**HTML 5**

**什么是 HTML5？**

HTML5 将成为 HTML、XHTML 以及 HTML DOM 的新标准。

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。

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。

**新特性**

HTML5 中的一些有趣的新特性：

* 用于绘画的 canvas 元素
* 用于媒介回放的 video 和 audio 元素
* 对本地离线存储的更好的支持
* 新的特殊内容元素，比如 article、footer、header、nav、section
* 新的表单控件，比如 calendar、date、time、email、url、search

**浏览器支持**

最新版本的 Safari、Chrome、Firefox 以及 Opera 支持某些 HTML5 特性。Internet Explorer 9 将支持某些 HTML5 特性。

# **让所有IE支持HTML5的解决方案**

在页面head元素内调用[html5.js](http://html5shiv.googlecode.com/svn/trunk/html5.js) (因为IE必须在元素解析前知道这个元素，所以这个js文件不能在页面底部调用。)  
可以使用IE条件注释来调用这个js文件，这样像FireFox等非IE浏览器就会忽视这段代码，也就不会有无谓的http请求了。下面这段代码仅会在IE浏览器下运行:

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

**新的Doctype声明**

HTML5的Doctype声明很短，看到这个声明相信你马上就能记住，不用浪费脑细胞去记那长的有点变态的XHTML的Doctype声明了。

<!DOCTYPE html>

HTML5的简短的DOCTYPE声明是让Firefox、Chrome等现代浏览器和IE6/7/8等浏览器都进入(准)标准模式，你可能会奇怪IE6/7居然也可以支持HTML5 Doctype，事实上，IE是只要doctype符合 这种格式，都会进入标准模式。

**html标签上不需要声明命名空间。**

<html xmlns="http://www.w3.org/1999/xhtml" lang="zh-CN">

<!--在HTML5中，这样写：-->

<html  lang="zh-CN">

**字符集编码声明也简单了**

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<!--在HTML5中，这样写：-->

<meta charset="UTF-8" />

**去掉了Javascript和CSS标签的type属性**

通常你会在<link>和<script>加上type属性：

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

在HTML5中，不再需要type属性了，因为这显得有点多余，去掉之后可以让代码更为简洁。

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

[**HTML 5 标签**](http://www.w3school.com.cn/html5/html5_reference.asp)

