

新增的主体结构元素

article元素

博客,文章 论坛帖子 一段用户评论 独立的插件 或其他任何独立的内容

可以嵌套使用

可以表示插件

```
1 <article>
2   <header>
3     <h1>我的世界</h1>
4     <p>欢迎来到我的世界</p>
5   </header>
6   <article>
7     <header>
8       <p>作者</p>
9     </header>
10    <p>评论</p>
11    <footer>time</footer>
12  </article>
13  <p>Hello</p>
14  <footer>
15    <p>这是底部</p>
16  </footer>
17 </article>
18
19 <article>
20   <h1>这是一个内嵌页面</h1>
21   <object>
22     <embed src="#" width="600" height="400"></embed>
23   </object>
24 </article>
```

section元素

用于网站或应用程序中页面的内容进行分块.

一个section元素通常由内容及标题组成.但section元素并非一个普通的容器元素,当一个容器需要被直接定义样式或通过脚本定义行为时,推荐使用div而不是元素

```
1 <section>
2   <h1>苹果</h1><!-- 不推荐对没有标题内容的进行分块-->
3   <p>这是一个苹果,可以吃,而且很好吃</p>
```

```

4 </section>
5
6 <article>
7   <h1>苹果</h1>
8   <p>这是一个苹果,可以吃,而且很好吃</p>
9   <section>
10     <h2>红富士</h2>
11     <p>这是代表一个很红的苹果,吃起来也不错</p>
12   </section>
13   <section>
14     <h2>国光</h2>
15     <p>这是代表一个很红的苹果,吃起来也不错</p>
16   </section>
17 </article>
18
19
20
21 section 与 article的区别
22
23 article 强调独立性, section强调分块 不要将section当作设置样式的容器,用div可以实现
24 不要对没有标题的内容使用section元素
25 article元素 set元素 nav 元素更符合 就不用section元素,(有更好的,不要使用section)
26 <section>
27   <h1>水果</h1>
28   <article>
29     <h2>苹果</h2>
30     <p>内容</p>
31   </article>
32   <article>
33     <h2>苹果</h2>
34     <p>内容</p>
35   </article>
36   <article>
37     <h2>苹果</h2>
38     <p>内容</p>
39   </article>
40 </section>
41
42
43

```

nav元素

是一个可以用作页面导航的连接组,其中的导航元素链接到其他页面或当前页面的其它部分. 并不是所有的连接组都要被放进nav元素,只需要将主要的 基本的连接组放进nav元素即可

nav元素的应用场景

传统导航条

侧边栏导航

页内导航

翻页操作

```
1  <nav>
2      <ul>
3          <li><a href="#">主页</a></li>
4          <li><a href="#">开发文档</a></li>
5      </ul>
6  </nav>
7
8  <article>
9      <header>
10         <h1>html5</h1>
11         <nav>
12             <ul>
13                 <li><a href="#">Html5的历史</a></li>
14                 <li><a href="#">Html5的发展</a></li>
15             </ul>
16         </nav>
17     </header>
18     <section>
19         <h1>html5的历史</h1>
20         <p>....</p>
21     </section>
22     <section>
23         <h1>CSS3的历史</h1>
24         <p>....</p>
25     </section>
26     <footer>
27         <a href="#">删除</a>
28         <a href="#">修改</a>
29     </footer>
30 </article>
31 <footer>
32     <p><small>版权声明:</small></p>
33 </footer>
```

aside元素

aside元素 用来表示当前页面或文章的附属信息部分,引用 侧边栏 广告 导航条 以及其他类似

的有区别于主要内容部分.

```
1  用法:
2  1.在article内部使用
3  2.在article外部使用 侧边栏 文章列表等等
4  <header>
5      <h1>js入门</h1>
6  </header>
7  <article>
8      <h1>语法</h1>
9      <p>正文</p>
10     <aside>
11         <h1>名词解释</h1>
12         <p>语法:对语言来说很重要的内容体</p>
13     </aside>
14 </article>
15 <aside>
16     <nav>
17         <h2>评论</h2>
18         <ul>
19             <li><a href="#">2015-03-15</a></li>
20             <li><a href="#">
21                 大牛:真希望能好好的学一下.
22             </a></li>
23         </ul>
24     </nav>
25
```

time元素与微格式

```
1  <time datetime="2015-10-10">2015-10-10 </time>
2  <time datetime="2015-10-10T20:00">2015-10-10 </time>
3  <time datetime="2015-10-10T20:00Z">2015-10-10 </time>
4  <time datetime="2015-10-10T20:00+09:00">2015-10-10 </time>
5
6  pubdate属性(当前具体的发布时间)
7  可选属性,boolean类型的属性
8
9
```

