

js-cookie详解

什么是 Cookie?

Cookie 是一些数据, 存储于你电脑上的文本文件中。

当 web 服务器向浏览器发送 web 页面时, 在连接关闭后, 服务端不会记录用户的信息。

Cookie 的作用就是用于解决 "如何记录客户端的用户信息":

- 当用户访问 web 页面时, 他的名字可以记录在 cookie 中。
- 在用户下一次访问该页面时, 可以在 cookie 中读取用户访问记录。

基础篇

```
<script>
    // 创建 cookie
    document.cookie = 'username=John Doe'
    // 还可以为 cookie 添加一个过期时间 (以 UTC 或 GMT 时间)。默认情况下, cookie 在浏览器关闭时删除
    document.cookie = 'username=John Doe; expires=Thu, 18 Dec 2043 12:00:00 GMT'
    // 可以使用 path 参数告诉浏览器 cookie 的路径。默认情况下, cookie 属于当前页面。
    document.cookie = 'username=John Doe; expires=Thu, 18 Dec 2043 12:00:00 GMT; path=/'
    // 查看cookie
    document.cookie;
    // 修改cookie类似于创建cookie

    // 删除cookie
    document.cookie = "username=; expires=Thu, 01 Jan 1970 00:00:00 GMT";
</script>
```

进阶篇

1、设置 cookie 值的函数:

```
/**
 *cname cookie中的key
 *cvalue cookie中的value
```

```

    *expiresDay cookie中的key和value过期时长
    */
function setCookie(cname, cvalue, exdays) {
    // 获取当前的系统时间
    let d = new Date()
    // console.log(d.getDate())
    d.setTime(d.getTime() + exdays * 24 * 60 * 60 *
1000)
    // console.log(exdays)
    // console.log(d.getDate())
    let expires = 'expires=' + d.toGMTString()
    //expires:格林威治时间
    document.cookie = cname + '=' + cvalue + '; ' +
expires
}

```

以上的函数参数中，cookie 的名称为 cname，cookie 的值为 cvalue，并设置了 cookie 的过期时间 expires。该函数设置了 cookie 名、cookie 值、cookie 过期时间。

2、获取 cookie 值的函数：

```

function getCookie(cname) {
    let name = cname + '='
    // console.log(name)
    // split() 方法用于把一个字符串分割成字符串数组。
    let ca = document.cookie.split(';')
    // console.log(ca)
    for (var i = 0; i < ca.length; i++) {
        // trim()去除字符串的头尾空格
        let c = ca[i].trim()
        // console.log(c)
        // indexOf() 方法可返回某个指定的字符串值在字符串中首次
出现的位置。
        if (c.indexOf(name) == 0) {
            // substring() 方法用于提取字符串中介于两个指定下标
之间的字符。
            return c.substring(name.length, c.length)
        }
    }
    return ''
}

```

cookie 名的参数为 cname。

创建一个文本变量用于检索指定 cookie :cname + "="。

使用分号来分割 document.cookie 字符串，并将分割后的字符串数组赋值给 ca (ca = document.cookie.split(';'))。

循环 ca 数组 (i=0;i<ca.length;i++)，然后读取数组中的每个值，并去除前后空格 (c=ca[i].trim())。

如果找到 cookie(c.indexOf(name) == 0)，返回 cookie 的值 (c.substring(name.length,c.length))。

如果没有找到 cookie, 返回 ""。

3、检测 cookie 值的函数

```
function checkCookie() {  
    let user = getCookie('username')  
    if (user != '') {  
        alert('欢迎 ' + user + ' 再次访问')  
    } else {  
        // 提示框经常用于提示用户在进入页面前输入某个值。  
        user = prompt('请输入你的名字:', '')  
        if (user != '' && user != null) {  
            setCookie('username', user, 30)  
        }  
    }  
}
```

最后，我们可以创建一个检测 cookie 是否创建的函数。

如果设置了 cookie，将显示一个问候信息。

如果没有设置 cookie，将会显示一个弹窗用于询问访问者的名字，并调用 setCookie 函数将访问者的名字存储 30天