logo.png

jquery

H5 web前端

[www.bufanui.com](http://www.bufanui.com)

# 一、HTML5

## 认识HTML5

HTML5 是html4.0 升级版

**结构 Html5 、样式 css3 、行为: API 都有所增强**

HTML5并不仅仅只是做为HTML标记语言的一个最新版本，更重要的是它制定了Web应用开发的一系列标准，成为第一个将Web做为应用开发平台的HTML语言。

HTML5定义了一系列新元素，如新语义标签、智能表单、多媒体标签等，可以帮助开发者创建富互联网应用，还提供了一些Javascript API，如地理定位、重力感应、硬件访问(页面要求访问相册)等，可以在浏览器内实现类原生应用，甚至结合Canvas我们可开发网页版游戏。

广义概念：HTML5代表浏览器端技术的一个发展阶段。在这个阶段，浏览器呈现技术得到了一个飞跃发展和广泛支持，它包括：HTML5，CSS3，Javascript，API在内的一套技术组合 技术栈 H5 H5前段开发工程师 html5前段开发工程师

https://threejs.org/

## 语法规范

HTML5在语法规范上也做了比较大的调整，去除了许多冗余的内容，书写规则更加简洁、清晰。

见代码实例

**特点：**

1、更简洁

2、更宽松

单标签不用写关闭符号 <input> </br>  
 双标签省略结束标签 ~~<div></div>~~  **可以但是不推荐**

html、head、body、colgroup、tbody(可以不写)可以完全省略 **不推荐**

# 二、语义标签

### 2.1 推出的理由及目标

HTML5的出现，对于WEB来说意义重大。因为它的意图是想要把目前WEB上存在的各种问题一并解决掉。

* Web浏览器之间的兼容性很低（目前只要是IE）
* 文档结构不够明确（增加了很多结构, 语义化的标签）(原来的时候前端div+css)

全是div布局，靠什么区分页面结构？class 最重要！class名字不要随便命名

* Web应用程序的功能受到了限制

面试题: 怎么看待html的语义化?

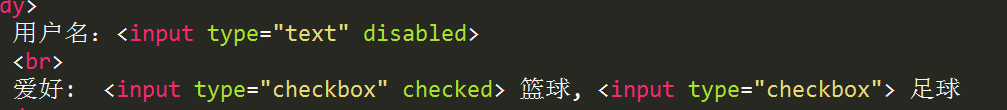
### 2.2 语法的改变

DOCTYPE声明

指定字符编码

可以省略结束（全部）标记的元素

具有boolean值的属性（disabled，checked）



省略引号 （标签内的属性值可以省略引号）

### 2.3 新增的结构化(语义化)标签元素

Section、article、aside、header、hgroup、footer、nav、figure

**section元素**  
表示页面中的一个内容区块,比如章节、页眉、页脚或页面的其他部分。可以和h1、 h2……等元素结合起来使用，表示文档结构。例：HTML5中<section>……</section>;HTML4中<div> ……</div>。

* **article元素**   
  表示页面中一块与上下文不相关的独立内容。比如一篇文章。
* **aside元素**   
  表示article元素内容之外的、与article元素内容相关的辅助信息。

[HTML](http://www.100sucai.com/web/wangyezhizuo/javascript/" \t "_blank)<aside>元素表示一个页面的一部分， 它的内容跟这个页面的其它内容的关联性不强，或者是没有关联，单独存在。<aside>元素通常显示成侧边栏(sidebar)或一些插入补充内容。通常用来在侧边栏显示一些定义，比如目录、索引、术语表等；也可以用来显示相关的广告宣传，作者的介绍，[Web应用](http://www.100sucai.com/web/jquerytexiao/guanggao/" \t "_blank)，相关链接，当前页内容简介等。

* **header元素**   
  表示页面中一个内容区块或真个页面的标题。
* **hgroup元素**   
  表示对整个页面或页面中的一个内容区块的**标题**进行组合。
* **footer元素**   
  表示整个页面或页面中一个内容区块的脚注。一般来说，他会包含创作者的姓名、创作日期以及创作者的联系信息。
* **nav元素**   
  表示页面中导航链接的部分。
* **figure元素**表示一段独立的流内容，一般表示文档主体流内容中的一个独立单元。

