Window

window对象，表示浏览器的一个实例。
所有全局JS对象，函数和变量自动成为window对象的成员。全局变量是window对象的属性，全局函数是window 对象的方法，甚至DOM中的document对象也是window对象属性。window对象的属性和方法可以省略window而直接使用。
全局变量不能通过 **delete** 运算符删除，而直接在 window 对象上的定义的属性可以。*//使用 var 语句添加的 window 属性有一个名为 Configurable 的特性，这个特性的值被默认设置为 false，因此这样定义的属性不可以通过 delete 运算符删除。*
window.fetch():用于发起获取资源的请求，返回一个promise
window.postMessage(message,targetOrigin[,transfer]) 可以安全地实现跨域通信
window.**open**() - 打开新窗口
window.close() - 关闭当前窗口
1.在当前窗口打开，可后退： .**open**("url","_self")
2.在新窗口打开，可打开多个： .**open**("url","_blank")
3.在新窗口打开，只能打开一个： .**open**("url","自定义窗口名")

Screen

window.**screen**对象包含用户屏幕的信息
screen.width/**screen**.height:以像素计的访问者屏幕的宽/高
screen.availWidth:访问者屏幕的宽度，以像素计，减去诸如窗口工具条之类的界面特征。
screen.availHeight

Location

window.**location**: 保存当前窗口正在打开的url信息的对象
【属性】:
window.**location**.href 返回当前页面的完整url地址
window.**location**.host 返回 主机名 + 端口号
window.**location**.hostname 返回 主机名
window.**location**.port 返回 端口号
window.**location**.pathname 返回当前页面的相对路径
window.**location**.protocol 返回使用的 web 协议（http: / https:）
window.**location**.hash 返回当前页面哈希值(#锚点地址)
window.**location**.search 返回当前页面查询字符串(?及之后的字符串)
【方法】:
1. 在当前窗口打开，可后退:**location**.assign(url) => **location**.href=url => **location**=url
在当前窗口打开，禁止后退:**location**.replace(url)
3. 重新加载页面: 刷新: 2种:
    1. 普通刷新: 优先从浏览器本地缓冲获取资源:
        F5
        history.go(0)
        location.reload(false)
    2. 强制刷新: 无论本地是否有缓存，总是强制从服务器获取资源
        location.reload(true)

History

window.**history** 保存当前窗口打开后，成功访问过的url的历史记录栈。在当前窗口中，每访问一个新url，都会将新url压入。
history.back() - 等同于在浏览器点击后退按钮，加载历史列表中前一个 URL
history.forward() - 等同于在浏览器中点击前进按钮
go() 方法可以在用户的历史记录中任意跳转。这个方法接受一个参数，表示向后或向前跳转的页面数的一个整数值。
前进：**history.go**(1)
后退：**history.go**(-1)
刷新：**history.go**(0)

Navigator

【window.navigator 保存浏览器配置信息的对象】
navigator.cookieEnabled：判断当前浏览器是否启用cookie
navigator.appName:"Netscape",浏览器的应用程序名称，它是IE11、Chrome、Firefox 以及 Safari 的应用程序名称的统称。
navigator.appVersion 返回有关浏览器的版本信息：
navigator.appCodeName: "Mozilla"，它是 Chrome、Firefox、IE、Safari 以及 Opera 的应用程序代码名称。
navigator.userAgent 保存浏览器名称和版本号的字符串(判断浏览器名称和版本号 时使用)
navigator.platform 返回浏览器平台(操作系统)
navigator.language 属性返回浏览器语言：
【警告】：来自 navigator 对象的信息通常是误导性的，不应该用于检测浏览器版本

Cookie

Cookie 在客户端持久存储用户私密数据的小文件。默认情况下，在浏览器关闭时会删除 cookie。
内存中所有数据都是临时的。程序关闭，内存中一切变量都释放。只要希望在客户端持久保存数据，就可以使用cookie。它是为了解决“如何记住用户信息”而发明的，当用户访问网页时，他的名字可以存储在 cookie 中。下次用户访问该页面时，cookie 会“记住”他的名字。
当浏览器从服务器请求一个网页时，会将属于该页的 cookie 添加到该请求中。这样服务器就获得了必要的数据来“记住”用户的信息。
【创建 cookie】

```
document.cookie = "username=Bill Gates";//创建 cookie:
document.cookie = "username=John Doe; expires=Sun, 31 Dec 2017 12:00:00 UTC";//添加有效日期（UTC 时间）
document.cookie = "username=Bill Gates; expires=Sun, 31 Dec 2017 12:00:00 UTC; path=/";//通过 path 参数，您可以告诉浏览器 cookie 属于什么路径。默认情况下，cookie 属于当前页。
```

【读取 cookie】

```
let x = document.cookie;//它会在一条字符串中返回所有 cookie，比如：'cookie1=value; cookie2=value; cookie3=value;'
```

【修改 cookie】
可以像创建 cookie 一样改变它，在cookie键名相同的情况下，旧 cookie 将被覆盖。如果设置了新cookie，则旧的 cookie 不会被覆盖，新的 Cookie 会被添加到 document.cookie。
【删除 cookie】
删除 cookie 时不必指定 cookie 值，直接把 expires 参数设置为过去的日期即可。

```
document.cookie = "username=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/;";//删除cookie时应该定义 cookie 路径以确保删除正确的 cookie。如果不指定路径，一些浏览器不允许删除 cookie。
```