1.Video与audio元素：

<audio src="">你的浏览器不支持(为不支持autio元素的浏览器上显示替代）</audio>

<video autoplay preload="auto" poster="" loop>

<source src="" type='video/ogg; codecs="theora,vorbis"'> type格式，codecs编码，source选择支持的格式

<source src="" type='video/quicktime' >

</video>

公有属性：

①autoplay自动播,

②preload预加载：none，metadata(加载元数据)，auto(加载全部),

③poster视频独有，不可用时代替的,

④loop循环，

⑤controls是否添加自带的控制条

⑥error属性：值1：用户操作原因终止。值2：媒体可用，下载出错。值3：媒体可用，解码出错。值4：媒体格式不被支持。

⑦networkState属性：读取网络状态。值0：初始。值1编码格式已好，未建立网络连接。值2：数据加载中。值3没有支持的编码格式。

⑧currentSrc属性：只读属性，读取媒体数据的URL地址。

⑨buffered属性：只读属性表示是否缓存数据。

⑩readyState属性：只读属性表示当前播放位置的就绪状态。0.1.2.3.4

11.Seeking和seekable属性：均只读属性，seeking返回bool值表示特定位置的数据请求。Seekable表示数据请求时间范围

12.currentTime，startTime，duration属性：currentTime当前播放位置读写属性可修改。startTime开始播放位置时间，只读。Duration总的播放时间，只读。三者值均为时间。

13.played属性：读取已播放部分的时间段。只读属性

14.pause属性：是否暂停播放，返回bool。只读属性。

15.end属性：是否播放完毕。返回bool。只读属性。

16.volume属性：修改播放音量。值为0-1。

17.muted属性：修改静音状态，值为bool。

## 2.Video与audio的方法

公有方法：

1. Play：自动将元素的paused属性变为false。 Id.play();
2. Pause方法：自动将元素的paused属性变为true。
3. Load方法：自动将元素的playbackRate属性值变为defaultPlaybackRate属性的值。Error的值变为null。
4. canPlayType方法:测试浏览器是否支持指定的媒体类型。返回值：空字符串，maybe，probably  
   var support=videoElement.canPlayType(type); type与source的type相同。

## 事件

1.事件处理方式：

①监听：id.addEventListener(type,listener,useCapture)：type事件名称，listener绑定的函数，userCapture是一个bool，默认是false。

Id.addEventListener(“error”,function(){},false);

②获取事件句柄：

<video onplay=”fun()”></video>

Fun()

{}

`2.video和audio的事件

## Geolocation API

1.获取当前地理位置：

Void getCurrentPos(onSuccess,onError,options);

第一个参数：function(position){}。Position对象作为参数

第二个参数：function(error){}。error对象作为参数。Erro对象有2个属性：code属性(1:拒绝，2：获取不到，3：超时)，message属性(为一个空字符串)

第三个参数：可选属性列表。(enableHighAccuracy：是否要求高精度的位置信息)，(timeout：获取位置做一个超时限制，超时返回错误)，(maximumAge：对地理位置进行的缓存有效时间)

navigator.geolocation.getCurrentPosition(

function(position)

{

var coords=position.coords;

showMap(coords.latitude,coords.longitude,coords.accuracy);

},

function(error)

{

var errorTypes={

1:'拒绝',

2：'获取不到',

3:'获取信息超时'

};

alert(errorTypes[error.code]+":不能确定位置信息");

}

{

maximumAge:60\*1000\*2,

timeout:5000

}

);

2.持续监听地理位置

Int watchCurrentPosition(onSuccess,onError,options);

返回数字，可以被clearWatch方法使用，停止监视

3.停止监视

Void clearWatch(watchId);

4.Position对象

Latitude属性：地理位置的纬度

Longitude属性：位置精度

Altitude属性：海拔高度

Accuracy属性：经度和纬度的精度

altitudeAccurancy属性：海拔高度的精度

heading属性：设备前进的方向

speed属性：设备前进的速度

timestamp属性：获取地理位置信息时的时间