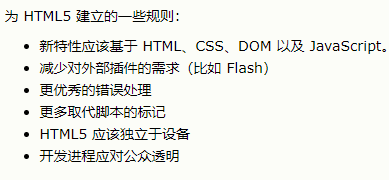
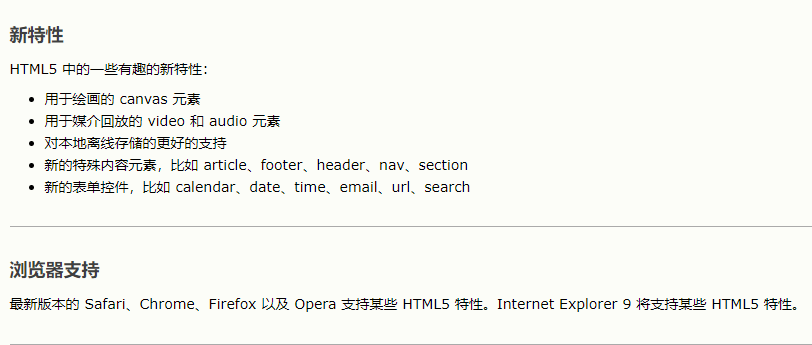
Css参考网站<http://phpstudy.php.cn/css3/>

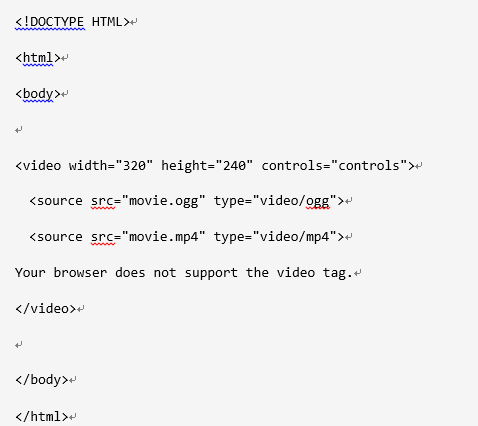
Jq参考网站<http://hemin.cn/jq/>

## 一、TML5简介

HTML 指超文本标签语言。HTML5 是下一代的 HTML。

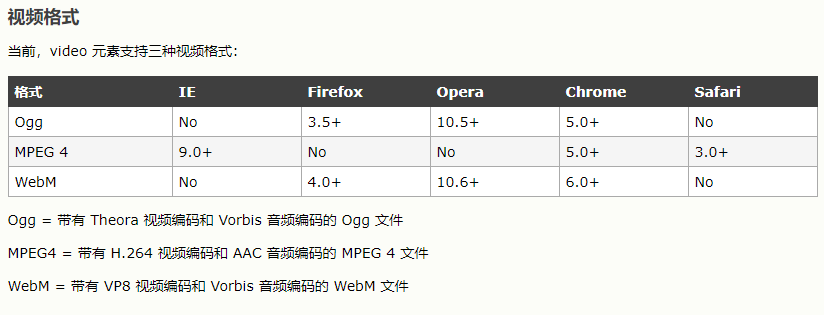




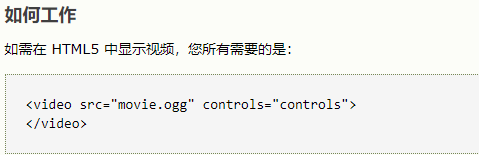


## 二、HTML5视频：

直到现在，仍然不存在一项旨在网页上显示视频的标准。今天，大多数视频都是通过插件（比如flash）来实现的，然而不是所有的浏览器都拥有同样的插件。HTML5 规定了一种通过 video 元素来包含视频的标准方法。



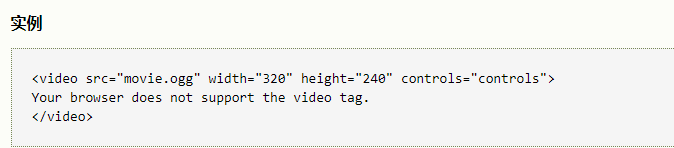
#### 1、如何工作：



controls 属性供添加播放、暂停和音量控件。

包含宽度和高度属性也是不错的主意。

<video> 与 </video> 之间插入的内容是供不支持 video 元素的浏览器显示的：



#### 2、html5的video元素支持多个source元素：



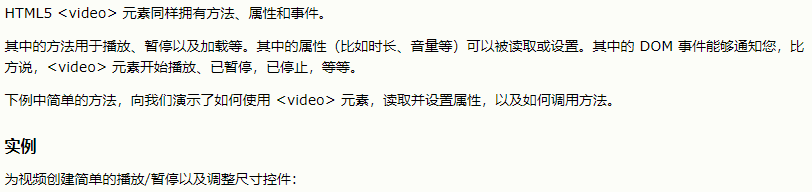
#### 3、Internet Explorer 8 不支持 video 元素。在 IE 9 中，将提供对使用 MPEG4 的 video 元素的支持。

#### 4、video标签属性：



## 三、HTML5 video=DOM：

#### 1、HTML5 <video> - 使用 DOM 进行控制：



<!DOCTYPE html>

<html>

<body>

<div style="text-align:center;">

<button onclick="playPause()">播放/暂停</button>

<button onclick="makeBig()">大</button>

<button onclick="makeNormal()">中</button>

<button onclick="makeSmall()">小</button>

<br />

<video id="video1" width="420" style="margin-top:15px;">

<source src="/example/html5/mov\_bbb.mp4" type="video/mp4" />

<source src="/example/html5/mov\_bbb.ogg" type="video/ogg" />

Your browser does not support HTML5 video.

