BOM浏览器对象模型

BOM简介

概述

浏览器对象模型，简称BOM，BOM提供了独立于内容而是与浏览器窗口进行交互的对象。BOM的核心是window对象。与ECMAScript不同的是，BOM针对的是浏览器，鉴于市面上的浏览器多种多样，

因此会影响BOM的兼容性

。

浏览器的兼容

浏览器因为它的多厂商，导致兼容问题是不容忽视的，主流浏览器中各个版本的兼容也不尽相同。因此，在BOM中运用时可能会出现结果不同或者毫无结果的情况。

window 对象

window对象的概念

window对象是BOM的核心对象，他除了可以使用全局对象的属性和方法之外，还可以执行浏览器的一些方法。

全局作用域

在全局作用域下创建的变量和函数都会成为window的属性和方法。

var a = 10;

console.log(window.a);

function b(){

return '你好';

}

console.log(window.b());

//在全局下window可省略

窗口大小

(1)视图的大小

可见内容视图相当于document.body的宽度和高度

//视口相当于document.body的宽度和高度

var w = document.body.clientWidth;

var h = document.body.clientHeight;

console.log(w + ',' + h);

(2)视口的大小

浏览器展现区域的宽高，包含滚动条所占区域.

var ww = window.innerWidth;

var wh = window.innerHeight;

console.log(ww + ',' + wh);

open

window.open('需要打开的地址','窗口打开的方式');这是最简单的open参数，还可以增加其他内容。使用open打开的页面基本都可以用close关闭，但open和close有兼容性问题。

document.getElementById('a').onclick = function(){

window.open('视口大小.html','\_top');

}

弹出框

alert('弹出框');

confirm('选择弹出框');

prompt('输入弹出框');

超时调用(一次性定时器)

setTimeout(回调函数，时间)也叫一次性定时器，创建这个定时器之后会在一定时间中调用函数，执行函数中的脚本。

setTimeout(function(){},1000);

间歇调用(循环定时器)

setInterval(回调函数,时间)也叫循环定时器,创建这个定时器之后会每隔一段时间触发,触发的时间间隔就是

注意:循环定时器存在bug,执行时间大于等待时间将会出现任务堆叠.一般情况下,使用setTimeout代替.

location对象

location对象的概念

window的location对象用于获取当前页面的地址(URL),并把浏览器重新定向到新的页面.在编写时可以不用加window前缀.

查看location对象的常用属性

href 返回当前完整的URL地址

host 返回的是服务器名称和端口号

port 返回端口号

pathname 返回的是目录和文件名

search 返回的是参数，也就是？后面的内容

protocol 返回使用页面的协议，如http/https

navigator对象

概念

window的navigator对象包含关于浏览器的信息,在编辑的时候不能使用window这个前缀,直接使用navigator,该对象代表浏览器本身的名称,版本,系统等信息,但存在兼容性问题,请慎用.

属性介绍

console.log(navigator);

navigator.appName : 浏览器名称

navigator.appVersion : 浏览器的版本

navigator.userAgen : 判断IE浏览器

鉴于兼容问题,在固定浏览器的特定版本进行自主寻找.

判断浏览器类型

if(navigator.userAgent.indexOf('MSIE')!=-1){

console.log('IE浏览器');

}else {

console.log('非IE浏览器');

}

对象

概念

screen对象表示一个屏幕的窗口，他会随着浏览器的不同，窗口大小的位置不同返回不同的值，迁移是返回的是当前打开窗口的内容。

属性

console.log(screen);

history对象

概念

history对象保存着从窗口被打开的历史记录，每个浏览器窗口，标签页，框架都有自己的history对象。为了保护用户隐私，对js访问该对象作出了一部分限制。

console.log(history);

常用方法

history.go(n)整数 0代表刷新，正数代表向前跳n步，负数代表向后跳n步。

history.back(n)代表向后跳n步。

(3)