比如：https://tympanus.net/Development/HoverEffectIdeas/index2.html

### 2.4 新增的“花样”标签元素

video、audio、canvas、embed、mark、progress、meter、time、ruby、rt、rp、wbr、command、details、datalist、datagrid、keygen、output、source、menu

### 2.5 新增的input类型

email、url、number、range、color

### 2.6 废除的标签（html是骨架，尽量不要有自己的样式，样式交个css）

* 能使用CSS替代的basefont、big、center、font、s、tt、u、strong、em、下划线、打折线等
* 不再使用frame框架 （用iframe代替）
* 只有部分浏览器支持的元素
* 其他被废除的元素

Seo是前端开发最重要也最需要注意的一个事项，怎么实现一个页面能更好的支持浏览器搜索，怎么使排名靠前。

1. 注意：h1标签一个页面只有一个，搜索引擎对h1标签的权重很大。
2. Description和keyword
3. 在页面中尽量多的地方显示当前页面的关键词 比如写到alt里面
4. 善用a标签和img，如果只展示图片，不展示内容，用 text-indent:-90000px,logo
5. Class命名很重要，百度搜索权重很大

怎么优化网页的seo?

1.keyword 要设置好

### 2.7 新增和废除的标签属性

* 新增的标签属性
* 表单内标签相关的属性
* 链接相关的属性
* 其他属性
* 废除的标签属性

|  |  |  |
| --- | --- | --- |
| **在HTML 4中使用的属性** | **使用该属性的元素** | **在HTML 5中的替代方案** |
| **rev** | link、a | rel |
| **charset** | link、a | 在被链接的资源的中使用HTTP Content-type头元素 |
| **shape、coords** | a | 使用area元素代替a元素 |
| **longdesc** | img、iframe | 使用a元素链接到校长描述 |
| **target** | link | 多余属性，被省略 |
| **nohref** | area | 多余属性，被省略 |
| **profile** | head | 多余属性，被省略 |
| **version** | html | 多余属性，被省略 |
| **name** | img | id |
| **scheme** | meta | 只为某个表单域使用scheme |
| **archive、chlassid、codebose、codetype、declare、standby** | object | 使用data与typc属性类调用插件。需要使用这些属性来设置参数时，使用param属性 |
| **valuetype、type** | param | 使用name与value属性，不声明之的MIME类型 |
| **axis、abbr** | td、th | 使用以明确简洁的文字开头、后跟详述文字的形式。可以对更详细内容使用title属性，来使单元格的内容变得简短 |
| **scope** | td | 在被链接的资源的中使用HTTP Content-type头元素 |
| **align** | caption、input、legend、div、h1、h2、h3、h4、h5、h6、p | 使用CSS样式表替代 |
| **alink、link、text、vlink、background、bgcolor** | body | 使用CSS样式表替代 |
| **align、bgcolor、border、cellpadding、cellspacing、frame、rules、width** | table | 使用CSS样式表替代 |
| **align、char、charoff、height、nowrap、valign** | tbody、thead、tfoot | 使用CSS样式表替代 |
| **align、bgcolor、char、charoff、height、nowrap、valign、width** | td、th | 使用CSS样式表替代 |
| **align、bgcolor、char、charoff、valign** | tr | 使用CSS样式表替代 |
| **align、char、charoff、valign、width** | col、colgroup | 使用CSS样式表替代 |
| **align、border、hspace、vspace** | object | 使用CSS样式表替代 |
| **clear** | br | 使用CSS样式表替代 |
| **compace、type** | ol、ul、li | 使用CSS样式表替代 |
| **compace** | dl | 使用CSS样式表替代 |
| **compace** | menu | 使用CSS样式表替代 |
| **width** | pre | 使用CSS样式表替代 |
| **align、hspace、vspace** | img | 使用CSS样式表替代 |
| **align、noshade、size、width** | hr | 使用CSS样式表替代 |
| **align、frameborder、scrolling、marginheight、marginwidth** | iframe | 使用CSS样式表替代 |
| **autosubmit** | menu |  |