</video>

</div>

<script type="text/javascript">

var myVideo=document.getElementById("video1");

function playPause()

{

if (myVideo.paused)

myVideo.play();

else

myVideo.pause();

}

function makeBig()

{

myVideo.width=560;

}

function makeSmall()

{

myVideo.width=320;

}

function makeNormal()

{

myVideo.width=420;

}

</script>

</body>

</html>



上面的例子调用了两个方法：play() 和 pause()。它同时使用了两个属性：paused 和 width。

#### 2、HTML5 <video> - 方法、属性以及事件：



## 四、HTML5音频：

直到现在，仍然不存在一项旨在网页上播放音频的标准。

今天，大多数音频是通过插件（比如 Flash）来播放的。然而，并非所有浏览器都拥有同样的插件。

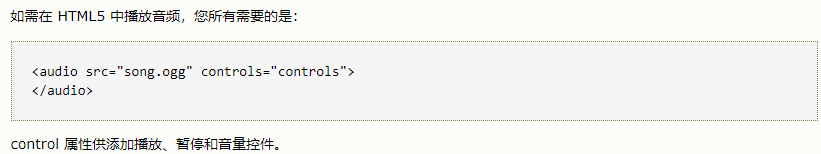
HTML5 规定了一种通过 audio 元素来包含音频的标准方法。

audio 元素能够播放声音文件或者音频流。

#### 目前，audio元素支持三种音频格式：



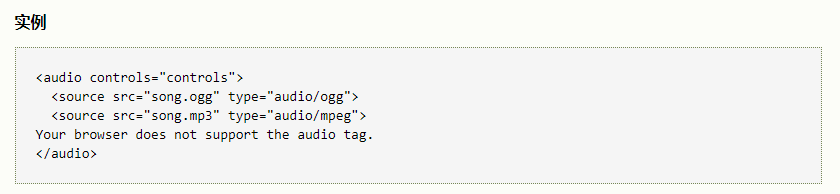
#### 2、如何工作：



<audio> 与 </audio> 之间插入的内容是供不支持 audio 元素的浏览器显示的：



#### 3、audio 元素允许多个 source 元素。source 元素可以链接不同的音频文件。浏览器将使用第一个可识别的格式：

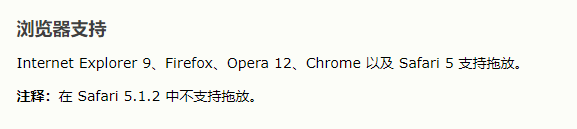
4、Internet Explorer 8 不支持 audio 元素。在 IE 9 中，将提供对 audio 元素的支持。



## 五、HTML5拖放：

拖放（Drag 和 drop）是 HTML5 标准的组成部分。拖放是一种常见的特性，即抓取对象以后拖到另一个位置。在 HTML5 中，拖放是标准的一部分，任何元素都能够拖放。

1. 浏览器支持：



#### 1、HTML5拖放实例：

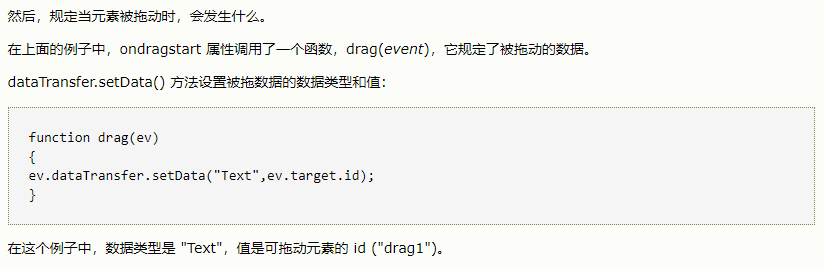


分别研究拖放事件的不同组成部分：

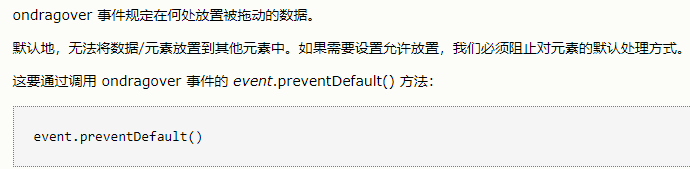
##### （1）设置元素为可拖放：首先，为了使元素可拖动，把 draggable 属性设置为 true ：



