

Browser Object Model				
知识点	涉及语法		功能	备注
BOM	Browser Object Model，浏览器对象模型，提供了独立于内容而与浏览器窗口进行交互的对象，核心对象是window。BOM缺乏标准，最初是Netscape浏览器标准的一部分。			
window对象	window对象是浏览器的顶级对象，具有双重角色： 1.它是JS访问浏览器窗口的一个接口 2.window对象是一个全局对象，定义在全局作用域中的变量、函数都会变成window对象的属性和方法。在调用的时候可以省略window，如alert(),prompt()等。			
window对象 常见事件	'load'	页面加载事件，当文档内容完全加载完成（包括图像、脚本文件、CSS文件等），才会触发该事件	注意： 1. 有了window.onload 就可把JS代码写到页面元素的上方，因为onload是等页面完全加载完毕，再去执行函数。 2. window.onload 传统注册事件方式只能写一次，多个的情况下，会以最后一个window.onload为准。使用addEventListener则没有次数限制。	
	'DOMContentLoaded'	仅当DOM结构加载完成，不包括样式图片等，即可触发事件	[1e9以上才支持] 如果页面的图片很多的话,从用户访问到 onload触发可能需要较长的时间,交互效果就不能实现,必然影响用户的体验,此时用 DOMContentLoaded事件比较合适。	
	'resize'	调整窗口大小事件	可配合window.innerWidth(当前视口宽度) 用于完成响应式布局	
window对象 常用函数	setTimeout(fn,ms)	倒计时定时器	clearTimeout(fn)	
	setInterval(fn,ms)	重复调用定时器	clearInterval(fn)	
location对象	window对象给我们提供了一个 location属性，用于获取或设置窗体的URL,并且可以用于解析URL。因为这个属性返回的是一个对象,所以我们将这个属性也称为 "location对象"。			
URL (Uniform Resource Locator) 统一资源定位符	URL的一般语法格式为 protocol://host[:port]/path/[?query] #fragment	protocol	通信协议：常用的http,ftp,maito等	
		host	主机(域名)	
		port	端口号：可选,省略时使用方案的默认端口,如http的默认端口为80	
		path	路径：由零或多个/符号隔开的字符串,一般用来表示主机上的一个目录或文件地址	
		query	参数：以键值对的形式通过&符号分隔开来	
		fragment	片段： #后面内容 常见于链接 锚点	
		location对象 属性	location.href	获取或重定义url
location.host	返回主机（域名）			
location.port	返回端口号			
location.pathname	返回路径			
location.search	返回参数		可用于跳转后显示欢迎词	
location.hash	返回#后面的内容			
location对象 常见方法	location.assign(url)	和href一样，可以跳转页面，可以后退		
	location.replace(url)	地址替换，不记录浏览历史，不可后退		
	location.reload()	重新加载页面，F5	参数为true时强制刷新，等于ctrl+F5(清除缓存)	
navigator对象	navigator对象包含有关浏览器的信息,它有很多属性,我们最常用的是 userAgent，该属性可以返回由客户机发送服务器的user-agent头部的值。	下面前端代码可以判断用户在哪一个终端打开页面,实现跳转： if((navigator.userAgent.match(/(phone pad pod iPhone iPod ios iPad Android Mobile BlackBerry IEMobile MQQBrowser JUC Fennec WOSBrowser BrowserNG WebOS Symbian Windows Phone)/i))) { window.location.href = "" ; //手机 } else { window.location.href = "" ; //电脑 }		
history对象	history对象一般在实际开发中比较少用，但是会在一些OA办公系统Office Automation中见到。	history.forward()	前进	
		history.back()	后退	
		history.go(1)	1: 前进 -1: 后退	
		history.go(-1)		