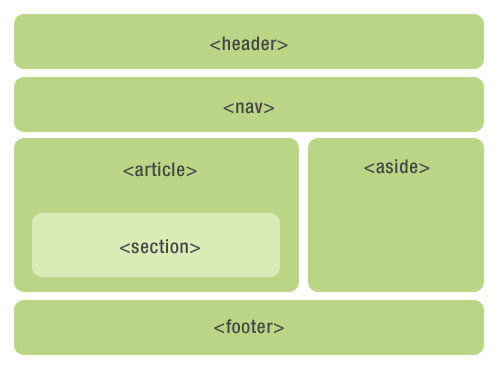
# HTML 5的革新之一：语义化标签一节元素标签。

在HTML 5出来之前，我们用div来表示页面章节，但是这些div都没有实际意义。（即使我们用css样式的id和class形容这块内容的意义）。这些标签只是我们提供给浏览器的指令，只是定义一个网页的某些部分。但现在，那些之前没“意义”的标签因为因为html5的出现消失了，这就是我们平时说的“语义”。

看下图没有用div标签来布局



html5的布局

嗯，如上图那个页面结构没有一个div，都是采用html5语义标签（用哪些标签，关键取决于你的设计目标）。

但是也不要因为html5新标签的出现，而随意用之，错误的使用肯定会事与愿违。所以有些地方还是要用div的，就是因为div没有任何意义的元素，他只是一个标签，仅仅是用来构建外观和结构。因此是最适合做容器的标签。

W3C定义了这些语义标签，不可能完全符合我们有时的设计目标，就像制定出来的法律不可能流传100年都不改变，更何况它才制定没多久，不可能这些语义标签对所以设计目标的适应。只是一定程度上的“通用”，我们的目标是让爬虫读懂重要的东西就够了。

结论：不能因为有了HTML 5标签就弃用了div，每个事物都有它的独有作用的。

节点元素标签因使用的地方不同，我将他们分为：**节元素标签**、**文本元素标签**、**分组元素标签**分开来讲解HTML5中新增加的语义化标签和使用总结。

**header元素**

header 元素代表“网页”或“section”的页眉。  
通常包含h1-h6元素或hgroup，作为整个页面或者一个内容块的标题。也可以包裹一节的目录部分，一个搜索框，一个nav，或者任何相关logo。

整个页面没有限制header元素的个数，可以拥有多个，可以为每个内容块增加一个header元素

<header>

<hgroup>

<h1>网站标题</h1>

<h1>网站副标题</h1>

</hgroup>

</header>

header的示例代码

header使用注意：

* 可以是“网页”或任意“section”的头部部分；
* 没有个数限制。
* 如果hgroup或h1-h6自己就能工作的很好，那就不要用header。

**footer元素**

footer元素代表“网页”或“section”的页脚，通常含有该节的一些基本信息，譬如：作者，相关文档链接，版权资料。如果footer元素包含了整个节，那么它们就代表附录，索引，提拔，许可协议，标签，类别等一些其他类似信息。

<footer>

COPYRIGHT@小北

</footer>

footer的示例代码

footer使用注意：

* 可以是“网页”或任意“section”的底部部分；
* 没有个数限制，除了包裹的内容不一样，其他跟header类似。

**hgroup元素**

hgroup元素代表“网页”或“section”的标题，当元素有多个层级时，该元素可以将h1到h6元素放在其内，譬如文章的主标题和副标题的组合

<hgroup>

<h1>这是一篇介绍HTML 5语义化标签和更简洁的结构</h1>

<h2>HTML 5</h2>

</hgroup>

hgroup示例代码

hgroup使用注意：

* 如果只需要一个h1-h6标签就不用hgroup
* 如果有连续多个h1-h6标签就用hgroup
* 如果有连续多个标题和其他文章数据，h1-h6标签就用hgroup包住，和其他文章元数据一起放入header标签

