**HTML5新特性**

1. HTML5建立的规则
2. 新特性应该基于 HTML、CSS、DOM 以及 JavaScript。
3. 减少对外部插件的需求（比如 Flash）
4. 更优秀的错误处理
5. 更多取代脚本的标记
6. HTML5 应该独立于设备
7. 浏览器的支持

Safari支持的html标签

<https://developer.apple.com/library/safari/#documentation/AppleApplications/Reference/SafariHTMLRef/Introduction.html>（参考网站）

|  |
| --- |
| Safari支持的标签(红色标签代表HTML5新增的，黄色代表html5不支持) |
| a, abbr,acronym,address, applet,area,audio |
| b,base,basefont,bdo,big,blockquote,body,br,button, |
| canvas,caption,center,cite,code,col,colgroup, |
| dd,del,dfn,dir,div,dl,dt |
| em,embed, |
| fieldset,font,form,frame,frameset, |
| h1~h6,head,hr,html, |
| i,iframe,img(image),input,ins,isindex, |
| kbd,keygen, |
| label,layer,legend,li,link,listing, |
| map,marquee,menu,meta, |
| nobr,noembed,noframes,nolayer,noscript, |
| object,ol,optgroup,option, |
| p,param,plaintext,pre, |
| q, |
| S,samp,script,select,small,source,span,strike,strong,style,  sub,sup, |
| table,tbody,td,textarea,tfoot,th,thead,title,tr,tt, |
| U,ul, |
| var,video, |
| wbr |
| Xmp |

1. HTML5的新特性
2. 可能用到的新增的功能标签

|  |  |  |  |
| --- | --- | --- | --- |
| 分类 | 标签 | 描述 | 属性 |
| 文档结构相关 | article | 用于代表页面上独立的内容 |  |
| section | 用于对页面内容分块 |  |
| **nav** | 定义页面上的导航条 |  |
| **figure** | 专门用于表示一块独立的图片区域，可包含一张或多张图片 |  |
| **figcaption** | 放在figure内部，用于定义标题 |  |
| 语义相关 | mark | 显示页面中重点显示的内容 |  |
| embed | 定义签入的内容，图片，插件等，src指定插件路径 |  |
| 视频音频相关 | audio | Html5推荐使用ogg vobis格式 |  |
| video | 推荐使用VP8格式 |  |
| 画布 | canvas | 用于绘图，但本身没有绘图能力，需用js控制，具体API以及实例可参考；  <http://dev.w3.org/html5/2dcontext/> |  |
| Input标签 | input | 可以解决让表单动态的提交到不同的URL，比如有两个按钮的表单，就可以通过该属性指定url值提交，而不需要通过脚本动态控制 | formaction |
| 可以控制表单的提交方式 | formmethod |
| 规定该input所属的表单 | form |
| 启用自动完成，默认为on(支持type=’text,email’) | autocomplete |
| 该属性控制标签自动获得焦点（一个页面最多只有一个表单控件可设置该属性） | autofocus |
| 在文本框中显示输入框的提示信息 | placeholder |
| 这些类型的input,safari支持的都不是很好 | type=color,date,time,datetime,  datetime-local,month,week,email,tel,url,number,range,search |
| 文件上传功能，可支持同时多文件上传，但是Safari支持的不是很完美。  Js可以通过files属性访问元素生成的文件上传域内的所有文件，返回一个filelist对象，相当于数组；每个file对象包含属性有：name,type,size | type=”file”拥有两个属性，accept(控制允许上传的文件类型)，multiple(设置是否允许选择多个文件) |
| 使用FileReader读取文件内容（safari支持的不好，火狐可以支持） |  |
| 客户端校验，但是safari支持的不好，火狐支持的还可以 | require（指定该空间必须）,pattern（指定控件的值是否符合指定的正则表达式）,min,max,step（只对数值类型，日期类型的有效） |
| 表单对象调用该方法是为了控制表单控件是否通过输入校验；如果调用返回true,则表明表单内的所有控件都有效，否则，有一个表单控件不能通过输入校验，则会返回false | checkValidity()方法(type类型判断是不是对应的类型数据) |
| 表单控件调用setCustomValidity（）来设置自定义的错误提示信息，但是safari貌似不支持 | 自定义错误提示的方法：setCustomValidity() |
| 为form表单提供该属性，表示希望关闭HTML5对表单提供的输入校验（但是因为safari对客户端校验支持的不好，所以就看不出效果，但是火狐可以），该属性的值可缺省，或novalidate | novalidate(针对form)  formnovalidate（针对type=’submit’的input或者button标签） |