新增的非主体结构元素

header元素

引导和导航作用的结构元素.放置整个页面或页面内的一个内容区块的标题,但是也可以包含



这图这么模糊
你该不是从粪坑里捞上来的吧

其他内容例如数据表格,搜索表单 或相关的logo图片

```
1  在html5之前的写法是这样
2
3  <div class="header"></div>
4  <div class="content"></div>
5  <div class="footer"></div>
6
7  现在的写法 直接写一个header
8  <header>
9      <h1>hhh</h1>
10     <h2>hhhh</h2>
11 </header>
12 <article>
13     <!--header 不仅仅只使用一次,还可以表示一个区域块的头部-->
14     <header>
15         <h1>Hello</h1>
16     </header>
17 </article>
18
19
20 书写一个比较完整的例子
21 <header>
22     <h1>it最新技术</h1>
23     <a href="http://www.baidu.com">百度一下</a>
24     <nav>
25         <ul>
26             <li><a href="#">学习</a></li>
27             <li><a href="#">技术</a></li>
28             <li><a href="#">好的</a></li>
29         </ul>
30     </nav>
31 </header>
```

footer元素

可以作为其上层父级内容区块或是一个根区块的脚注.footer通常包括其相关区块的脚注信息,如作者 相关阅读链接及版权信息

```
1  以前的书写方法
2  <div class="footer">
3      <ul>
4          <li><a href="#">版权信息</a></li>
5          <li><a href="#">站点地图</a></li>
6          <li><a href="#">联系方式</a></li>
7      </ul>
8  </div>
```

```
10 现在
11 <footer>
12     <ul>
13         <li><a href="#">版权信息</a></li>
14         <li><a href="#">站点地图</a></li>
15         <li><a href="#">联系方式</a></li>
16     </ul>
17 </footer>
```

```
18
19 同样可以用做一个区域块的底部
```

hgroup元素

是将标题及其子标题进行分组的元素.hgroup元素通常会将h1~h6元素进行分组,譬如一个内容区块的标题及其子元素算一组

```
1  <article>
2      <header>
3          <hgroup>
4              <h1>这是文章标题</h1>
5              <h2>子</h2>
6          </hgroup>
7          <p><time datetime="2015-11-11">2015-11-11</time></p>
8      </header>
9
10     <div>
11         这是内容
12     </div>
13     <footer>
14         这是底部
```

```
15     </footer>
16 </article>
```

address元素

用来在文档中呈现联系信息,包括文档作者或文档维护者的名字 他们的网站链接 电子邮箱或真实地址电话号码,不知这些: 还用来展示根文档相关的联系人的所有联系信息

```
1 <address>
2     <a href="#">iwen</a>
3     <a href="#">ime</a>
4     <a href="#">iwen</a>
5
6 </address>
7 <footer>
8     <div>
9         <address>
10             <a href="#">iwen</a>
11             海淀大厦
12         </address>
13         <time datetime="2015-10-10">2015-10-10</time>
14     </div>
15 </footer>
16
17
```

```
1 网页编排demo
2 <!DOCTYPE html>
3 <html>
4     <head>
5         <meta charset="UTF-8">
6         <title>网页编排实例</title>
7     </head>
8     <body>
9         <header>
10             <h1>网页标题</h1>
11             <nav>
12                 <ul>
13                     <li><a href="#">首页</a></li>
14                     <li><a href="#">帮助</a></li>
15                 </ul>
16             </header>
```

```

17     <article>
18         <hgroup>
19             <h1>文章主标题</h1>
20             <h2>文章副标题</h2>
21
22         </hgroup>
23         <p>文章正文</p>
24         <section>
25             <div>
26                 <article>
27                     <h1>评论标题</h1>
28                     <p>评论正文</p>
29                 </article>
30             </div>
31         </section>
32
33     </article>
34     <footer>
35         <small>版权所有</small>
36     </footer>
37 </body>
38 </html>

```

表单新增元素与属性

表单内元素的form属性

在html4中,表单内的从属元素必须书写在表单内部,而在html5中,可以把他们书写在页面的任何一个地方,然后为该元素制定以一个form属性,属性值为该表单的id 这样就可以声明该元素从属于指定表单了.

```

1  这是以前的写法
2
3  <form id="testform">
4      <input type="text"/>
5      <textarea form="testform"></textarea>
6  </form>
7  这是现在的写法
8
9  将textarea移到表单之外也是可以的
10 <form id="testform">
11     ...
12 </form>
13 <textarea form="testform"></testarea>

