

三、本地存储

在 JS 中我们把一部分信息存储到“当前浏览器”“某一个域/源”中的的某一个位置,浏览器关闭或者页面关闭,存储的信息不消失,以后重新打开页面依然可以从上一次存储信息的位置把内容获取到...

=>谷歌下存储的信息,IE 浏览器无法获取

=>京东在谷歌下存储的信息,百度再看谷歌下也获取不到

本地存储可以实现不同页面之间的信息通讯,在 A 页面中操作的一些信息,我们把它存储到本地,这样的话在 B 页面的时候我们可以再从本地把信息获取到(还有页面跳转的时候我们 URL 地址栏的问号传参其实也是在实现页面和页面之间的通信)

在当前市场上最常用的本地存储方式: COOKIE、
webStorage(localStorage/sessionStorage) =>你们之前听过的 session/数据库等都是服务器存储四、COOKIE PK webStorage

cookie 兼容所有的浏览器,而 webStorage 是 H5 新增加的,不兼容 IE 低版本浏览器

不管是使用 cookie 还是 webStorage 存储的内容,都可以在谷歌浏览器控制台的 Resources 选项中查看到(没有

Resources 选项的我们可以使用 Application 选项查看)

1、使用的语法

[cookie]

`document.cookie=''` 设置/修改

(NAME/VALUE/EXPIRES/DOMAIN/PATH...)

`document.cookie` 获取(还需要把全部的信息使用正则匹配出我们需要的结果)

删除的原理就是设置 `cookie` 信息存储的时间是今天以前一个时间即可,这样 `cookie` 就过期了,也就删除了

[webStorage] `localStorage/sessionStorage`

`localStorage.setItem([key],[value])` 设置/修改

`localStorage.getItem([key])` 获取

`localStorage.removeItem([key])` 移除

`localStorage.clear()` 它是把当前域/源下所有的使用 `localStorage` 存储的内容都移除掉

`localStorage.key([index])` 通过索引获取指定位置存储的 KEY

`localStorage` 是本地存储,不管是刷新页面还是关掉页面,存储的信息都不会消失

`sessionStorage` 是本地会话存储,只要页面关掉,证明会话结束了,浏览器会把它存储的信息也清空掉

2、安全

`cookie` 和 `localStorage` 都是明文存储,所以在使用它们做存储的时候保持以下的原则:

->尽量不要存储过于重要的信息(用户账号和密码、银行卡账号、身份证、手机...)

->如果非要存储的话,我们需要把存储的信息进行加密

不可逆转加密:密码... 最常用的是 **MD5**,但是其它也可以自己回去扩展一下

可逆转加密:手机、银行卡号、用户名... 自己回去查去,每个公司一般都用自己加密解密方法来做

3、cookie pk localStorage

[cookie]

->兼容所有的浏览器

->`cookie` 存储的大小有限制,一般同一个源下最多只能存储 **4KB** 的大小

->`cookie` 有过期的时间(过期时间自己可以设置,一般不要超过一个月)

->我们使用 **360** 或者浏览器自带清理垃圾的功能,很

多时候会把 **cookie** 清除掉;用户可能出于安全的角度禁止存储 **cookie**,例如: 无痕浏览器、隐身模式...

->**cookie** 不是单纯的本地存储,它的相关操作其实和服务器之间还有千丝万缕的联系

[localStorage]

->不兼容 IE 低版本的浏览器

->**localStorage** 也会存在大小的限制,一般同一个源下只能存储 5MB 的内容

->**localStorage** 不存在过期时间,只要我们不删除,就会永久存储到本地

->安全卫士和浏览器的清除功能不能清除

localStorage;无痕浏览器等模式对于 **localStorage** 没有影响;

->**localStorage** 是正统的本地存储,和服务器没有任何的关系

=>自动登录 OR 记住用户名密码

如果是做移动端的开发,那么我们使用 **cookie** 或者

`localStorage` 都可以;移动端或者是不需要考虑兼容的PC端, 我们使用 `localStorage` 还可以做一个数据缓存的优化....