1. 新增的功能标签(注：以下标签按照字母表顺序排列)

|  |  |  |  |
| --- | --- | --- | --- |
| 标签 | 描述 | 属性 | 注意 |
| <article> | 标记显示一个独立的文章内容，article可以嵌套，则内层的article对外层的article标签有隶属的关系。 | 支持HTML5的全局属性和事件属性，具体参考：<http://www.w3school.com.cn/html5/tag_article.asp>  <http://www.htmlv.cn/bbs/topic51.html> |  |
| <aside> | 标记定义页面内容部分的侧边栏，即定义其所处内容之外的内容 | 同上 |  |
| <audio> | 标记定义音频内容 | autoplay:表示音频在就绪后马上播放；  controls:表示向用户显示控件，比如播放按钮；  loop:表示每当音频结束时重新开始播放；  preload:表示音频在页面加载时进行加载，并预备播放，如果使用“autoplay”，则忽略该属性；  src：表示要播放的音频的URL;  支持全局实行和事件属性： |  |
| <bdi> | 定义文本的文本方向，使其脱离其周围文本的方向设置 |  |  |
| <canvas> | 标记定义图片，它只是图形容器，必须使用脚本来绘制图形；其本身没有绘图能力 | Height:设置canvas的高度；  Width:设置canvas的宽度；  支持全局属性和事件属性；  实例看：<http://www.mhtml5.com/resources> |  |
| <command> | 标记定义一个命令按钮，表示用户能够调用的命令，只有当command元素位于menu元素内时，该元素才是可见的，否则不会显示这个元素，但是可以用它规定键盘快捷键； | checked:checked;定义是否被选中，仅用于radio或checkbox类型；  disabled: disabled;定义是否可用；  icon:url;定义作为command来显示的图像的URL；  label：定义可见的label;  radiogroup：定义所属组名，仅在类型为radio时使用；  type:checkbox/command/radio:默认为command; |  |
| <datalist> | 标记定义一个下拉列表,与input元素配合使用该元素，来定义input可能的值，datalist及其选项不会被显示出来，它仅仅是合法的输入值列表，实现自动完成的效果 | <http://www.w3school.com.cn/html5/tag_datalist.asp>  当前只是 Firefox 和 Opera 浏览器支持  <http://www.mhtml5.com/2012/06/5116.html> |  |
| <details> | 标记定义一个元素的详细内容 | 目前只有chrome支持该标签，与<summary>标签配合使用可以为details定义标题，标题是可见的，用户点击标题时，会显示details;  open:open;定义是否可见 |  |
| <dialog> | 标记定义一个对话框(会话框) | 对话中的每个句子必须属于<dt>标签所定义的部分； |  |
| <embed> | 标记定义外部的可交互的内容或插件 | height;src;type;width; |  |
| <figcaption> | 定义 figure 元素的标题,应被置于figure元素的第一个或者最后一个子元素的位置； |  |  |
| <figure> | 标记定义一组媒体内容以及它们的标题<footer> 标记定义一个页面或一个区域的底部 |  |  |
| <footer> | 定义 section 或 page 的页脚 |  |  |
| <header> | 标记定义一个页面或一个区域的头部 |  |  |
| <hgroup> | 标记定义文件中一个区块的相关信息；用于对网页或区段的标题元素进行组合 |  |  |
| <keygen> | 标记定义表单里一个生成的键值，规定用于表单的密钥对生成器字段，当提交表单时，私钥存储在本地，公钥发送到服务器； |  |  |
| <mark> | 标记定义有标记的文本，定义带有记号的文本； |  |  |
| <meter> | 标记定义。定义度量衡，仅用于已知最大和最小值的度量 | <http://www.w3school.com.cn/html5/tag_meter.asp> |  |
| <nav> | 标记定义导航链接，放入一些当前页面的主要导航链接 |  |  |
| <output> | 标记定义一些输出类型，比如脚本的输出 | <http://www.w3school.com.cn/html5/tag_output.asp> |  |
| <progress> | 标记定义任务的过程，定义运行中的进度（进程）；可以使用该标签显示javascript中耗费时间的函数的进度； |  |  |
| <rp> | 标记是用在Ruby 注释 ，告诉那些不支持 Ruby元素的浏览器如何去显示 |  |  |
| <rt> | 标记定义对ruby 注释的解释 |  |  |
| <ruby> | 标记定义 ruby 注释 |  |  |
| <section> | 标记定义一个区域，比如章节、页眉、页脚和文档中的其他部分 | <http://www.w3school.com.cn/html5/tag_section.asp> |  |
| <source> | 标记定义媒体资源 | <http://www.w3school.com.cn/html5/tag_source.asp> |  |
| <summary> | 定义details元素的标题 |  |  |
| <time> | 标记定义一个日期/时间 | 目前所有浏览器都不支持time标签 |  |
| <track> | 定义用在媒体播放器中的文本轨道 | 目前所有主流浏览器不支持该标签 |  |
| <video> | 标记定义一个视频 | <http://www.w3school.com.cn/html5/tag_video.asp> |  |