### 2.8 新增的全局属性

是指对所有的标签都可以使用的属性

**contentEditable**属性 单独某一个标签的属性 可以使内容能被编辑

**designMode**属性（这个在js中进行使用，让页面中所有的元素开启可编辑模式） 让页面所有的标签都可以被编辑 比如div、p、h1等等

window.onload = function() {

document.designMode = "on";

}

**hidden**属性

**~~spellcheck~~**~~属性 （input和textarea提供的新属性，做了语法检查）~~

**tabindex**属性 (让标签获取到焦点)



# 三、兼容处理

CSS hack

(我们在测试ie 的兼容的时候，有一个叫做ietester 的软件，这个软件可以模拟ie6-ie11)

在不支持HTML5新标签的浏览器里，会将这些新的标签解析成行内元素(inline)对待，所以我们只需要将其转换成块元素(block)即可使用，但是在IE9版本以下，并不能正常解析这些新标签，但是却可以识别通过document.createElement('tagName')创建的自定义标签，于是我们的解决方案就是将HTML5的新标签全部通过document.createElement('tagName')来创建一遍，这样IE低版本也能正常解析HTML5新标签了，在实际开发中我们更多采用的是通过检测IE浏览器的版本来加载三方的一个JS库来解决兼容问题。

**lte：**就是Less than or equal to的简写，也就是小于或等于的意思。  
**lt ：**就是Less than的简写，也就是小于的意思。  
**gte：**就是Greater than or equal to的简写，也就是大于或等于的意思。  
**gt ：**就是Greater than的简写，也就是大于的意思。  
**!：**就是不等于的意思，跟javascript里的不等于判断符相同

**注意：**在非IE浏览器中是看不到效果的

|  |
| --- |
| <!--[if gte IE 7]> <link rel="stylesheet" href="ie10.css"> <![endif]-->  <!--[if lte IE 8]> <script src="html5shiv.min.js"></script> <![endif]--> |

# 四、表单

伴随着互联网富应用以及移动开发的兴起，传统的Web表单已经越来越不能满足开发的需求，所以HTML5在Web表单方向也做了很大的改进，如拾色器、日期/时间组件等，使表单处理更加高效。

## 4.1输入类型 (表单类型，表单元素，表单属性,表单事件.)

email 输入email格式

tel 手机号码

url 只能输入url格式

number 只能输入数字

search 搜索框

range 范围 滑动条

color 拾色器

* time 时间
* date 日期 不是绝对的
* --datetime 时间日期(移动支持)
* month 月份
* week 星期

部分类型是针对移动设备生效的，且具有一定的兼容性，在实际应用当中可选择性的使用。

## 4.2表单元素（标签）

**<datalist>** 数据列表 自动补全

实际开发中：需要自动补全的内容列表项可能很多，不可能挨个展示在页面中，一般是通过ajax局部页面刷新技术实现的。

与input 配合使用

|  |
| --- |
| <input type=”text” list=”data”>  <datalist id=”data”>  <option>男</option>  <option>女</option>  <option>不详</option>  </datalist> |

~~<~~**~~output>~~** 不可当做数据提交 目前没发现怎么用

|  |
| --- |
| <form onchange="x.value=parseInt(a.value)+parseInt(b.value)">  0<input type="range" id="a" value="50">100  +<input type="number" id="b" value="50">  =<output name="x" for="a b"></output>  </form> |

**<meter>** meter元素用来表示规定范围内的数量值，如磁盘使用量比例饿、关键词匹配程度等。需要注意的是，meter不可以用来标记那些没有已知范围的任意值，如重量、高度，除非已经设定了它们值的范围。表示度量器，不建议用作进度条

|  |
| --- |
| <meter value="81" min="0" max="100" low="60" high="80" ></meter> |