4: 指示在 HTML 4.01 中定义了该元素 ，5: 指示在 HTML 5 中定义了该元素

|  |  |  |  |
| --- | --- | --- | --- |
| **标签** | **描述** | **4** | **5** |
| <!--...--> | 定义注释。 | 4 | 5 |
| <!DOCTYPE> | 定义文档类型。 | 4 | 5 |
| <a> | 定义超链接。 | 4 | 5 |
| <abbr> | 定义缩写。 | 4 | 5 |
| <acronym> | HTML 5 中不支持。定义首字母缩写。 | 4 |  |
| <address> | 定义地址元素。 | 4 | 5 |
| <applet> | HTML 5 中不支持。定义 applet。 | 4 |  |
| <area> | 定义图像映射中的区域。 | 4 | 5 |
| <article> new | 定义 article。 |  | 5 |
| <aside> new | 定义页面内容之外的内容。 |  | 5 |
| [<audio>](#audio)  new | 定义声音内容。 |  | 5 |
| <b> | 定义粗体文本。 | 4 | 5 |
| <base> | 定义页面中所有链接的基准 URL。 | 4 | 5 |
| <basefont> | HTML 5 中不支持。请使用 CSS 代替。 | 4 |  |
| <bdo> | 定义文本显示的方向。 | 4 | 5 |
| <big> | HTML 5 中不支持。定义大号文本。 | 4 |  |
| <blockquote> | 定义长的引用。 | 4 | 5 |
| <body> | 定义 body 元素。 | 4 | 5 |
| <br> | 插入换行符。 | 4 | 5 |
| <button> | 定义按钮。 | 4 | 5 |
| [<canvas>](#canvas)  new | 定义图形。 |  | 5 |
| <caption> | 定义表格标题。 | 4 | 5 |
| <center> | HTML 5 中不支持。定义居中的文本。 | 4 |  |
| <cite> | 定义引用。 | 4 | 5 |
| <code> | 定义计算机代码文本。 | 4 | 5 |
| <col> | 定义表格列的属性。 | 4 | 5 |
| <colgroup> | 定义表格列的分组。 | 4 | 5 |
| <command> new | 定义命令按钮。 |  | 5 |
| <datalist> new | 定义下拉列表。 |  | 5 |
| <dd> | 定义定义的描述。 | 4 | 5 |
| <del> | 定义删除文本。 | 4 | 5 |
| <details> new | 定义元素的细节。 |  | 5 |
| <dfn> | 定义定义项目。 | 4 | 5 |
| <dir> | HTML 5 中不支持。定义目录列表。 | 4 |  |
| <div> | 定义文档中的一个部分。 | 4 | 5 |
| <dl> | 定义定义列表。 | 4 | 5 |
| <dt> | 定义定义的项目。 | 4 | 5 |
| <em> | 定义强调文本。 | 4 | 5 |
| <embed> new | 定义外部交互内容或插件。 |  | 5 |
| <fieldset> | 定义 fieldset。 | 4 | 5 |
| <figcaption> new | 定义 figure 元素的标题。 |  | 5 |
| [<figure>](#figure) new | 定义媒介内容的分组，以及它们的标题。 |  | 5 |
| <font> | HTML 5 中不支持。 | 4 |  |
| [<footer>](#header)  new | 定义 section 或 page 的页脚。 |  | 5 |
| <form> | 定义表单。 | 4 | 5 |
| <frame> | HTML 5 中不支持。定义子窗口（框架）。 | 4 |  |
| <frameset> | HTML 5 中不支持。定义框架的集。 | 4 |  |
| <h1> to <h6> | 定义标题 1 到标题 6。 | 4 | 5 |
| <head> | 定义关于文档的信息。 | 4 | 5 |
| [<header>](#header)  new | 定义 section 或 page 的页眉。 |  | 5 |
| [<hgroup>](#hgroup) new | 定义有关文档中的 section 的信息。 |  | 5 |
| <hr> | 定义水平线。 | 4 | 5 |
| <html> | 定义 html 文档。 | 4 | 5 |
| <i> | 定义斜体文本。 | 4 | 5 |
| <iframe> | 定义行内的子窗口（框架）。 | 4 | 5 |
