BOM：浏览器对象模型

window对象：BOM的核心对象，它表示浏览器的一个实例。也是ECMA规定的Global对象。这意味着网页中定义的任何一个对象、变量和函数，都以window作为其Global对象，因此有权访问parseInt()等方法

全局作用域

所有在全局作用域中声明的变量、函数都会变成window对象的属性和方法

定义全局变量和在window对象上直接定义的差别：全局变量不能通过delete操作符删除，而直接在window对象上定义的就可以

var age = 20;

window.color = ‘red’;

delete window.age; //抛出错误

delete window.color;

窗口关系及框架

页面中包含框架，则每个框架都拥有自己的window对象，并且保存在frames集合中，可以通过数值索引（从0开始，从左至右，从上到下）或者框架名称来访问相应的window对象，每个window对象都有一个name属性，其中包含框架的名称

<html>

<head>

<title> JS</title>

</head>

<frameset rows="160,\*">

<frame src="1.html" name="topframe">

<frameset cols="50%,50%">

<frame src="2.html" name="leftframe">

<frame src="3.html" name="rightframe">

</frameset>

</frameset>

</html>

window.frames[0] window.frame[“topframe”]但最好使用 top来引用这些框架，top对象始终指向最高（最外）层的框架，也就是浏览器窗口

parent对象始终指向当前框架的直接上层框架，parent有可能等于top,但在没有框架的情况下，parent一定等于top，此时它们都等于window

3.html中又包含了框架

<html>

<head>

<title> test</title>

</head>

<frameset cols="50%,50%">

<frame src="4.html" name="4frame">

<frame src="5.html" name="5frame">

</frameset>

</html>

间歇调用和超时调用

JS是单线程语言，

间歇：每隔指定的时间就执行一次代码。

使用setInterval()方法，接受的参数和setTimeout()相同

超时：在指定的时间过后执行代码。

使用window对象的setTimeout()方法，接受两个对数，要执行的代码，和以毫秒表示的时间

setTimeout(“console.log(‘Hello World!’)”,10000);

推荐以下调用方式：

setTimeout(function(){

console.log(“Hello World!”)

},10000);

为了控制要执行的代码，就有一个JS任务队列，这些任务会按照将它们添加到队列的顺序执行。调用setTimeou()方法后，该方法会返回一个数值ID，表示超时调用，这个超时调用ID是计划执行代码的唯一标识符，可取消超时调用。

clearTimeout(数值ID)

超时调用的代码都是在全局作用域中执行的，this指向window对象

系统对话框

alert()

confirm()

prompt()

find()

print()

location对象

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。location既是window对象的属性，也是document对象的属性，window.location document.location引用的是同一个对象。

location对象的用处不只表现在它保存着当前文档的信息，它还将URL解析为独立的片段，让开发人员可以通过不同的属性访问这些片段。

host 返回服务器名称和端口号

hostname 返回不带端口号的服务器名称

href 返回当前加载页面的完整URL

pathname 返回URL中的目录或文件名

port 返回URL中指定的端口号，如URL不包含端口号，返回空字符串

protocol 返回页面使用的协议 http: https:

search 返回URL的查询字符串，这个字符串以？开头

查询出神入化串参数search

不能逐个访问其中的每个查询字符串参数，创建一个函数用以解析查询字符串，然后返回包含所有的参数的一个对象

function getQuerySringArgs(){

var qs = (location.search.length > 0 ? location.search.substring(1) : “”);

args = {};

items = qs.length ? qs.split(“&”) : [],

item = null,

name = null,

value = null,

I = 0,

len = items.length;

for(I = 0; I < len; i++){

item = item[i].split(“=”);

name = decodeURIComponent(item[0]);

value = decodeURIComponent(item[1]);

if(name.length){

args[name] = value;

}

}

return args;

}

split(“分割符”)把字符串以分割符为标识分割马字符串数组

decodeURIComponent(参数)对参数内容进行解码

http%3A%2F%2Fwww.w3school.com.cn%2 ＝＝＝〉http://www.w3school.com.cn/