1. 表单新的属性
2. 新的form属性

* **autocomplete**

说明：该属性规定form或input域应该拥有自动完成的功能

注释：该标签适用于<form>标签，以及以下类型的<input>标签：text,search,url,telephone,email,password,datepickers,range以及color.

* **novalidate**（chrome/safari不支持）

说明：规定在提交表单时是否需要验证form或input域

注释：适用于form以及以下类型的input标签：text,search,url,telephone,email,password,date pickers,range以及color

1. 新的input属性

* **autocomplete**

见1）中的autocomplete说明

* **autofocus**

说明：规定在页面加载时，域自动获得焦点

注释：适用于所有input标签

* **form(chrome/safari不支持)**

说明：规定输入域所属的一个或多个表单；

注释:适用于所有input标签的类型；

* **form overrides（chrome/safari不支持）**

说明：表单重写属性，允许重写form元素的某些属性设定

重写属性有：

* formaction-重写表单的action属性
* formenctype-重写表单的enctype属性
* formmethod-重写表单的method属性
* formnovalidata-重写表单的novalidate属性
* formtarget-重写表单的target属性

注释：表单重写属性适用于以下类型的input标签：submit 和image

* **height width**

说明：规定用于image类型的input标签的图像高度和宽度

注释：只适用于image类型的input标签

* **list（chrome/safari不支持）**

说明：规定输入域的datalist.

注释：适用于以下类型的input标签：text，search，url，telephone,email,date pickers,number,range,color.

* **min max step**

max:规定输入域所允许的最大值

min:规定输入域所允许的最小值

step:规定输入域规定合法的数字间隔

注释：适用于以下类型input标签，date pickers，number,range

* **multiple**

说明：属性规定输入域中可选择多个值

注释：适用于以下类型的input标签：email和file

* **pattern（safari不支持）**

说明：规定用于验证input域的模式（正则表达式）

注释：适用于以下类型的input标签：text,search,url,telephone,email,password

* **placeholder**

说明：该属性提供一种提示，描述输入域所期待的值。

注释：适用于以下类型的input标签：text,search,url,telephone,email,password

* **required（safari不支持）**

说明：规定必须在提交之前填写输入域（不能为空）

注释：适用于以下input标签：text,search,url,telephone,email,password,date pickers,number,checkbox,radio ,file