##### 拖动什么 - ondragstart 和 setData()：



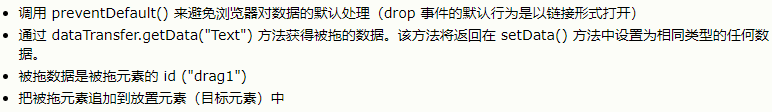
#### 放到何处-ondragover：



##### （4）进行放置：



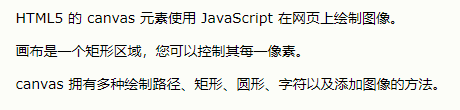
##### （5）代码解释：



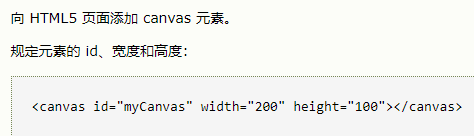
## 六、HTML5 canvas

canvas 元素用于在网页上绘制图形。

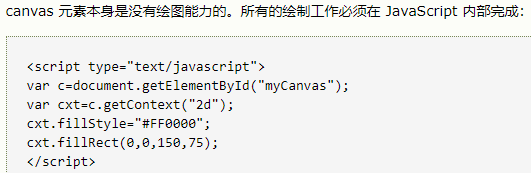
#### 什么是canvas：



#### 创建canvas元素：



#### 通过JavaScript来绘制：



##### （1）JavaScript 使用 id 来寻找 canvas 元素：



##### （2）然后，创建context对象：

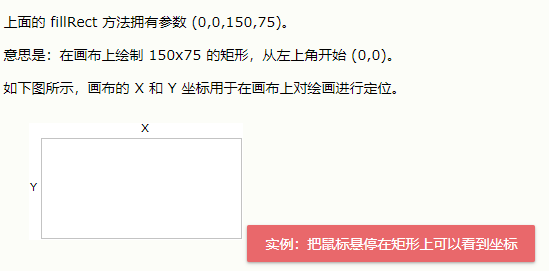


##### （3）getContext("2d") 对象是内建的 HTML5 对象，拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法。下面的两行代码绘制一个红色的矩形：



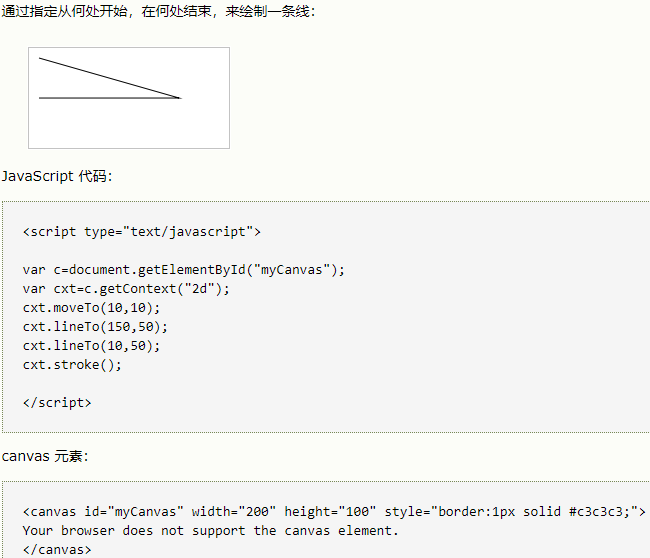
fillStyle 方法将其染成红色，fillRect 方法规定了形状、位置和尺寸。