```


表单内元素的formaction属性

在html4中,一个表单内的所有元素只能通过表单的action属性被同一提交到另一个页面,而在html5中可以为所有的提交按钮,增加不同的formaction属性,使得单击不同的按钮时可以将表单提交到不同的页面

```
1 <form id="testform">
2   <input type="submit" name="s1" value="v1" formaction="xx.jsp">提交到xx.jsp
   页面</input>
3   <input type="submit" name="s1" value="v1" formaction="xx.jsp">提交到xx.jsp
   页面</input>
4   <input type="submit" name="s1" value="v1" formaction="xx.jsp">提交到xx.jsp
   页面</input>
5
6 </form>
7
8
9 而以前是不这样添加的
10 而是
11 <form id="testform" action="xx.jsp">
12   ...
13 </form>
```

表单内元素的formmethod属性

在html4中 一个表单内只能有一个action属性用来堆表单内所有元素统一指定提交页面,所以每个表单内页只有一个method属性来统一指定提交方法.在html5中 可以使用formmethod属性来对每一个表单元素分别指定不同的提交方法.

```
1 <form id="testform">
2   <input type="submit" name="s1" value="v1" formmethod="post"
   formaction="xx.jsp">提交</input>
3   <input type="submit" name="s2" value="v2" formmethod="get"
   formaction="xx.jsp">提交</input>
4
5
6 </form>
7 post 和get有哪些区别??自己学习.
```

表单内元素的formenctype属性

在html4中 表单元素具有一个enctype属性,该属性用于指定在表单发送到服务器之前应如何堆表单内的数据进行编码

在html5中,可以使用formenctype属性堆表单元素**分别**指定不同的编码方式

```
1 <form>
2   <input type="test" enctype="text/plain"><!-- 表单中的空格转换为+号,但不
  对表单数据中的数据进行编码-->
3   <input type="test" enctype="multipart/form-data"><!-- 不对字符进行编码,在
  使用包含文件上传空间的表单时 必须使用get值???-->
4   <input type="test" enctype="application/x-www-form-urlencoded"><!-- 发
  送前编码所有字符,当表单元素的action为get时,浏览器则用当前编码方式 将
  数据转化为字符-->
5
6 </form>
```

表单内元素的formtarget属性

在html4中,表单元素具有一个target属性,该属性用于指定在何处打开表单提交后所需要加载的页面

在html5中,可以对多个提交按钮分别使用formtarget属性来指定提交后在何处打开所需加载的页面.

可以对每一个提交按钮分别使用formtarget属性来指定...

```
1 <form id="testform">
2   <input type="submit" name="s1" value="v1" formtarget="_blank"/>
3   <input type="submit" name="s2" value="v2" formtarget="_self"/>
4   <input type="submit" name="s3" value="v3" formtarget="_parent"/>
5   <input type="submit" name="s1" value="v1" formtarget="_top"/>
6   <input type="submit" name="s1" value="v1" formtarget="frameName"/>
7 </form>
```

表单内元素的autofocus属性

为文本框 选择框或按钮控件加上autofocus属性,当画面打开时,该空间自动获得光标焦点.

```
1 <form>
2   <input type="text"/>
3   <input type="text" autofocus/>
4 </form>
```

表单内元素的required属性

html5中新增的required属性可以应用在大多数输入元素上,在提交时,如果元素中内容为空白,则不允许提交,同时在浏览器中显示信息提示文字.

```

1 <form>
2   <input type="text" required="required"/>
3   <button type="submit">提交</button>
4 </form>

```

表单内元素的labels属性

在html5中, 为所有可使用标签的表单元素, button, select元素等, 定义一个labels属性, 属性值为一个NodeList对象, 代表该元素所绑定的标签元素所构成的集合.

```

1 <form id="testform">
2   <script>
3     function Valuedate() {
4       var txtName = document.getElementById("txt_name");
5       var button= document.getElementById("btnValidate");
6       var form = document.getElementById("testform");
7       if(txtName.value.trim()=="") {
8         var label = document.createElement("label");
9         label.setAttribute("form", "txt_name");
10        form.insertBefore(label, button);
11        txtName.labels[1].innerHTML = "请输入姓名";
12        txtName.labels[1].setAttribute("style", "font-
size:9px;color:red");
13      }
14    }
15  </script>
16  <label id="label" for="txt_name">姓名</label>
17  <input id="txt_name"/>
18  <input type="button" id="btnValidate" value="验证" onclick="Valuedate()"/>
19 </form>

```