1. 拥有新属性的标签

|  |  |
| --- | --- |
| 标签 | 属性 |
| ol | Starta: 指定列表项的起始数字  Type: 指定使用哪种类型的编号，该属性在html5中已经不推荐使用了  Reversed: 指定是否将排序反转，目前没有浏览器支持 |
| Input  注：(type=”file”,增加了两新属性：accept（该属性控制允许上传的文件类型）和multiple（该属性设置是否允许选择对个文件）)；  Html5新增了客户端校验：required,pattern,min,max,step,其中，min,max，step只对数值类型，日期类型的input元素有效，firefox暂时不支持type=”number”的input的元素，因此对min,max,step,校验的规则也不是很理想 | color:让标签元素生成一个颜色选择器  date：生成一个日期选择器  time：生成一个时间选择器  datetime：生成一个UTC日期、时间选择器  datetime-local：生成一个本地日期、时间选择器  week：生成一个供用户选择第几周的文本框  month：生成一个月份选择器  email：生成一个email输入框，可自动检测  tel：生成输入电话号码的文本框，不会自测  url：生成一个url输入框，自测  number：生成一个只能输入数字的文本框  range(min,max,step)：生成一个拖动条  search：生成一个专门用于输入搜索关键字的文本框 |

1. 无需插件支持的音频和视频

HTML5新增了两个元素，即<audio>和<video>，用法比较简单，可参考：

<http://www.w3school.com.cn/html5/html_5_audio.asp>

<http://www.w3school.com.cn/html5/html_5_video.asp>

1. 画布功能

HTML5新增了一个<canvas>元素，该元素专门用于绘制图形。（这个元素本身不会绘制图形，只是相当于一个画布，要再canvas上绘制图形，必须使用js脚本控制）

具体可参考：

如下链接来自苹果官网介绍：

[https://developer.apple.com/library/safari/#documentation/AppleApplications/Conceptual/SafariJSProgTopics/Tasks/Canvas.html#//apple\_ref/doc/uid/30001236](https://developer.apple.com/library/safari/#documentation/AppleApplications/Conceptual/SafariJSProgTopics/Tasks/Canvas.html)

如下链接来自HTML5研究小组：

<http://www.mhtml5.com/resources>

1. 可编辑内容和拖放

* **内容可编辑**：是通过全局属性来控制的，即contenteditable属性，

|  |  |
| --- | --- |
| 属性值 | 描述 |
| true | 可以编辑元素内容 |
| false | 无法编辑元素内容 |
| classname | 继承父元素的 contenteditable 属性 |

注：全局属性可以用于任何html5元素，具体全局属性可参考如下网站

<http://www.w3school.com.cn/html5/html5_ref_globalattributes.asp>

* **可拖放**：html5新增了关于拖放的API，通过拖放API可以让HTML页面的任意元素都变成可拖动的。
* 首先了解“拖”：

1. <img>元素默认是可拖动的
2. <a>元素只要设置了href属性，也是默认可拖动的。
3. 普通元素设置可拖动性，只需要把该元素的draggable属性设置为true就可以拖动了，但是这只设置了draggable=”true”只表示元素可拖动，但并不能携带数据，为此，需要为该被拖动元素的ondragstart事件指定监听器，在监听器中让拖动操作携带数据

* 其次了解“放”：

为了让document可以接受“放”，应该为其ondragover事件指定监听器，在监听器中取消其对拖动事件的默认行为。

* 最后了解 DataTransfer对像

通过DataTransfer对象，可以让拖放开始时将拖放源的数据存入DataTransfer对象中，然后再拖放结束时从DataTransfer对象中读取数据，这样就可以完成更复杂的拖放操作。

1. 数据存储

Html5新增了Web Storage功能，通过Web Storage可以让应用程序在客户端运行时在客户端保存程序数据。

Session Storage

Web Storage

Local Storage

1. **Storage接口分类：**

Session Storage:基于Session的Web Storage。Session Storage保存的数据生存期限与用户Session期限相同，用户Session结束时Session Storage保存的数据也会丢失了。

