# 一、BOM对象

BOM 对象实际上指的是window对象，所有的浏览器都是支持BOM对象，可以通过js的BOM对象实现对浏览器之间的对话。简单的说，可以通过js来操作浏览器窗口。

## 1、实际的应用

① 通过BOM 对象获取浏览器窗口的 尺寸

② 通过BOM 对象实现访问资源，浏览历史记录（前进、后退）

③ 通过BOM 对象操作窗口提示框（提示框、输入框、确定和取消按钮的确认框等）

④ 通过BOM 对象实现定时操作

## 2、window对象

### ① 获取窗体的宽度和高度

|  |
| --- |
| <**script**>  **var *height*** = ***window***.innerHeight;  **var *width*** = ***window***.innerWidth;  *alert*(**"宽度："**+***width***+**",高度："**+***height***); </**script**> |

### ② 打开新窗口：open() 方法

|  |
| --- |
| <**script**>  */\*var height = window.innerHeight;  var width = window.innerWidth;  alert("宽度："+width+",高度："+height);\*/  /\*定义函数名字时，要避免出现 DOM 或 BOM 中已经出现的函数名或属性名\*/* **function** *openWindow*(){  */\*  open 方法，有以下常见的参数设置  ① url：新打开资源的地址  ② name：被打开的新资源打开方式， 当前窗口(\_self)、新窗口(\_blank 默认值)、或指定框架中打开  \*/* ***window***.open(**"https://www.baidu.com"**,**"\_blank"**,**"scrollbars=0,width=600px,height=400px,"**);  }  </**script**> </**head**> <**body**>  <**button onclick="***openWindow*()**"**>打开新窗口</**button**> </**body**> |

### ③ 关闭window对象 可以使用close()方法

|  |
| --- |
| <**script**>  **var *newWindow*** ; *// 新的窗口对象* **function** *openWindow*(){  ***newWindow*** = ***window***.open(**"https://www.baidu.com"**,**""**  **width=600px,height=400px,"**);  }  **function** *closeWindow*() {  */ \*关闭新的窗口对象\*/* ***newWindow***.close();  }    </**script**> </**head**> <**body**>  <**button onclick="***openWindow*()**"**>打开新窗口</**button**>  <**button onclick="***closeWindow*()**"**>关闭新窗口</**button**> </**body**> |

### ④ 定时器对象

1. setInterval(函数,毫秒数) 按指定毫秒数 反复的执行对应的函数脚本
2. clearInterval() 清除定时对象
3. setTimeout(函数,毫秒数) 规定在指定毫秒数后执行对应的函数脚本
4. clearTimeout() 清除定时对象

|  |
| --- |
| <**style**>  **#warp**{  **display**: **none**;  **position**: **relative**;  **top**: 160**px**;  **left**: 520**px**;  **width**: 300**px**;  **height**: 120**px**;  **border**: **#BDBDBD**;  **box-shadow**: 0**px** 0**px** 2**px #373B3C**;  **border-radius**: 5**px**;  **padding**: 10**px**;  }  **h3**{  **margin**: 0**px**;  **font-size**: 15**px**;  **color**: **orangered**;  **text-align**: **center**;  }  .**con**{  **text-align**: **center**;  **padding**: 15**px**;  **margin-top**: 15**px**;  }  </**style**>  <**script**>  **function** *show*() {  **var** num = ***document***.getElementById(**"num"**);  **var** warp = ***document***.getElementById(**"warp"**);  warp.style.display = **"block"**;  */\* var interval;  var index = 7;  interval = setInterval(function () {  num.innerHTML = ""+index;  if(index==0){  warp.style.display = "none";  clearInterval(interval);  }  index--;  },1000)\*/  /\*需要在3秒后将提示框隐藏\*/  setTimeout*(**function** () {  warp.style.display = **"none"**;  },3000)  }   </**script**> </**head**> <**body**>  <**button onclick="***show*()**"**>显示提示框</**button**>  <**div id="warp"**>  <**h3**>提示框</**h3**>  <**div class="con"**>下午好，<**span id="num"**>3</**span**>秒后弹窗关闭</**div**>  </**div**> </**body**> |

