



做真实的自己，用良心做教育

H5教学部

Cookie



目录

做真实的自己，用良心做教育

1

Cookie会话跟踪技术介绍

2

Cookie的使用

3

Cookie封装

4

JSON

Cookie会话跟踪技术介绍

1, cookie是什么

cookie 也叫 HTTPCookie , 是客户端与服务器端进行会话使用的一个能够在浏览器本地化存储的技术。

2, cookie的作用

cookie的作用主要是在浏览器存储少量数据, 利用cookie我们可以实现一些保存数据的功能.

比如: 1, 用户登录的记住密码功能(下次再访问网站时无需输入密码了);

2, 购物车, 加入购物车的商品没有及时付款, 使用cookie保存后, 可以在一定时间后再访问网站, 会发现购物车里还有之前的商品列表;

Cookie的使用

cookie由键值对形式的文本组成：name=value。

完整格式为：

name=value;[expires=date];[path=路径];[domain=域名];[secure]

其中中括号[]表示该值是可选。

name=value: 为你要保存的键值对(必选)

expires=date: 表示cookie的失效时间, 默认是浏览器关闭时失效(可选)

path=路径: 访问路径, 默认为当前文件所在目录(可选)

domain=域名: 访问域名, 限制在该域名下访问(可选)

secure: 安全设置, 如果设置了则必须使用https协议才可获取cookie(可选)

Cookie的使用

1, 获取和设置cookie

使用document对象来获取和设置cookie:

//设置cookie和获取cookie

```
document.cookie = "user="+ "张三"; //设置
```

```
console.log(document.cookie); //获取
```

//URI编码后设置cookie, 和URI解码后获取cookie

```
document.cookie = "user2=" + encodeURIComponent("张三");
```

```
console.log(decodeURIComponent(document.cookie));
```

Cookie的使用

2, expires=失效时间

失效时间: 表示cookie会在该时间被删除掉, 默认是浏览器关闭的时候;

可以自己设置cookie的失效时间。 如设置7天后再删除cookie

```
var date = new Date();  
date.setDate(date.getDate() + 7);  
document.cookie = "user=张三;expires=" + date;
```

主动删除cookie

失效时间设置在现在时间或现在之前的时间即可删除指定cookie

name值为指定要删除的那个cookie

```
var date = new Date();  
document.cookie = "user=张三;expires=" + date;
```

Cookie的使用

3, path=路径

设置路径后, 则只有设置的那个路径文件才可以访问cookie, 默认为当前文件所在目录
一般设置path=/, 表示磁盘(域名)根目录, 则其他路径也可以获取到该cookie值

```
document.cookie = "user=abc;expires=" + date + ";path=/";
```

注意: 在设置路径path时, 要记得设置失效时间expires

4, domain=域名

用于限制只有设置的域名才可以访问, 没有设置则默认为当前域名。

```
document.cookie = "user=张三;domain=www.baidu.com" ;
```

5, 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=张三;secure" ;
```


Cookie的封装

将cookie相关的操作封装在函数中, 方便以后调用, 一般我们只要设置name, value, expires, path即可;

//设置cookie

```
function setCookie(name, value, expires, path) {  
    var cookieText = encodeURIComponent(name) + "=" + encodeURIComponent(value);  
    if (expires instanceof Date) {  
        cookieText += ";expires=" + expires;  
    }  
    if (path) {  
        cookieText += ";path=" + path;  
    }  
    document.cookie = cookieText;  
    return decodeURIComponent(document.cookie);  
}
```


Cookie的封装

//获取cookie

```
function getCookie(name) {  
    var cookie = decodeURIComponent(document.cookie);  
    var arr = cookie.split( ";" );  
    for (var i=0; i<arr.length; i++) {  
        var arr2 = arr[i].split( "=" );  
        if (arr2.length >= 2) {  
            if (arr2[0] == name) {  
                return arr2[1];  
            }  
        }  
    }  
    return "" ;  
}
```

//删除cookie

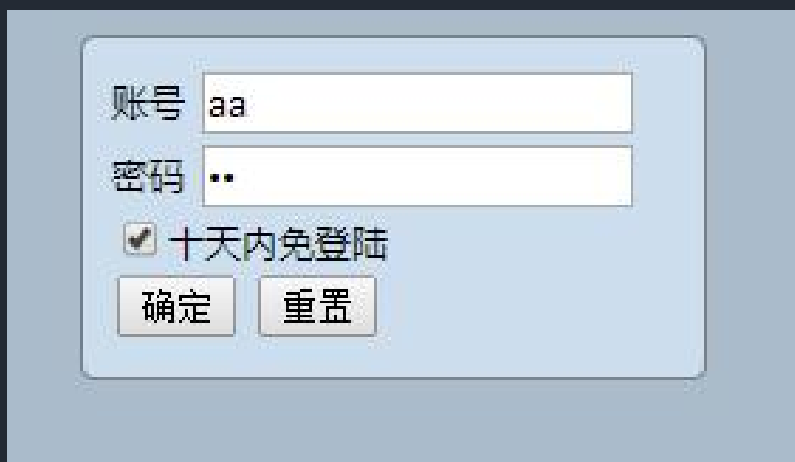
```
function removeCookie(name) {  
    document.cookie=encodeURIComponent(name)+ "=; expires=" + new Date();  
}
```

Cookie的封装

练习

1, 十天内免登录:

如下图, 有账户,密码输入框, 当用户点击确定按钮, 如果选择了十天内免登陆, 则将账户和密码保存在cookie中. 当下一次再进入该页面时, 自动填写上次保存的账户和密码



账号 aa

密码 ..

☒ 免登陆

确定 重置

JSON

1, JSON

JSON 是一种结构化的数据表示方式(数据格式). 通过JavaScript中的一些模式来表示结构化数据, JSON 并不是 JavaScript 独有的数据格式, 其他很多语言都可以对 JSON 进行解析和序列化, 除了JSON外, 还有XML也是一种数据表示方式; 目前很少使用XML的数据格式, 在这里我们只介绍JSON.

JSON

2, JSON的写法

对象的表现形式: 包括数组和对象

字符串的表现形式: 包括数组和对象的字符串

对象的写法: 使用双引号

