

后盾网 人人做后盾

www.houdunwang.com

HTML 5

后盾网 2011-2015

什么是HTML5

- HTML5 将成为 HTML、XHTML 以及 HTML DOM 的新标准。
- HTML5是针对目前web发展对原有HTML的补充与升级，并不是颠覆
- 将元素语义与元素内容呈现结果的影响分开，如去除font标签添加article
- 使用标签完全从标签的语义出发，而不是表现形态

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。

什么是HTML 5

提高可用性和改进用户的友好体验;

有几个新的标签,这将有助开发人员定义重要的内容;

可以给站点带来更多的多媒体元素(视频和音频);

可以很好的替代FLASH和Silverlight;

当涉及到网站的抓取和索引的时候,对于SEO很友好;

将被大量应用于移动应用程序和游戏。

HTML5 优势

1. **WebStorage**: 比Cookies更大、更有弹性的的储存。
2. **Web SQL Database**: 本地端的SQL数据库。
3. **Indexed DB**: Key-value的本地数据库。
4. **Application Cache**: 将部分常用的网页内容cache起来。
5. **WebSocket**: 实时的socket联机。
6. **Web Workers**: 以往JavaScript都是single thread, 透过Worker可以有多个运算。
7. **Notifications**: 原生的提示讯息, 类似像OS X的Growl提示。
8. **Dragn Drop**: HTML元素的拖拉。
9. **File API**: 读取用户本机计算机的内容。
10. **Geolocation**: 地理定位。
11. **Device orientation**: 手持装置的方向。
12. **Speech input**: 语音输入。
13. **New tags**: 新的标签, 像是header、section等。
14. **Application tags**: 也是新的标签, 像是meter、progress等。
15. **Microdata**: 加入语义的数据让搜索引擎等网站可以正确显示。
16. **Form type**: form可以加入的type便多了, 包含email和tel等属性
17. **Audio video**: 影片和音乐的原生播放支持。
18. **Canvas**的绘图功能支持。
19. **CSS 3**技术

HTML5 主要包括部分

Html5在chrome,ie9+, firefox, safai等浏览器都表现良好。在chrome与firefox及safai表示优秀。



HTML5支持的浏览器

原有的html中有一些标签造成效果与表现的混淆，所以在html5中废除了，不建议使用。

废除标签：

- `big center font s u strike frame frameset noframes mqrquee`

HTML5 中废除的标签

DOCTYPE

新写法: `<!DOCTYPE html>`

meta设置编码

新写法: `<meta charset="utf-8">`

link标签

新写法: `<link rel="stylesheet" src="hd.css"/>`

script标签

可以不用写type: `<script>...</script>`

Html旧标签的改变

contenteditable属性

- 修改页面中的标签内容

示例：

```
<p contenteditable="true" style="width:300px;height:300px;border:solid 1px #f00f00;"></p>
```

hidden属性

- 隐藏元素,hidden属性加上之后可以隐藏属性，通过js可以去掉hidden元素让其显示

示例：

```
<p hidden style="width:300px;height:300px;border:solid 1px #f00f00;"></p>
```

全局属性

article

- article 字面意思为“文章”。在web页面中表现为独立的内容，如一篇新闻，一篇评论，一段名言，一段联系方式
- 代码独立的、完整的相关内容块，一般来说， article 会有标题部分(通常包含在 header 内)，有时也会 包含 footer

示例：

1. <!doctype html>
2. <title>html5站案例</title>
3. <body>
4. <article>
5. <h1></h1>
6. <p>主内容</p>
7. </article>
8. <aside>
9. 侧边内容
10. </aside>
11. </body>

article标签

section

- section 字面上理解为“块”，“部分”。在html5网页中表现的意思跟字面理解差不多，即部分，块，模块，主要作用为对页面的内容进行分块或者对文章的内容进行分段。

示例：

1. <!doctype html>
2. <article>
3. <h1>Web编程语言比较</h1>
4. <p>web编程语言常用的有asp.net,php,jsp...</p>
5. <section>
6. <h2>asp</h2>
7. <p>asp全称Active Server Page</p>
8. </section>
9. <section>
10. <h2>php</h2>
11. <p>动态语言，免费，强大</p>
12. </section>
13. </article>

