/\*

JavaScript 事件参考



\*/

1. j1

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>hello world!!!</title>

<script type="text/javascript">

// alert("1hello world!!!");//弹窗

// document.write("2hello world!!");//页面

// console.log("3hello world!!!");//控制台

//int a=10,b=20;

var a=10,b=20;

//int temp=a+b;

var temp=a+b;

// System.out.print();

alert(temp);

</script>

</head>

<body>

</body>

</html>

Sj2

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>引用js代码，数据类型，正则表达式</title>

<script type="text/javascript" src="js/s1.js"></script>

</head>

<body>

</body>

</html>

Sj3

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>全局函数，自定义函数,自定义对象</title>

<script type="text/javascript" src="js/s2.js"></script>

</head>

<body>

</body>

</html>

Sj4

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>BOM对象 定时器 location</title>

<script type="text/javascript" src="js/s3.js"></script>

</head>

<body>

</body>

</html>

Sj5

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>hello包含</title>

</head>

<body>

<!--可以将js代码编写到标签的onclick属性中

当我们点击按钮时，js代码才会执行

虽然可以写在标签的属性中，但是他们属于结构与行为耦合，不方便维护，不推荐使用

-->

<button onclick="alert('你好')">hi</button>

<a href="javascript:alert('hhhh')">ohhh</a>

<!--可以将js代码写在超链接的href属性中，这样当点击超链接时，会执行js代码-->

<a href="javascript:">ohhh</a>

</body>

</html>

Js1

// alert("hello world!!!");

// var str="hello world";输出

// alert(str);输出

// document.write(str);输出

/\*document.writeln(typeof(str));

str=10;

document.writeln(typeof(str));

str=true;

document.writeln(typeof(str));

str=12.03;

document.writeln(typeof(str));

str=null;

document.writeln(typeof(str));\*/

//typeof()显示数据类型

/\*var str=10;

var str2="10";

document.writeln(str==str2);

document.writeln(str===str2);

===全等，必须要类型和值都相同\*/

/\* var reg=/^\s\*$/;//0到多个空格

document.write(reg.test(" a "));

必须全部字符符合正则返回true

有一个字符不符合正则就会返回false\*/

/\*var reg=/^\s+$/;//直接方式

document.write(reg.test(" we"));

var reg1=/\s+/;//普通方式

document.write(reg1.test(" qeq"));\*/

/\*/^表达式$/

只要有无法成立正则的字符存在，即为false。。

全部符合为true.

(检查严格，眼睛不揉沙子)。

适用于表单校验。

/表达式/

只要有成立正则的字符存在，即为true。。

全部不符合为false

(检查不严格，懒人思想)。

适用于字符串查找、替换。\*/

/\*var arr= [1,2,3,"a",true];//常用的JS数组。长度5。

var arr= new Array();创建一一个 数组对象，数组长度默认为0。\*/

/\*var ar=[1,2,33,4,"qeq"];

document.write(ar.length);

ar[1]="qqq";

ar[ar.length]="qqqwq";//可以给ar直接增加空间，添加元素

document.writeln(ar);

ar[10]=10;

document.writeln(ar);

document.writeln(ar.length);\*/

/\*var arr=['a','b','cc'];

document.writeln(arr.join('@'));//链接数组

document.writeln(arr.reverse());//反转数组

document.writeln(arr.pop());//删除最后一个元素并且返回数组长度

document.writeln(arr.push('qwq','mmm')); //向数组的末尾添加一个或更多元素，并返回新的长度。

document.writeln(arr.reverse())//颠倒数组中元素的顺序\*/

Js2

/\*

var msg="https://www.bilibili.com/video/av73840600?p=60小飞";

var temp=encodeURI(msg);//16进制编码转换

document.writeln(temp);

document.writeln(decodeURI(temp));//解码

\*/

/\*

var str="10.123";

var str2="10.123a";

var num1=parseInt(str);

var num2=parseFloat(str);

var num3=parseInt(str2);

var num4=parseFloat(str2);

document.writeln(typeof (num1)+num1);

document.writeln(typeof (num2)+num2);

document.writeln(typeof (num3)+num3);

document.writeln(typeof (num4)+num4);

/!\*如果字符串的某个字符从字面值上无法转为数字,那么从该字符开始停止转换,仅返回

前面正确的转换值。(例如: 11.5a55， parselnt 结果11，parseFloat 结果11.5)。

+

如果字符串的第-一个字符就无法以字面值上转为数字，那么停止转换，返回NaN.

NaN (Not A Number, -一个数字类型的标识，表示不是一个正确的数字)。

\*!/\*/

/\*

function getSum(a,b){

return a+b;

}

function getSum(a,b,c){

return a+b+c;

}

document.writeln(getSum(1,2));

document.writeln(getSum(1,2,3));

/!\*

JavaScript函数定义必须用小写的function;

JavaScript函数无需定义返回值类型，直接在function后面书写方法名;。

参数的定义无需使用var关键字，否则报错;

JavaScript函数体中,return可以不写，也可以return具体值,或者仅仅写return;。

JavaScript函数调用执行完毕一定有返回值，值及类型根据return决定，如果未return

具体值，返回值为undefined;

JavaScript函数若同名，则不存在方法重载，只有方法相互覆盖，最后定义的函数覆盖

之前的定义;。

因为JavaScript不存在函数重载,所以JavaScript仅根据方法名来调用函数,即使实参与

函数的形参不匹配，也不会影响正常调用;

\*!/\*/

/\*

function Peron(){

this.name="小飞";

this.age=20;

}

var a =new Peron();

document.writeln(a.name);

document.writeln(a.age);

\*/

/\*var per={name:"小飞",age:"20"};

document.writeln(per.age+per.name);\*/

Js3

/\*

确认窗口，有返回值

var a;

a=confirm("hello");

document.writeln(a);\*/

/\*var id;

function run1(){

alert("hello");

clearInterval(id);/!\*取消定时器\*!/

}

setInterval("run1()",2000);/!\*定时器，时间单位是毫秒\*!/

id=setInterval("run1()",2000);\*/

/\*

1,尽可能在setInterval中自己清除（停止）。（设置判断）

2、如果将setInterval赋给一个全局变量，

在其他地方ClearInterval进行清除有可能达不到效果。

(原因：那边刚清除，这边可能又重新赋值了，惨痛的经历！）

var timesRun = 0;

var interval = setInterval(function(){

timesRun += 1;

if(timesRun === 2){

clearInterval(interval);

}

document.writeln("hello");

//do whatever here..

}, 2000);\*/

/\*function run(){

document.writeln("hello");

}

setTimeout("run()",2000);/!\*一次性定时器 取消和循环定时器一样\*!/

\*/

/\*var a=location.href;

document.writeln(a);/!\*输出当前地址\*!/

location.href="study24.html";/!\*跳转对应地址,设置href属性，浏览器就会跳转到对应的路径。\*!/

\*/

/\*function a{

location.href="https://www.bilibili.com/";

}

setTimeout(a,2000);/!\*延迟跳转\*!/\*/