

一、cookie

cookie算是比较早的技术，最初是为了记录http的状态，提高访问速度。cookie是服务器"种植"在客户端的key-value形式文本文件。但同时客户端也能操作cookie。特点：

大小：cookie的大小限制在4k。每个域名下cookie的个数现在在20个。

在客户端请求服务器端和服务器响应时，cookie始终被携带在http请求中，即使不需要(造成流量浪费)。这也是限制cookie大小的原因。

客户端可以通过document.cookie操作cookie，并不安全。

cookie可以设置过期时间、路径、域和httpOnly等字段。如果设置了过期时间，cookie会保存在硬盘里，知道到了设定的过期时间才会失效。若未设置过期时间，在浏览器窗口关闭时，cookie就失效了。路径和域两个字段限制了cookie的作用范围。httpOnly设置为true，则js不能通过document.cookie操作cookie。

二、localStorage

它也是采用key-value的形式存储数据，但是它与cookie有很大的区别特点：

对比着来，localStorage能保存更大的数据，标准浏览器是5Mb。

localStorage保存在客户端，不随着请求发送给服务器，避免了流量的浪费。

客户端可以通过：setItem、getItem方法访问localStorage。

并且，localStorage没有过期时间，如果不手动清除，数据就永远不会过期，一直保存在浏览器当中。存储的信息在同一域中是共享的。

三、sessionStorage

特点：

与localStorage不同的是，sessionStorage并不持久化，在窗口关闭那一刻，sessionStorage会被清除。

存储的信息是会话级别的，同域也是不能共享的。关闭当前标签页，sessionStorage即失效。

四、session(附加)

上面提到了cookie，顺带提一下session。客户端第一次访问服务器，服务器种植一个cookie，保存唯一的sessionId。后面客户端再次访问，会读取此sessionId，随即能在服务端读取到此id保存的会话对象。

特点：

session是基于cookie的，由于session在客户端不可被修改，相对于cookie来说安全，所以可存放一些重要数据。

数据保存在服务器端，客户端通过sessionId，读取到相对应的数据。

五、用法

对浏览器来说，使用 Web Storage 存储键值对比存储 Cookie 方式更直观，而且容量更大，它包含两种：localStorage 和 sessionStorage

sessionStorage(临时存储)：为每一个数据源维持一个存储区域，在浏览器打开期间存在，包括页面重新加载

localStorage(长期存储)：与 sessionStorage 一样，但是浏览器关闭后，数据依然会一直存在
sessionStorage 和 localStorage 的用法基本一致，引用类型的值要转换成JSON