 MP3 版本，因此。请注意 IE，每往常一样，不支持这些格式，Opera 10 和以及以下版本只能使用.wav 文件。

十七、Video 支持

与<audio>元素很类似，在新的浏览器中也存在 Video！事实上，就在最近，YouTube 宣

告了新的 HTML5 视频嵌入，当然，是为支持此功能浏览器。因为 HTML5 的规范没有指定特定的视频编解码器，它留给了浏览器来决定。虽然 Safari 和 Internet Explorer9 可以预期支持 H.264 格式的视频（其中 Flash 播放器可以播放），Firefox 和 Opera 是坚持开源 Theora 和 Vorbis 格式。因此，当显示 HTML5 的视频，您必须提供这两种格式。

<video controls preload> <source src="cohagenPhoneCall.ogv" type="video/ogg; codecs='vorbis, theora'" /> <source src="cohagenPhoneCall.mp4" type="video/mp4; 'codecs='avc1.42E01E, mp4a.40.2'" /> Your browser is old. Download this video instead. </video>无论是”ogg”格式还是”mp4”格式的视频 Chrome 浏览器都能正确编码

还有一个值得注意的一些事情：

我们技术上不需要来设置 type 属性，但是，如果我们不这样做，浏览器不得不自己去寻找类型。节省一些带宽，还是你自己声明下吧。

不是所有的浏览器理解 HTML5 视频。在资源元素的下面，我们可以提供一个下载链接，或嵌入视频的 Flash 版本代替。这取决于你。

controls 和 preload 属性就会在下面提及。

有方法可以让所有的浏览器支持 video 标签 具体参见我前面的“让所有浏览器支持 HTML5 video 视频标签”一文。

十八、视频预载(Preload Videos)

预载属性不完全是你想的那个样子，虽然，你应该先决定是否要在浏览器预装的视频。是否有必要？或许吧。如果访问者访问一个专门展示了一个视频的页面，你一定要预载的视频，节约参观者等待的一部分时间。影片可以通过设置 preload=”preload”或是简单地添加 preload 进行预载。我更喜欢后者的解决方案，它少了一点多余的东西。

<video preload>

十九、显示控制条

如果你使用过上面的每一个提到的技术点，你可能已经注意到，使用上面的代码，视频仅仅显示的是张图片，没有控制条。为了渲染出播放控制条，我们必须在 video 元素内指定 controls 属性。

<video preload controls>

<video>

<video controls preload>

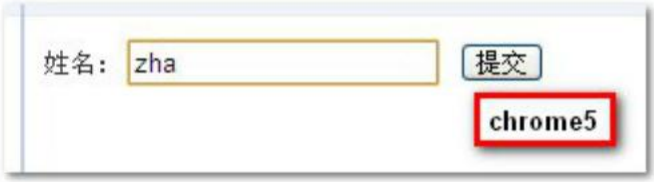
请注意，不同浏览器渲染出来的进度条的模样都是不一样的。

二十、正则表达式

你发现自己多久匆匆编写一些正则表达式验证一个特定的文本。多亏了新的 pattern 属性，我们可以在标签处直接插入一个正则表达式。

<form action="" method="get"> <label for="username">姓名:</label> <input id="username" name="username" type="text" placeholder="4-10 个英文字母" pattern="[A-Za-z]{4,10}" required="required" autofocus /> <button type="submit">提交</button> </form>如果你熟悉正则表达式，那么应该清楚 [A-Za-z]{4,10}表示接受 4-10 位不区分大小写的英文字母。如果浏览器支持 pattern 属性，

则提交表单时 ,如果文本框中的内容不符合其正则表达式 ,文本框会高亮显示。如下图所示。



//zxx:我自己小测了下，貌似目前只在 Chrome 下有效(win 系统)

注意到，我们已经开始组合使用这些很棒的属性。

如果您对正则表达式概念模糊了，可以参见这里。

二十一、属性支持检测

如果我们没有方法检测浏览器是否支持这些属性，这些就不能称之为好的属性。恩，不错的观点，事实上我们是有几种方法的，这里我们讨论 2 个。第一个是利用优秀的 Modernizr 库 ,或者 ,我们可以创建和分析这些元素 ,以确定浏览器的能力。例如 ,在我们前面的例子，如果我们要确定浏览器是否能使用 pattern 的属性，我们可以添加一小段 JavaScript 到我们的页面上：

alert('pattern' in document.createElement('input')); // boolean 事实上，这是一种确定浏览器兼容的常用方法。jQuery 库了利用这种伎俩。在上面，我们创建了一个新的 input 元素 ,并确定了里面的 pattern 属性浏览器是否认得。如果是 ,浏览器则支持此功能。否则，当然就不支持了。

<script> if (!'pattern' in document.createElement('input')) { // do client/server side validation } </script>谨记此方法依赖于 JavaScript。

二十二、mark 元素(Mark Element)

试想<mark>元素作为高亮。此标签包裹的字符串应该与用户当前的行动相关联。例如，我在一些博客上搜索“北川景子” ，我就可以使用一些 JavaScript 将当前的每个结果字符串用 mark 标签包裹。

