第 29 章 Cookie 与存储

学习要点：

1.cookie

2.cookie 局限性

3.其他存储

主讲教师：李炎恢

合作网站：http://www.ibeifeng.com

讲师博客：http://hi.baidu.com/李炎恢

随着 Web 越来越复杂，开发者急切的需要能够本地化存储的脚本功能。这个时候，第一个出现的方案：cookie 诞生了。cookie 的意图是：在本地的客户端的磁盘上以很小的文件形式保存数据。

一. Cookie

cookie 也叫 HTTP Cookie，最初是客户端与服务器端进行会话使用的。比如，会员登录，下次回访网站时无须登录了；或者是购物车，购买的商品没有及时付款，过两天发现购物车里还有之前的商品列表。

HTTP Cookie 要求服务器对任意 HTTP 请求发送 Set-Cookie，因此，Cookie 的处理原则上需要在服务器环境下进行。当然，现在大部分浏览器在客户端也能实现 Cookie 的生成和获取。(目前 Chrome 不可以在客户端操作，其他浏览器均可)

cookie 的组成

cookie 由名/值对形式的文本组成：name=value。完整格式为：

name=value; [expires=date]; [path=path]; [domain=somewhere.com]; [secure]

中括号是可选，name=value 是必选。

```
document.cookie = 'user=' + encodeURIComponent('李炎恢');   //编码写入
alert(decodeURIComponent(document.cookie));                 //解码读取
```

expires=date 失效时间，如果没有声明，则为浏览器关闭后即失效。声明了失效时间，那么时间到期后方能失效。

```
var date = new Date();                                  //创建一个
date.setDate(date.getDate() + 7);
document.cookie = "user= " + encodeURIComponent('李炎恢') +";expires=" + date;
```

PS：可以通过 Firefox 浏览器查看和验证失效时间。如果要提前删除 cookie 也非常简单，只要重新创建 cookie 把时间设置当前时间之前即可：date.getDate() - 1 或 new Date(0)。

path=path 访问路径，当设置了路径，那么只有设置的那个路径文件才可以访问 cookie。

```
var path = '/E:/%E5%A4%87%E8%AF%BE%E7%AC%94%E8%AE%B0/JS1/29/demo';
document.cookie = "user= " + encodeURIComponent('李炎恢') + ";path=" + path;
```

PS：为了操作方便，我直接把路径复制下来，并且增加了一个目录以强调效果。

domain=domain 访问域名，用于限制只有设置的域名才可以访问，那么没有设置，会默认限制为创建 cookie 的域名。

```
var domain = 'yc60.com';
document.cookie = "user= " + encodeURIComponent('李炎恢') + ";domain=" + domain;
```

PS：如果定义了 yc60.com，那么在这个域名下的任何网页都可访问，如果定义了 v.yc60.com，那么只能在这个二级域名访问该 cookie，而主域名和其他子域名则不能访问。

PS：设置域名，必须在当前域名绑定的服务器上设置，如果在 yc60.com 服务器上随意设置其他域名，则会无法创建 cookie。

secure 安全设置，指明必须通过安全的通信通道来传输(HTTPS)才能获取 cookie。

```
document.cookie = "user= " + encodeURIComponent('李炎恢') + ";secure";
```

PS：https 安全通信链接需要单独配置。

JavaScript 设置、读取和删除并不是特别的直观方便，我们可以封装成函数来方便调用。

```
//创建 cookie
function setCookie(name, value, expires, path, domain, secure) {
    var cookieText = encodeURIComponent(name) + '=' + encodeURIComponent(value);
    if (expires instanceof Date)  {
        cookieText += '; expires=' + expires;
    }
    if (path) {
        cookieText += '; expires=' + expires;
    }
    if (domain) {
        cookieText += '; domain=' + domain;
    }
    if (secure) {
        cookieText += '; secure';
    }
    document.cookie = cookieText;
}

//获取 cookie
function getCookie(name) {
    var cookieName = encodeURIComponent(name) + '=';
    var cookieStart = document.cookie.indexOf(cookieName);
    var cookieValue = null;

    if (cookieStart > -1) {
```

```
            var cookieEnd = document.cookie.indexOf(';', cookieStart);
            if (cookieEnd == -1) {
                cookieEnd = document.cookie.length;
            }
            cookieValue = decodeURIComponent(
                document.cookie.substring(cookieStart + cookieName.length, cookieEnd));
        }
        return cookieValue;
    }

    //删除 cookie
    function unsetCookie(name) {
        document.cookie = name + "= ; expires=" + new Date(0);
    }

    //失效天数，直接传一个天数即可
    function setCookieDate(day) {
        if (typeof day == 'number' && day > 0) {
            var date = new Date();
            date.setDate(date.getDate() + day);
        } else {
            throw new Error('传递的 day 必须是一个天数，必须比 0 大');
        }
        return date;
    }
```

