

JavaScript BOM

学习目标

- 1、掌握什么是BOM
- 2、掌握BOM的核心-window对象
- 3、掌握window对象的控制、弹出窗口方法

什么是bom

BOM (browser object model) 浏览器对象模型

window

window是浏览器的一个实例，在浏览器中，window对象有双重角色，它既是通过JavaScript访问浏览器窗口的一个接口，又是ECMAScript规定的Global对象。

Window对象的方法

语法：`window.alert (" content")`

功能：显示带有一段消息和一个确认按钮的警告框

语法：`window.confirm("message")`

功能：显示一个带有指定消息和OK及取消按钮的对话框

返回值：如果用户点击确定按钮，则`confirm ()`返回`true`

如果用户点击取消按钮，则`confirm ()`返回`false`

Window对象的方法

语法：`window.prompt("text,defaultText")`

参数说明：

text：要在对话框中显示的纯文本（而不是HTML格式的文本）

defaultText：默认的输入文本

返回值：如果用户单击提示框的取消按钮，则返回null

如果用户单击确认按钮，则返回输入字段当前显示的文本

Window对象的方法

语法：`window.open(pageURL,name,parameters)`

功能：打开一个新的浏览器窗口或查找一个已命名的窗口

参数说明：

pageURL：子窗口路径

name：子窗口句柄。

parameters：窗口参数(各参数用逗号分隔)

Window对象的方法

语法：`window.close()`

功能：关闭浏览器窗口

学习目标

1、掌握超时调用

2、掌握间歇调用

超时调用

语法：`setTimeout (code,millisec)`

功能：在指定的毫秒数后调用函数或计算表达式

参数说明：

1、code：要调用的函数或要执行的JavaScript代码串

2、millisec：在执行代码前需等待的毫秒数

说明：`setTimeout ()`只执行code一次。如果要多次调用，请使用

`setInterval ()`或者让code自身再次调用`setTimeout ()`

清除超时调用

语法：`clearTimeout (id_of_settimeout)`

功能：取消由`setTimeout ()`方法设置的timeout

参数说明：

1、 id_of_settimeout : 由`setTimeout ()`返回的ID值，该值标识要取消的延迟执行代码块

间歇调用

语法：`setInterval(code,millisec)`

功能：每隔指定的时间执行一次代码

参数说明：

- 1、code：**要调用的函数或要执行的代码串。
- 2、millisec：**周期性执行或调用code之间的时间间隔，以毫秒计

学习目标

- 1、掌握location对象
- 2、掌握location对象的常用属性

location对象

location对象提供了与当前窗口中加载的文档有关的信息，还提供了一些导航的功能，它既是window对象的属性，也是document对象的属性。

location对象的常用属性

语法：location.href

功能：返回当前加载页面的完整URL

说明：location.href与window.location等价

语法：location.hash

功能：返回URL中的hash（#号后跟零或多个字符），如果不包含
则返回空字符串。

location对象的常用属性

语法：location.host

功能：返回服务器名称和端口号（如果有）

语法：location.hostname

功能：返回不带端口号的服务器名称。

语法：location.pathname

功能：返回URL中的目录和（或）文件名。

location对象的常用属性

语法：location.port

功能：返回URL中指定的端口号，如果没有，返回空字符串。

语法：location.protocol

功能：返回页面使用的协议。

语法：location.search

功能：返回URL的查询字符串。这个字符串以问号开头。

学习目标

- 1、掌握位置操作
- 2、掌握`location.replace()`
- 3、掌握`location.reload()`

位置操作

改变浏览器位置的方法：

location.href属性

location对象其他属性也可改变URL：

location.hash

location.search

location.replace()

语法：location.replace(url)

功能：重新定向URL。

说明：使用location.replace不会在历史记录中生成新纪录。

语法：location.reload()

功能：重新加载当前显示的页面。

说明：

- location.reload()有可能从缓存中加载
- location.reload(true)从服务器重新加载

学习目标

- 1、掌握对象字面量表示法
- 2、掌握编码与解码方法
- 3、掌握解析地址栏中的参数

对象字面量

语法: `var obj={key1:value1,key2:value2...}`

功能：使用对象字面量的形式创建一个对象

说明：

- 1、可以使用`obj.key1`或`obj["key1"]`的形式访问对象中的键值
- 2、可以用`obj.key1=value1`的形式为obj添加值。

encodeURIComponent()

语法：encodeURIComponent(URIstring)

功能：可把字符串作为 URI 进行编码。

参数：必需。一个字符串，含有 URI 或其他要编码的文本。

说明：

- 该方法不会对 ASCII 字母和数字进行编码
- 不会对 ASCII 标点符号进行编码：- _ . ! ~ * ' () 。
- 如 URI 组件中含有分隔符，如 ? 和 #

应使用 encodeURIComponent() 方法分别对各组件进行编码

decodeURI()

语法：decodeURI(URIstring)

功能：可对 encodeURI() 函数编码过的URI 进行解码。

参数：必需。一个字符串，含有要解码的 URI 或其他要 解码的文本。

encodeURIComponent()

语法：encodeURIComponent(URIstring)

功能：可把字符串作为 URI 组件进行编码。

参数：必需。一个字符串，含有 URI 组件或其他要编码的文本。

说明：

- 不会对 ASCII 字母和数字进行编码，
- 也不会对这些 ASCII 标点符号进行编码： - _ . ! ~ * ' () 。

decodeURIComponent()

语法：decodeURIComponent(URIstring)

功能：对 encodeURIComponent() 函数编码的 URI 进行解码。

参数：必需。一个字符串，含有编码 URI 组件或其他要解码的文本。

学习目标

掌握BOM中的history对象

history历史对象

语法：history.back()

功能：回到历史记录的上一步

说明：相当于使用了history.go(-1)

history历史对象

语法：`history.forward()`

功能：回到历史记录的下一步

说明：相当于使用了`history.go(1)`

history历史对象

语法：`history.back()`

功能：回到历史记录的上一步

说明：相当于使用了`history.go(-1)`

语法：`history.forward()`

功能：回到历史记录的下一步

说明：相当于使用了`history.go(1)`

history历史对象

语法：`history.go(-n)`

功能：回到历史记录的前n步

语法：`history.go(n)`

功能：回到历史记录的后n步

学习目标

掌握Screen对象及其常用属性

Screen对象属性

语法：`screen.availWidth`

功能：返回可用的屏幕宽度

语法：`screen.availHeight`

功能：返回可用的屏幕高度

Navigator对象

学习目标

- 1、掌握Navigator对象的userAgent属性
- 2、掌握如何判断浏览器的类型
- 3、掌握如何判断设备的终端是移动还是PC

Navigator

UserAgent：用来识别浏览器名称、版本、引擎 以及操作系统等信息的内容。