函数和window对象

2019年8月20日

10:28

一、什么是函数

类似于java中的方法，是完成特定功能的代码块

function/java中的方法

1、自定义函数

function 函数名（参数1，参数2）{

……

return 值；

}

调用：函数名（实参1，实参2）

2、系统函数

**//toString()转换为字符串(?)**

**//parseInt()转换为整数**

document.write(parseInt(100.54)+"<br>");//100

document.write(parseInt("100)+"<br>");//100

document.write(parseInt("123abc456")+"<br>");//123

document.write(parseInt("abc123")+"<br>");//NaN

document.write(parseInt("true")+"<br>");//NaN

如果遇到不能转换的地方就停止，开始不能转换返回NaN

**//parseFloat（）转换为小数**

document.write(parseFloat("123.0")+"<br>");//123.0

注意：没有小数点或小数点后全是零就返回整数

document.write(parseFloat("12a3.0")+"<br>");//12

document.write(parseFloat("123.456.789")+"<br>");//123.456

注意：第二个小数点之后的数舍去

**//isNaN()判断参数是否是非数字，参数数字：false参数非数字：true**

document.write(isNaN(123));//false

document.write(isNaN("abc"));//true

document.write(isNaN("123abc"));//true

document.write(isNaN(true));//true

注意：isNaN()能将参数转换为数字，能转换的结果为false，不能转换的true

3、匿名函数

var abc = function(){}

调用：abc(3,4);

把一个函数赋给一个变量，而这个函数是没有名字的，即匿名函数。

<script>

var abc=function(x,y){

return x+y;

}

alert(abc(3,4));

</script>

二、响应事件

onclick:当标签被点击的时候触发，单击

ondbclick:双击的时候触发

onload:当一个页面或一个图片被加载完成的时候触发

onchange:当标签内容发生变化时触发

onkeydown:当键盘按下时触发

onmouseover:当鼠标滑过时触发

onblur:当标签失去焦点时触发

onfocus:当标签得到焦点时触发

onscroll:当滚动条滚动时触发

三、window对象

window对象表示浏览器中打开的一个窗口。

1、常用属性

screen:包含有关客户端显示屏幕的信息。可利用width和height获取屏幕的宽高。

alert(window.screen.width);//浏览器所在屏幕的宽度

alert(window.screen.height);//浏览器所在屏幕的高度

2、history:历史信息

length:浏览器历史列表中的url数量

back():加载历史列表中的上一个url

window.history.back();

forward();加载历史列表中的下一个url

window.history.forward();

go(n):单击n次后退按钮

3、location :包含有关当前url的信息，

1、常用属性

alert(window.location.href);//返回当前HTML文档的完整URL

**window.location.href="http://www.baidu.com";//跳转到百度**

**window.location.reload():**重新载入当前页面，等同于刷新

**window.location.replace("Url"):**在当前页面打开，不保留历史访问记录

2、常用函数

1)alert():

2)confirm(string message):

3)prompt()

4)close()

function close\_win(){

window.close();

}

5)open(url,打开方式,窗口特征)

function open\_win(){

window.open("demo03.html","\_blank","width=200px,height=300px,resizable=no");

}

窗口特征：宽，高，各种栏（工具栏，标题栏……），能否改变大小

6)setTimeout('要执行的操作',等待的时间)，在一定的时间之后执行相应的操作

<button type="button"onclick="setTimeout('wait()',2000)">点我等2秒显示</button>

7)setInterval('要执行的操作',间隔时间),每隔一段时间执行相应的操作。

<button type="button"onclick="setInterval('wait()',2000)">点我隔2秒显示</button>

**注意：时间单位为毫秒！**

案例：制作时钟特效

//调用

<body onload="setInterval('showTime()',1000)">

</body>

//显示时间函数

functionshowTime(){

//获取系统当前时间

var date=newDate();

var y=date.getFullYear();//年

var m=date.getMonth()+1;//月

var d=date.getDate();//日

var h=date.getHours();//时

var mi=date.getMinutes();//分

var s=date.getSeconds();//秒

var myClock=document.getElementById("myClock");//找到div

myClock.innerHTML=y+":"+m+":"+d+":"+h+":"+mi+":"+s;

}

**监听滚动条滚动事件(注意:如果滚动条是左右滚动,那么scrollTop应改为scrllLeft)**

window.onscroll = function () {

var sTop = document.body.scrollTop + document.documentElement.scrollTop;

if (sTop % 100 == 0) {

console.log(sTop)

}

};

scrollTop 和scrollLeft是可以赋值的,并会立刻滚动到网页的相应位置;element.scrollIntoView()

也有类似的作用,使元素立刻出现在浏览器的左上角

**获取元素的相对位置(窗口左上角)和绝对位置(document左上角)**

1. 使用getBoundingClientRect()方法

相对位置

var X= this.getBoundingClientRect().left;

var Y =this.getBoundingClientRect().top;

绝对位置

var X= this.getBoundingClientRect().left+document.documentElement.scrollLeft;

var Y =this.getBoundingClientRect().top+document.documentElement.scrollTop;

**监听键盘按键事件,其中,enter键的keyCode ==13;**

window.onkeydown = function (e) {

console.log(e);

console.log(window.event);

var keyCode = window.event ? e.keyCode : e.which;

if(keyCode == 13){

window.alert("按下了enter键")

