## 网络状态

我们可以通过window. navigator.onLine来检测，用户当前的网络状况，返回一个布尔值

addEventListener 进行绑定online用户从无网的网络状态被连接时调用

addEventListener 进行绑定.offline用户网络从有网到断开时被调用

事件是给window绑订的

## 地理定位

在HTML规范中，增加了获取用户地理信息的API，这样使得我们可以基于用户位置开发互联网应用，即基于位置服务 (Location Base Service)

### 获取地理信息方式

1、IP地址

2、三维坐标

GPS（Global Positioning System，全球定位系统）

Wi-Fi

手机信号

3、用户自定义数据

如下图对不同获取方式的优缺点进行了比较，浏览器会自动以最优方式去获取用户地理信息。



### 隐私

HTML5 Geolocation 规范提供了一套保护用户隐私的机制。必须先得到用户明确许可，才能获取用户的位置信息。

### API详解

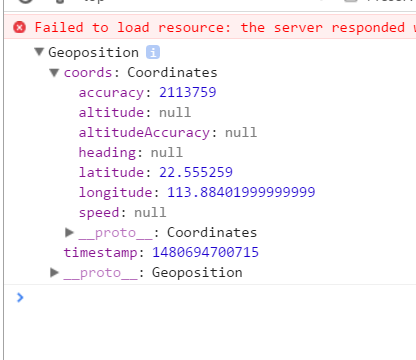
navigator.geolocation.getCurrentPosition(successCallback, errorCallback, options) 获取当前地理信息

navigator.geolocation.watchPosition(successCallback, errorCallback, options) 重复获取当前地理信息

1、当成功获取地理信息后，会调用succssCallback，并返回一个包含位置信息的对象position。

position.coords.latitude纬度

position.coords.longitude经度





1. 当获取地理信息失败后，会调用errorCallback，并返回错误信息error\



3、可选参数 options 对象可以调整位置信息数据收集方式

a)timeout 超时设置，单位为ms

百度地图的用法：

官网：http://lbsyun.baidu.com/

1. 进入官网 直接找到javascript API



1. 直接找到示例DEMO，复制源代码
2. 需要获取密钥 （自己申请，需要一到两个工作日）
3. 创建应用（填写浏览器端）
4. 利用密钥去替换script标签里面的“你的密钥”

## Web存储

随着互联网的快速发展，基于网页的应用越来越普遍，同时也变的越来越复杂，为了满足各种各样的需求，会经常性在本地存储大量的数据，传统方式我们以document.cookie来进行存储的，但是由于其存储大小只有4k左右，并且解析也相当的复杂，每一次发送请求都会携带上cookie，会造成带宽的浪费，给开发带来诸多不便，HTML5规范则提出解决方案。

web存储的含义是将数据存储到用户的电脑上，这样可以缓解服务器的压力，并且提高体验

### 特性

1、设置、读取方便

2、容量较大，sessionStorage约5M、localStorage约20M

3、只能存储字符串，可以将对象JSON.stringify() 编码后存储

### window.sessionStorage

1、生命周期为关闭当前页面窗口

2、不能多窗口下数据可以共享 （同源策略，百度家的local阿里他们家是拿不到）

补充：通过跳转可以

运用场景：

页面跳转的时候可以通过session实现数据共享

在一些单页面（spa）的运用中，进行页面传值的时候比较有用

### window.localStorage

1、永久生效，除非手动删除或用代码删除

2、可以多窗口共享（同源策略）

运用场景：一些不涉及到安全的一些数据（不要太过庞大）都可以存储到本地

### 方法详解

setItem(key, value) 设置存储内容

getItem(key) 读取存储内容

removeItem(key) 删除键值为key的存储内容

clear() 清空所有存储内容

key(n) 以索引值来获取键名

length 存储的数据的个数

区别：cookie数据始终在同源的[http请求](https://www.baidu.com/s?wd=http%E8%AF%B7%E6%B1%82&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmhfvnjIWmWI-PWPbnWDk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3Erj04PjRvnWndPHm1PWTkrHTz" \t "http://zhidao.baidu.com/_blank)中携带（即使不需要），即cookie在浏览器和服务器间来回传递。而sessionStorage和localStorage不会自动把数据发给服务器，仅在本地保存。cookie数据还有路径（path）的概念，可以限制cookie只属于某个路径下。存储大小限制也不同，cookie数据不能超过4k，同时因为每次[http请求](https://www.baidu.com/s?wd=http%E8%AF%B7%E6%B1%82&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmhfvnjIWmWI-PWPbnWDk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3Erj04PjRvnWndPHm1PWTkrHTz" \t "http://zhidao.baidu.com/_blank)都会携带cookie，所以cookie只适合保存很小的数据，如会话标识。sessionStorage和localStorage 虽然也有存储大小的限制，但比cookie大得多，可以达到5M或更大。数据有效期不同，sessionStorage：仅在当前浏览器窗口关闭前有效，自然也就不可能持久保持；localStorage：始终有效，窗口或浏览器关闭也一直保存，因此用作持久数据；cookie只在设置的cookie过期时间之前一直有效，即使窗口或浏览器关闭。作用域不同，sessionStorage不能在不同的浏览器窗口中共享，即使是同一个页面；localStorage 在所有同源窗口中都是共享的；cookie也是在所有同源窗口中都是共享的。

