1. **语义化标签**

**<header> </header> ：用在页面的头部或者版块的头部**

**<footer> </footer> ：用在页面的头部或者版块的底部**

**<section> </section> ：板块标签；**用来划分区域

**<nav></nav> ：导航标签**

**<hgroup></hgroup> ：标题组合标签；例：**<hgroup><h1>妙味课堂</h1><h2>带您进入富有人情味的IT培训</h2></hgroup>

**<article> </article> ：在页面中表示一套结构完整且独立的内容部分**（可以用来呈现论坛的一个帖子，杂志或报纸中的一篇文章，一篇博客，用户提交的评论内容，可互动的页面模块挂件等）

**<aside></ aside> ： 元素标签可以包含与当前页面或主要内容相关的引用、侧边栏、广告、nav元素组，以及其他类似的有别与主要内容的部分**

aside 的内容应该与 article 的内容相关。

被包含在<article>中作为主要内容的附属信息部分，其中的内容 以是与当前文章有关的引用、词汇列表等

在<article>之外使用，作为页面或站点全局的附属信息部分；最典型的形式是侧边栏(sidebar)，其中的内容可以是友情链接、附属导航或广告单元等。

**<figure> < figure > ： 用于对元素进行组合。一般用于图片或视频**

**<figcaption> <figcaption> figure的子元素 用于对figure的内容 进行说明**

例：<figure>

<img src=“miaov.png”/>(注意没有alt)

<figcaption> 妙味课堂 photo &copy 妙味趣学</figcaption>

</figure>

**<time></time> 用来表现时间或日期**

例：<p> 我们在每天早上 <time>9:00</time> 开始营业。 </p>参数

<p> 我在 <time datetime="2008-02-14">情人节</time> 有个约会。 </p>

**<datalist></datalist> 选项列表。与 input 元素配合使用，来定义 input 可能的值。（提示信息）**

例：<input type="text" list="valList" /> （红色部分要对应）

<datalist id="valList">

<option value="javascript">javascript</option>

<option value="html">html</option>

<option value="css">css</option>

</datalist>

**<details></details>用于描述文档或文档某个部分的细节**

该元素用于摘录引用等 应该与页面的主要内容区分开的其他内容

Open 属性默认展开

**< summary></summary> details 元素的标题**

例：<details>

<summary>妙味课堂</summary>

<p>国内将知名的IT培训机构</p>

</details>

**<dialog></dialog>定义一段对话**

Open 属性默认展开

例：<dialog>

<dt>老师</dt>

<dd>2+2 等于？</dd>

<dt>学生</dt>

<dd>4</dd>

<dt>老师</dt>

<dd>答对了！</dd>

</dialog>

**address></address> 定义文章 或页面作者的详细联系信息**

**<mark></mark> 需要标记的词或句子**（可以通过改变background改变标记颜色）

**<keygen>给表单添加一个公钥**

例：<form action="http://www.baidu.com" method="get">

用户: <input type="text" name="usr\_name" />

公钥: <keygen name="security" />

<input type="submit" />

</form>

**<progress><progress>定义进度条**

例：<progress max="100" value="76"> （max为总进度可以为任意值，value为当前进度）

<span>76</span>% （此处span是为了向下兼容，在谷歌下不显示，在ie下显示）

</progress>

1. **新增表单**

**新的输入型表单控件,<input type=”” />**

**email**  : 电子邮箱文本框（跟普通的没区别，当输入不是邮箱的时候，验证通不过，移动端的键盘会变成英文键盘）

**tel**  : 电话号码（跟普通的没区别，不验证，移动端的键盘会变成数字键盘）

**url** : 网页的URL（跟普通的没区别，当输入不是网址的时候，验证通不过，移动端的键盘会变成英文键盘）

**search** : 搜索引擎（跟普通的没区别，chrome下输入文字后，会多出一个关闭的X）

**range** : 特定范围内的数值选择器属性：min（起始值）、max（终止值）、step( 步数 )、value（当前值）

**number** : 只能包含数字的输入框  属性：min（最小值）、max（最大值）、value（当前值）

**color**  : 颜色选择器 

**datetime** : 显示完整日期(utc为格林尼治时间)

**datetime-local** : 显示完整日期，不含时区（此处是当地时间，没有utc了）

**time** : 显示时间，不含时区 （只有小时和分钟数）

**date** : 显示日期

**week**  : 显示周

**month** : 显示月

**新的表单特性和函数**

**placeholder**  : 输入框提示信息 例 : 微博的密码框提示

**autocomplete** : 是否保存用户输入值，用作再次输入时的提示 （默认为on，关闭提示选择off）

**autofocus** : 指定表单获取输入焦点

**list和datalis**t : 为输入框构造一个选择列表 （list值为datalist标签的id）

**required**  : 此项必填，不能为空，用作验证。

**Pattern** : 正则验证 pattern="\d{1,5}“ ，用作验证输入格式是否满足要求

**Formaction**  ：在submit里定义提交地址 ，例：保存至草稿箱；（原来只能在form标签中定义提交地址）

**表单验证**

validity对象，通过下面的valid可以查看验证是否通过，如果八种验证都通过返回true，一种验证失败返回false

Invalid事件 : 验证反馈 input.addEventListener('invalid',fn,false)