section标签

- `div section article`，语义是从无到有，逐渐增强的。`div`无任何语义，仅仅用作样式化或者脚本，对于一段主题性的内容，则就适用 `section`，而假如这段内容可以脱离上下文，作为完整的独立存在的一段内容，则就适用 `article`。
- 当一个标签只是为了样式化或者方便脚本使用时，应该使用 `div`
- `div`是用来布局网页，划分大的区块，在其中定义`article`、`session`
- 将内容划分为几块（段）时使用`session`，
- 内容独立完成时使用`article`
- `article`是文章，文章就是一段完整的独立的内容
- `section`就是块，某种意义上可以理解为`div`，但是比`div`的意思更加明确一点
- `section`和`div`都可以对内容进行分块，但是`section`是进行有意义的分块，无意义的分块应该由`div`来做，例如用作设置样式的页面容器
- 新的HTML 标签 `header`, `footer`, `dialog`, `aside`, `figure` 等的使用，将使内容创作者更加语义地创建文档，之前的开发者在实现这些功能时一般都是使用`div`
- 就好比以前容器都是`div`，但是在HTML里，头部的容器就是`header`，脚部的容器就是`footer`，这些标签功能上和`div`并没有什么区别，但是可以增加代码的可读性。

article、section、div的异同

nav

- 页面导航的链接组

注：

- 并不是所有链接都要放在nav中，只需要将主要的链接放到nav中

适用场合：

1. 传统顶部导航条（首页，关于我们等）
2. 商城左侧导航条
3. 页面分页部分

示例：

```
<nav>  
    <a href="">首页</a>  
    <a href="">关于后盾网</a>  
</nav>
```

新增标签nav

aside

- aside字面理解为“旁边”，在html5中范围更广一点，是跟主内容相关，但是又可以独立的内容，可以是广告、引用、侧边栏等等。

适用场合：

1. 广告、引用、侧边栏

示例：

```
<article>
  <h2>新闻列表</h2>
  <ul>
    主内容
  </ul>
</article>
<aside>
  <section>
    <h3>Html5最新动态</h3>
  </section>
  <section>
    <h3>Html5新增元素</h3>
  </section>
  <section>
    <h3>Html5新增Api</h3>
  </section>
  <section>
    <h2>Html5文章推荐</h2>
  </section>
</aside>
<footer></footer>
```

新增标签aside

time

- time 字面理解为“时间”，在html5中用于定义元素的时间、日期或者日期时间。

注：

- pubdate表示发布日期，让搜索引擎知道是时间，而且是发布时间

示例：

```
<article>
```

```
  <h2>快学网上线了</h2>
```

```
  <p>今天是今天是<time datetime="2014-03-06">2014-3-6</time>
```

```
    <time datetime="2014-02-25" pubdate="pubdate">2014年2月25日</time>
```

```
</article>
```

Time标签

header

- header字面意思为头部，在html5可以理解为页头，文档头，页眉。

示例：

```
<header>
  <h1>Web编程语言比较</h1>
  <nav>
    <ul>
      <li><a href="#asp">asp</a></li>
      <li><a href="aspnet">asp.net</a></li>
      <li><a href="php">php</a></li>
    </ul>
  </nav>
</header>
```

header标签

footer

- footer字面意思为“页脚”，“结尾”，在html5中跟字面意思类似，是页脚和结尾的意思。它与header是相对的一对。footer可以包含该区块相关的作者信息、相关文档的链接、版权信息、导航、附录，索引，长的版本记录，冗长的许可协议和其他的诸于此类的内容等。

示例：

```
<footer>
  <section>
    <h1>友情链接</h1>
  </section>
  后盾网版权所有 copyright houdunwang.com
</footer>
```

footer标签

address

- address字面理解为“地址”，在html5中，它用于文档或者文章作者或拥有者的联系信息。注意，这里放的不是字面上理解的“地址”，而是指“联系信息”，可以包括文档创建者的名字、站点链接、电子邮箱、真实地址、电话号码等各类联系信息。

示例：

```
<footer>
  后盾网
  <address>
    <ul>
      <li>网址：http://www.houdunwang.com</li>
      <li>电话：010-64825057</li>
      <li>网址：houdunwang.com</li>
    </ul>
  </address>
</footer>
```

address标签

code

- 用于显示程序代码。

示例：

```
<code>  
    &lt;?php echo "hello word"?&gt;  
</code>
```

code标签

mark

- 突出显示文本,标记,默认是黄色,可以通过style改变背景色

示例:

后盾网 <mark>houdunwang.com</mark>

mark标签

ins与del

- Ins与del用于显示突出或删除的文本

示例：

后盾网址：<ins>houdunwang.com</ins> 下划线

百度网址：baidu.com 删除线

ins与del标签

figure

- 标签规定独立的流内容（图像、图表、照片、代码等等）

示例：

```
<figure>  
  <figcaption>刘德华毕业作品展示</figcaption>  
    
</figure>
```

figure标签

colgroup与col

- 表格是使用tr定义的所以定义行样式很方便，但定义列就不方便了，而colgroup与col就是为了解决定义列表样式产生的
- colgroup 是把列分为组
- col是列
- span属性是跨列，如果span是3就跨3列，从当前位置开始