| <img> | 定义图像。 | 4 | 5 |
| <input> | 定义输入域。 | 4 | 5 |
| <ins> | 定义插入文本。 | 4 | 5 |
| <keygen> new | 定义生成密钥。 |  | 5 |
| <isindex> | HTML 5 中不支持。定义单行的输入域。 | 4 |  |
| <kbd> | 定义键盘文本。 | 4 | 5 |
| <label> | 定义表单控件的标注。 | 4 | 5 |
| <legend> | 定义 fieldset 中的标题。 | 4 | 5 |
| <li> | 定义列表的项目。 | 4 | 5 |
| <link> | 定义资源引用。 | 4 | 5 |
| <m> new | 定义有记号的文本。 |  | 5 |
| <map> | 定义图像映射。 | 4 | 5 |
| [<mark>](#Mark)  new | 定义有记号的文本。 |  | 5 |
| <menu> | 定义菜单列表。 | 4 | 5 |
| <meta> | 定义元信息。 | 4 | 5 |
| <meter> new | 定义预定义范围内的度量。 |  | 5 |
| <nav> new | 定义导航链接。 |  | 5 |
| <noframes> | HTML 5 中不支持。定义 noframe 部分。 | 4 |  |
| <noscript> | 定义 noscript 部分。 | 4 | 5 |
| <object> | 定义嵌入对象。 | 4 | 5 |
| <ol> | 定义有序列表。 | 4 | 5 |
| <optgroup> | 定义选项组。 | 4 | 5 |
| <option> | 定义下拉列表中的选项。 | 4 | 5 |
| <output> new | 定义输出的一些类型。 |  | 5 |
| <p> | 定义段落。 | 4 | 5 |
| <param> | 为对象定义参数。 | 4 | 5 |
| <pre> | 定义预格式化文本。 | 4 | 5 |
| <progress> new | 定义任何类型的任务的进度。 |  | 5 |
| <q> | 定义短的引用。 | 4 | 5 |
| <rp> new | 定义若浏览器不支持 ruby 元素显示的内容。 |  | 5 |
| <rt> new | 定义 ruby 注释的解释。 |  | 5 |
| <ruby> new | 定义 ruby 注释。 |  | 5 |
| <s> | HTML 5 中不支持。定义加删除线的文本。 | 4 |  |
| <samp> | 定义样本计算机代码。 | 4 | 5 |
| <script> | 定义脚本。 | 4 | 5 |
| [<section>](#section)  new | 定义 section。 |  | 5 |
| <select> | 定义可选列表。 | 4 | 5 |
| [<small>](#small) | 定义小号文本。 | 4 | 5 |
| <source> new | 定义媒介源。 |  | 5 |
| <span> | 定义文档中的 section。 | 4 | 5 |
| <strike> | HTML 5 中不支持。定义加删除线的文本。 | 4 |  |
| <strong> | 定义强调文本。 | 4 | 5 |
| <style> | 定义样式定义。 | 4 | 5 |
| <sub> | 定义下标文本。 | 4 | 5 |
| <summary> new | 定义 details 元素的标题。 |  | 5 |
| <sup> | 定义上标文本。 | 4 | 5 |
| <table> | 定义表格。 | 4 | 5 |
| <tbody> | 定义表格的主体。 | 4 | 5 |
| <td> | 定义表格单元。 | 4 | 5 |
| <textarea> | 定义 textarea。 | 4 | 5 |
| <tfoot> | 定义表格的脚注。 | 4 | 5 |
| <th> | 定义表头。 | 4 | 5 |
| <thead> | 定义表头。 | 4 | 5 |
| <time> new | 定义日期/时间。 |  | 5 |
| <title> | 定义文档的标题。 | 4 | 5 |
| <tr> | 定义表格行。 | 4 | 5 |
| <tt> | HTML 5 中不支持。定义打字机文本。 | 4 |  |
| <u> | HTML 5 中不支持。定义下划线文本。 | 4 |  |
| <ul> | 定义无序列表。 | 4 | 5 |
| <var> | 定义变量。 | 4 | 5 |
| [<video>](#video) new | 定义视频。 |  | 5 |
| <xmp> | HTML 5 中不支持。定义预格式文本。 | 4 |  |

