HTML5 Web存储（Web Storage）技术以及用法

Web Storage的目的是为了克服由cookie带来的一些限制，当数据需要被严格控制在客户端上时，无须持续地将数据发回服务器。Web Storage的两个主要目标是：

a 提供一种在cookie之外存储会话数据的途径。

b 提供一种存储大量可以跨会话存在的数据的机制。

     Web Storage又分为两种： sessionStorage 和localStorage ，即这两个是Storage的一个实例。从字面意思就可以很清楚的看出来，sessionStorage将数据保存在session中，浏览器关闭也就没了；而localStorage则一直将数据保存在客户端本地； 不管是sessionStorage，还是localStorage，可使用的API都相同，常用的有如下几个（以localStorage为例）：

保存数据：localStorage.setItem(key,value);

读取数据：localStorage.getItem(key);

删除单个数据：localStorage.removeItem(key);

删除所有数据：localStorage.clear();

得到某个索引的key：localStorage.key(index);

***如上，key和value都必须为字符串，换言之，web Storage的API只能操作字符串。***

1、sessionStorage对象

      sessionStorage对象存储特定于某个对话的数据，也就是它的生命周期为当前窗口或标签页，一旦窗口或标签页被永久关闭了，那么所有通过sessionStorage存储的数据也就被清空了。存储在sessionStorage中的数据可以跨越页面刷新而存在，同时如果浏览器支持，浏览器崩溃并重启之后依然可以可用（注意：Firefox和Weblit都支持，IE则不行）。

      因为sessionStorage对象绑定于某个服务器会话，所以当文件在本地运行的时候是不可用的。存储在sessionStorage中的数据只能由最初给对象存储数据的页面访问到，所以对多页面应用有限制。



 

sessionStorage的生命周期 ：用户访问浏览器页面，从进入到离开的这段时间；

sessionStorage仅在当前会话有效，关闭页面或浏览器后被清除；

Firefox和Webkit实现了同步写入，所以添加到存储空间中的数据是立刻被提交的。

IE的实现则是异步写入数据，所以在设置数据和将数据实际写入磁盘之间可能有一些延迟。

**2、localStorage 对象**

     localStorage一直将数据保存在客户端本地。localStorage 和sessionStorage的用法相似，这里就不做详细的说明。无法手动删除的时候，除非被清除 clear()，否则永久保存。

**不同浏览器无法共享localStorage和sessionStorage中的信息。**

**同一浏览器的相同域名和端口的不同页面间可以共享相同的 localStorage，**

**但是不同页面间无法共享sessionStorage的信息。**

**这里需要注意的是，页面仅指顶级窗口，如果一个页面包含多个iframe且他们属于同源页面，那么他们之间是可以共享sessionStorage的。**