##### 理解坐标：



#### 更多实例：

##### （1）实例---线条：



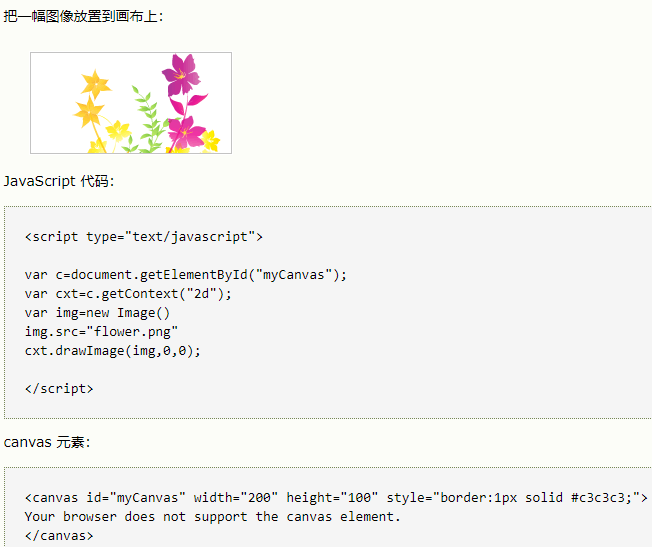
##### 实例---圆



##### （3）实例---渐变：



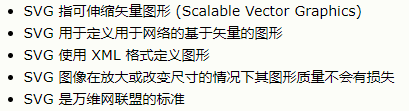
##### （4）实例---图像：



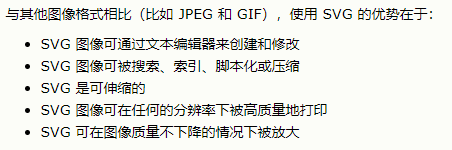
## 七、HTM5 内联SVG

HTML5支持内联SVG。

#### 什么是SVG：



#### SVG的优势：



#### 3、浏览器支持：

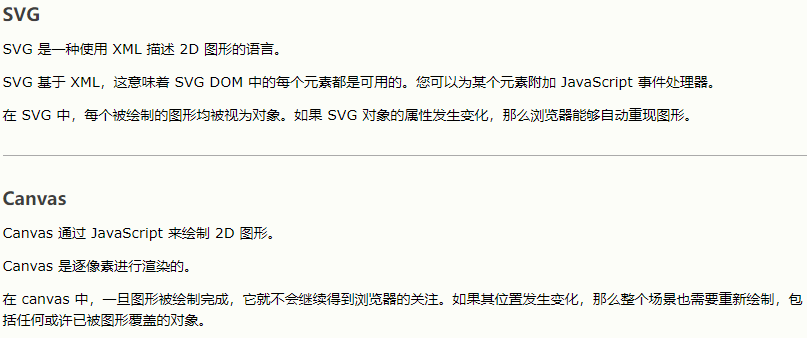


#### 4、把SVG直接嵌入HTML页面：

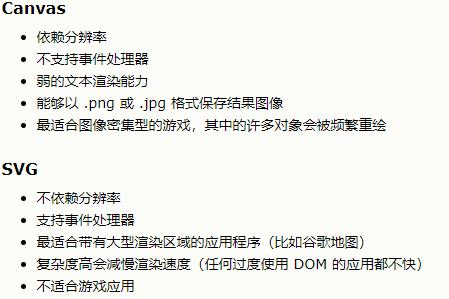


## 八、HTML5 canvas和SVG

#### 1、canvas和SVG性质区别：



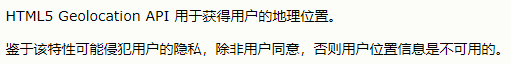
#### 2、canvas和SVG的比较：



## 九、HTML5 地理定位

HTML5 Geolocation用于定位用户的位置。

#### 定位用户的位置：



#### 浏览器支持：



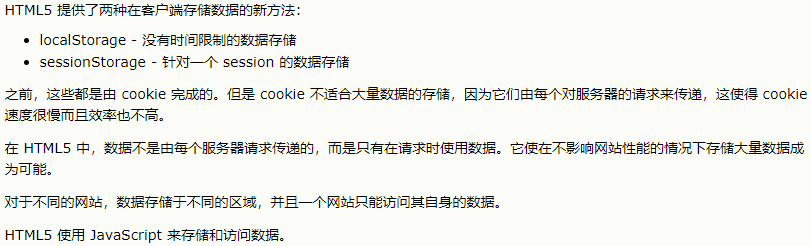
#### 3、HTML5使用地理位置：



这里只做简单的介绍，详情请在33school查阅

## 十、HTML5 web存储

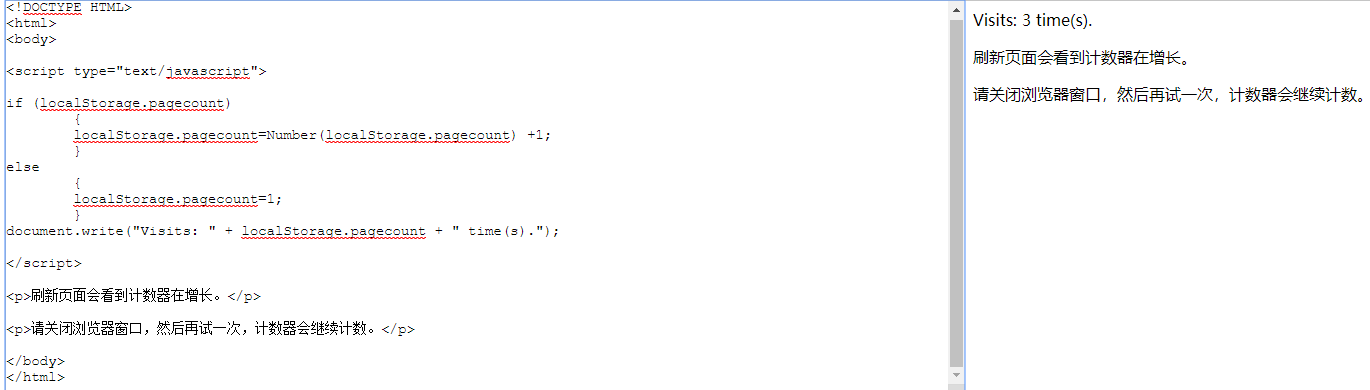