**<progress></progress>**

|  |
| --- |
| <progress value="22" max="100"></progress> |

## 4.3表单属性

**placeholde**r 占位符

**autofocus** 获取焦点

**multiple** 文件上传多选或多个邮箱地址

**autocomplete** 自动完成，用于表单元素，也可用于表单自身(on/off)

**form** 指定表单项属于哪个form，处理复杂表单时会需要 一般一个页面只有一个form

**novalidate** 关闭验证，可用于<form>标签

**required** 必填项

**pattern** 正则表达式 验证表单

手机号:<input type="tel" name="tel" required="required" pattern="^(\+86)?1[3,5,8](\d{9})$">

**formaction**应用于提交按钮上，如：<input type="submit" **formaction**="xxx.action">

## 4.4表单事件

oninput 用户输入内容时触发，可用于移动端输入字数统计

oninvalid 验证不通过时触发

|  |
| --- |
| <form action="abc">  手机号:  <input type="text" name="phone" maxlength="11" pattern="^(0|86|17951)?1[0-9]{10}"  oninvalid="setCustomValidity('请输入11位手机号');"/>  <br>  <input type="submit">  </form> |

## 4.5案例练习

作业：注册表单

用户名：

密码：

电话号码：

邮箱

验证不能为空而且符合条件才能提交

# 五、多媒体

在HTML5之前，在网页上播放音频/视频的通用方法是利用Flash来播放，但是大多情况下，并非所有用户的浏览器都安装了Flash插件，由此使得处理音频/视频播放变的非常复杂，并且移动设备的浏览器并不支持Flash插件。

Flash最早用来做网页酷炫的各种动画效果。Qq秀 、banner 、包括视频、网页游戏

Html提供了音视频标签，提供了canvas画布，基本上以前能用flash实现的现在h5都能实现。

白鹭引擎 建模 物理 物体碰撞 光线投影 粒子效果等....

游戏

Media palyer

## 5.1音频

HTML5通过<audio>标签来解决音频播放的问题。

|  |
| --- |
| <audio src="素材/小手拍拍.mp3" controls>不支持</audio> |

并且可以通过附加属性可以更友好控制音频的播放，如：

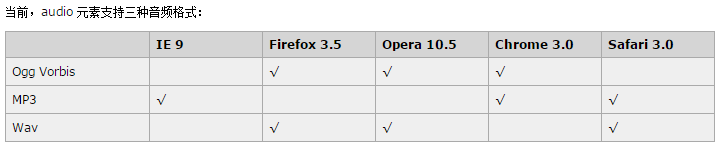
autoplay 自动播放

controls 是否显不默认播放控件

loop 循环播放

preload 预加载 同时设置autoplay时此属性失效

由于版权等原因，不同的浏览器可支持播放的格式是不一样的



**多浏览器支持的方案，如下图**

|  |
| --- |
| <audio>  <source src="素材/小手拍拍.mp3">  <source src="素材/小手拍拍.wav">  <source src="素材/小手拍拍.ogg">  </audio> |

## 5.2视频

HTML5通过<video>标签来解决音频播放的问题。

同音频播放一样，<video>使用也相当简单，如下图

|  |
| --- |
| <video src="素材/movie.ogg" width="400" height="300" controls></video> |

同样，通过附加属性可以更友好的控制视频的播放

autoplay 自动播放

controls 是否显示默认播放控件

loop 循环播放

preload 预加载，同时设置了autoplay，此属性将失效

width 设置播放窗口宽度

height 设置播放窗口的高度

由于版权等原因，不同的浏览器可支持播放的格式是不一样的



**多浏览器支持的方案，如下图**

|  |
| --- |
| <video controls>  <source src="素材/movie.ogg" type="">  <source src="素材/movie.mp4" type="">  您的浏览器不支持  </video> |

# DOM扩展

## 6.1获取元素

document.querySelector(‘div’) html5新选择器，参数是css选择器参数,选择选中的第一个

document.querySelectorAll('selector') 选择多个

很好用，但是出现太晚了。。。

Java基础语法

Ssh struts(1.X 2.X) spring hibernate 复杂 (老)