```
{ "name" : "Zhang" , "age" :100}
```

数组的写法:

```
[ "张三" , "李四" , "王五" ]
```

对象和数组结合的写法:

```
[
```

```
  { "name" : "张三" , "age" : 33 },
```

```
  { "name" : "李四" , "age" : 44 }
```

```
]
```

JSON

3, JSON的两种表现形式

JSON对象:

```
var jsonObj = [  
    { "name" : "张三" , age" : 33 },  
    { "name" : "李四" , "age" : 44 }  
]
```

JSON字符串:

```
var jsonStr = '[{ "name" : "张三" , "age" : 33}, { "name" : "李四" , "age" : 44}]' ;
```

(注意单双引号, 将单引号写在外面, 里面用双引号)

JSON

4, JSON解析

JSON解析就是将JSON字符串变成JSON对象(对象或数组)的过程;

`eval()`: 但这个方法可能会造成安全问题

`JSON.parse()`: 常用该方法进行解析. (兼容IE8+)

例如:

```
var jsonStr = '[{"name": "a", "age" : 1}, {"name" : "b", "age" : 2}]';  
var jsonObj = JSON.parse(jsonStr);  
console.log(jsonStr);  
console.log(jsonObj);
```

JSON

5, JSON序列化

JSON序列化就是将JSON对象(对象或数组)变成JSON字符串的过程; 和JSON解析相反;

`JSON.stringify()`: JSON序列化

例如:

```
var jsonObj = [{name : 'a', age : 1},{name : 'b', age : 2}];
```

```
var jsonStr = JSON.stringify(jsonObj);
```

```
console.log(jsonStr);
```

```
console.log(jsonStr);
```


Cookie实现购物车

练习

1, 使用cookie实现购物车功能:

有以下商品, 点击加入购物车即可加入到购物车中(cookie中保存),
点击查看购物车, 进入另一个页面, 并显示之前加入购物车的商品.

商品列表

- id1 单车 \$100 [加入购物车](#)
- id2 摩托 \$1000 [加入购物车](#)
- id3 汽车 \$10000 [加入购物车](#)
- id4 坦克 \$100000 [加入购物车](#)
- id5 飞机 \$1000000 [加入购物车](#)
- id6 飞船 \$10000000 [加入购物车](#)

[查看购物车](#)

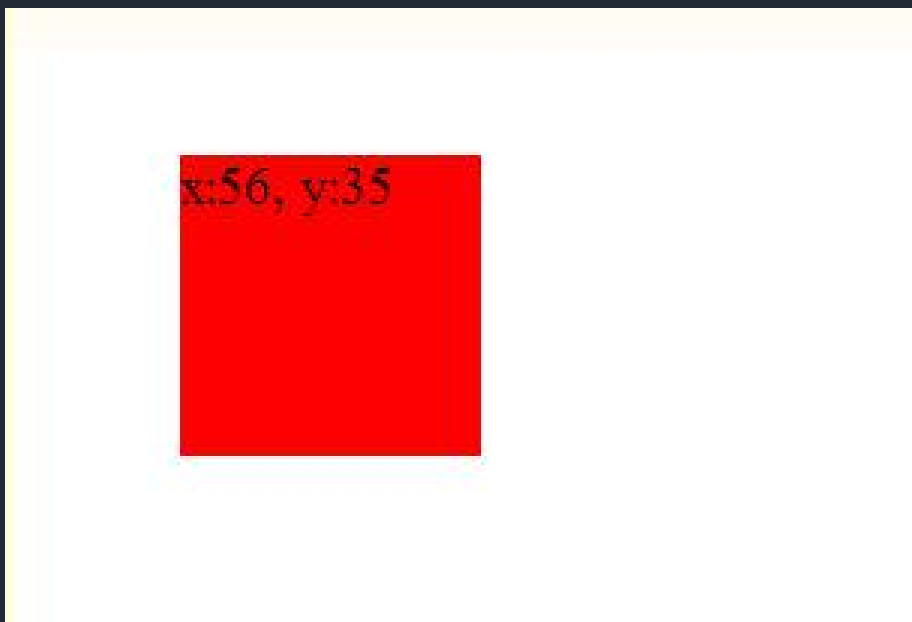
我的购物车

- 商品名称：摩托，商品价格：\$1000, 商品数量：1
- 商品名称：坦克，商品价格：\$100000, 商品数量：1
- 商品名称：飞机，商品价格：\$1000000, 商品数量：1

练习

1, 记录图片位置:

页面上有一个红色div, 可以对该div实现拖拽, 每次拖拽后需要保存位置
下次重新进入该页面时, 该红色div还是在最后一次拖拽松开时的位置



THANK YOU



做真实的自己，用良心做教育