#### 1、在客户端存储数据：



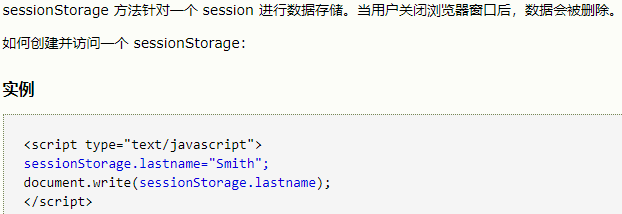
#### 2、localStorage：（存储在本地）



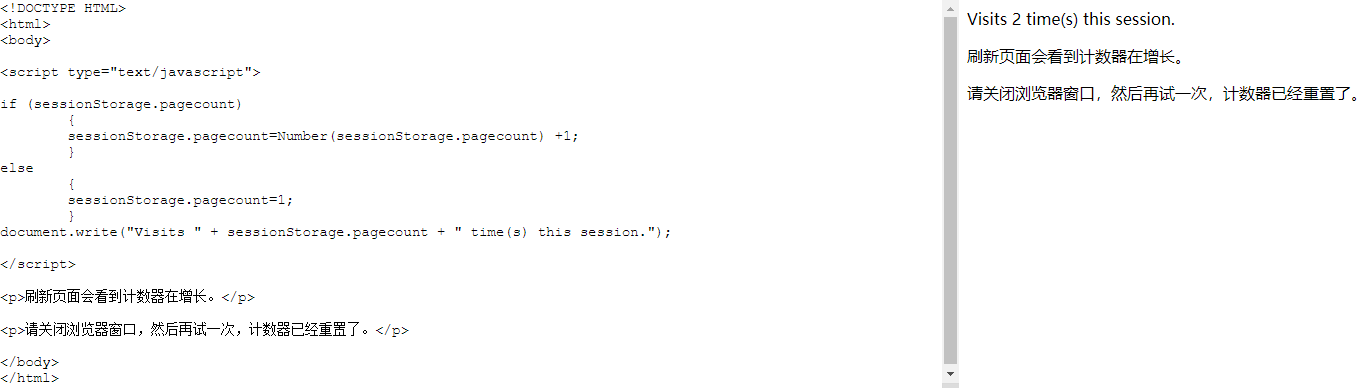
下面例子对用户访问页面的次数进行计数：



#### 3、sessionStorage方法：



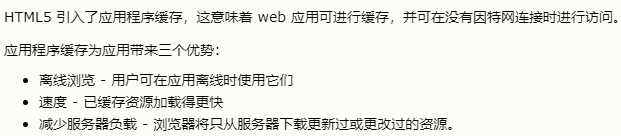
下面的例子针对用户在session中访问页面的次数进行计数：



## 十一、HTML5应用缓存

使用 HTML5，通过创建 cache manifest 文件，可以轻松地创建 web 应用的离线版本。

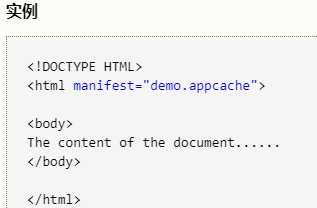
#### 什么是应用程序缓存（Application Cache）：



#### 浏览器支持：



#### HTML5 Cache Manifest 实例：



#### Cache Manifest基础：

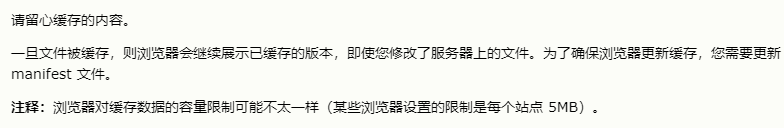


1. Manifest 文件：  
   

#### 更新缓存：



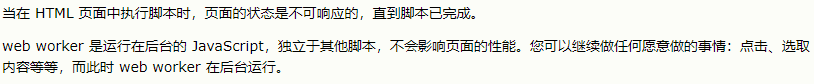
#### 6、关于应用程序缓存的注释：



## 十二、HTML5 Web Workers

web worker 是运行在后台的 JavaScript，不会影响页面的性能。

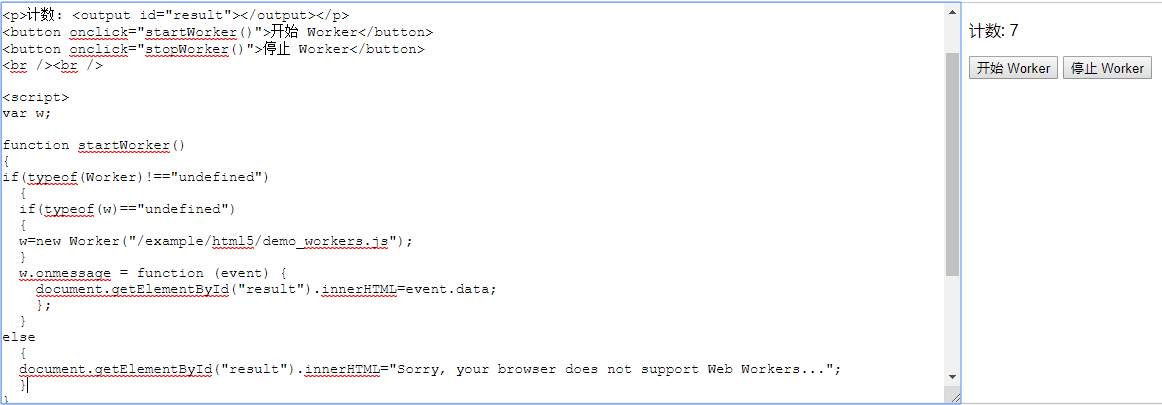
#### 什么是Web Workers：

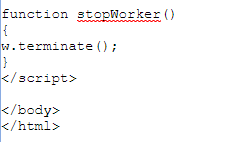
2、浏览器支持：



