

数据存储技术

数据存储技术(Cookie)简单介绍

web存储相关的技术：[cookie](#) | [sessionStorage](#) | [localStorage](#) | [应用缓存](#)

cookie是一种会话跟踪技术，用于在进行网页访问的时候，存储页面中的某些数据信息。

cookie的使用注意点

HTML

001 使用cookie来进行数据存储的大小有限制，4KB

002 每个网页中存储cookie的个数（最多50）和每个网站中存储cookie的个数都有限制（200）

003 各个不同的浏览器对cookie的数量也不相同

IE6.0：每个域为20个，大小为4095个字节

IE7.0/8.0：每个域为50个，大小为4095个字节

Opera：每个域为30个，4096个字节

FF：每个域为50个，大小为4097个字节

Safari：没有个数限制，大小为4097个字节

Chrome：每个域为53个，大小为4097个字节

总结：在进行页面cookie操作的时候，应该尽量保证cookie个数小于20个，总大小 小于4KB

004 cookie数据的过期时间

1) 默认情况下，cookie存储的内容是一次性的，它的有效期间是当前会话（需要把整个浏览器都关闭会话就结束）

2) 设置过期时间：通过`expires=time;`的格式来进行指定设置之后没超过过期时间则数据一直在。

005 cookie是不可以跨浏览器的(在IE中保存的cookie，不可以在火狐中使用)

006 cookie是不可以跨域的(跨域名)

示例：127.0.0.1/code/test.html 和127.0.0.1/code/test1.html可以访问

网络请求的常见路径：

`http://www.baidu.com:80/资源`

协议://域名.后缀:端口号/资源

不能跨域的含义：就是只有（协议+域名+后缀+端口）号都相同才能相互访问

`http://www.baidu.com:80/a.html`

`http://www.baidu.com:80/b.html` 能

`http://www.baidu.com:81/a.html`

`http://www.baidu.com:80/b.html` 不能（端口号不相同）

`http://mp3.baidu.com:80/a.html`

`http://map.baidu.com:80/b.html` 不能（二级域名不相同）

`https://www.baidu.com:80/a.html`

`http://www.baidu.com:80/b.html` 不能（协议不相同）

`http://www.bdu.com:80/a.html`

`http://www.dbu.com:80/b.html` 不能（以及域名不相同）

Cookie的设置和使用

设置方法

设置cookie数据: `document.cookie = "name=zhangsan";`

设置cookie数据和过期的时间: `document.cookie = "name=zhangsan; expires="+date+"";`

过期7天的设置: `var date = new Date(); // date.setDate(date.getDate() + 7);`

使用注意

在设置cookie的时候, 一次只能设置一个数据 (一个键值对), 不能进行批量设置

错误的演示: `window.cookie = "name=zhangsan&age=18";`

```
//设置三个cookie的值, 默认的过期时间是session (会话内)
document.cookie = "name=wendingding";
document.cookie = "age=18";
document.cookie = "des=12345";

//设置cookie的值, 并指定过期的时间
var date = new Date();
date.setDate(date.getDate() + 3); //3天的过期时间

//注意: 在设置过期时间的时候, 中间使用; 分隔开发
document.cookie = "color=red;expires="+date;

//删除cookie
//删除cookie的原理: 只要过期时间超过了当前的时间, 数据就会被删除
date.setDate(date.getDate() + -1);
document.cookie = "age=18; expires="+date+"";
```

Cookie操作的封装

```
let Cookie = {
  getItem(name) {
    let cookies = document.cookie.split("; ");
    for (let i = 0; i < cookies.length; i++) {
      let arr = cookies[i].split("=");
      if (name == arr[0]) return arr[1];
    }
  },
  setItem(name, value, days) {
    if (days) {
      var date = new Date();
      date.setDate(date.getDate() + days);
      document.cookie = `${name}=${value};expires=${date}`;
    } else {
      document.cookie = `${name}=${value}`;
    }
  },
  removeItem(name) {
```

```

        this.setItem(name, null, -1);
    },
    hasItem(name) {
        let keys = this.keys();
        let flas = false;
        if (keys) {
            flas = keys.includes(name);
        }
        return flas;
    },
    clear() {
        let keys = this.keys();
        keys.forEach(ele => {
            this.removeItem(ele);
        })
    },
    keys() {
        let result = [];
        let cookies = document.cookie.split("; ");
        cookies.forEach(ele => {
            let arr = ele.split("=");
            result.push(arr[0]);
        });
        return result;
    }
}