}

}

**自定义右键事件**

window.oncontextmenu = function (e) {

console.log(e);

if(!e) e = window.event;

e.preventDefault();

};//阻止默认的右键弹出框

window.onmouseup = function (e) {

if (e.button == 2) {

window.alert("右键事件")

}

}//新增右键事件

**自定义双击事件(注意:方法是 ondblclick 而不是 dbclick)**

window.ondblclick = function(e){

console.log(e);

alert("双击事件")

}

注意: 由于浏览的像素是只有整数的,所以当滚动条滚动到底部时, scrollBottom 不一定和clientHeight相等

,有可能略大于clientHeight

**阻止离开页面**

window.isCloseHint = true;

//初始化关闭

window.addEventListener("beforeunload", function(e) {

if (window.isCloseHint) {

var confirmationMessage = "要记得保存！你确定要离开我吗？";

(e || window.event).returnValue = confirmationMessage; // 兼容 Gecko + IE

return confirmationMessage // 兼容 Gecko + Webkit, Safari, Chrome

}

});

**关闭页面前发送请求(做统计功能)**

1. 添加监听事件, 在文本卸载前发送请求

window.addEventListener("beforeunload", dealMethod)

或者

window.addEventListener("unload", dealMethod)

或者

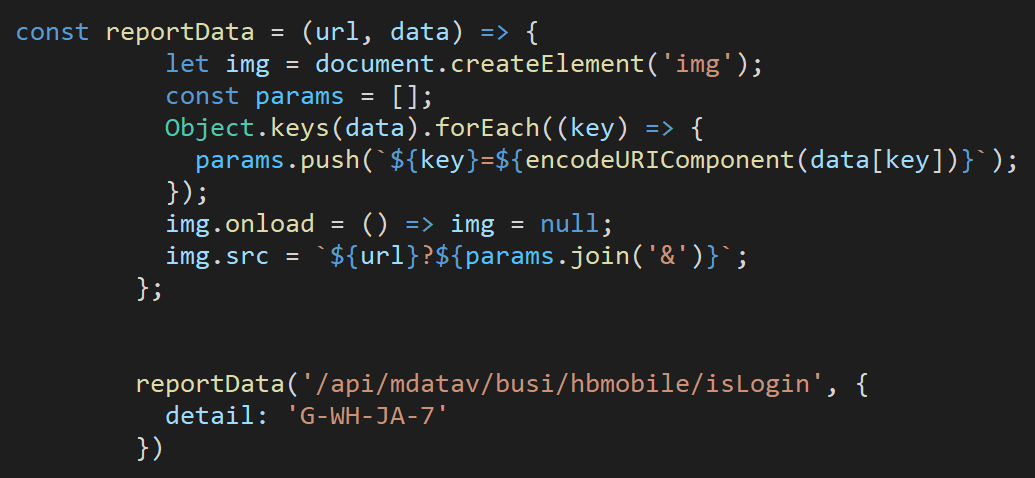
window.addEventListener("visibilitychange", function() {

if (document.visibilityState === 'hidden') {

dealMethod()

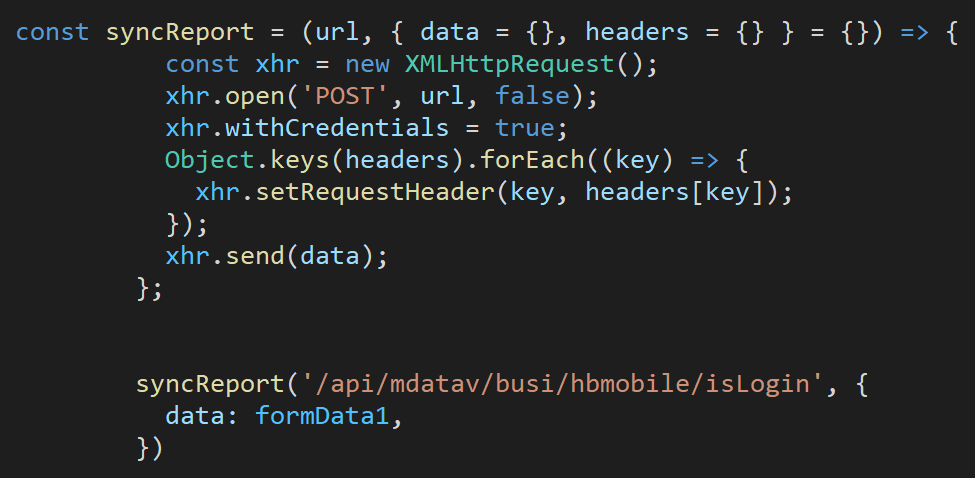
}  
})

1. 处理方法, 发送请求
   * 利用img加载发送get请求



特点: 兼容性好, 只能发送get请求

* 发送同步请求

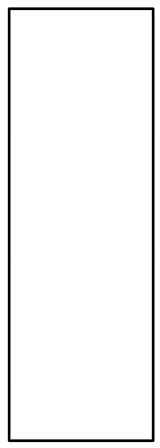


特点: 未能成功发送请求, 但兼容性好, 能发送POST请求

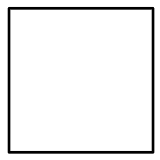
* sendBeacon



特点: 只能发送POST请求, 不能兼容IE, 能成功发送



scrollTop



scrollHeight

clientHeight



scrollBottom