二．cookie 局限性

cookie 虽然在持久保存客户端用户数据提供了方便，分担了服务器存储的负担。但是还有很多局限性的。

第一：每个特定的域名下最多生成 20 个 cookie（根据不同的浏览器有所区别）。

1.IE6 或更低版本最多 20 个 cookie

2.IE7 和之后的版本最多可以 50 个 cookie。IE7 最初也只能 20 个，之后因被升级不定后增加了。

3.Firefox 最多 50 个 cookie

4.Opera 最多 30 个 cookie

5.Safari 和 Chrome 没有做硬性限制。

PS：为了更好的兼容性，所以按照最低的要求来，也就是最多不得超过 20 个 cookie。当超过指定的 cookie 时，浏览器会清理掉早期的 cookie。IE 和 Opera 会清理近期最少使用的 cookie，Firefox 会随机清理 cookie。

第二：cookie 的最大大约为 4096 字节(4k)，为了更好的兼容性，一般不能超过 4095 字节即可。

第三：cookie 存储在客户端的文本文件，所以特别重要和敏感的数据是不建议保存在 cookie 的。比如银行卡号，用户密码等。

三．其他存储

IE 提供了一种存储可以持久化用户数据，叫做 userData，从 IE5.0 就开始支持。每个数据最多 128K，每个域名下最多 1M。这个持久化数据存放在缓存中，如果缓存没有清理，那么会一直存在。

```
<div style="behavior:url(#default#userData)" id="box"></div>

addEvent(window, 'load', function () {
    var box = document.getElementById('box');
    box.setAttribute('name', encodeURIComponent('李炎恢'));
    box.expires = setCookieDate(7);
    box.save('bookinfo');

    //box.removeAttribute('name');              //删除 userDate
    //box.save('bookinfo');

    box.load('bookinfo');
    alert(decodeURIComponent(box.getAttribute('name')));
});
```

PS：这个数据文件也是保存在 cookie 目录中，只要清除 cookie 即可。如果指定过期日期，则到期后自动删除，如果没有指定就是永久保存。

Web 存储

在比较高版本的浏览器，JavaScript 提供了 sessionStorage 和 globalStorage。在 HTML5 中提供了 localStorage 来取代 globalStorage。而浏览器最低版本为：IE8+、Firefox3.5+、Chrome 4+和 Opera10.5+。

PS：由于这三个对浏览器版本要求较高，我们就只简单的在 Firefox 了解一下，有兴趣的可以通过关键字搜索查询。

```
//通过方法存储和获取
sessionStorage.setItem('name', '李炎恢');
alert(sessionStorage.getItem('name'));

//通过属性存储和获取
sessionStorage.book = '李炎恢';
alert(sessionStorage.book);

//删除存储
sessionStorage.removeItem('name');
```

PS：由于 localStorage 代替了 globalStorage，所以在 Firefox、Opera 和 Chrome 目前的最新版本已不支持。

```
//通过方法存储和获取
localStorage.setItem('name', '李炎恢');
alert(localStorage.getItem('name'));

//通过属性存储和获取
localStorage.book = '李炎恢';
alert(localStorage.book);

//删除存储
localStorage.removeItem('name');
```

PS：这三个对象都是永久保存的，保存在缓存里，只有手工删除或者清理浏览器缓存方可失效。在容量上也有一些限制，主要看浏览器的差异，Firefox3+、IE8+、Opera 为 5M，，Chrome 和 Safari 为 2.5M。

感谢收看本次教程！

本课程是由北风网(ibeifeng.com)

瓢城 Web 俱乐部(yc60.com)联合提供：

本次主讲老师：李炎恢

我的博客：hi.baidu.com/李炎恢/

我的邮件：yc60.com@gmail.com