

浏览器环境中提供了两个对象，localStorage 和 sessionStorage，专门用于在客户端存储数据，数据是以键值对的形式被存储在浏览器的缓存中。

只能存储字符串类型的值。

永久存储

有效时长

使用 localStorage 对象存储的数据会被永久的保存到浏览器的缓存数据中，除非 web 应用自己删除或者用户清除浏览器缓存，否则数据将会被永久保存，永不过期。

可见范围

同源的文档间共享相同的localStorage数据。它们可以互相读取、修改和删除对方存储的数据。但是非同源的文档之间存储的数据是相互对立的，彼此无法共享。

同一个文档在不同的浏览器中存储的数据是无法共享的。例如，在 Chrome 浏览器中存储的数据是无法在 Firefox 浏览器中获取的。

存储方法

通过键值对的方式存储数据

```
1 localStorage.setItem('name', '张三');  
2 localStorage.setItem('age', '20');
```

根据存储时使用的键名获取对应的值

```
1 localStorage.getItem('name'); // '张三'  
2 localStorage.getItem('age'); // '20'
```

根据存储时使用键名修改键对应的值

```
1 localStorage.setItem('name', '李四');  
2 localStorage.setItem('age', '18');
```

根据存储时使用键名删除对应的值

```
1 localStorage.removeItem('name');
```

清空同源文档存储的所有键值对

```
1 localStorage.clear();
```

遍历同源文档存储的所有键值对

```
1 // length 表示键值对的总数  
2 // 使用 key(index) 方法获取索引对应的 key  
3 for(var i = 0; i < localStorage.length; i++) {  
4   var key = localStorage.key(i);           // 键名  
5   var value = localStorage.getItem(key);    // 键名对应的值  
6 }
```

临时存储

有效时长

使用 sessionStorage 对象存储的数据只在页面被打开期间有效，一旦浏览器窗口或者标签页被关闭，那么所有存储的数据都将被删除。

需要注意的是，现代的浏览器已经具备了重新打开最近关闭的标签页随后恢复上一次浏览的会话功能，因此，这些标签页以及与之相关的 `sessionStorage` 的有效期可能会更加长些。

可见范围

与 `localStorage` 一样，`sessionStorage` 的可见范围也是限定在文档源中，因此非同源文档间都是无法共享 `sessionStorage` 的。

不仅如此，`sessionStorage` 的可见范围还被限定在窗口中。如果同源的文档渲染在不同的浏览器标签页中，那么它们互相之间拥有的是各自的 `sessionStorage` 数据，无法共享；一个标签页中的脚本是无法读取或者覆盖由另一个标签页脚本写入的数据，哪怕这两个标签页渲染的是同一个页面，运行的是同一个脚本也不行。

要注意的是，这里提高到的基于窗口可见范围的 `sessionStorage` 指的窗口只是顶级窗口。如果一个浏览器标签页包含两个 `<iframe>` 元素，它们所包含的文档是同源的，那么这两者之间是可以共享 `sessionStorage` 的。

存储方法

`sessionStorage` 对象的方法与 `localStorage` 对象的方法是相同的，参考 `localStorage` 的存储方法，只需要将 `localStorage` 替换成 `sessionStorage` 即可。

存储事件

无论什么时候存储在 `localStorage` 或 `sessionStorage` 的数据发生改变，浏览器都会在其他对该数据可见的窗口对象上触发存储事件（但是，在对数据进行改变的窗口对象上是不会触发的）。

如果浏览器有两个标签页都打开了来自同源的页面，其中一个页面在 `localStorage` 上存储了数据，那么另一个标签页就会接收到一个存储事件。要记住的是 `sessionStorage` 的可见范围是限制在顶层窗口的，因此对 `sessionStorage` 的改变只有当有相牵连的窗口的时候才会触发存储事件。还需要注意的是，只有当存储数据真正发生改变时才会触发存储事件。像给已经存在的存储项设置一个一模一样的值，抑或是删除一个本来就不存在的存储项都是不会触发存储事件的。

为存储事件注册处理程序可以使用 `addEventListener()` 方法。在绝大多数浏览器中，还可以使用给 `window` 对象设置 `onstorage` 属性的方式。

与存储事件相关的事件对象有5个非常重要的属性：

- key：修改或删除的key值，如果调用clear()时，该属性值为null
- newValue：新设置的值，如果调用removeItem()时，该属性值为null
- oldValue：调用改变前的value值；添加新项时，该属性值为null
- storageArea：当前的storage对象
- url：触发该存储变化脚本所在文档的URL（也就是说哪个页面修改的数据）

作业

1. 下面哪两个 URL 都拥有相同的文档源？

http://www.foo.com:8088/path/to/file1.html
https://www.foo.com:8088/path/to/file2.html
https://www.bar.com:8088/path/to/file3.html
https://www.foo.com:8080/path/to/file4.html
https://www.foo.com:8088/path/to/file5.html

2. 请简述 localStorage 与 sessionStorage 的区别。