Ssm struts sping（框架之王） ibatis mybatis 简单(新)

SpringMVC spring （很好用，但是出现的晚啦！ 自己干活用）

前端：重复的轮子特别多。有甄别能力。

不是越新越好，也不是越简单越好，真正要看市场哪个占用率高。

比如：angular.js 本来是一统江湖，google认为angular.js难度高，坡度大，不适合新手，于是就you除了个angular.js2(重做)， 这个时候出现了facebook ，react.js 和ng平分市场。

比如vue.js weex（阿里的）….

前端的日子目前并好过因为各种重复的轮子太多，市场也没有最终定位。需要我们自己去逐个了解并确定方向。

前端就是追新的东西，前提是最后能占住市场。

## 6.2类名操作2

1、Node.classList.add('class') 添加class

2、Node.classList.remove('class') 移除class

3、Node.classList.toggle('class') 切换class，有则移除，无则添加

4、Node.classList.contains('class') 检测是否存在class

非常好用 但是出现的太晚了 。。。

## 6.3自定义属性

在HTML5中我们可以自定义属性，其格式如下data-\*=""，例如

data-info="我是自定义属性"，通过Node.dataset['info'] 我们便可以获取到自定义的属性值。

Node.dataset是以类对象形式存在的

当我们如下格式设置时，则需要以驼峰格式才能正确获取

data-my-name="mm"，获取Node.dataset['myName']

## 案例练习 (tab 切换)

# 附件1：

你难道还没有考虑使用HTML5？ 当然我猜想你可能有自己的原因；它现在还没有被广泛的支持，在IE中不好使，或者你就是喜欢写比较严格的XHTML代码。HTML5是web开发世界的一次重大的改变，事实上不管你是否喜欢，它都是代表着未来趋势。其实HTML5并不难理解和使用。我们这里能列出许多原因为什么现在要开始使用HTML5。

　　目前有很多的文章介绍使用HTML5并且介绍了使用它的优势和好处，没错，我们这篇文章也类似。随着更多这样的文章，以及Apple的支持， Adobe围绕HTML5的产品开发，以及移动flash的死亡，如此多网站的支持，我想对那些仍旧没有或者不想接受它的人说一些话。我认为主要得原因是，它看起来像一个神秘的东西。很多感觉它像喷气背包或者飞行汽车。一个未经验证的非凡想法但是并不实际。但是事实上现在已近非常的实际了。

　　为了解密HTML5并且帮助顽固的开发设计人员，我这里写了列出了使用HTML5的几大原因，希望对大家有帮助！

**第十大原因：易用性**

　　俩个原因使得使用HTML5创建网站更加简单：语义上及其ARIA。新的HTML标签像<header>, <footer>，<nav>，<section>， <aside>等等，使得阅读者更加容易去访问内容。在以前，即使你定义了class或者ID你的阅读者也没有办法去了解给出的一个div究竟是什么。使用新的语义学的定义标签，你可以更好的了解HTML文档，并且创建一个更好的使用体验。

　　ARIA是一个W3C的标准主要用来对HTML文章中的元素指定“角色“，通过角色属性来创建重要的页面地形例如,header，footer，navigation或者aritcle很有必要。这一点曾经被忽略掉了并且没有被广泛使用，因为事实上并不验证。然而，HTML5将会验证这样属性。同时，HTML5将会内建这些角色并且无法不覆盖。更多的HTML5和ARIA讨论，请大家查看这里。

**第九大原因：视频和音频支持**

　　忘了flash和其它第三方应用吧，让你的视频和音频通过HTML5标签<video>和<audio>来访问资源。正确播放媒体一直都是一个非常可怕的事情，你需要使用<embed>和<object>标签，并且为了它们能正确播放必须赋予一大堆的参数。你的媒体标签将会非常复杂，大堆得令人迷惑的代码。而且HTML5视频和音频标签基本将他们视为图片：<video src=”"/>。但是其它参数例如宽度和高度或者自动播放呢？不必担心，只需要像其它HTML标签一样定义：<video src=”url” width=”640px” height=”380px” autoplay/>。