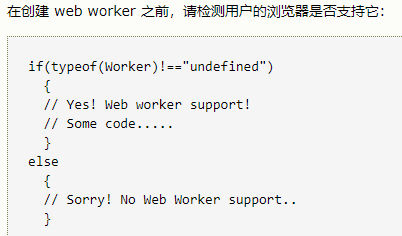
#### 3、HTML5 Web Workers实例：

下面的实例创建了一个简单的Web Worker，在后台计数：





##### （1）检查Web Worker支持：



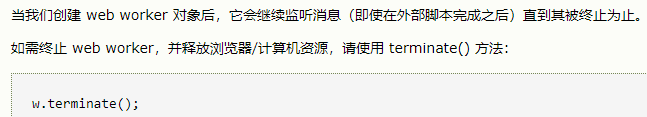
#### （2）创建Web Worker文件：



##### 创建Web Worker对象：



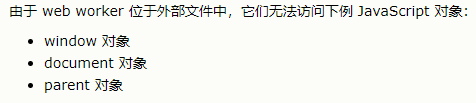
##### 终止Web Worker：



##### 完整的Web Worker实例代码：



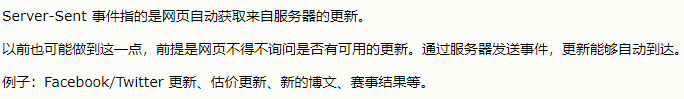
##### Web Workers 和Dom：



## 十三、HTML5服务器发送事件：

HTML5 服务器发送事件（server-sent event）允许网页获得来自服务器的更新。

#### Server-Sent 事件 - 单向消息传递



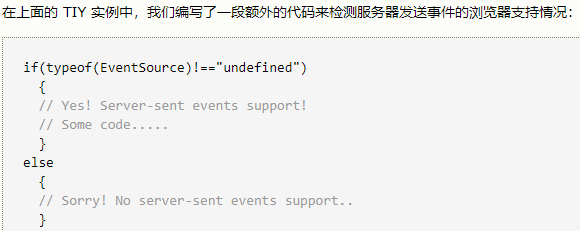
#### 浏览器支持



#### 接收Server-Sent事件通知



#### 检测Server-Sent事件支持



#### 服务器端代码实例



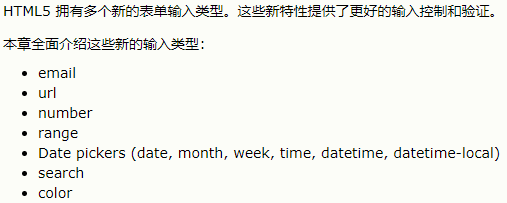
#### Event Source通知



## 十四、HTML5表单

### HTML5 Input类型

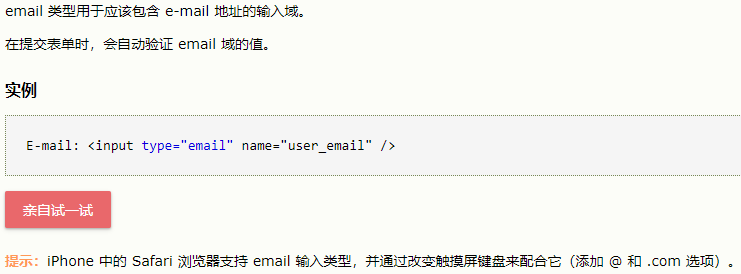
#### HTML5新的Input类型

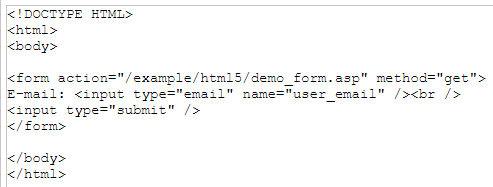