<h3> 搜索结果 </h3>

 我很喜欢《零秒出手》里面那个拉小提琴的女孩，原来她叫做 <y;mark>北川景子 </mark>。 二十三、什么时候使用 div

我们有些人开始质问到底何时该使用 div。现在我们可以使用 header, article, section, 和 footer，还有机会使用 div...吗？当然可以。

div 应该用在没有更好的元素的时候。

例如，如果你发现你需要包裹一段代码块在对内容定位处理的包装单元内。不过如果你是包裹一个博客文章，或者，可能是，底部的链接列表，则需考虑分别使用&article&和&nav&元素，因为其更具语义。

二十四、什么可以开始立即使用

一直谈论到现在的 HTML5 要到 2022 年才能全部完成，许多人完全忽视它，这是个巨大的错误。事实上，有少量的 HTML5 的功能，我们可以在我们所有的项目中使用！更简单，更干净的代码总是一件好事。在今天的视频快速展示的技巧中，我将告诉你一些可用的选项。

//zxx:YouTuBe 视频，需要翻墙。

二十五、哪些不是 HTML5(What is Not HTML5)

那些仅凭自己的假设形象将 JavaScript 变少的过渡被全部归为 HTML5 的人是可以理解的，嘿，甚至苹果无意中推动这一想法。对于非开发人员，谁管这个呢，它是一个简单的方法适用于现代网页标准。不过，对于我们来说，尽管它可能只是语义，重要的是要准确理解什么不是 HTML5。

SVG:不是 HTML5，至少 5 岁了。

CSS3:不是 HTML5，它是...CSS。

Geolocation:不是 HTML5.//zxx:Geolocation（地理位置）：通过 HTML 5，您应该能够

使 Web 应用程序可确定您的位置，并为您提供更多的相关信息。

Client Storage(客户端存储):非 HTML5，虽说有一点切合，但被排除在规范之外，原因在于，担忧其作为一个整体，会变得过于复杂。它现在有自己的规范。

Web Sockets:不是 HTML5，同样的，有着自己的一套准则。

不管你需求有多大的区别，所有这些技术可以归为现代网络堆栈。事实上，不少这些分支规范的管理着还是同一人。

二十六、data 属性(The Data Attribute)

我们现在可以很正式地让所有的 HTML 元素支持自定义属性。然而，以前，我们可能会这样：

<h1 id=someId customAttribute=value> 小样，胆儿挺肥的呢 </h1>...校验器会小题大做！但是现在，只要我们以”data”为前缀定义我们的自定义属性，盗版属性立马变成正牌的了。如果你发现你曾经把一个重要的数据附加在诸如 class 的属性上，可能为了 JavaScript 之用，那么，本属性将大有帮助啊。

HTML 片段

 巴拉巴拉，lady 嘎嘎 检索自定义属性的价值

var theDiv = document.getElementById('myDiv'); var attr = theDiv.getAttribute('data-custom-attr'); alert(attr); // My Value 此属性还可以用在 CSS 中，例如下面这个有些傻里傻气的 CSS 文字改变的例子：

CSS 代码：.data_custom { display:inline-block; position:

relative; } .data_custom:hover { color: transparent; } .data_custom:hover:after { content: attr(data-hover-response); color: black; position: absolute; left: 0; }HTML 代码 :<a data-hover-response="我说过不要碰我 !" href="#">不要碰我 ,雅蠛蝶~~ 如果你的浏览器支持 after 伪类 以及 content 的 attr 属性 则可以看到类似下面的效果(IE8 不一样) :

要查看上图所示的效果，您可以狠狠地点击这里：CSS 与 HTML5 自定义属性 demo

还有，content 属性其实是一个非常强大的属性，由于低版本的 IE 不支持，所以此属性尚未流行，关于 content 内容生成技术，可以参见我之前的“CSS content 内容生成技术以及应用”这篇文章。

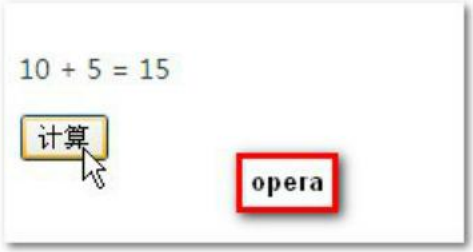
二十七、Output 元素

正如你可能预料到的，output 元素被用来显示部分计算，例如，如果你想显示一个鼠标的位置，或者是一系列数字的总和坐标，这个数据应被插入到 output 元素中。

举个简单的例子，当提交按钮被按下，我们用 JavaScript 将两个数字相加值插入到空的 output 中。

<form action="" method="get"> 10 + 5 = <output name="sum"></output> <button type="submit">计算</button> </form> (function() { var f = document.forms[0]; if (typeof f['sum'] !== 'undefined')

{ f.addEventListener('submit', function(e) { f['sum'].value = 15; e.preventDefault(); }, false); } else { alert('你的浏览器尚未准备好 !'); } })();自己测试了下 ,貌似现在只有在 Opera 浏览器下有上佳的效果：