**nav元素**

nav元素代表页面的导航链接区域。用于定义页面的**主要导航部分**。

<nav>

<ul>

<li>HTML 5</li>

<li>CSS3</li>

<li>JavaScript</li>

</ul>

</nav>

nav实例

但是我在有些时候却情不自禁的想用它，譬如：侧边栏上目录，面包屑导航，搜索样式，或者下一篇上一篇文章，但是事实上规范上说nav只能用在页面主要导航部分上。页脚区域中的链接列表，虽然指向不同网站的不同区域，譬如服务条款，版权页等，这些footer元素就能够用了。

nav使用注意：

* 用在整个页面主要导航部分上，不合适就不要用nav元素；

**aside元素（旁白，离题的话）**

aside元素被包含在article元素中作为主要内容的附属信息部分，其中的内容可以是与当前文章有关的相关资料、标签、名次解释等。（特殊的section）

在article元素之外使用作为页面或站点全局的附属信息部分。最典型的是侧边栏，其中的内容可以是日志串连，其他组的导航，甚至广告，这些内容相关的页面。

<article>

<p>内容</p>

<aside>

<h1>作者简介</h1>

<p>小北，前端一枚</p>

</aside>

</article>

aside实例

aside使用总结：

* aside在article内表示主要内容的附属信息，
* 在article之外则可做侧边栏，没有article与之对应，最好不用。
* 如果是广告，其他日志链接或者其他分类导航也可以用

**section元素**

section元素代表文档中的“节”或“段”，“段”可以是指一篇文章里按照主题的分段；“节”可以是指一个页面里的分组。

section通常还带标题，虽然html5中section会自动给标题h1-h6降级，但是最好手动给他们降级。如下：

<section>

<h1>section是啥？</h1>

<article>

<h2>关于section</h2>

<p>section的介绍</p>

<section>

<h3>关于其他</h3>

<p>关于其他section的介绍</p>

</section>

</article>

</section>

section示例代码

section使用注意：

一张页面可以用section划分为简介、文章条目和联系信息。不过在文章内页，最好用article。section不是一般意义上的容器元素，如果想作为样式展示和脚本的便利，可以用div。

* 表示文档中的节或者段；
* article、nav、aside可以理解为特殊的section，所以如果可以用article、nav、aside就不要用section，没实际意义的就用div

**article元素（文章）**

article元素最容易跟section和div容易混淆，其实article代表一个在文档，页面或者网站中自成一体的内容，其目的是为了让开发者独立开发或重用。譬如论坛的帖子，博客上的文章，一篇用户的评论，一个互动的widget小工具。（特殊的section）

除了它的内容，article会有一个标题（通常会在header里），会有一个footer页脚。我们举几个例子介绍一下article，好更好区分article、section、div

<article>

<h1>一篇文章</h1>

<p>文章内容..</p>

<footer>

<p><small>版权：html5jscss网所属，作者：小北</small></p>

</footer>

</article>

一篇简单文章的article示例代码

上例是最好简单的article标签使用情况，如果在article内部再嵌套article，那就代表内嵌的article是与它外部的内容有关联的，如博客文章下面的评论，如下：

<article>

<header>

<h1>一篇文章</h1>

<p><time pubdate datetime="2012-10-03">2012/10/03</time></p>

</header>

<p>文章内容..</p>

<article>

<h2>评论</h2>

<article>

<header>

<h3>评论者: XXX</h3>

<p><time pubdate datetime="2012-10-03T19:10-08:00">~1 hour ago</time></p>

</header>

<p>哈哈哈</p>

</article>

<article>

<header>

<h3>评论者: XXX</h3>

<p><time pubdate datetime="2012-10-03T19:10-08:00">~1 hour ago</time></p>

</header>

<p>哈？哈？哈？</p>

</article>

</article>

</article>

文章里的评论，一个article嵌套article来表示的实例

article内部嵌套article，有可能是评论或其他跟文章有关联的内容。那article内部嵌套section一般是什么情况呢。如下：

<article>

<h1>前端技术</h1>

<p>前端技术有那些</p>

<section>

<h2>CSS</h2>

<p>样式..</p>

</section>

<section>

<h2>JS</h2>

<p>脚本</p>

</section>

</article>

文章里的章节，一个article里的section实例

因为文章内section部分虽然也是独立的部分，但是它门只能算是*组成整体的一部分*，从属关系，article是大主体，section是构成这个大主体的一部分。本网站的全部文章都是article嵌套一个个section章节，这样能让浏览器更容易区分各个章节所包括的内容。

