

cookie、sessionStorage、localStorage

相同：sessionStorage、localStorage、cookie都是在浏览器端存储的数据

sessionStorage 和 localStorage 是HTML5 Web Storage API 提供的，可以方便的在web请求之间保存数据。

区别

1.cookie数据始终在同源的http请求中携带（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。

2.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。

sessionStorage 和 localStorage 之间的区别

3.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。

4.作用域不同，sessionStorage不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。

5.Web Storage 支持事件通知机制，可以将数据更新的通知发送给监听者。

6.Web Storage 的 api 接口使用更方便。

默认各种浏览器都支持，缺陷就是每次请求，浏览器都会把本机存的cookies发送到服务器，无形中浪费带宽。

作用

1.就可以避免数据在浏览器和服务器间不必要地来回传递。

2.快速显示数据：性能好，从本地读数据比通过网络从服务器获得数据快得多，本地数据可以即时获得。

sessionStorage

适用场合：临时存储：很多时候数据只需要在用户浏览一组页面期间使用，关闭窗口后数据就可以丢弃了，这种情况使用sessionStorage非常方便。

特点：sessionStorage是在同源的同窗口（或tab）中，始终存在的数据。也就是说只要这个浏览器窗口没有关闭，即使刷新页面或进入同源另一页面，数据仍然存在。关闭窗口后，sessionStorage即被销毁。同时“独立”打开的不同窗口，即使是同一页面，sessionStorage对象也是不同的。

sessionStorage用于本地存储一个会话（session）中的数据，这些数据只有在同一个会话中的页面才能访问并且当会话结束后数据也随之销毁。因此sessionStorage不是一种持久化的本地存储，仅仅是会话级别的存储。

localStorage用于持久化的本地存储，除非主动删除数据，否则数据是永远不会过期的。

与Cookie相比，Web Storage存在不少的优势，概括为以下几点：

- 存储空间更大：IE8下每个独立的存储空间为10M，其他浏览器实现略有不同，但都比Cookie要大很多。
- 存储内容不会发送到服务器：当设置了Cookie后，Cookie的内容会随着请求一并发送到服务器，这对于本地存储的数据是一种带宽浪费。而Web Storage中的数据则仅仅是存在本地，不会与服务器发生任何交互。
- 更多丰富易用的接口：Web Storage提供了一套更为丰富的接口，使得数据操作更为简便。
- 独立的存储空间：每个域（包括子域）有独立的存储空间，各个存储空间是完全独立的，因此不会造成数据混乱。