　　实际上这个过程非常简单，然而我们的老浏览器可能并不喜欢我们的HTML5，你需要添加更多代码来让他们正确工作。但是这个代码还是比<embed>和<object>来的简单的多。

**第八大原因：Doctype**

　　没错，就是doctype，没有更多内容了。是不是非常简答？不需要拷贝粘贴一堆无法理解的代码，也没有多余的head标签。最大的好消息在于，除了简单，它能在每一个浏览器中正常工作即使是名声狼藉的IE6。

**第七大原因：更清晰的代码**

　　如果你对于简答，优雅，容易阅读的代码有所偏好的话，HTML5绝对是一个为你量身定做的东西。HTML5允许你写出简单清晰富于描述的代码。符合语义学的代码允许你分开样式和内容。看看这个典型的简单拥有导航的heaer代码：<div id="header"> <h1>Header Text</h1> <div id="nav">  <ul>   <li><a href="#">Link</a></li>   <li><a href="#">Link</a></li>   <li><a href="#">Link</a></li>  </ul> </div></div>是不是很简单？但是使用HTML5后会使得代码更加简单并且富有含义：<header> <h1>Header Text</h1> <nav>  <ul>   <li><a href="#">Link</a></li>   <li><a href="#">Link</a></li>   <li><a href="#">Link</a></li>  </ul> </nav></header>

　　使用HTML5你可以通过使用语义学的HTML header标签描述内容来最后解决你的div及其class定义问题。 以前你需要大量的使用div来定义每一个页面内容区域，但是使用新的<section>，<article>，<header>，<footer>，<aside>和<nav>标签，需要你让你的代码更加清晰易于阅读。

**第六大原因：更聪明的存储**

　　HTML5中最酷的特性就是本地存储。有一点像比较老的技术cookie和客户端数据库的融合。它比cooke更好用因为支持多个windows存储，它拥有更好的安全和性能，即使浏览器关闭后也可以保存。

　　因为它是个客户端的数据库，你不用担心用户删除任何cookie，并且所有主流浏览器都支持。

　　本地存储对于很多情况来说都不错， 它是HTML5工具中一个不需要第三方插件实现的。能够保存数据到用户的浏览器中意味你可以简单的创建一些应用特性例如：保存用户信息，缓存数据，加载用户上一次的应用状态。

**第五大原因：更好的互动**

　　我们都喜欢更好的互动，我们都喜欢对于用户有反馈的动态网站，用户可以享受互动的过程。输入<canvas>，HTML5的画图标签允许你做更多的互动和动画，就像我们使用Flash达到的效果。

　　除了<canvas>，HTML5同样也拥有很多API允许你创建更加好的用户体验并且更加动态的web应用程序。 这里有一个列表：

    Drag and Drop (DnD)

    Offline storage database

    Browser history management

    document editing

    Timed media playback

**第四大原因：游戏开发**

　　没错， 你可以使用HTML5的<canvas>开发游戏。HTML5提供了一个非常伟大的，移动友好的方式去开发有趣互动的游戏。如果你开发Flash游戏，你就会喜欢上HTML5的游戏开发。

　　Script-tutorials目前提供了4个不部分的HTML5游戏开发教程，这里看看他们开发的有趣游戏：

    HTML5 Gaming Development Lesson One

    HTML5 Gaming Development Lesson Two

    HTML5 Gaming Development Lesson Three

    HTML5 Gaming Development Lesson Four

**第三大原因： 遗留及其跨浏览器支持**

　　你的现代流行浏览器都支持HTML5（Chrome，Firefox，Safari，IE9和Opera），并且创建了HTML5 doctype这样所有的浏览器，即使非常老非常令人厌恶浏览器像IE6都可以使用。但是因为老的浏览器能够识别doctype并不意味它可以处理HTML5标签和功能。幸运的是，HTML5已经使得开发更加简单了，更多支持更多浏览器，这样老的IE浏览器可以通过添加javascript代码来使用新的元素：

