**一、本地存储由来的背景**

由于HTML4时代Cookie的大小、格式、存储数据格式等限制，网站应用如果想在浏览器端存储用户的部分信息，那么只能借助于Cookie。但是Cookie的这些限制，也就导致了Cookie只能存储一些ID之类的标识符等简单的数据。

下面是Cookie的限制：

　　　　大多数浏览器支持最大为 4096 字节的 Cookie。

　　　   浏览器还限制站点可以在用户计算机上存储的 Cookie 的数量。大多数浏览器只允许每个站点存储 20 个Cookie；如果试图存储更多 Cookie，则最旧的 Cookie 便会被丢弃。

　　　　有些浏览器还会对它们将接受的来自所有站点的 Cookie 总数作出绝对限制，通常为 300 个。  
　　　　Cookie默认情况都会随着Http请求发送到后台服务器，但并不是所有请求都需要Cookie的，比如：js、css、图片等请求则不需要Cookie。  
　　　　为了破解Cookie的一系列限制，HTML5通过JS的新的API就能直接存储大量的数据到客户端浏览器，而且支持复杂的本地数据库，让JS更有效率。 HTML5支持两种的WebStorage：

　　　　永久性的本地存储（localStorage）  
　　　　会话级别的本地存储（sessionStorage）

**二、本地存储的分类**

　　H5本地存储有两个API，一个是Web Storage，还有一个是Web SQL。不管是哪一个，都是基于JavaScript语言来使用，接下来我就教你怎么使用Web Storage

**三、Web Storage**

　　　HTML5 定义了本地存储规范 Web Storage ， 提供了两种存储类型 API：  sessionStorage 和 localStorage，二者的差异主要是数据的保存时长及数据的共享方式。

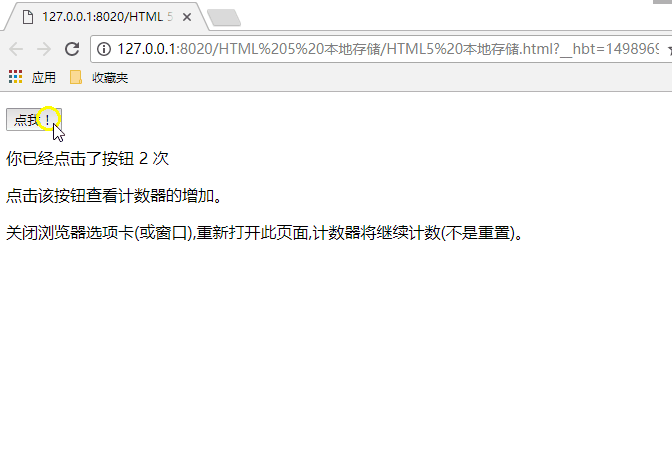
**1.localStorage** 一直存储在本地，数据存储是永久的，除非用户或程序对其进行删除操作；localStorage 对象存储的数据没有时间限制。第二天、第二周或下一年之后，数据依然可用。

**特点：①**域内安全、永久保存。即客户端或浏览器中来自同一域名的所有页面都可访问localStorage数据且数据除了删除否则永久保存，但客户端或浏览器之间的数据相互独立。

**②** 数据不会随着Http请求发送到后台服务器；

**③**存储数据的大小机会不用考虑，因为在HTML5的标准中要求浏览器至少要支持到4MB。

看一个例子：

+

代码如下：

[复制代码](javascript:void(0);)

function clickCounter(){

if(typeof(Storage)!=="undefined"){

if(localStorage.clickcount){

localStorage.clickcount=Number(localStorage.clickcount)+1;

}else{

localStorage.clickcount=1;

}

document.getElementById("result").innerHTML=" 你已经点击了按钮 " + localStorage.clickcount + " 次 ";

}else{

document.getElementById("result").innerHTML="对不起，您的浏览器不支持 web 存储。";

}

}

</script>

<p><button onclick="clickCounter()" type="button">点我！</button></p>

<div id="result"></div>

<p>点击该按钮查看计数器的增加。</p>

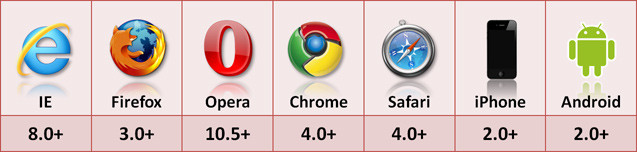
<p>关闭浏览器选项卡(或窗口),重新打开此页面,计数器将继续计数(不是重置)。</p>

[复制代码](javascript:void(0);)

**2.sessionStorage**在会话期内有效，数据在浏览器关闭后自动删除；

**特点：**会话控制、短期保存。会话概念与服务器端的session概念相似，短期保存指窗口或浏览器或客户端关闭后自动消除数据。

**兼容性**

****

　　注意：IE9 localStorage不支持本地文件，需要将项目署到服务器，才可以支持！

　　目前所有主流的浏览器都在一定程度上支持 HTML5 的 Web Storage特性。 由上图可以看出，基本上所有现代浏览器都已经支持 Web Storage。

　　Android平台和 IOS 平台各自的浏览器都基本上支持 Web Storage 本地存储特性。 目前市场上的移动设备， 除了 android 手机和 iphone 手机外，越来越多的平板电脑面世，而且基本上依赖着两种平台。在移动端使用 Web Storage 我们几乎不需要考虑浏览器是否支持， 当然从代码的严谨来说，建议最好在使用前先检查浏览器是否支持

