

浏览器存储

Cookie, sessionStorage, localStorage这三者都可以被用来在浏览器端存储数据，而且都是字符串类型的键值对！

注意：session和sessionStorage不是一个概念！！！在服务器有一种存储方式叫做：session会话存储，常常被简称session
后期Node课程中会对cookie和后端所使用的session会话存储进行详细讲解

session: 会话

sessionStorage: 浏览器端用于存储数据的容器，常常被前端人员简称为session。

session会话存储：服务器端一种存储数据的方式，常常被后端人员简称为session。

Web Storage

sessionStorage和localStorage都是浏览器本地存储，统称为web storage，存储内容大小一般支持5-10MB

浏览器端通过 window.sessionStorage 和 window.localStorage 属性来实现本地存储机制。

相关API:

1. xxxxxStorage.setItem('key', 'value');

该方法接受一个键名和值作为参数，将会把键值对添加到存储中，如果键名存在，则更新其对应的值。

2. var data = xxxxxStorage.getItem('person');

该方法接受一个键名作为参数，返回键名对应的值。

3. xxxxxStorage.removeItem('key');

该方法接受一个键名作为参数，并把该键名从存储中删除。

4. xxxxxStorage.clear()

调用该方法会清空存储中的所有键名

备注：sessionStorage存储的内容会随着浏览器窗口关闭而消失。

localStorage存储的内容，需要手动清除才会消失。

storage事件:

1. Storage对象发生变化时触发（即创建/更新/删除数据项时，Storage.clear() 只会触发一次）
2. 在同一个页面内发生的改变不会起作用
3. 在相同域名下的其他页面发生的改变才会起作用。（修改的页面不会触发事件，与它共享的页面会触发事件）

key	: 修改或删除的key值，如果调用clear(),为null
newValue	: 新设置的值，如果调用clear(),为null
oldvalue	: 调用改变前的value值,如果调用clear(),为null
url	: 触发该脚本变化的文档的url
storageArea	: 当前的storage对象

使用方法:

```
window.addEventListener('storage',function (event) {  
    //此处写具体业务逻辑  
})
```

浏览器储存量的支持

<http://dev-test.nemikor.com/web-storage/support-test/>

数据同步（跨页签通信）

数据同步只能实现同一网站的不同页签的数据同步，不能跨网站同步。

```
// 数据同步的实现：
// 页签1
<input type="text" id="input">

let inputNode = document.getElementById('input')
inputNode.onblur = function () {
  localStorage.setItem('data',inputNode.value)
}

// 页签2
<input type="text" id="input2">

let inputNode = document.getElementById('input2')
window.addEventListener('storage',function (event) {
  inputNode.value = event.newValue
  //inputNode.value = localStorage.getItem('data')
})
```