1. Js中的cookie使用

设置cookie:document.cookie=” name=value; expires=Tue, 10 Oct 2017 16:14:47 GMT; path=/; domai=“域名””

使用的场景：1、保存用户登录的状态

2、跟踪用户的行为

3、定制页面。例如换肤

4、创建购物车，例如淘宝网就使用cookie记录了用户曾经浏览过的商品，方便随时进行比较

Cookie的缺点

（1）cookie可能被禁用。当用户非常注重个人隐私保护时，他很可能禁用浏览器的cookie功能；   
（2）cookie是与浏览器相关的。这意味着即使访问的是同一个页面，不同浏览器之间所保存的cookie也是不能互相访问的；   
（3）cookie可能被删除。因为每个cookie都是硬盘上的一个文件，因此很有可能被用户删除；   
（4）cookie安全性不够高。所有的cookie都是以纯文本的形式记录于文件中，因此如果要保存用户名密码等信息时，最好事先经过加密处理。

1. 服务其中的cookie

设置setcookie(“变量名”，“值”)

获取$\_COOKIE(“存储时变量名”)

获取：document.cookie

1. sessionStorage的使用

sessionStorage.getItem(key):获取指定key本地存储的值

sessionStorage.setItem(key,value)：将value存储到key字段

sessionStorage.removeItem(key):删除指定key本地存储的值

sessionStorage.length是sessionStorage的项目数

4、LocalStorage的使用

设置值：LocalStorage.setItem(“name”,”值”)

获取：LocalStorage.getItem(name)

LoclStorage.removeItem(name)

清除：LocalStorage.clear( )

sessionStorage 、localStorage 和 cookie 之间的区别

共同点：都是保存在浏览器端，且同源的。

区别：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；cookie数据还有路径（path）的概念，可以限制cookie只属于某个路径下。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。

而sessionStorage和localStorage不会自动把数据发给服务器，仅在本地保存。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。

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。

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。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。Web Storage 的 api 接口使用更方便

Json是一种一种轻量级的数据交换格式

JSON.stringify方法用于将一个值转为字符串。该字符串符合 JSON 格式，并且可以被JSON.parse方法还原

**JSON对值的类型和格式有严格的规定**

* 复合类型的值只能是数组或对象，不能是函数、正则表达式对象、日期对象。
* 简单类型的值只有四种：字符串、数值（必须以十进制表示）、布尔值和null（不能使用NaN, Infinity, -Infinity和undefined）。
* 字符串必须使用双引号表示，不能使用单引号。
* 对象的键名必须放在双引号里面。
* 数组或对象最后一个成员的后面，不能加逗号。

　　PS: 需要注意的是，空数组和空对象都是合格的JSON值，null本身也是一个合格的JSON值。

字符串的语法规则：

* 数组（Array）用方括号(“[]”)表示。
* 对象（Object）用大括号（”{}”）表示。
* 名称/值对（name/value）组合成数组和对象。
* 名称（name）置于双引号中，值（value）有字符串、数值、布尔值、null、对象和数组。
* 并列的数据之间用逗号（“,”）分隔

1 var obj = {

2 'prop1': 'value1',

3 'prop2': 'value2',

4 'prop3': 'value3'

5 };

6

7 var selectedProperties = ['prop1', 'prop2'];

8

9 JSON.stringify(obj, selectedProperties)

10 // "{"prop1":"value1","prop2":"value2"}"

[复制代码](javascript:void(0);)

　　上面代码中，JSON.stringify方法的第二个参数指定，只转prop1和prop2两个属性。

这个类似“白名单”的数组，只对对象的属性有效，对数组无效。

http://yunkus.com/physical-pixel-device-independent-pixels/