

千锋HTML5学院

第二阶段javascript课程课件

1

BOM的概念

2

window 对象

3

延时调用

4

Location对象

5

history对象

6

onScroll事件

BOM的概念

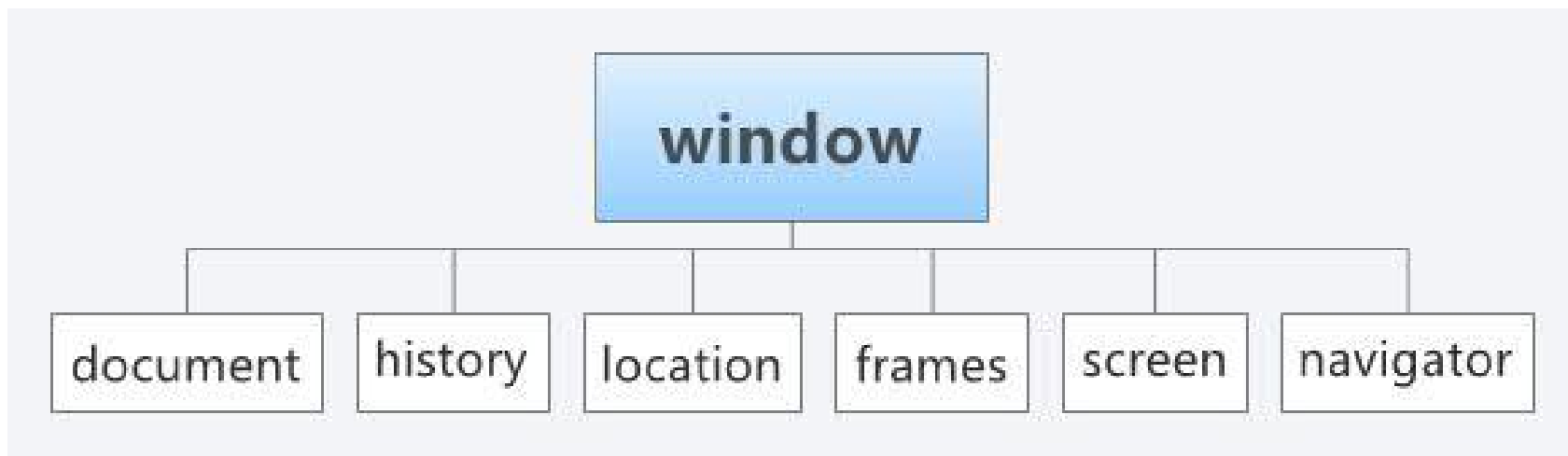
□ BOM (Browser Object Model)

BOM 是Browser Object Model的缩写，简称浏览器对象模型，提供了独立于内容而与浏览器窗口进行交互的对象，用于访问浏览器的功能。BOM由一系列相关的对象构成，并且每个对象都提供了很多方法与属性。

JavaScript语法的标准化组织是ECMA，DOM的标准化组织是W3C，但是BOM缺乏标准。由于BOM 缺少规范，每个浏览器提供商又按照自己想法去扩展它，那么浏览器共有对象就成了事实的标准。

window对象

window是BOM的核心, **window**对象表示浏览器窗口的一个实例, 每个浏览器窗口都有一个**window**对象与之对应. **window**对象是BOM的顶层(核心)对象, 所有对象都是通过它延伸出来的.



window对象

window对象的子对象:

document(核心): 文档对象, 让我们可以在js脚本中直接访问页面元素(DOM)

history(重要): 历史对象, 包含窗口的浏览历史, 可以实现后退

location(重要): 包含浏览器当前的地址信息, 可以用来刷新本页面或跳转到新页面

frames: 框架对象, 可以获取页面框架内容

screen: 包含有关客户端显示屏幕的信息

navigator: 导航对象, 包含所有有关访问者浏览器的信息

window对象

window对象的部分方法:

alert(text): 弹出提示框(警告框)

confirm(): 创建一个需要用户确认的对话框

prompt(text,defaultInput) :创建一个对话框要求用户输入信息

open(url,name,[options]) : 打开一个新窗口并返回新 window 对象

setInterval(): 设置定时器

clearInterval() : 移除定时器

setTimeout() : 设置延时器

clearTimeout(): 移出延时器

close(): 关闭窗口

print(): 调出打印对话框

window对象

window对象的属性和方法的调用:

window对象的属性和方法的调用方式，可以使用 `window.属性`、`window.方法()` 或者一般可以直接使用属性、方法()的方式调用。

例如: `window.alert()` 和 `alert()` 效果是一样的

系统对话框

浏览器可以通过`alert()`、`confirm()`和`prompt()`方法可以调用系统对话框向用户显示信息。

弹出警告框

```
alert("Lee");
```

提示框 :确定和取消

```
confirm("请确定或者取消");
```

```
if (confirm(("请确定或者取消")){
```

```
    alert("您按了确定！");    //按确定返回true
```

```
} else{
```

```
    alert("您按了取消！")    //按取消返回false
```

```
}
```