示例：

```
<style type="text/css">
    #colgroup1{background: #f00f00;}
    #col1{background:blue;}
    .col2{background:#8a727d;}
</style>
<table width="100%" border="1">
    <colgroup id="colgroup1" span="1" ></colgroup>
    <colgroup id="colgroup2">
    <col id="col1" span="1"/>
        <col class="col2"/>
    </colgroup>
    <tr>
        <th>php</th>
        <th>smarty</th>
        <th>hdphp</th>
    </tr>
</table>
```

colgroup与col标签

optgroup

- 对select标签中的option进行分组

示例：

```
<select name="select" id="select">
  <optgroup label="城市">
    <option value="beijing">北京</option>
    <option value="nanjing">南京</option>
  </optgroup>
  <optgroup label="sex">
    <option value="boy">男</option>
    <option value="girl">女</option>
  </optgroup>
</select>
```

optgroup标签

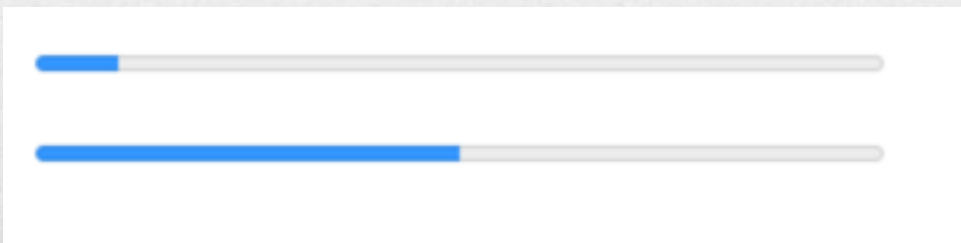
progress

- 显示完成的进度

示例：

```
<progress value="10" max="100"></progress>
```

```
<progress value="50" max="100"></progress>
```



progress标签

meter

- 度量给定范围内的数据

示例：

密码: <input type="password" id="password" />

```
<meter id="meter" value="0" min="1" low="3" optimum="15" high="8" max="10" ></meter>
```

```
<script>
```

```
    document.getElementById("password").onkeyup=function(){  
        document.getElementById("meter").value=this.value.length;  
    };
```

```
</script>
```