#### 浏览器支持

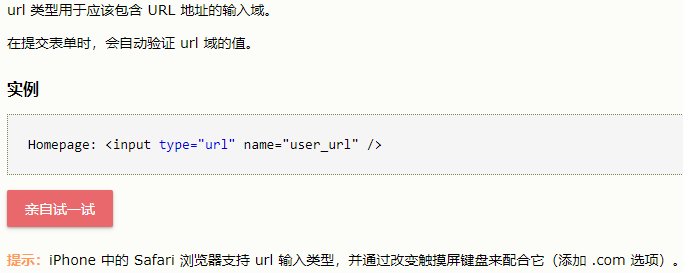


#### Input类型-email



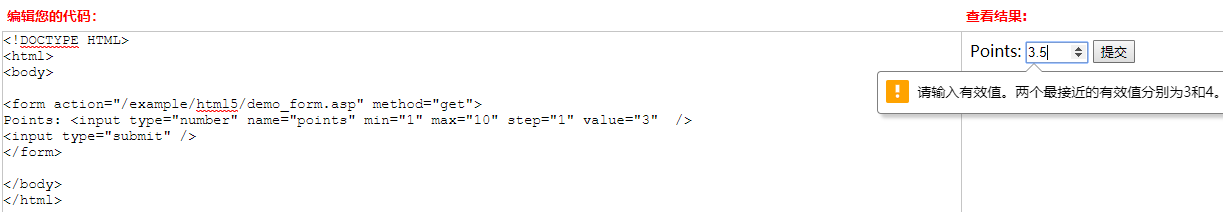


#### Input类型-url



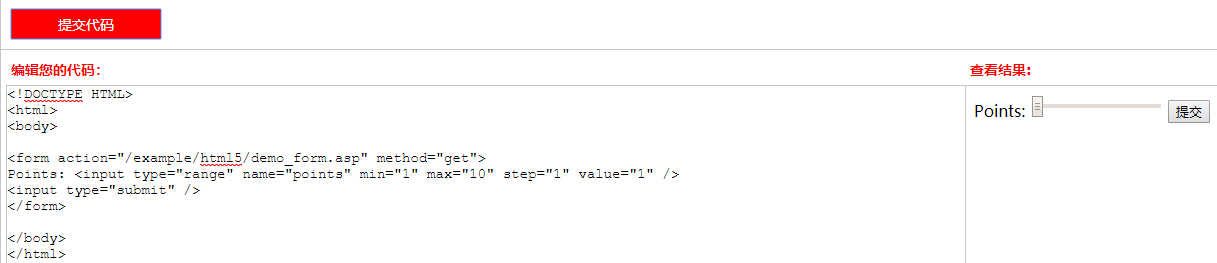
#### Input类型-number





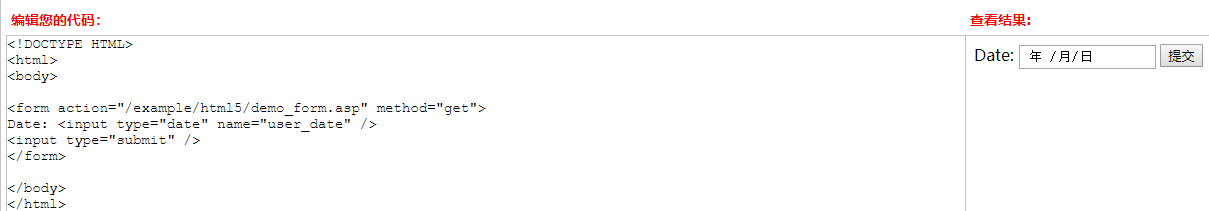
#### Input类型-range

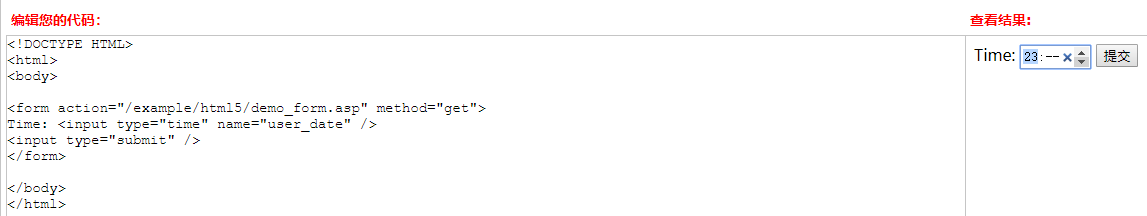




1. Input类型-Date Pickers（日期选择器）





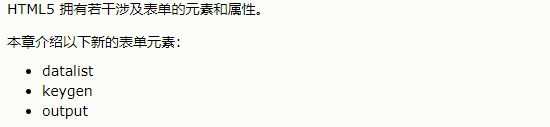


#### Input类型-search



### （二）HTML5 表单元素

#### 1、HTML5的新的表单元素

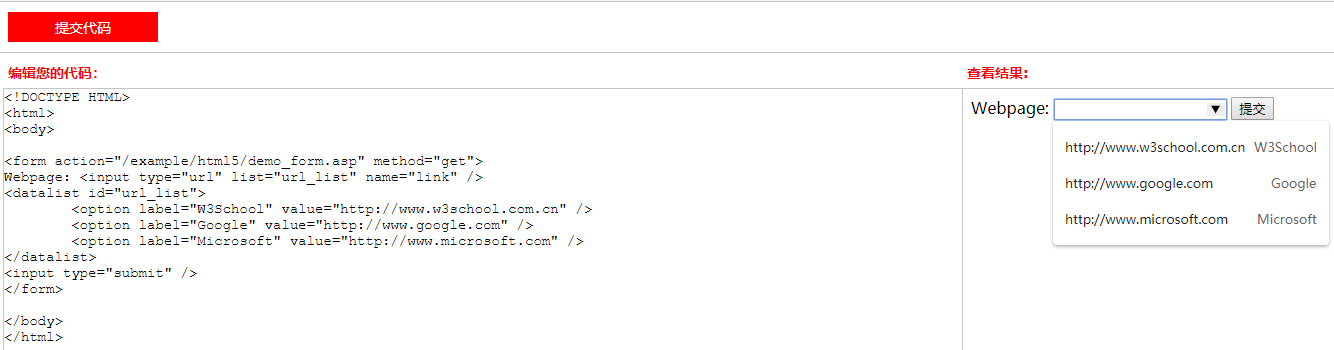


#### 浏览器支持

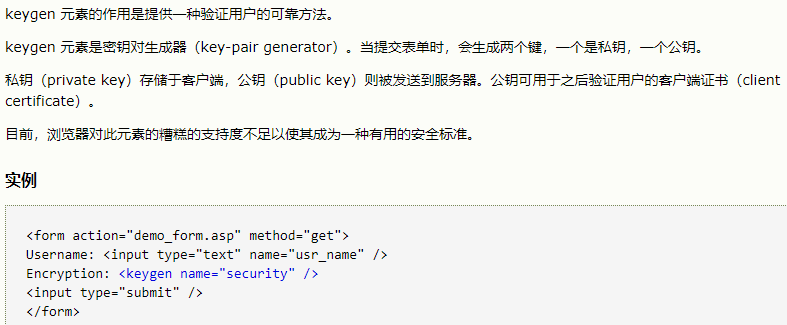


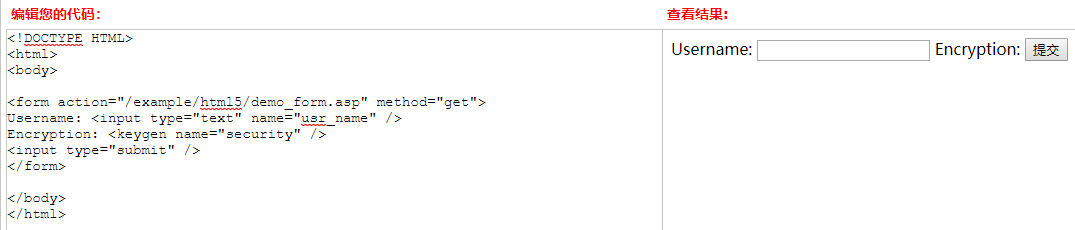
#### datalist元素



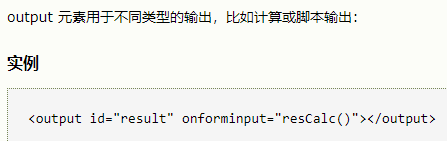


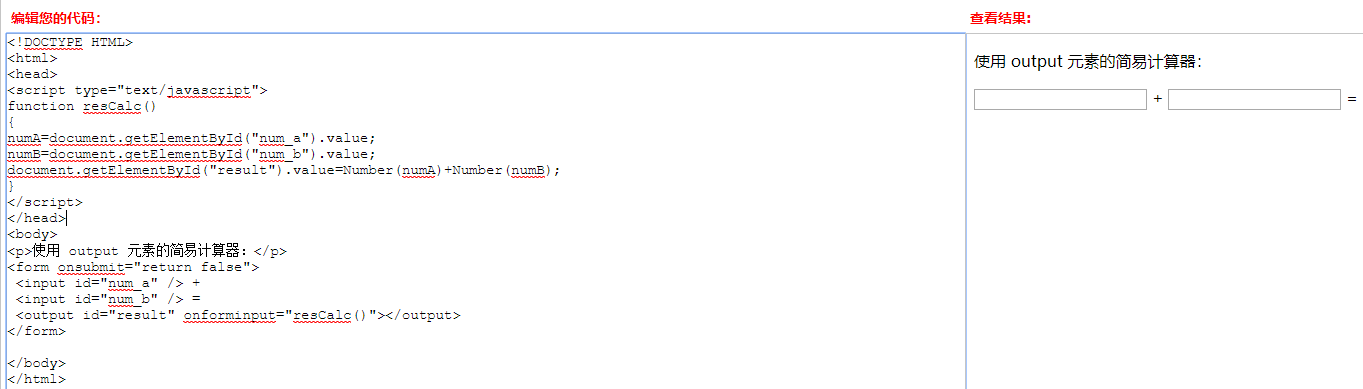
#### keygen元素





#### output元素





### （三）HTML5表单属性

#### HTML5的新的表单属性

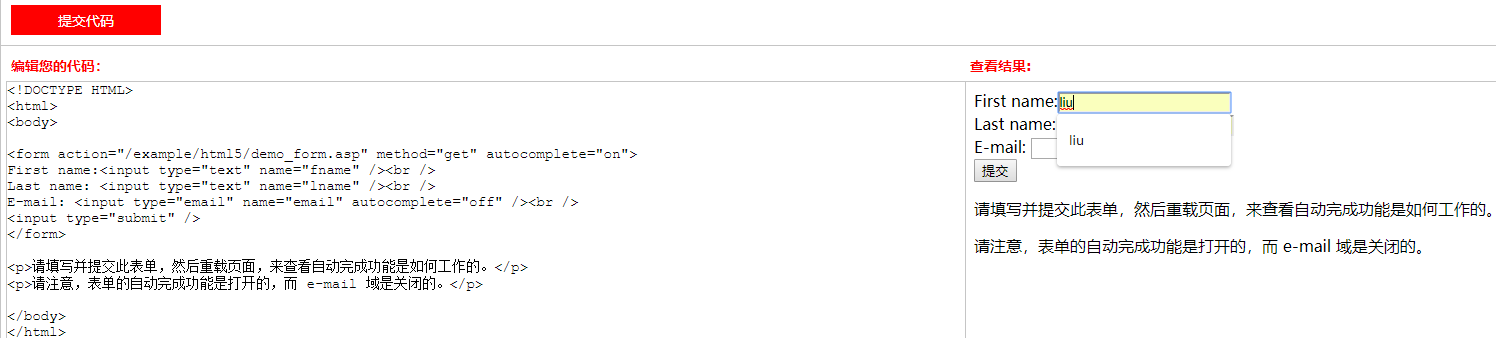


