

千锋HTML5学院

# 第二阶段 Javascript

马军成 18911329780



## 主要内容

什么是**cookie**

**Cookie**的设置与读取

**Cookie**的封装

**Cookie**的应用

## Cookie与存储

cookie 也叫 HTTP Cookie，最初是客户端与服务器端进行会话使用的(会话跟踪)。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。

HTTPCookie 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基础

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。
- 3.Firefox 最多 150 个 cookie
- 4.Safari 和 Chrome 不确定（150左右）。

## Cookie的局限性

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：每个cookie 的最大长度大约为 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

# Cookie的存储位置与查看方法

火狐：选项 -- 隐私 -- 移除单个cookie|| 右键--查看页面信息

Chrome：选项 -- 高级设置 -- 隐私设置 -- 内容设置 -- 所有cookie

...

隐私设置

内容设置...

清除浏览数据...

Google Chrome 浏览器可能会使用网络服务改善您的浏览体验。您可以视情况停用这些服务。

内容设置

Cookie

☒ 允许设置本地数据（推荐）

☐ 仅将本地数据保留到您退出浏览器为止

☐ 阻止网站设置任何数据

☐ 阻止第三方 Cookie 和网站数据

管理例外情况...

所有 Cookie 和网站数据...

IF

Elements Sources Console Network Timeline Profiles Resources

Frames

Web SQL

IndexedDB

Local Storage

Session Storage

Cookies

localhost

Application Cache

Name	Value
cool	1aaaaaaaaaaaaa
	PATH /
	EXPIRES 2017/5/9 22:33:09

文件(F) 查找(N) 禁用(S) 查看(V) 图像(I) 缓存(C) 工具(T) 验证(A) 浏览器模式: IE10(B) 文档模式(M): Quirks

HTML CSS 控制台 脚本 探查器

始终从服务器中刷新(R)

清除浏览器缓存(B)... Ctrl+R

清除此域的浏览器缓存(D)... Ctrl+D

禁用 Cookie(C)

清除会话 Cookie(S)

清除域的 Cookie(O)

查看 Cookie 信息(I)

## Cookie的添加与获取

**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，“键= 值 ”是必选。

```
document.cookie = "username=123";  
//一次写入一个键值对  
document.cookie = "username=123;password=456";  
//一次写入多个，只有第一个生效
```

```
alert(typeof document.cookie);  
alert(document.cookie);
```

## 失效时间

`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。

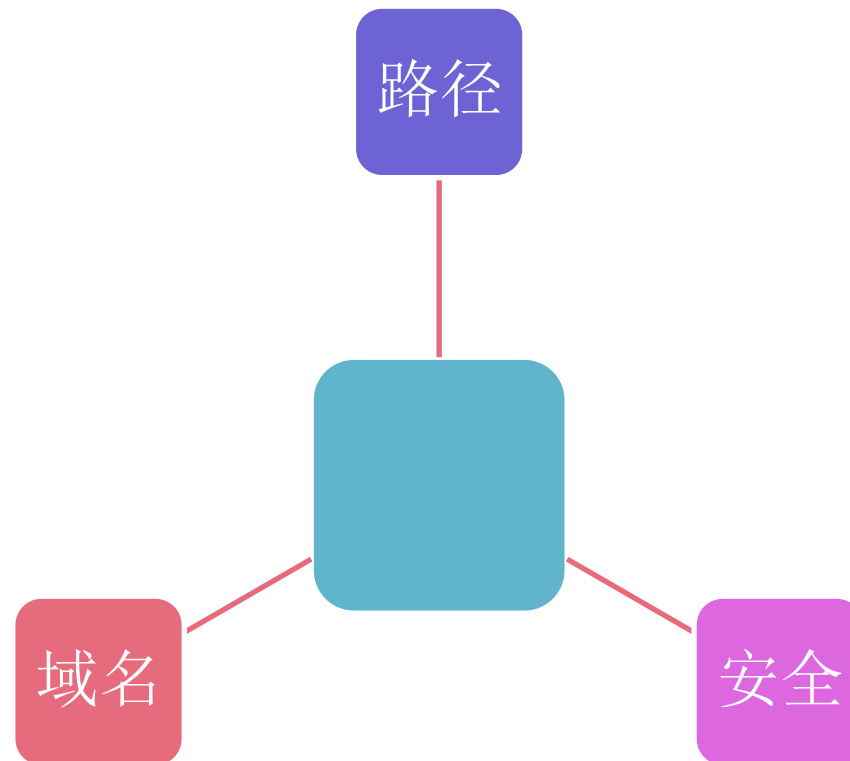
```
var date=newDate(); //创建一个  
date.setDate(date.getDate() +7);  
document.cookie="user= abc;expires=" +date;
```