- optimum 最佳值

如果该值高于 "high" 属性，则意味着值越高越好。如果该值低于 "low" 属性的值，则意味着值越低越好。

- min 最小值

min 属性的值必须小于 max 属性的值。如果没有规定 min 属性的值，则默认值是 0。提示：min 属性与 max 属性一起规定了度量的完整范围。

- max 最大值

- low 低的值

low 属性的值必须大于 min 属性的值，且必须小于 high 和 max 属性的值。

- high

被界定为高的值。high 属性的值必须小于 max 属性的值，且必须大于 low 和 min 属性的值。

meter标签



details

- 标签用于描述文档或文档某个部分的细节。

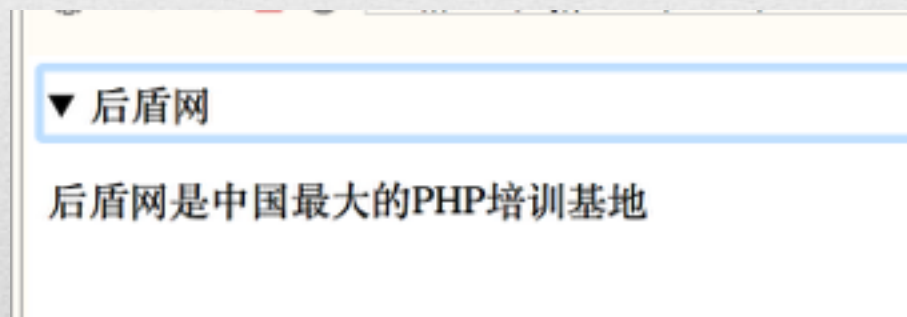
示例：

```
<details>
```

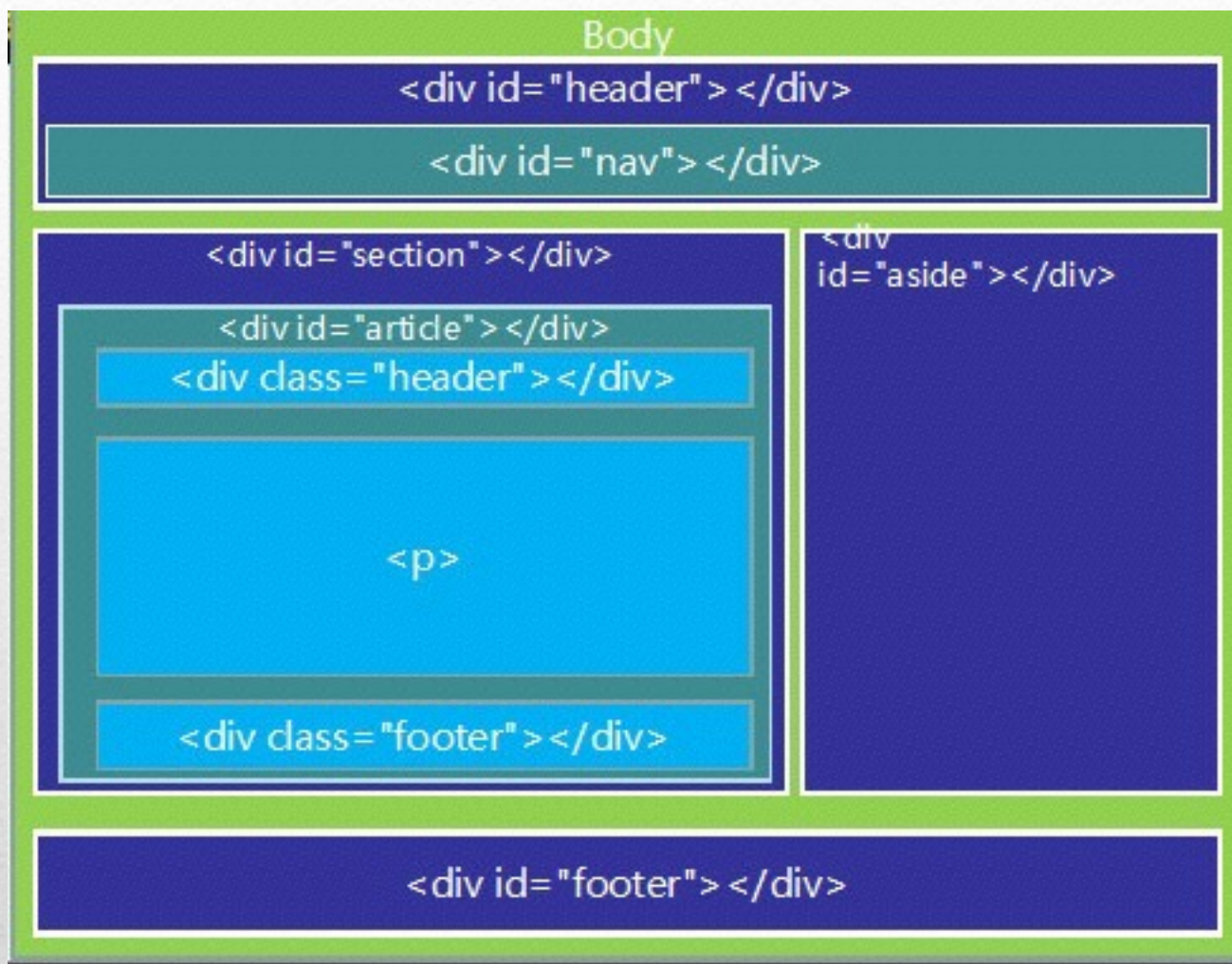
```
    <summary>后盾网</summary>
```

```
    <p>后盾网是中国最大的PHP培训基地</p>
```