#### 浏览器支持

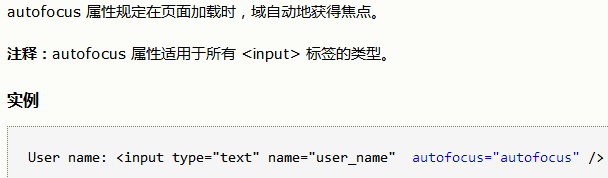


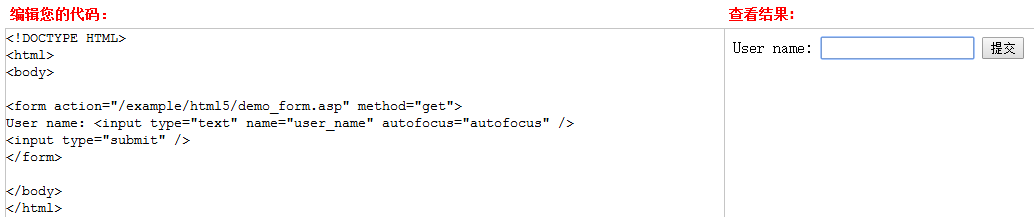
#### 3、autocomplete





#### 4、autofocus属性





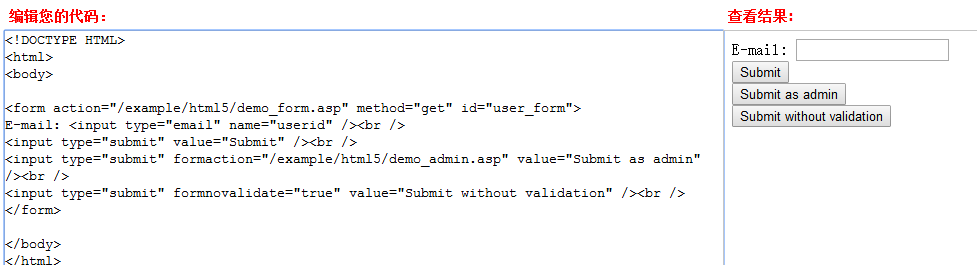
#### 5、Form属性





#### 6、表单重写属性





#### 7、hight和width属性

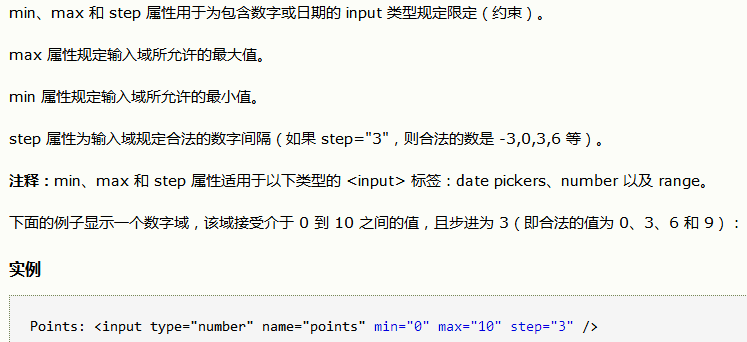


#### 8、list属性

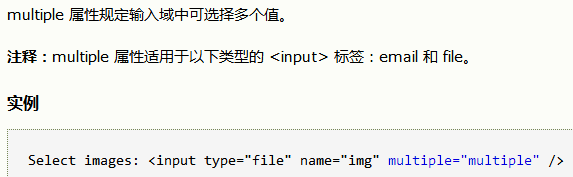


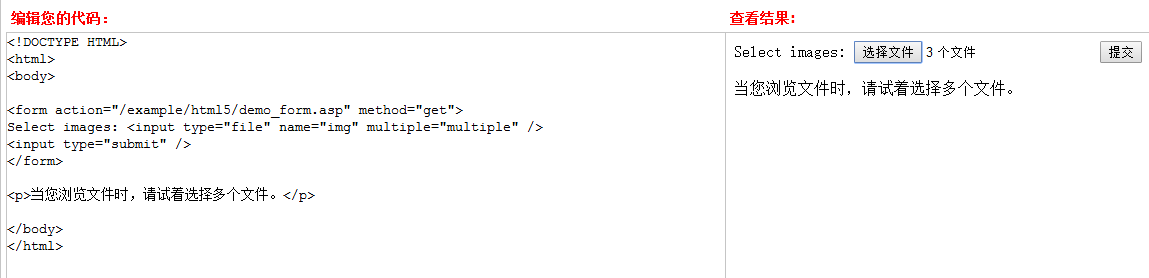


#### 9、min、max和step属性

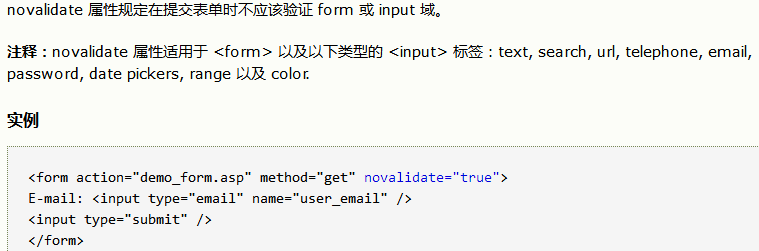


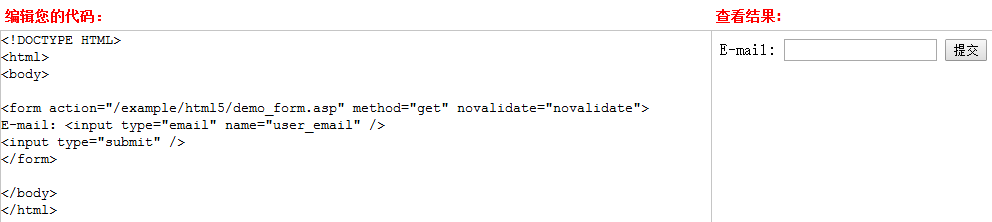
#### 10、multiple属性



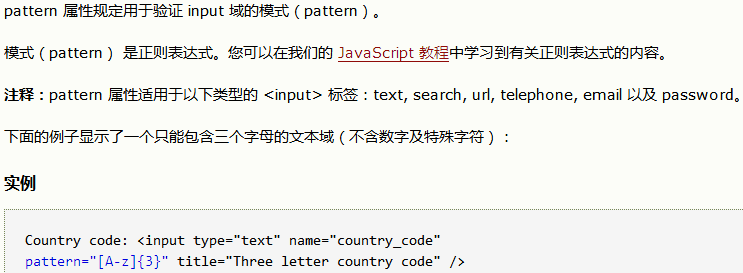


#### 11、novalidate属性

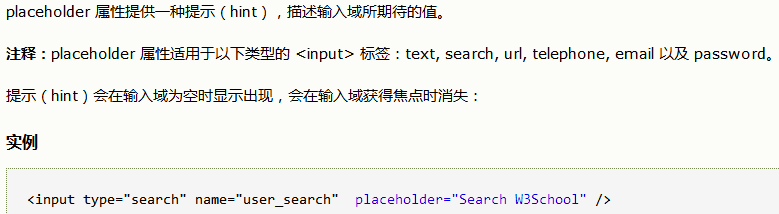


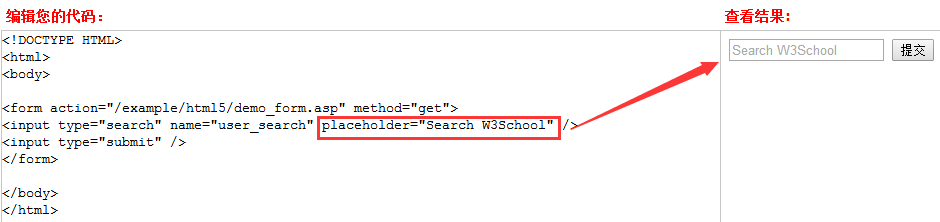


#### 12、pattern属性



#### 13、placeholder属性





#### 14、required属性

