

一、BOM

1. BOM 介绍

2. window对象常用方法

- 1) 网页弹框
- 2) 窗口的打开和关闭
- 3) 定时器方法

window对象常用属性

- 1) history
- 2) location
- 3) document

二、DOM节点操作

- 1) 节点对象
- 2) 常用节点分类
- 3) 获取元素节点
- 4) 操作元素内容
- 5) 操作元素属性
- 6) 操作元素样式

一、BOM

1. BOM 介绍

BOM全称为“Browser Object Model”，浏览器对象模型。提供一系列操作浏览器的属性和方法。核心对象为window对象，不需要手动创建，跟随网页运行自动产生，直接使用，在使用时可以省略书写。

2. window对象常用方法

1) 网页弹框

```
alert()      //警告框
prompt()     //带输入框的弹框
confirm()    //确认框
```

2) 窗口的打开和关闭

```
window.open("URL") //新建窗口访问指定的URL
window.close()     //关闭当前窗口
```

3) 定时器方法

1. 间歇调用(周期性定时器)

作用：每隔一段时间就执行一次代码

开启定时器：

```
var timerID = setInterval(function, interval);  
/*  
参数 :  
    function : 需要执行的代码, 可以传入函数名; 或匿名函数  
    interval : 时间间隔, 默认以毫秒为单位 1s = 1000ms  
返回值 : 返回定时器的ID, 用于关闭定时器  
*/
```

关闭定时器:

```
//关闭指定id对应的定时器  
clearInterval(timerID);
```

2. 超时调用(一次性定时器)

作用: 等待多久之后执行一次代码

```
//开启超时调用:  
var timerId = setTimeout(function, timeout);  
//关闭超时调用:  
clearTimeout(timerId);
```

window 对象常用属性

window的大部分属性又是对象类型

1) history

作用: 保存当前窗口所访问过的URL

属性:

length 表示当前窗口访问过的URL数量

方法:

```
back() 对应浏览器窗口的后退按钮, 访问前一个记录  
forward() 对应前进按钮, 访问记录中的下一个URL  
go(n) 参数为number值, 翻阅几条历史记录, 正值表示前进, 负值表示后退
```

2) location

作用: 保存当前窗口的地址栏信息(URL)

属性:

href 设置或读取当前窗口的地址栏信息

方法:

reload(param) 重载页面(刷新)

参数为布尔值, 默认为false, 表示从缓存中加载, 设置为true, 强制从服务器根目录加载

3) document

提供操作文档HTML 文档的方法, 参见DOM

二、DOM节点操作

DOM全称为“Document Object Model”，文档对象模型，提供操作HTML文档的方法。（注：每个html文件在浏览器中都视为一篇文档,操作文档实际就是操作页面元素。）

1) 节点对象

JS 会对html文档中的元素，属性，文本内容甚至注释进行封装，称为节点对象，提供相关的属性和方法。

2) 常用节点分类

- 元素节点（操作标签）
- 属性节点（操作标签属性）
- 文本节点（操作标签的文本内容）

3) 获取元素节点

1. 根据标签名获取元素节点列表

```
var elems = document.getElementsByTagName("");  
/*  
参数：标签名  
返回值：节点列表, 需要从节点列表中获取具体的元素节点对象  
*/
```

2. 根据class属性值获取元素节点列表

```
var elems = document.getElementsByClassName("");  
/*  
参数：类名(class属性值)  
返回值：节点列表  
*/
```

3. 根据id属性值取元素节点

```
var elem = document.getElementById("");  
/*  
参数：id属性值  
返回值：元素节点  
*/
```

4. 根据name属性值取元素列表

```
var elems = document.getElementsByName("");  
/*  
参数：name属性值  
返回：节点列表  
*/
```

4) 操作元素内容

元素节点对象提供了以下属性来操作元素内容

innerHTML : 读取或设置元素文本内容,可识别标签语法
innerText : 设置元素文本内容,不能识别标签语法
value : 读取或设置表单控件的值

5) 操作元素属性

1. 通过元素节点对象的方法操作标签属性

```
elem.getAttribute("attrname");//根据指定的属性名返回对应属性值  
elem.setAttribute("attrname", "value");//为元素添加属性, 参数为属性名和属性值  
elem.removeAttribute("attrname");//移除指定属性
```

2. 标签属性都是元素节点对象的属性,可以使用点语法访问, 例如:

```
h1.id = "d1";           //set 方法  
console.log(h1.id);     //get 方法  
h1.id = null;           //remove 方法
```

注意:

- 属性值以字符串表示
- class属性需要更名为className,避免与关键字冲突,例如:
h1.className = "c1 c2 c3";

6) 操作元素样式

1. 为元素添加id, class属性, 对应选择器样式
2. 操作元素的行内样式,访问元素节点的style属性, 获取样式对象; 样式对象中包含CSS属性, 使用点语法操作。

```
p.style.color = "white";  
p.style.width = "300px";  
p.style.fontSize = "20px";
```

注意:

- 属性值以字符串形式给出,单位不能省略
- 如果css属性名包含连接符,使用JS访问时,一律去掉连接符,改为驼峰. font-size -> fontSize