* **layout**

1. **Doctype声明**

|  |
| --- |
| <!DOCTYPE html> |

doctype，是为了告诉验证器，让验证器按照该doctype来验证我的文档，有效的doctype触发的是标准模式，而不是兼容模式（quiks mode）

按照HTML5的另一个设计原理，它必须向前向后兼容，兼容未来的HTML版本——不管是HTML6、HTML7，还是其他什么——都要与当前的HTML版本，HTML5，兼容。因此，把一个版本号放在doctype里面没有多大的意义。

1. **meta**

|  |
| --- |
| <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  可以简写成  <meta charset="utf-8"> |

斜杠也可以省略

1. **去掉了Javascript和CSS标签的type属性**

在HTML5中，不再需要type属性了

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

1. **script属性**

HTML5中除了具备HTML5标准属性以外，与HTML4.01相比移除了language属性，修改了type属性为可选的（默认text/javascript），并新增了一个属性async。

* **defer**

defer 可选。延迟脚本执行，相当于将script标签放入页面body标签的底部。

HTML4.01中定义的defer属性除IE和较新版本的Firefox外，其他浏览器并未支持。

* **async**

async定义脚本是否异步执行，取值true或false。在head标签内插入script，它可以让脚本与文档同时下载，脚本、文档可用时便执行。

* + - 如果 async 属性为 true，则脚本会相对于文档的其余部分异步执行，这样脚本会可以在页面继续解析的过程中来执行。
* 如果 async 属性为 false，而 defer 属性为 true，则脚本会在页面完成解析时得到执行。
* 如果 async 和 defer 属性均为 false，那么脚本会立即执行，页面会在脚本执行完毕继续解析。
* **现在的script存在的问题：**
* 把它放在head标签内。

在下载script引入的外部脚本时，浏览器处于阻塞状态，网络不好或者script文件过大时，页面处于空白停顿状态，体验是并不够好。

* 把它放入页面底部

这是大家公认的提高前端页面性能和体验的方法，但还是存在一定的问题，放在页面底部的脚本是要等到页面文档流下载完毕才去下载、执行，页面中的交互会存在一个延迟的时间去实现。虽然页面显示时间变短了，但交互却被延后了。体验也不够好。

* 按需执行

在head标签内引入部分公共脚本，在每一个需要交互的HTML元素之后插入script立即执行，需要特定条件才执行的脚本放到页 面最底部。这也不是一个完美的解决方案，一则页面内穿插过多的script标签引起维护不便，二来底部脚本还未加载完时用户便触发了某个条件该怎么办？虽 有方法实现，但体验仍不够好。

1. **是否使用双引号**

HTML5可以省去标签中的双引号

* **表单**

1. **input的type属性**

input type=”number/search/range/email/date/url”，现在的任何浏览器当不支持这些type值时，都会把它解析成type=”text”，所以当使用搜索框时，完全可以写成type=”search”,safai会将其解析成一个系统级的搜索控件，而不识别它的浏览器则显示的是一个文本框，没有其它副作用。这可以实现浏览器平稳退化。

1. **电子邮件输入框**

检测输入的内容是否符合电子邮件的书写格式，到目前为止，包括现代浏览器在内都不支持该属性

|  |
| --- |
| <input id="email" type="email" name="email" /> |

1. **URL输入框**

|  |
| --- |
| <input type="url"></input> |

1. **日期输入框**

|  |
| --- |
| <input type="date"></input> |

1. **select数据绑定**

HTML5将支持data属性，为select控件外联数据源

|  |
| --- |
| <select data="http://domain/getmyoptions"></select> |

1. **必填项属性**

在HTML5中，新增了一个“必须填写”的属性：required, 如果输入框为空，表单将无法提交成功。

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

1. **自动获取焦点**

HTML5也不再需要Javascript来解决输入框的自动获取焦点, HTML5新增了自动获取焦点属性autofocus

autofocus也同样可以写成autofocus=”autofocus”

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

1. **使用正则表达式**

在HTML5中，我们可以直接使用正则表达式。

|  |
| --- |
| <input id="username" type="text" name="username"  placeholder="4 <> 10" required="required"  autofocus="autofocus" pattern="[A-Za-z]{4,10}"> |

1. **占位符**

HTML5中新增了占位符属性placeholder

目前只有Chrome和Safari支持该属性，Firefox和Opera不支持该属性。

|  |
| --- |
| <input type="email" name="email" placeholder="doug@givethesepeopleair.com"> |

**2010-12-30_105652.png 2010-12-30_105704.png**

1. **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。IE8， Safari 4， Firefox 3.5支持这个功能。

* **标签/属性**

1. **<figure>**

<figure> 标签用于对元素进行组合,使代码更加语义化.

"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

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

1. **标题群( hgroup)**

类似于figure, 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> |

1. **Header、footer**

HTML5中可以直接使用<header>和<footer>标签，不需要再给header和footer定义一个div，然后再添加一个id=”header/footer”了。

1. **section article aside nav**

最重要的是它们的语义，跟位置没有关系,即footer不一定要放在底部，放在footer中的内容也可以是署名，文章作者之类的，它只是你使用的一个元素。这个元素并没有说“必须把我放在文档或者分区的下面。”

section-分区 article-文章 aside-侧边栏 nav-导航

使用分区嵌套，还可以重复使用h1 h2 h3等标签。

1. **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

|  |
| --- |
| <p>Do not forget to buy <mark>milk</mark> today.</p> |

1. **time标签**

定义日期或时间，或者两者。

|  |
| --- |
| <p>我们在每天早上 <time>9:00</time> 开始营业。</p>  <p>我在 <time datetime="2008-02-14">情人节</time> 有个约会。</p> |

datetime: 定义元素的日期和时间。如果未定义该属性，则必须在元素的内容中规定日期或时间。

* **媒体**

1. **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。

1. **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'" />  <object data="movie.swf">  <a href="movie.mp4">download</a>  </object>  </video> |

上面的代码中包含了4个不同的层次，用于兼容浏览器：

1. 如果浏览器支持video元素，也支持H264，没什么好说的，用第一个视频。
2. 如果浏览器支持video元素，支持Ogg，那么用第二个视频。
3. 如果浏览器不支持video元素，那么就要试试Flash影片了。
4. 如果浏览器不支持video元素，也不支持Flash，我还给出了下载链接。

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。

1. **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> |

1. **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> |

* **浏览器兼容**

1. **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JS来检测该浏览器是否支持这些属性。

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

1. **IE对HTML5的支持**

IE浏览器目前对HTML5的支持并不好, 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> |

* **其它**

1. **整块内容加链接**

|  |
| --- |
| <a href="/path/to/resource">  <h2>Headline text</h2>  <p>Paragraph text.</p>  </a> |

以前这样写是不合乎规范的，但是HTML5把它规范化了。