此元素也可以接受一个属性，它反映了输出相关元素的名称，类似 label 工作原理。

二十八、使用区域 input 创建滑块(Create Sliders with the Range Input)

HTML5 引进了 range 类型的 input。

<input type="range">最值得注意的是，它可以接收 min, max, step, 和 value 属性，等等。虽然现在似乎只有 Opera 浏览器充分支持这种输入类型，但是当我们可以实际使用时，这将是美妙无比的！

参见下面的快速演示：

第一步：标签

首先，创建标签

```
<form method="post">
    <h4>音量控制</h4>
    <input type="range" name="range" min="0" max="10" step="1" value="" />
    <output name="result">  </output>
</form>
```

第二步：CSS

下面，我们要使用一点点的样式。我们将使用:before 和:after 去告知用户我们制定的最大值和最小值。

input { font-size: 14px; font-weight: bold; } input[type=range]:before { content: attr(min); padding-right: 5px; } input[type=range]:after { content: attr(max); padding-left: 5px;} output { display: block; font-size: 5.5em; font-weight: bold; }第三步：JavaScript

最后，我们

检测我们的浏览器是否认识 range input，如果不，显示提示。

当用户移动滑块的时候，动态改变 output 的值。

监听，当用户离开滑块，插入值，同时本地存储。

然后，下次用户刷新页面的时候，选择的区域和值会自动地设置成他们最后一次选择。

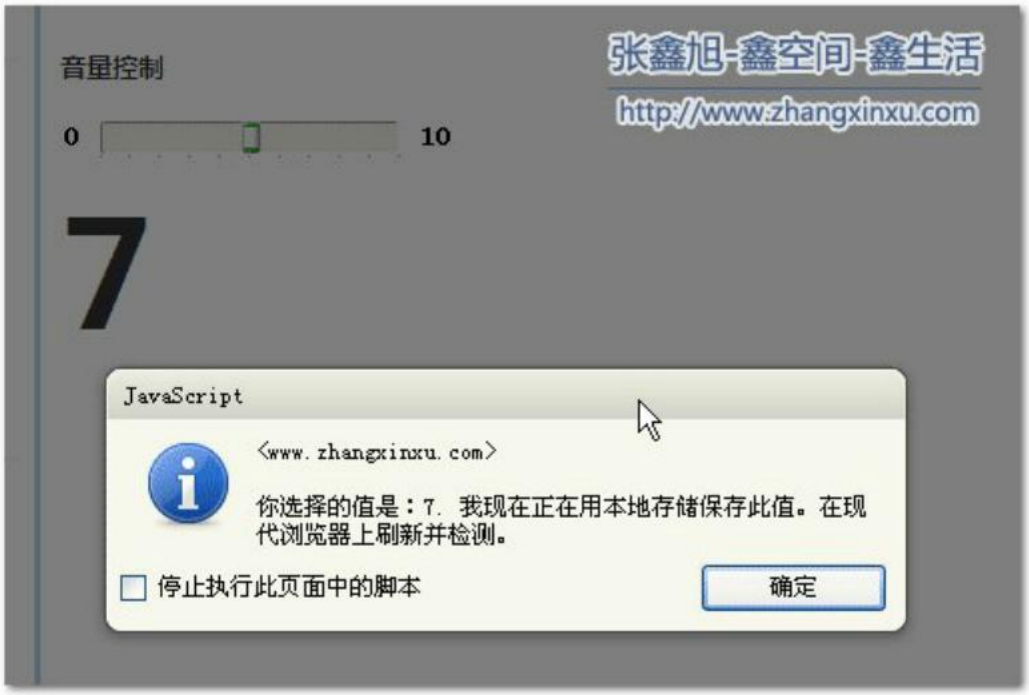
(function() { var f = document.forms[0], range = f['range'], result = f['result'], cachedRangeValue = localStorage.rangeValue ? localStorage.rangeValue : 5; // 检测浏览器是否是足够酷 // 识别 range input. var o = document.createElement('input'); o.type = 'range'; if (o.type === 'text') alert('不好意思，你的浏览器还不够酷，试试最新的 Opera 浏览器吧。'); // 设置初始值 // 无论是否本地存储了，都设置值为 5 range.value = cachedRangeValue; result.value = cachedRangeValue; // 当用户选择了个值，更新本地存储 range.addEventListener("mouseup", function() { alert("你选择的值是：" + range.value + ". 我现在正在用本地存储保存此值。在现代浏览器上刷新并检测。"); localStorage ? (localStorage.rangeValue = range.value) : alert("数据保存到了

数据库或是其他什么地方。"); }, false); // 滑动时显示选择的值

range.addEventListener("change", function() { result.value = range.value; },

false); })();您可以狠狠地点击这里：HTML5 range input 炫酷效果 demo

我的电脑是 xp 系统，默认主题，在滑块松开后 Opera 下的效果如下图所示，酷吧：



感谢您的阅读！我们已经讨论了很多，但可能只是触及到 HTML5 的皮毛，全当抛砖引玉，

希望能对您的学习有所帮助！