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 `newDate(0)`。

可以用于删除cookie



# Cookie参数使用



## path访问路径

**cookie** 一般都是由于用户访问页面而被创建的，可是并不是只有在创建 **cookie** 的页面才可以访问这个 **cookie**。

默认情况下，只有与创建 **cookie** 的页面在**同一个目录或子目录**下的网页才可以访问，这个是因为安全方面的考虑，造成不是所有页面都可以随意访问其他页面创建的 **cookie**。

那么如何让这个 **cookie** 能被其他目录或者父级的目录访问类，通过设置 **cookie** 的路径就可以实现

```
document.cookie = "name=value;path=相对路径"  
document.cookie = "name=value;expires=date;path=相对路径"
```

最常用的例子就是让 **cookie** 在根目录下,这样不管是哪个子页面创建的 **cookie**，所有的页面都可以访问到了：

```
document.cookie = "name=Darren;path=/"
```

## domain访问域名

**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。

```
document.cookie = "name=value;path=path;domain=xxx.com"
```

例如 "www.qq.com" 与 "sports.qq.com" 公用一个关联的域名 "qq.com"，我们如果想让 "sports.qq.com" 下的 cookie 被 "www.qq.com" 访问，我们就需要用到 **cookie 的 domain 属性**，并且需要把 **path 属性设置为 "/"**。

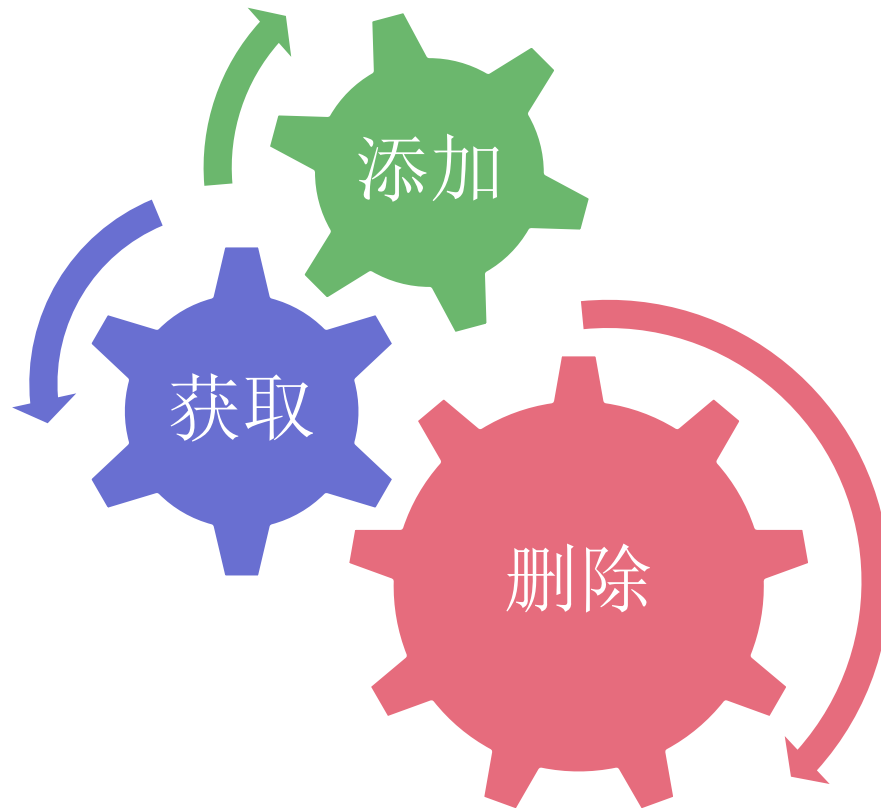
```
document.cookie = "username=Darren;path=/;domain=qq.com"
```

注：

- ✓ 必须在绑定域名的服务器上才可以设置域名
- ✓ 并且只能设置绑定的域名，也就是说，不同服务器间的 cookie 文件不共享

# 封装函数

将生成cookie，获取cookie的步骤方法，封装成函数，以便调用



## 扩展--信息的多项存储

```
document.cookie="id1=10001";  
document.cookie="num1=10";  
document.cookie="id2=10002";  
document.cookie="num2=20";  
alert(document.cookie);  
id1=10001; num1=10; id2=10002; num2=20
```

```
document.cookie=  
"cart=[{id:1001,num:10},{id:1002,num:2}]";  
var list= eval(document.cookie.split("=")[1]);  
alert(document.cookie); //  
cart=[{id:1001,num:10},{id:1002,num:2}];  
alert(list.length); // 2  
alert(list[0].id); // 1001
```

## 作业

- 例子1：在拖拽例子中使用cookie记录上一次Div的位置  
鼠标抬起——记录位置  
window.onload——读取位置
- 例子2：用cookie记录上次登录的用户名  
提交时——记录用户名  
window.onload——读取用户名
- 例子3：使用购物车记录用户的信息