带输入的提示框: 提示文字和输入框默认值

```
var num = prompt("请输入一个数字", 0);
```

```
alert(num);    //点击确定得到输入的值, 点击取消返回null
```


open()新建窗口

window.open() 方法可以导航到一个特定的URL，也可以打开一个新的浏览器窗口，一般可以接受三个参数：

1. 要加载的URL
2. 窗口的名称或者窗口的目标
3. 一个特性的字符串

```
open("http://www.baidu.com");           //新建页面并打开百度
open("http://www.baidu.com", "baidu");   //新建页面并打开百度,窗口命名为baidu
open("http://www.baidu.com", "_parent"); //在本页窗口打开,_blank是新建(默认)
```

【注】不命名会每次打开新窗口，命名的第一次打开新窗口,之后在这个窗口中加载。

open()新建窗口

第三个参数字符串

设置	值	说明
width	数值	新窗口的宽度。不能小于100
height	数值	新窗口的高度。不能小于100
top	数值	新窗口的Y坐标。不能是负值
left	数值	新窗口的X坐标。不能是负值

```
open('http://www.baidu.com', 'baidu',  
'width=400,height=400,top=200,left=200' );
```

子窗口对象和父窗口对象

open()方法返回打开的子窗口的window对象

```
var box = open();
```

```
box.document.write(subWindow.name);
```

opener: 父窗口对象(谷歌浏览器不支持, IE/火狐浏览器支持)

```
document.onclick = function(){
```

```
    opener.document.write("调用父窗口对象输出!");
```

```
}
```

延时调用

`setTimeout()` 指定的时间过后执行代码

写法一:

```
setTimeout("alert('hello')", 1000); //不建议直接使用字符串
```

写法二:

```
function box(){  
    alert("hello");  
}
```

```
setTimeout(box, 1000); //直接传入函数名即可
```

写法三:

```
setTimeout(function(){ //推荐做法  
    alert("hello");  
}, 1000);
```

【注】直接使用函数传入的方法，扩展性好，性能更佳。

上述方法会返回唯一标识符，可以通过它来取消超时调用。

要取消尚未执行的超时调用计划，可以调用`clearTimeout()`方法并将相应的超时调用ID作为参数传递给它。

```
clearTimeout(box);
```

close() 关闭浏览器

close() 关闭浏览器

注意: 火狐浏览器 不支持

//5秒后关闭浏览器

```
setTimeout(function(){  
    window.close();  
    alert("窗口关闭");  
}, 5000);
```

print(): 调出打印对话框

Location对象

location是BOM对象之一，它提供了与当前窗口中加载的文档有关的信息，还提供了一些导航功能。

事实上，location对象是window对象的属性，也是document对象的属性；所以window.location和document.location等效。

```
alert(window.location);    //获取当前的URL  
alert(location);           //window可以省略  
alert(window.document.location);
```

Location对象的属性

属性	描述的 URL 内容
hash	如果该部分存在, <u>表示锚点部分</u>
host	主机名: 端口号
hostname	主机名
<u>href</u>	整个 URL
pathname	路径名
port	端口号
protocol	协议部分
search	查询字符串

Location对象的属性

`location.hash = '#1';` //设置#后的字符串, 并跳转
`alert(location.hash);` //获取#后的字符串

`location.port = 8888;` //设置端口号, 并跳转
`alert(location.port);` //获取当前端口号

`location.hostname = 'hello';` //设置主机名(域名), 并跳转
`alert(location.hostname);` //获取主机名(域名)

`location.pathname = 'hello';` //设置当前路径(服务器地址后部分), 并跳转
`alert(location.pathname);` //获取当前路径(服务器地址后部分)

`location.search = "?id=5";` //设置?后面的字符串, 并跳转
`alert(location.search);` //获取?后面的字符串

`location.href = "http://www.baidu.com";` //设置跳转的URL, 并跳转
`alert(location.href);` //获取当前的URL

在Web开发中，我们经常需要获取诸如?id=5&search=ok这种类型的URL键值对,那么通过location,我们可以写一个函数,来一一获取。

```
function getArray(){  
    //创建一个存放键值对的数组  
    var array = [];  
    //取出?号  
    var str = location.search.length > 0 ? Location.search.substring(1) :"";  
    //按&字符串拆分数组  
    var items = str.split("&");  
    //遍历  
    for(var i = 0; i < items.length; i++){  
        var itemArray = items[i].split("=");  
        array[i] = itemArray[1];  
    }  
    return array;  
}
```