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

**第二大原因： 移动，移动还是移动**

　　你可以称之为“直觉”，但是我认为移动技术将会变得更加的流行。我知道，这里有些非常疯狂的猜测，有些可能你也想到了 – Mobile是一个时尚！移动设备将占领世界。更多的接受移动设备将会增长的非常迅速。这意味着更多的用户会选择使用移动设备访问网站或者web应用。HTML5是最移动化的开发工具。随着Adobe宣布放弃移动flash开发，你将会考虑使用HTML5来开发webp应用。当手机浏览器完全支持HTML5那么开发移动项目将会和设计更小的触摸显示一样简单。这里有很多的meta标签允许你优化移动：viewport: 允许你定义viewport宽度和缩放设置；全屏浏览器： ISO指定的数值允许Apple设备全屏模式显示；Home screen icons:  就像桌面收藏，这些图标可以用来添加收藏到IOS和Android移动设备的首页。

**第一大原因： 它是未来，开始用吧！**

　　最大的原因今天你就开始使用HTML5是因为它是未来，不要掉队了！HTML5不会往每个方向发展，但是更多的元素已经被很多公司采用，并且开始着手开发。HTML5其实更像HTML，它不是一个新的技术需要你重新学习！如果你开发XHTML strict的话你现在就已经在开发HTML5了。为什么不更完整的享受HTML5的功能呢？

你实际上没有任何借口不接受HTML5。事实上我唯一一个原因使用HTML5是因为它书写代码简单清晰。其它的特性其实我也没有真正使用。你可以考虑现在开始使用HTML5书写代码，它能帮助你改变书写代码的方式及其设计方式。开始用HTML5代码编写web应用吧，说不定下一个移动应用或者游戏应用就是用HTML5开发的！

# 附件2：

**<article>**标签定义外部的内容。比如来自一个外部的新闻提供者的一篇新的文章，或者来自 blog 的文本，或者是来自论坛的文本。亦或是来自其他外部源内容。  
**HTML5:**<article></article>  
**HTML4:**<div></div>

**<aside>**标签定义 article 以外的内容。aside 的内容应该与 article 的内容相关。  
**HTML5:**<aside>Aside 的内容是独立的内容，但应与文档内容相关。</aside>  
**HTML4:**<div>Aside 的内容是独立的内容，但应与文档内容相关。</div>

**<audio>** 标签定义声音，比如音乐或其他音频流。  
**HTML5:**<audio src="someaudio.wav">您的浏览器不支持 audio 标签。</audio>  
**HTML4:**<object type="application/ogg" data="someaudio.wav"><param name="src" value="someaudio.wav"></object>

**<canvas>** 标签定义图形，比如图表和其他图像。这个 HTML 元素是为了客户端矢量图形而设计的。它自己没有行为，但却把一个绘图 API 展现给客户端 JavaScript 以使脚本能够把想绘制的东西都绘制到一块画布上。  
**HTML5:**<canvas id="myCanvas" width="200" height="200"></canvas>  
**HTML4:**<object data="inc/hdr.svg" type="image/svg+xml" width="200" height="200"></object>

**<command>** 标签定义命令按钮，比如单选按钮、复选框或按钮。  
**HTML5:**<command onclick=cut()" label="cut">  
**HTML4:**none

**<datalist>** 标签定义可选数据的列表。与 input 元素配合使用，就可以制作出输入值的下拉列表。  
**HTML5:**<datalist></datalist>  
**HTML4:**see combobox.

**<details>** 标签定义元素的细节，用户可进行查看，或通过点击进行隐藏。与 <legend> 一起使用，来制作 detail 的标题。该标题对用户是可见的，当在其上点击时可打开或关闭 detail。  
**HTML5:** <details></details>  
**HTML4:**<dl style="display:hidden"></dl>

**<embed>** 标签定义嵌入的内容，比如插件。  
**HTML5:**<embed src="horse.wav" />  
**HTML4:** <object data="flash.swf"  type="application/x-shockwave-flash"></object>