Local Storage:保存在用户磁盘的Web Storage。通过Local Storage保存的数据生存期限很长，除非用户或程序显式地清除这些数据，否则这些数据会一直存在。

实践发现：如果浏览器设置了清除痕迹的选项，那么即使是不关闭浏览器localStorage也不会存储数据，如果没有设置，关闭浏览器，数据还是存在的。（ie貌似不支持），但它保存的数据不会进行加密，所以不适宜保存比较敏感的数据

1. **Storage接口提供的属性和方法**

|  |  |
| --- | --- |
| 属性或方法 | 描述 |
| length | 该属性返回该Storage里保存了多少组key-value对 |
| key(index) | 该方法返回该Storage中第index个key |
| getitem(key) | 该方法返回该Storage中指定key对应的value |
| set(key,value) | 该方法用于向该Storage存入指定的key-value对 |
| removeitem(key) | 该方法用于从该Storage删除指定key对应的key-value对 |
| clear() | 该方法用于删除该Storage中所有的key-value对 |

实例参考：<http://www.w3school.com.cn/html5/html_5_webstorage.asp>

**Html5离线应用**

1. 客户端存储通常需要与离线应用相结合使用，最通用的应用方式可细分为：

* 当用户在线。能连服务器时，直接把数据提交给服务器，直接与服务器交互
* 当用户离线、不能连服务器时，用户浏览、操作的是离线应用；
* 当用户再次接入网络时，Web应用程序控制把本地存储中得保存的数据提交给远程服务器。

1. 离线应用与浏览器缓存的区别

* 服务范围不同：离线应用控制对整个Web应用进行缓存。离线应用提供的是一种不在线的网站服务功能；而浏览器缓存只是单纯的缓存网页。
* 可靠性不同：离线应用可以精确地控制浏览器需要缓存哪些资源，它是非常可靠的；但浏览器缓存则完全依赖浏览器行为，具有一定的不可靠性。
* 可控制性不同：离线应用可准确地控制缓存哪些资源，并可控制刷新缓存，但浏览器缓存则完全依赖于浏览器行为，程序无法控制缓存行为。

1. 为了给页面增加离线应用功能，需要在<html>标签中添加manifest属性，每个指定了 manifest 的页面在用户对其访问时都会被缓存。如果未指定 manifest 属性，则页面不会被缓存（除非在 manifest 文件中直接指定了该页面）。
2. 判断在线状态：HTML5提供了两种方法（navigator.onLine属性，online/offline事件）

* navigator.onLine属性：navigator.onLine属性可返回当前是否在线，如果返回true,则表示在线，如果返回false，则表示离线，当网络发生变化时，navigator.onLine属性的值也随之改变。
* online/offline事件：如果开发者需要在网络状态发生变化时立刻得到通知，则可以通过该事件监测，当在线/离线状态切换时，body元素上得online/offline事件将会被触发，并沿着document.body、document和window冒泡。因此开发者可通过他们的online/offline事件来检测网络状态的变化

1. 开启离线应用后，js可以通过applicationCache来控制离线缓存
2. 废弃的标签

<big>，<frameset>，<frame>，<noframes>

可参考网站

1. HTML5规范：<http://dev.w3.org/html5/spec/single-page.html>
2. <http://www.w3.org/TR/html5/>
3. HTML5研究小组：<http://www.mhtml5.com/resources>
4. W3school中文版：<http://www.w3school.com.cn/html5/html_5_intro.asp>
5. Safari支持的html标签：

[https://developer.apple.com/library/safari/#documentation/AppleApplications/Reference/SafariHTMLRef/Articles/HTMLTags.html#//apple\_ref/doc/uid/30001262-SW2](https://developer.apple.com/library/safari/#documentation/AppleApplications/Reference/SafariHTMLRef/Articles/HTMLTags.html)

1. 当前浏览器HTML5支持情况在线测试：<http://www.html5test.com/>
2. 当前浏览器对html5 css3的支持情况：<http://www.findmebyip.com/>
3. 各个浏览器对html5 css3支持情况大全：<http://www.caniuse.com/>