Location对象的方法

方法	功能
<code>assign()</code>	跳转到指定页面，与 <u><code>href</code></u> 等效
<code>reload()</code>	重载当前 URL
<u><code>replace()</code></u>	用新的 URL 替换当前页面

`location.assign('http://www.baidu.com');` //跳转到指定的URL

`location.reload();` //最有效的重新加载,有缓存加载

`location.reload(true);` //强制加载,从服务器源头重新加载

`location.replace("http://www.baidu.com");` //可以避免产生历史记录

history对象

【history对象】

history对象是window对象的属性，它保存这用户上网的记录,从窗口被打开的那一刻算起。

属性

history.length; //history对象中的记录数

方法

history.back(); //前往浏览器历史条目前一个URL,类似后退

history.forward(); //前往浏览器历史条目下一个URL,类似前进

history.go(-1); //浏览器在history对象中向前或向后(-1代表后退, 1代表向前,0表示刷新当前页面)

练习

1, 利用location和history实现跳转, 效果如图:



练习



夏颂——纳清风台榭开怀，傍流水亭轩赏心
夏初——槐柳成阴雨洗尘，樱桃乳酪并尝新
夏热——日轮当午凝不去，万国如在洪炉中
夏雨——风声撼山翻怒涛，雨点飞空射强弩
夏水——涧影见藤竹，潭香闻芰荷
夏夜——散发乘夕凉，开轩卧闲敞
夏花——荷叶罗裙一色裁，芙蓉向脸两边开
夏鸟——漠漠水田飞白鹭，阴阴夏木啭黄鹂
夏蝉——垂緌饮清露，流响出疏桐
夏节——彩线轻缠红玉臂，小符斜挂绿云鬟
夏游——松风涧水杂清音，空山如弄琴
夏趣——懒摇白羽扇，裸体青林中

[返回上一页](#)

[前往春天](#)

[前往秋天](#)

[前往冬天](#)

navigator对象

【navigator对象】

navigator对象是window对象的属性，它保存了浏览器的信息，如：浏览器名称,版本,浏览器所在的电脑操作系统等。

属性

```
console.log(navigator.appName); //浏览器名称  
console.log(navigator.appVersion); //浏览器版本  
console.log(navigator.platform); //操作系统  
//最新的浏览器已经全面放弃以上这些属性
```

属性

```
console.log(navigator.userAgent);  
//用户代理信息, 通过该属性可以获取浏览器及操作系统信息
```

onScroll滚动事件

onScroll滚动事件: 实时获取距离顶部的距离

//document.documentElement.scrollTop 不支持谷歌

//document.body.scrollTop: 支持谷歌

```
window.onscroll = function() {  
    var scrollTop = document.documentElement.scrollTop  
                    || document.body.scrollTop;  
    var box = document.getElementById("box");  
    box.innerHTML = scrollTop;  
}
```

示例: 滚动页面, 显示和顶部的距离



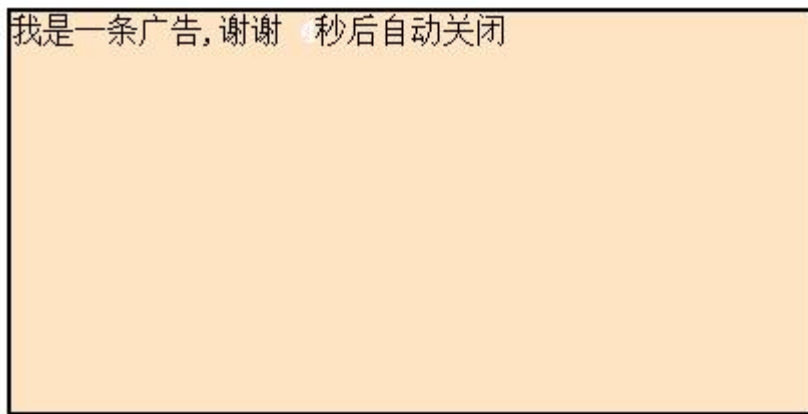
练习

1, 回到顶部, 当滚动条滚动到距离顶部超过2000时, 显示回到顶部(红色部分), 并当点击回到顶部时, 让页面回到顶部



作业

- 1, 将今天课堂代码写一遍
- 2, 广告窗口自动关闭, 倒计时5秒后关闭窗口 (分别使用定时器和延时器)



- 3, 顶部悬浮, 当滚动条滚动到较下方时, 留言信息一直显示在顶部