## 3、navigator对象

用于获取浏览器相关的信息

|  |
| --- |
| <**script**>  **function** *browserInfo*(){  */\*获取到navigator 对象，用户得到浏览器相关信息\*/* **var** info = ***window***.navigator;   *//① appCodeName：浏览器代码名* **var** appCodeName = info.appCodeName;  *alert*(appCodeName);   */\*② 返回浏览器的名称\*/* **var** APPName = info.appName;  *alert*(APPName);   */\*③ 浏览器的平台和版本信息\*/* **var** appVersion = info.appVersion;  *alert*(appVersion);   */\*④ 客户端发送到服务器的头部信息\*/* **var** userAgent = info.userAgent;  *alert*(userAgent);  }  </**script**> </**head**> <**body**>  <**button onclick="***browserInfo*()**"**>浏览器信息</**button**> </**body**> |

## 4、history对象

用于实现浏览器前进、后退以及访问历史页面的功能

|  |
| --- |
| <**script**>  **function** *backUrl*(){  ***window***.history.back(); *//后退一页* }  **function** *forwordUrl*() {  ***window***.history.forward(); *//前进一页* }  **function** *lookHistory*() {  **var** id = *prompt*(**"历史记录共有"**+***window***.history.length+**"个，请输入访问的历史页面的标识："**);  ***window***.history.go(id);  }  </**script**> </**head**> <**body**>  <**button onclick="***backUrl*()**"**>后退</**button**>  <**button onclick="***forwordUrl*()**"**>前进</**button**>  <**button onclick="***lookHistory*()**"**>访问历史</**button**> </**body**> |

## 5、location对象

用于操作窗口加载的资源信息，包括URL地址、主机名、端口号、资源地址、协议等

|  |
| --- |
| <**script**>  */\*  定义接口方法时 命名规范  ① 添加 前缀 addXxx insertXxx  ② 删除 前缀 removeXxx deleteXxx  ③ 更新 前缀 updateXxx  ④ 查询 前缀 查询一条 getXxx  查询多条 listXxx  统计数量 count  \*/* **function** *getUrl*(){  **var** url = ***window***.location.href; *//获取当前窗口加载的资源全地址  alert*(url);  }  **function** *getHost*(){  **var** host = ***window***.location.host; *//获取当前窗口加载的地址的主机名和端口  alert*(host);  }  **function** *getHostName*() {  **var** hostName = ***window***.location.hostname;*//获取当前窗口加载的地址的主机名  alert*(hostName);  }  **function** *getPort*() {  **var** port = ***window***.location.port;*//获取当前资源的端口号  alert*(port);  }  **function** *getPathName*() {  **var** pathName= ***window***.location.pathname;*//访问 具体的资源地址，主机名后面的部分  alert*(pathName);  }  **function** *getProtocol*() {  **var** protocol = ***window***.location.protocol;*//获取协议  alert*(protocol);  }  **function** *updateHref*() {  ***window***.location.href = **"https://www.baidu.com"**;  }  </**script**> </**head**> <**body**>  <**button onclick="***getUrl*()**"**>获取URL地址</**button**>  <**button onclick="***getHost*()**"**>获取URL中的主机名和端口号</**button**>  <**button onclick="***getHostName*()**"**>获取URL中的主机名</**button**>  <**button onclick="***getPort*()**"**>获取端口号</**button**>  <**button onclick="***getPathName*()**"**>获取资源地址</**button**>  <**button onclick="***getProtocol*()**"**>获取协议</**button**>  <**button onclick="***updateHref*()**"**>重新设置当前窗口加载的资源地址</**button**> </**body**> |