**<figcaption>** 标签定义 figure 元素的标题。”figcaption” 元素应该被置于 “figure” 元素的第一个或最后一个子元素的位置。  
**HTML5:**<figure><figcaption>PRC</figcaption></figure>  
**HTML4:**none

**<figure>** 标签用于对元素进行组合。使用 <figcaption> 元素为元素组添加标题。  
**HTML5:**<figure><figcaption>PRC</figcaption><p>The People's Republic of China was born in 1949...</p></figure>  
**HTML4:** <dl><h1>PRC</h1><p>The People's Republic of China was born in 1949...</p></dl>

**<footer>** 标签定义 section 或 document 的页脚。典型地，它会包含创作者的姓名、文档的创作日期以及/或者联系信息。  
**HTML5:** <footer></footer>  
**HTML4:**<div></div>

**<header>** 标签定义 section 或 document 的页眉。  
**HTML5:** <header></header>  
**HTML4:**<div></div>

**<hgroup>** 标签用于对网页或区段（section）的标题进行组合。  
**HTML5:** <hgroup></hgroup>  
**HTML4:** <div></div>

**<keygen>** 标签定义生成密钥。  
**HTML5:** <keygen>  
**HTML4:** none

**<mark>**主要用来在视觉上向用户呈现那些需要突出的文字。<mark>标签的一个比较典型的应用就是在搜索结果中向用户高亮显示搜索关键词。  
**HTML5:** <mark></mark>  
**HTML4:** <span></span>

**<meter>** 标签定义度量衡。仅用于已知最大和最小值的度量。必须定义度量的范围，既可以在元素的文本中，也可以在 min/max 属性中定义。  
**HTML5:** <meter></meter>  
**HTML4:** none

**<nav>** 标签定义导航链接的部分。  
**HTML5:** <nav></nav>  
**HTML4:**<ul></ul>

**<output>** 标签定义不同类型的输出，比如脚本的输出。  
**HTML5:**<output></output>  
**HTML4:** <span></span>

**<progress>** 标签运行中的进程。可以使用 <progress> 标签来显示 JavaScript 中耗费时间的函数的进程。  
**HTML5:** <progress></progress>  
**HTML4:** none

**<rp>** 标签在 ruby 注释中使用，以定义不支持 ruby 元素的浏览器所显示的内容。  
**HTML5:** <ruby>漢 <rt><rp>(</rp>ㄏㄢˋ<rp>)</rp></rt></ruby>  
**HTML4:** none

**<rt>** 标签定义字符（中文注音或字符）的解释或发音。  
**HTML5:** <ruby>漢 <rt> ㄏㄢˋ </rt></ruby>  
**HTML4:** none

**<ruby>** 标签定义 ruby 注释（中文注音或字符）。  
**HTML5:**<ruby>漢 <rt><rp>(</rp>ㄏㄢˋ<rp>)</rp></rt></ruby>  
**HTML4:** none

**<section>** 标签定义文档中的节（section、区段）。比如章节、页眉、页脚或文档中的其他部分。  
**HTML5:** <section></section>  
**HTML4:** <div></div>

**<source>** 标签为媒介元素（比如 <video> 和 <audio>）定义媒介资源。  
**HTML5:** <source>  
**HTML4:** <param>

**<summary>** 标签包含 details 元素的标题，”details” 元素用于描述有关文档或文档片段的详细信息。”summary” 元素应该是 “details” 元素的第一个子元素。  
**HTML5:** <details><summary>HTML 5</summary>This document teaches you everything you have to learn about HTML 5.</details>  
**HTML4:** none

**<time>** 标签定义日期或时间，或者两者。  
**HTML5:** <time></time>  
**HTML4:** <span></span>

**<video>** 标签定义视频，比如电影片段或其他视频流。  
**HTML5:** <video src="movie.ogg" controls="controls">您的浏览器不支持 video 标签。</video>  
**HTML4:**<object type="video/ogg" data="movie.ogv"><param name="src" value="movie.ogv"></object>