**使网页内容可以编辑**

HTML 5中的User Interaction 用来描述页面内容交互工作的新方式。它的contenteditable 属性可以让开发者决定，页面哪部分内容允许进行用户更改，这对于wiki类的网站更为有用。

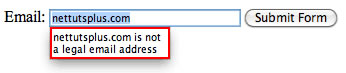
代码示例：

<div contentEditable="true">这是一个可以编辑的DIV 元素.</div>

**电子邮件输入框**

HMTL5中新增了一个输入框的电子邮件属性，可以检测输入的内容是否符合电子邮件的书写格式，功能越来越强大了吧，在HTML5之前只能依靠JS来检测。虽然内置的表单验证功能很快就会成为现实，但这个属性很多浏览器都还不支持，只会当作普通的文本输入框来处理。

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



到目前为止，包括现代浏览器在内都不支持该属性，所以这个属性暂时还是靠不住的。

**占位符**

文本框中的占位符(看看本博的搜索框效果)有利于提升用户体验，之前，我们只能依靠JS来实现占位符的效果，在HTML5中新增了占位符属性placeholder。

<input type="email" name="email" placeholder="[doug@givethesepeopleair.com](mailto:doug@givethesepeopleair.com)">

同样，目前的主流现代浏览器对该属性的支持不大好，暂时只有Chrome和Safari支持该属性，Firefox和Opera不支持该属性。



**必填项属性（改进页面表单操作）**

前端人员肯定做过不少表单验证的项目，其中很重要的一点就是有些输入框的内容是必须填写的，这里就需要使用Javascript来检查。在 HTML5中，新增了一个“必须填写”的属性：required。required属性有两种使用方法，第二种方法显得更有结构性，而第一种更简洁。

<input type="text" name="someInput" required>

<input type="text" name="someInput" required="required">

有了这个属性，使表单的提交验证变得更简单了，看看下面简单的例子：

<form method=post>  
    <label for=someInput> your name: </label>  
<input id=someInput type=text name=someInput placeholder="Douglas Quaid" required="required">  
    <button type="submit">Go</button>  
</form>

http://www.blueidea.com/articleimg/2010/12/8160/05.jpg

如果输入框为空，表单将无法提交成功。

**自动获取焦点**

同样的，HTML5也不再需要Javascript来解决输入框的自动获取焦点，如果某个输入框应当被选择或是获取到输入焦点，HTML5新增了自动获取焦点属性autofocus：

<input type="text" name="someInput" placeholder="douglas quaid" required="required" autofocus="autofocus">

autofocus也同样可以写成”autofocus=autofocus”，这样看起来标准些，这个根据自己的个人喜好而定。

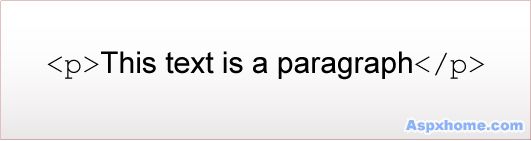
**本地存储**

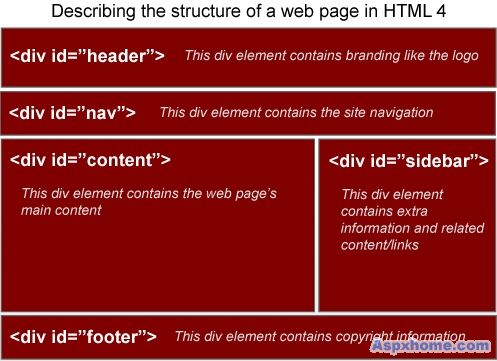
HTML5的本地存储功能，可以让现代浏览器“记住”我们输入的，就算浏览器关闭和刷新也不会受影响。虽然这个功能有些浏览器不支持，但是IE8, Safari 4, 还有 Firefox 3.5还是支持这个功能的，你可以测试下。



**新的HTML元素，增强描述内容的能力**

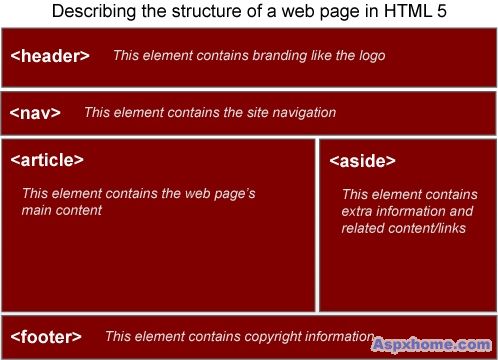
HTML的主任务是描述页面的架构，例如在<p></p>元素之间的文本内容，HTML将告诉浏览器这些文本是一个段落。

  
在HTML 5中将增加更多的HTML组件，给网页开发者提供更好的，更精确的方式来描述数据对象。请看以下的对比情况：



该布局的问题是，对于浏览器来说，所有的一切都是一个<div>元素。浏览器将平等对待<div>元素里面的内容，因为他无法进行区分，包括“级别”和“标识”，比如content, sidebar,和footer不同的网站内容各不相同。