差异性：（面试经常问）

相同点：都是存储数据，存储在web端，并且都是同源

不同点：

（1）cookie 只有4K 小 并且每一次请求都会带上cookie 体验不好，浪费带宽

（2）session和local直接存储在本地，请求不会携带，并且容量比cookie要大的多

（3）session 是临时会话，当窗口被关闭的时候就清除掉 ，而 local永久存在，cookie有过期时间

（4）cookie 和local都可以支持多窗口共享，而session不支持多窗口共享 但是都支持a链接跳转的新窗口

## 全屏

HTML5规范允许用户自定义网页上任一元素全屏显示。

requestFullScreen() 开启全屏显示

cancelFullScreen() 关闭全屏显示 绑定给document的

## 文件读取

通过FileReader对象我们可以读取本地存储的文件，可以使用 [File](https://developer.mozilla.org/zh-CN/docs/DOM/File" \o "此页面仍未被本地化, 期待您的翻译!" \t "_blank)对象来指定所要读取的文件或数据。其中File对象可以是来自用户在一个 [<input>](https://developer.mozilla.org/zh-CN/docs/HTML/Element/input" \o "This article hasn't been written yet. Please consider contributing!" \t "_blank)元素上选择文件后返回的[FileList](https://developer.mozilla.org/zh-CN/docs/DOM/FileList" \o "一个FileList对象通常来自于一个HTML input元素的files属性,你可以通过这个对象访问到用户所选择的文件.该类型的对象还有可能来自用户的拖放操作,查看DataTransfer对象了解详情." \t "_blank)对象，也可以来自由拖放操作生成的  [DataTransfer](https://developer.mozilla.org/zh-CN/DragDrop/DataTransfer" \o "En/DragDrop/DataTransfer" \t "_blank)

### FileList对象

由于HTML5中我们可以通过为表单元素添加multiple属性，因此我们通过<input>上传文件后得到的是一个FileList对象（伪数组形式）。

### FileReader对象

Base64

http://www.tuicool.com/articles/IbYJvin

HTML5新增内建对象，可以读取本地文件内容。

var reader = new FileReader; 可以实例化一个对象

**实例方法**

1、[readAsDataURL](https://developer.mozilla.org/zh-CN/docs/Web/API/FileReader" \l "readAsDataURL())() 以DataURL形式读取文件

**事件监听**

onload 当文读取完成时调用

**属性**

result 文件读取结果

参考资料

<https://developer.mozilla.org/zh-CN/docs/Web/API/FileReader#toc>

http://www.cnblogs.com/jscode/archive/2013/04/27/3572239.html

## 1.6拖拽

在HTML5的规范中，我们可以通过为元素增加draggable="true"来设置此元素是否可以进行拖拽操作，其中图片、链接默认是开启的。

### 1.6.1拖拽元素

页面中设置了draggable="true"属性的元素，其中<img>、<a>标签默认是可以被拖拽的

### 1.6.2目标元素

页面中任何一个元素都可以成为目标元素

### 1.6.3事件监听

根据元素类型不同，需要设置不同的事件监听

**1、拖拽元素**

ondrag 应用于拖拽元素，整个拖拽过程都会调用

ondragstart 应用于拖拽元素，当拖拽开始时调用

ondragend 应用于拖拽元素，当拖拽结束时调用

**2、目标元素**

ondragenter 应用于目标元素，当拖拽元素进入时调用

ondragleave 应用于目标元素，当鼠标离开目标元素时调用

ondragover 应用于目标元素，当停留在目标元素上时调用

ondrop 应用于目标元素，当在目标元素上松开鼠标时调用

## 1.7多媒体

方法：load()、play()、pause()

属性：currentSrc、currentTime、duration

事件：

**参考文档**

<http://www.w3school.com.cn/tags/html_ref_audio_video_dom.asp>

http://www.jianshu.com/p/404d01b8e713

## 1.8应用

1、自定义视频播放器