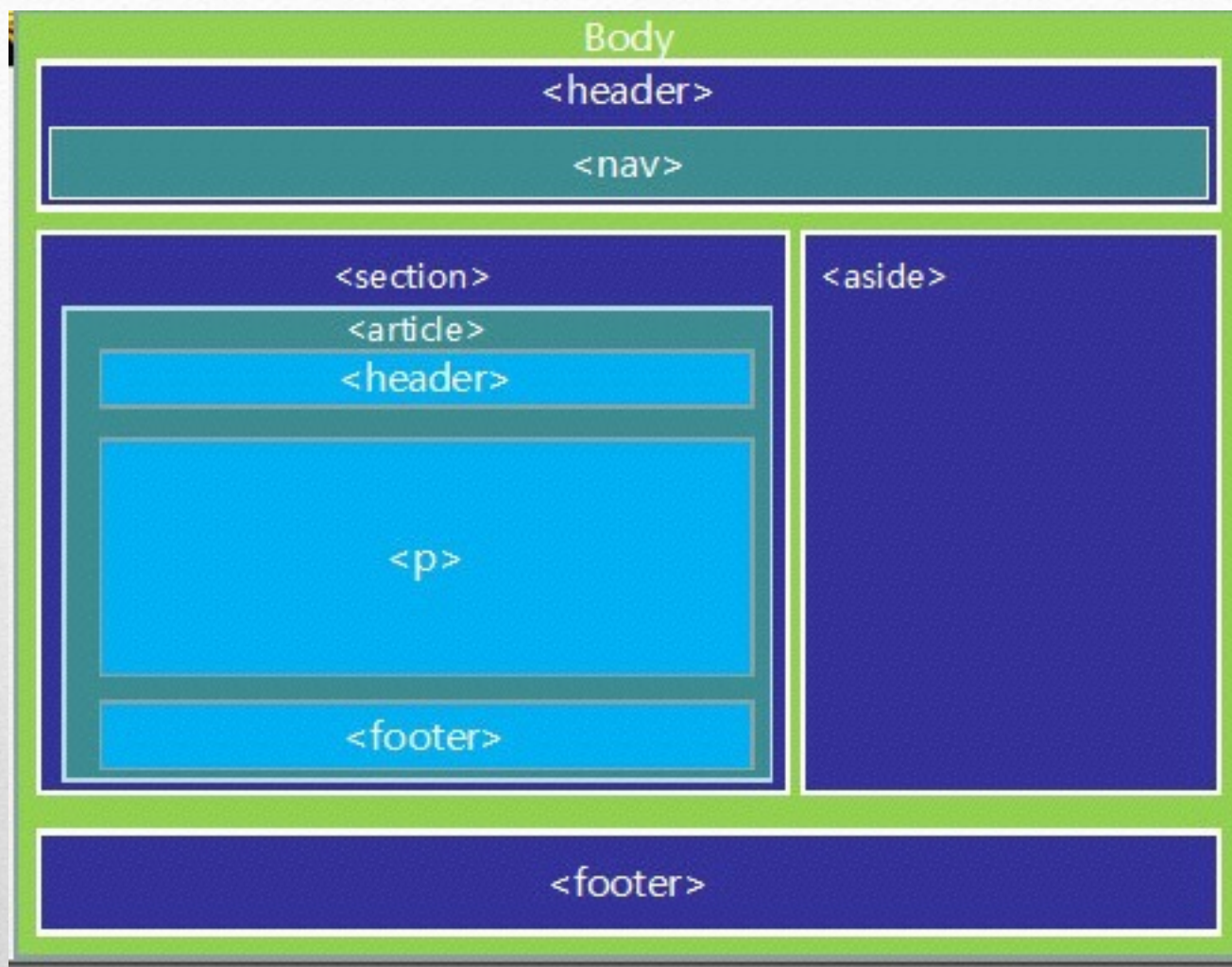
```
</details>
```



details标签



原来布局



html5布局

Div和新结构标签的区别和意义

DIV

mz

yf

xz

HTML5

帽子

衣服

鞋子

区别

兼容器

- 越来越多的站点开始使用 HTML5 标签。但情况是还有很多人在使用 IE6, IE7, IE8。为了让所有网站浏览者都能正常的访问网站使用 html5shiv.js 可以解决这个问题。

示例：

```
<!--[if lt IE9]>  
    <script src="http://cdn.bootcss.com/html5shiv/r29/html5.min.js"></script>  
<![endif]-->
```

解决低版本浏览器兼容器

JSON.parse

- parse用于从一个字符串中解析出json对象
- 注意：属性必须有引号

示例：

```
<script>  
    var json = '{"name":"后盾网","url":"http://www.houdunwang.com}';  
    alert(JSON.parse(json).name);  
</script>
```

JSON.stringify

- 用于从一个JSON对象解析出字符串

示例：

```
<script>  
    var json = '{"name":"后盾网","url":"http://www.houdunwang.com}';  
    var obj = JSON.parse(json);  
    alert(JSON.stringify(obj));  
</script>
```

JSON处理

defer属性

- defer 属性规定是否对脚本执行进行延迟，直到页面加载为止

示例：

```
<script src="9.js" defer="defer"></script>
```

async属性

- 异步加载外部JS文件

示例：

```
<script src="9.js" async="async"></script>
```

Js加载处理



蓝色线代表网络读取，红色线代表执行时间，这俩都是针对脚本的；绿色线代表 HTML 解析。

Js加载处理

练习一下各种新标签

练习