那section内部嵌套article又有哪些情况呢，如下

<section>

<h1>介绍: 网站制作成员配备</h1>

<article>

<h2>设计师</h2>

<p>设计网页的...</p>

</article>

<article>

<h2>程序员</h2>

<p>后台写程序的..</p>

</article>

<article>

<h2>前端工程师</h2>

<p>给楼上两位打杂的..</p>

</article>

</section>

一个section里的article实例

设计师、程序员、前端工程师都是一个独立的整体，他们组成了网站制作基本配备，当然还有其他成员~~。设计师、程序员、前端工程师就像article，是一个个独立的整体，而section将这些自成一体的article包裹，就组成了一个团体。

article和section和例子就例举这么多了，具体情况具体分析，不易深究。漏了div，其实div就只是想用来把元素组合或者给它们加样式时使用。

article使用注意：

* 自身独立的情况下：用article
* 是相关内容：用section
* 没有语义的：用div

**HTML5其他结构元素标签**

HTML5节元素标签包括body article nav aside section header footer hgroup ，还有h1-h6 address。

* address代表区块容器，必须是作为联系信息出现，邮编地址、邮件地址等等,一般出现在footer。
* h1-h6因为hgroup，section和article的出现，h1-h6定义也发生了变化，允许一张页面出现多个h1。

HTML5语义化标签还有第二篇[HTML 5的革新——语义化标签(二)](http://www.html5jscss.com/html5-semantics-rich-60.html)。我们在构造语义化和结构化的标签时的选择也变得有些不慎重。也就是说，我们不应该滥用超语义化的元素。

HTML 5的革新之一：语义化标签二文本元素标签、分组元素标签。

[HTML 5的革新——语义化标签(一)](http://www.html5jscss.com/html5-semantics-section-30.html)中介绍了一些HTML5新加的一些节元素，一张页面中结构元素构成网页大体，但是也需要其他内容来填充，譬如图片、文本、代码、表单、交互等等。所以接下来我们就来详细了解HTML5的新增加的和重新定义的文本元素标签、分组元素标签等其他元素标签。

## 文本字体元素

文字对SEO影响很大，而HTML5对一些文本字体元素的语义又重新定义了一遍，也增加了一些新的。那我们就来重新认识他们。

* **a**（anchor 的缩写）： 用于定义超链接
* **em**(emphasis 的缩写)：em 是句意强调，加与不加会引起语义变化，也可以理解为局部强调，用在语句某个单词上来改变句子的侧重。
* **strong**：strong表示重要，strong 的强调则是一种随意无顺序的，看见某文时，立刻就凸显出来的关键词句。
* **p**：p元素
* **b**（bold 的缩写）：b 元素原本就是加粗，现在表示“文体突出”文字，通俗将是用来在文本中高亮显示某个或者几个字符，旨在引起用户的特别注意，无强调作用。譬如文档概要中的关键字，评论中的产品名，以及分类名。
* **i**（italic 的缩写）：i 元素原本只是倾斜，现在描述为在普通文章中突出不同意见或语气或其他的一段文本，就像剧本里的话外音（外语、译音），或也可以用做排版的斜体文字。
* **code**：定义计算机代码文本。
* **q**（quote 的缩写）：用于定义一段引用的内容（短内容）
* **cite**：用于定义引用内容出自书籍或杂志等的标题，不允许其他信息，如作者，日期等。
* **u**（underline 的缩写）：定义下划线文本
* **abbr**（abbreviation 的缩写）：定义一个缩写文本，建议在 abbr 的 title 属性中描述缩写的全称
* **dfn**（defining instance 的缩写）：用于定义一个术语
* **var**：定义计算机代码中的变量
* **samp**（sample 的缩写）：由程序输出的示例文本
* **kbd**（keyboard 的缩写）：定义由键盘输入的文本
* **wbr**(word break)的缩写：定义换行的时机
* **span**：没有任何语义
* **br**：定义一个换行符

以上很多只是W3C定义，有些还不支持，但是为了彻底贯彻W3C的语义化，还是慢慢熟悉用起来吧。

文本字体元素标签使用注意:

* 在下面这些元素都不适合的时候：表重要的 strong ，表强调的 em ，表标题的 h1–h6，表高亮或标记文本的 p 等，就用 b 来表示。
* em 的强调是用在语句某个单词上来改变句子的侧重，可以说是局部的，而strong 和局部还是全局无关，局部强调用strong也可以，strong强调的是重要性，不会改变句意。

### time元素

time元素也是文本标签，因为是全新的标签，所以我们单独来介绍。time元素用来标记一篇文章的发布时间。

<time datetime="2012-02-15" pubdate>2012年02月15日</time>

形如如上代码，知道time标签一般有三个组成部分

1. 机器可识别的时间戳：格式必须是年月日的数字以减号相隔，如果增加时间，那就在日期后面加字母T然后跟24小时格式的时间值以及时区偏移量，形如datetime="2012-2-15T22:49:40+08:00"
2. 人可识别的文本内容：格式随意，只要能看懂。
3. 一个可选的pubdata标记：pubdata是个布尔值，如果需要，写上属性名就好pubdata。但是为了美观，我们也可以写成pubdata=""