下面是检测方式：

[复制代码](javascript:void(0);)

if (window.localStorage) {

// 浏览器支持 localStorage

}else{

// 不支持

}

if (window.sessionStorage) {

// 浏览器支持 sessionStorage

}else{

// 不支持

}

[复制代码](javascript:void(0);)

**三、localStorage**

　　HTML5 的本地存储 API 中的 localStorage 与 sessionStorage 在使用方法上是相同的，区别在于 sessionStorage 在关闭页面后即被清空，而 localStorage 则会一直保存。我们这里以 localStorage 为例，简要介绍下 html5 的本地存储，并针对如遍历等常见问题作一些示例说明。 localStorage 是 Html5 本地存储的 API，使用键值对的方式进行存取数据，存取的数据只能是字符串。不同浏览器对该 API 支持情况有所差异，如使用方法、最大存储空间等。

**存储方式：以键值对(Key-Value)的方式存储字符串。**

**主要应用：**购物车、客户登录、游戏存档。。。

**可储存的数据类型：**数组，图片，json，样式，脚本。。。（只要是能序列化成字符串的内容都可以存储）

**储存地址：**C:\Users\15014\AppData\Local\Google\Chrome\User Data\Default\Local Storage（不同电脑不一样，需要打开隐藏文件显示，但是在C盘搜索localStorage就能搜出这个文件夹。）

 localStorage提供了四个方法来辅助我们进行对本地存储做相关操作。

（1）localStorage.setItem(键名，键值)在本地客户端存储一个字符串类型的数据，其中，第一个参数“键名”代表了该数据的标识符，而第二个参数“键值”为该数据本身。如：

localStorage.setItem("name", "张三"); //存储键名为name和键值为"张三"的数据到本地

localStorage.setItem("age", "28"); //存储键名为age和键值为"28"的数据到本地

　　（2）localStorage.getItem(键名) 读取已存储在本地的数据，通过键名作为参数读取出对应键名的数据。如：

var data = localStorage.getItem("name");

alert(data);//张三

　　（3）localStorage.removeItem(键名)移除已存储在本地的数据，通过键名作为参数删除对应键名的数据。如：

var data2 = localStorage.removeItem("name");//从本地存储中移除键名为name的数据

alert(data2); //undefined

　　（4）localStorage.clear() 移除本地存储所有数据。如：

localStorage.clear() 移除本地存储所有数据。如：

localStorage.clear(); //保存着的"age/28"和"name/张三"的键/值对也被移除了，所有本地数据拜拜

 　　（5）另外，sessionStorage中的四个函数与以上localStorage类的函数用法基本一致，就不再详解。

**下面是一个小实例：**

[复制代码](javascript:void(0);)

<script type="text/javascript">

localStorage.setItem("name", "张三");

localStorage.setItem("age", "28");

verify(); //验证本地存储

localStorage.removeItem("name");

verify(); //验证name是否存在

localStorage.clear();

verify(); //验证name和age是否存在

//自定义验证函数，验证name和age的数据是否存在

function verify(){

var type = localStorage.getItem("name");

var price = localStorage.getItem("age");

type = type ? type : '不存在';

price = price ? price : '不存在';

alert( "name: " + type + "\n\n" + "age: " + price );

}

</script>

[复制代码](javascript:void(0);)

**三、localStorage过期策略**

　　由于html5没有给本地存储设置过期策略，那么在处理数据的过期策略的时候可以编写自己过期策略程序，如下：

[复制代码](javascript:void(0);)

<!DOCTYPE>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta http-equiv="Access-Control-Allow-Origin" content="anonymous">

<title>locstorage 过期策略</title>

</head>

<body>

</body>

</html>

<script>

function set(key,value){

var curtime = new Date().getTime();//获取当前时间

localStorage.setItem(key,JSON.stringify({val:value,time:curtime}));//转换成json字符串序列

/\*

说明：

JSON.parse用于从一个字符串中解析出json对象,如

var str = '{"name":"huangxiaojian","age":"23"}'

结果：

JSON.parse(str)

Object

age: "23"

name: "huangxiaojian"

\_\_proto\_\_: Object

注意：单引号写在{}外，每个属性名都必须用双引号，否则会抛出异常。

JSON.stringify()用于从一个对象解析出字符串，如

var a = {a:1,b:2}

结果：

JSON.stringify(a)

"{"a":1,"b":2}"

\*/

}

function get(key,exp)//exp是设置的过期时间

{

var val = localStorage.getItem(key);//获取存储的元素

var dataobj = JSON.parse(val);//解析出json对象

if(new Date().getTime() - dataobj.time > exp)//如果当前时间-减去存储的元素在创建时候设置的时间 > 过期时间

{

console.log("expires");//提示过期

}

else{

console.log("val="+dataobj.val);

}

}

</script>

[复制代码](javascript:void(0);)

使用操作如下图所示：