```

sessionStorage技术简单介绍

- (1) 在存储用户数据的时候，因为cookie可以存储的数据比较小才4KB左右，所以也可以考虑使用 **sessionStorage** 来存储，大小限制为2M左右
- (2) 该技术和localStorage是h5推出的，因此在使用的时候，需要考虑到兼容性的问题
- (3) 相关资料：http://www.w3school.com.cn/html5/html_5_webstorage.asp
- (4) 简单介绍和使用：

001 sessionStorage: 会话存储技术 (session-会话)
 002 该技术和cookie一样，一样浏览器退出了，那么保存的数据就会
 003 sessionStorage技术的相关操作：添加|获取|更新|删除|清空

```

<input type="text">
<button class="btn1">添加</button>
<button class="btn2">获取</button>
<button class="btn3">更新</button>
<button class="btn4">删除</button>
<button class="btn5">清空</button>

<script>
    window.sessionStorage.setItem("age", "18");
    window.sessionStorage.setItem("color", "red");
    window.sessionStorage.setItem("des", "no des");
    var oInput = document.querySelector("input");
    var oBtn1 = document.querySelector(".btn1");

```

HTML

```

oBtn1.onclick = function () {
    //添加操作
    window.sessionStorage.setItem("name",oInput.value);
};
var oBtn2 = document.querySelector(".btn2");
oBtn2.onclick = function () {
    //获取操作
    console.log(window.sessionStorage.getItem("name"));
};

var oBtn3 = document.querySelector(".btn3");
oBtn3.onclick = function () {
    //更新操作
    window.sessionStorage.setItem("name",oInput.value);
};
var oBtn4 = document.querySelector(".btn4");
oBtn4.onclick = function () {
    //删除操作
    window.sessionStorage.removeItem("name");
};
var oBtn5 = document.querySelector(".btn5");
oBtn5.onclick = function () {
    //清空操作
    window.sessionStorage.clear();
};

</script>

```

localStorage技术简单介绍

- (1) 简单对比：保存在本地 + 没有时间限制
- (2) 大小比较：4K(cookie) - 5M(sessionStorage) - 20M(localStorage)
- (3) 使用方式和sessionStorage相同

HTML

```

<input type="text">
<button class="btn1">添加</button>
<button class="btn2">获取</button>
<button class="btn3">更新</button>
<button class="btn4">删除</button>
<button class="btn5">清空</button>

```

```

<script>
    window.localStorage.setItem("age", "18");
    window.localStorage.setItem("color", "red");
    window.localStorage.setItem("des", "no des");
    var oInput = document.querySelector("input");
    var oBtn1 = document.querySelector(".btn1");

    oBtn1.onclick = function () {
        //添加操作
        window.localStorage.setItem("name",oInput.value);
    };
    var oBtn2 = document.querySelector(".btn2");

```

```
oBtn2.onclick = function () {  
    //获取操作  
    console.log(window.localStorage.getItem("name"));  
};  
  
var oBtn3 = document.querySelector(".btn3");  
oBtn3.onclick = function () {  
    //更新操作  
    window.localStorage.setItem("name",oInput.value);  
};  
var oBtn4 = document.querySelector(".btn4");  
oBtn4.onclick = function () {  
    //删除操作  
    window.localStorage.removeItem("name");  
};  
var oBtn5 = document.querySelector(".btn5");  
oBtn5.onclick = function () {  
    //清空操作  
    window.localStorage.clear();  
};  
  
</script>
```

应用程序缓存简单介绍

简单介绍：h5推出的应用程序缓存，可以对Web页面进行缓存，并且可以在没有网络的情况下进行访问，所有主流浏览器均支持应用程序缓存，除了 Internet Explorer，主要优点

- 1) 支持离线浏览（没有网络的情况下依然可以查看）
- 2) 速度更快，性能更好（使用缓存文件可以提高访问的速度）
- 3) 减轻服务器端的压力 - 浏览器将只从服务器下载更新过或更改过的资源

代码和配置文件演示

- 1) 在页面中加载资源文件（图片），调试工具窗口选择（none network）表示网络无法连接
- 2) 新创建xx.appcache文件，在该文件配置要缓存的内容
- 3) 设置html页面中的manifest属性为配置文件路径

CACHE MANIFEST

CACHE:

需要缓存的列表

0.jpg

NETWORK:

不需要缓存的

1.jpg

FALLBACK:

访问缓存失败后，备用访问的资源