在HTML5中，可以通过以下的方式进行同一个例子的布局：



这样浏览器就能区分各个部分，页面的主要内容在<article>元素中，导航栏在<nav>元素中等等。除了更清晰和更 符合语意的标记，它还增强了标记的互用性，比如搜索引擎能更精确的确定页面上什么内容比较重要，它可以忽略掉<nav>元素 和<footer>里的内容，因为他们通常不包含页面的重要内容，提高了搜索引擎的效率。

另外对于页面开发者来说，他们能更简单统计页面的数据，比如统计视频数量时只需在<video>元素里统计就行。对于手持设备等主要进行文章浏览时，可以直接定位到<article>元素中的内容。

**IE对HTML5的支持**

IE浏览器目前对HTML5的支持并不好，也是阻碍HTML5的更快普及的一大绊脚石，不过，IE9对HTML5的支持度还是很不错的。

IE把HTML5新增的标签都解析成内联元素，而实际上它们是块级元素，所以有必要为它们定义一个样式：

header, footer, article, section, nav, menu, hgroup {  
   display: block;  
}

尽管如此，IE还是不能解析这些新增的HTML5标签，这个时候就需要借助Javascript来解决这个问题：

document.createElement("article");  
document.createElement("footer");  
document.createElement("header");  
document.createElement("hgroup");  
document.createElement("nav");  
document.createElement("menu");

你可以借助这一段Javascript代码来修复IE更好的解析HTML5

<script mce\_src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>

**<****canvas>**

<canvas> 标签定义图形，比如图表和其他图像。

实例

如何通过 canvas 元素来显示一个红色的矩形：

<canvas id="myCanvas"></canvas>

<script type="text/javascript">

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

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

ctx.fillStyle='#FF0000';

ctx.fillRect(0,0,80,100);

</script>

HTML 5 canvas 的基础教程

**标题群(** **hgroup)**

这个类似于第二点技巧。如果用h1和h2标签分别表示网站的名称和副标题，但这会让两个本义上密切相关的标题并没有关联起来。这个时候可以使用<hgroup>标签将它们组合起来，这样代码会更有语义。

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

**<****figure>标签**

看看下面一段简单的代码：

<img alt="About image" src="path/to/image">   
<h6>image of Mars.</h6>

遗憾的是，这里的h6标签和img标签好像没有什么关系，语义不够明确。HTML5意识到了这一点，于是就采用了<figure>标 签。当<figure>结合<figcaption>标签的使用，可以让h6标签和img标签组合起来，代码就更具语义化了。

<figure>  
<img alt="about image" src="path/to/image">   
<figcaption>  
<h6>This is an image of something interesting. </h6>  
</figcaption>  
</figure>

**重新定义<****small>**

在HTML5中重新定义了<small>标签，使之更能表现语义化，在<small>的字号都会变小，想想如果这个标签用于网站的底部的版权信息还是个不错的做法。

**<section>**

<section> 标签定义文档中的节（section）。比如章节、页眉、页脚或文档中的其他部分。

<section>

  <h1>PRC</h1>

  <p>The People's Republic of China was born in 1949...</p>

</section>

**音频播放的支持**

HTML5中提供了<audio>标签，解决了以往必须依靠第三方插件才能播放音频文件的问题。目前为止，还只有少数的最新浏览器支持该标签。

<audio controls="controls" autoplay="autoplay">  
    <source src="file.ogg" />  
    <source src="file.mp3" />  
    <a href="file.mp3">Download this file.</a>  
</audio>

为什么会有两种格式的音频文件？因为Firefox和Webkit浏览器所支持的格式存在差异，Firefox只能支持.ogg文件，而 Webkit只支持.mp3的文件，解决的办法就是创建两个版本的音频文件，这样就可以兼容Firefox和Webkit的浏览器了，需要注意的是IE不 支持该标签。