阻止默认验证：ev.preventDefault()

var oText=document.getElementById("text");

oText.addEventListener("invalid",fn,false);

function fn()

{

alert(this.validity.valid);

ev.preventDefault();

}

var oText=document.getElementById("text");

oText.addEventListener("invalid",fn,false);

function fn()

{

alert(this.validity.valueMissing);

ev.preventDefault();

}

valueMissing : 输入值为空时，返回true

typeMismatch : 控件值与预期类型不匹配，返回true

patternMismatch : 输入值不满足pattern正则，返回true

tooLong : 超过maxLength最大限制，返回true

rangeUnderflow : 验证的range最小值，返回true

rangeOverflow：验证的range最大值，返回true

stepMismatch: 验证range 的当前值 是否符合min、max及step的规则，返回true

customError 不符合自定义验证

setCustomValidity(); 自定义验证

var oText=document.getElementById("text");

oText.addEventListener("invalid",fn,false);

oText.oninput=function(){

if(this.value=="敏感词")

{

this.setCustomValidity("请不要输入敏感词");

}

else

{

this.setCustomValidity("");

}

};

function fn(){

alert(this.validity.customError);

ev.preventDefault();

}

**formnovalidate属性 : 关闭验证 （一般给保存草稿时加）**

1. **新特性**

**新增js选择器**

1. 通过class名获取一组元素 document/元素.getElementsByClassName('class名称') （类数组）

2. 通过CSS 选择器获取一个元素 document/元素.querySelector('css选择器'); 如果选择一组的话，只能选中一组中的第一个元素

3. 通过CSS 选择器获取一组元素 document/元素.querySelectorAll('css选择器');

**获取class列表属性**

方法：元素.classList 类似于数组的对象

* + - * length : class的长度
      * add() : 添加class方法
      * remove() : 删除class方法
      * toggle() : 切换class方法 （如果有的话就是删除，反之为添加）

**JSON的新方法**

JSON是一种类似JS中的对象格式的字符串，JSON的类型是字符串，只不过长得像对象数据

JSON.parse

作用

把JSON类型转换成JS中对象类型

如果要转化的数据，不符合JSON格式，那么就会报错

语法：JSON.parse(JSONString)

JSON.stringify(object)

作用：把object转成JSON格式的字符串

eval() : 可以解析任何字符8串变成JS; parse() : 只能解析JSON形式的字符串变成JSON (安全性要高一些)

**新方法的应用：深度克隆新对象**

var a = {

name : { age : 100 }

};

var str = JSON.stringify(a);

var b = JSON.parse(str);

b.name.age = 200;

alert( a.name.age ); //100

如何其他浏览器做到兼容：<http://www.json.org/>去下载json2.js

**data自定义数据**

**dataset**

data-name : dataset.name

data-name-first : dataset.nameFirst

Data数据在jquery mobile中有着重要作用

<div id="div1" data-miaov="妙味" data-miaov-all="妙味课堂">div</div>

var oDiv = document.getElementById('div1');

alert( oDiv.dataset.miaov ); //妙味

alert( oDiv.dataset.miaovAll ); //妙味课堂

**加载JS**

JS的加载会影响后面的内容加载

很多浏览器都采用了并行加载JS，但还是会影响其他内容

Html5的defer和async (只能用于外链js)

defer : **延迟加载**，会按顺序执行，在onload执行前被触发

async : **异步加载**，加载完就触发，有顺序问题

<script src="a.js" defer =" defer "></script>

<script src="b.js" async ="async"></script>

<script src="c.js" async ="async"></script>

Labjs库

**4.历史管理**

触发历史管理的方法 : 1.通过跳转页面 2.改变hash 3.pushState

**onhashchange** ：改变hash值时触发该事件

history ：

服务器下运行

pushState : 三个参数 ：数据 标题(都没实现) 地址(可选)

popstate事件 : 读取数据 event.state

注意：网址是虚假的，需在服务器指定对应页面，不然刷新找不到页面