我们还要注意的是，如果该time位于一个article中，那么它表示这篇文章的发布时间；如果不在article之中表示整个文档的发布时间。

<article>

<header>

<h1>html5jscss网<time datetime="2012-02-14">2月14日</time>成立</h1>

<p>发布 <time datetime="2012-02-15" pubdate>2012年02月15日</time> 分类：前端交流</p>

</header>

<p>你好世界！</p>

</article>

time示例代码

## 分组元素标签

我们熟悉的div、 p 、dl 、dt、dd、ol、ul、li、hr都是分组元素标签，我们接下来看一些不常用的和新加的分组元素标签

* **blockquote**：标记一段长引文。标记短引文（行内引文），应采用 q 元素！
* **pre**：pre 元素可定义预格式化的文本。被包围在 pre 元素中的文本通常会保留空格和换行符。而文本也会呈现为等宽字体。pre 标签的一个常见应用就是用来表示源代码。他跟code的关系好比blockquote和q的关系。

### ol元素

ol元素在HTML5有改良，增加了两个属性：

1. “start”：start属性用来定义列表编号的起始位置，
2. “reversed”：reversed属性表示将列表进行反转，但是目前还没有任何一款浏览器对其提供支持，在这里就不细说了。

### figure元素与figcaption元素

figure元素用来包含一块独立内容，该内容如果被移除掉不会对周围的内容有影响。具体来说它可以用来表示图片，统计图，图表，音频，视频，代码片段等。如果需要你也可以给该内容添加一个标题，这个标题使用figcaption来表示。figcaption只能作为figure元素的子元素，可以放在figure元素内的任何位置。形如：

<figure>

<img src="" alt="" />

<figcaption>html5jscss前端网是刚建立的小站</figcaption>

</figure>

不是所有图片都用figure来包裹，img 标签也有语义的。如果纯粹只是为了呈现的图，也不在文档其他地方引用，那就绝对不要用figure。如果和上下文有关，也可以把它移动到附录，那就别用figure，aside可能适合。

figure元素和aside元素看起来表达的内容差不多，但是aside所能包含的内容比figure要广。当你不知道如何选择的时候可以这样来做：这段内容对周围的内容来说是一个要点，或者很重要，不可少，那么可以使用figure，否则使用aside。

注意：

一个figure元素内最多只允许放置一个figcaption元素，也可以不放，但是其他元素可无限放置。注意不是所有图片都得用figure元素。

## 嵌入元素标签

在页面中除了显示文档活字符外，还需要放入一些其他元素，就是我们所说的嵌入元素。

嵌入元素包括img（图片），页面（iframe），多媒体对象将不再全部绑定在 object 或 embed 标签 中，而是由有 video(视频)audio（音频），用于绘画的 canvas 元素

这里我们不详细介绍它们的属性了，以后专门介绍。

HTML5中增加了一些表单元素和一些交互元素等新的标签，但是着重点不同，所以我们放在另一片文章来继续了解。HTML5语义标签到此结束。

**1、什么是HTML语义化？**

<基本上都是围绕着几个主要的标签，像标题（H1~H6）、列表（li）、强调（strong em）等等>

　　根据内容的结构化（内容语义化），选择合适的标签（代码语义化）便于开发者阅读和写出更优雅的代码的同时让浏览器的爬虫和机器很好地解析。

**2、为什么要语义化？**

* 为了在没有CSS的情况下，页面也能呈现出很好地内容结构、代码结构:为了裸奔时好看；
* 用户体验：例如title、alt用于解释名词或解释图片信息、label标签的活用；
* 有利于[SEO](http://baike.baidu.com/view/1047.htm)：和搜索引擎建立良好沟通，有助于爬虫抓取更多的有效信息：[爬虫](http://baike.baidu.com/view/998403.htm)依赖于标签来确定上下文和各个关键字的权重；
* 方便其他设备解析（如屏幕阅读器、盲人阅读器、移动设备）以意义的方式来渲染网页；
* 便于团队开发和维护，语义化更具可读性，是下一步吧网页的重要动向，遵循W3C标准的团队都遵循这个标准，可以减少差异化。