**视频播放的支持**

和<audio>标签一样，HTML5也提供了<video>标签对播放视频文件的支持。YouTube也宣布了一项新的 HTML5的视频嵌入。不过有点遗憾，HTML5的规范并没有指定特定的视频解码器，而是让浏览器自己来决定。这就造成了个浏览器的兼容问题，虽然 Safari和IE9都支持还H.264格式的视频( Flash 播放器可以可以播放)，Firefox和Opera则支持开源的Theora和Vorbis格式。因此，当显示HTML5视频的时候，也得准备2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'" />   
<div> your browser is old. <a href="cohagenPhoneCall.mp4">download this video instead.</a> </div>  
   
</video>

需要注意的是，type属性虽然可以省略掉，但是如果加上的话，浏览器就可以更快的准确的解析该视频文件。并不是所有的浏览器都支持HTML5的视频，所以得做好使用Flash版本来代替，当然，这个决定权在于你。

**预加载视频**

预加载属性：preload，首先要确定是否需要预先加载视频，假如，访客在访问一个有很多视频展示的页面，那么就有必要预先加载一段视频，这样可 以节省访客的等待时间，提高用户体验。你可以给<video>标签添加一个preload属性来实现预先加载的功能。

<video preload="preload">  
 ...  
</video>

**显示控件**

显示控件属性可以给视频添加一个播放暂停的控件，需要注意的是每个浏览器显示的效果可能会有些差异。

<video controls="controls" preload="preload">  
...  
</video>

**Mark标签**

<mark>标签用于高亮显示那些需要在视觉上向用户突出其重要性的文字，包裹在此标签里的字符串必须与用户当前的行为相关。例如，如 果我在一些博客中搜索“Open your Mind” ，我可以使用在<mark>标签里使用JavaScript 来包裹每一次动作。

<h3> search results </h3>  
<h6> They were interrupted, just after Quato said, <mark>"Open your Mind"</mark>. </h6>

**使用正则表达式**

在HTML5中，我们可以直接使用正则表达式。

<form method=post action="">  
    <label for="username">create a username: </label>   
<input id="username" type="text" name="username" placeholder="4 <> 10" required="required" autofocus="autofocus" pattern="[A-Za-z]{4,10}">  
    <button type="submit">Go </button>  
</form>

**是否使用双引号**

这有点让人纠结，HTML5并不是XTHML，你可以省去标签中的双引号。相信大多数同志也包括我都习惯了加上双引号，因为这让代码看起来会更标准。不过，这可以根据你的个人喜好来确定是到底要不要双引号。

<h6 id="someid" class="myclass"> start the reactor. </h6>

**该如何正确的使用div标签**

有些人可能会有疑问，有了<header>和<footer>等这些标签，<div>标签在HTML5中还有 用吗？答案是肯定的，比如你想创建一个能包裹特殊内容的容器自由灵活的<div>肯定是首选，而你要创建一篇文章或者一个导航菜单，建议你使 用更有语义的<article>和<nav>标签。

很多人认为HTML5可能还是很遥远的事，所以直接无视，其实不然，现在很多网站都已经开始使用HTML5了，事实上，HTML5的一些新增属性和 功能是让代码变得更简洁，这总归是一件好事，应该值得我们推崇。

**检测浏览器对HTML5属性的支持**

由于各浏览器对HTML5属性的支持度不同，这就造成了一些兼容问题。但是可以使用方法来检测该浏览器是否支持这些属性，上例中的代码如果要检测pattern属性是否被浏览器识别，可以使用Javascript代码来检测。

alert( 'pattern' in document.createElement('input') ) // boolean;

其实这是确定浏览器兼容常用的方法，jQuery库就经常使用这种方法。上面的代码中创建了一个input标签，并检测pattern属性是否被浏览器支持，如果能支持的话，浏览器就支持这个功能，否则就不支持。

<script>    
 if (!'pattern' in document.createElement('input') ) {   
    // do client/server side validation   
 }   
</script>