

知识点1【Date对象】（了解）

1、setTimeout进行定时

2、setInterval设置定时

3、setTimeout和setInterval的区别

知识点2【Math 对象】（了解）

知识点3【string对象】（了解）

知识点4【全局函数】（了解）

知识点1【Date对象】（了解）

(1) Date 对象方法

getFullYear(): 返回当前年份

getMonth(): 返回当前月份, 0~11

getDay(): 返回星期中的某一天, 0~6,0 表示周日

getDate(): 返回一月中的某一天

getHours(): 返回当前时间的小时, 0~23

getMinutes(): 返回当前时间的分钟, 0~59

getSeconds(): 返回当前时间的秒, 0~59

1、setTimeout进行定时

index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Document</title>
6     <script type="text/javascript" src="08_js.js"></script>
7 </head>
8 <body onload="start_timer()">
9     <div align="center">
10         <input type="text" id="time">
11         <br>
12         <input type="button" value="开启计时" onclick="start_timer()">
```

[illegible]

08_js.js

```
1
2 var t;
3
4 function myTimeOut() {
5
6     var d = new Date();
7     var h = d.getHours();
8     var m = d.getMinutes();
9     var s = d.getSeconds();
10    document.getElementById("time").value = h + ":" + m + ":" + s;
11    //设置超时时间
12    t = setTimeout("myTimeOut()", 1000);
13 }
14
15 function start_timer() {
16     myTimeOut();
17 }
18 function stop_timer() {
19     clearTimeout(t);
20 }
```

2、setInterval设置定时

index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Document</title>
6     <script type="text/javascript" src="08_js.js"></script>
7 </head>
8 <body onload="start_timer()">
9     <div align="center">
10         <input type="text" id="time">
11         <br>
12         <input type="button" value="开启计时" onclick="start_timer()">
```

[illegible]

08_js.js

```
1
2 var t;
3
4 function myTimeOut() {
5
6     var d = new Date();
7     var h = d.getHours();
8     var m = d.getMinutes();
9     var s = d.getSeconds();
10    document.getElementById("time").value = h + ":" + m + ":" + s;
11 }
12
13 function start_timer() {
14     t = setInterval("myTimeOut()", 1000);
15 }
16 function stop_timer() {
17     clearTimeout(t);
18 }
```

3、setTimeout和setInterval的区别

setTimeout: 设置一次只能调用一次

setInterval: 设置一次多次调用

知识点2 【Math 对象】 (了解)

使用 Math 的属性和方法的语法：

```
var pi_value=Math.PI; var sqrt_value=Math.sqrt(15);
```

Math 对象属性

属性	描述
E	返回算术常量 e，即自然对数的底数（约等于2.718）。
LN2	返回 2 的自然对数（约等于0.693）。
LN10	返回 10 的自然对数（约等于2.302）。
LOG2E	返回以 2 为底的 e 的对数（约等于 1.414）。
LOG10E	返回以 10 为底的 e 的对数（约等于0.434）。
PI	返回圆周率（约等于3.14159）。
SQRT1_2	返回返回 2 的平方根的倒数（约等于 0.707）。
SQRT2	返回 2 的平方根（约等于 1.414）。

方法	描述
abs(x)	返回数的绝对值。
acos(x)	返回数的反余弦值。
asin(x)	返回数的反正弦值。
atan(x)	以介于 -PI/2 与 PI/2 弧度之间的数值来返回 x 的反正切值。
atan2(y,x)	返回从 x 轴到点 (x,y) 的角度（介于 -PI/2 与 PI/2 弧度之间）。
ceil(x)	对数进行上舍入。
cos(x)	返回数的余弦。
exp(x)	返回 e 的指数。
floor(x)	对数进行下舍入。
log(x)	返回数的自然对数（底为e）。
max(x,y)	返回 x 和 y 中的最高值。
min(x,y)	返回 x 和 y 中的最低值。
pow(x,y)	返回 x 的 y 次幂。
random()	返回 0 ~ 1 之间的随机数。
round(x)	把数四舍五入为最接近的整数。
sin(x)	返回数的正弦。
sqrt(x)	返回数的平方根。
tan(x)	返回角的正切。
toSource()	返回该对象的源代码。
valueOf()	返回 Math 对象的原始值。

知识点3 【string对象】（了解）

String 对象属性

FF: Firefox, IE: Internet Explorer

属性	描述	FF	IE
constructor	对创建该对象的函数的引用	1	4
<u>length</u>	字符串的长度	1	3
prototype	允许您向对象添加属性和方法	1	4

方法	描述	FF	IE
anchor()	创建 HTML 锚。	1	3
big()	用大号字体显示字符串。	1	3
blink()	显示闪烁字符串。	1	
bold()	使用粗体显示字符串。	1	3
charAt()	返回在指定位置的字符。	1	3
charCodeAt()	返回在指定的位置的字符的 Unicode 编码。	1	4
concat()	连接字符串。	1	4
fixed()	以打字机文本显示字符串。	1	3
fontcolor()	使用指定的颜色来显示字符串。	1	3
fontSize()	使用指定的尺寸来显示字符串。	1	3
fromCharCode()	从字符编码创建一个字符串。	1	4
indexOf()	检索字符串。	1	3
italics()	使用斜体显示字符串。	1	3
lastIndexOf()	从后向前检索字符串。	1	3
link()	将字符串显示为链接。	1	3
localeCompare()	用本地特定的顺序来比较两个字符串。	1	4
match()	找到一个或多个正在表达式的匹配。	1	4
replace()	替换与正则表达式匹配的子串。	1	4
search()	检索与正则表达式相匹配的值。	1	4
slice()	提取字符串的片断，并在新的字符串中返回被提取的部分。	1	4
small()	使用小字号来显示字符串。	1	3
split()	把字符串分割为字符串数组。	1	4
strike()	使用删除线来显示字符串。	1	3
sub()	把字符串显示为下标。	1	3
substr()	从起始索引 号提取字符串中指定数目的字符。	1	4
substring()	提取字符串中两个指定的索引号之间的字符。	1	3
sup()	把字符串显示为上标。	1	3
toLocaleLowerCase()	把字符串转换为小写。	-	-
toLocaleUpperCase()	把字符串转换为大写。	-	-
toLowerCase()	把字符串转换为小写。	1	3
toUpperCase()	把字符串转换为大写。	1	3
toSource()	代表对象的源代码。	1	-
toString()	返回字符串。	-	-
valueOf()	返回某个字符串对象的原始值。	1	4

知识点4【全局函数】（了解）

函数	描述	FF	IE
<u>decodeURI()</u>	解码某个编码的 URI。	1	5.5
<u>decodeURIComponent()</u>	解码一个编码的 URI 组件。	1	5.5
<u>encodeURI()</u>	把字符串编码为 URI。	1	5.5
<u>encodeURIComponent()</u>	把字符串编码为 URI 组件。	1	5.5
<u>escape()</u>	对字符串进行编码。	1	3
<u>eval()</u>	计算 JavaScript 字符串，并把它作为脚本代码来执行。	1	3
<u>getClass()</u>	返回一个 JavaObject 的 JavaClass。		
<u>isFinite()</u>	检查某个值是否为有穷大的数。	1	4
<u>isNaN()</u>	检查某个值是否是数字。	1	3
<u>Number()</u>	把对象的值转换为数字。	1	
<u>parseFloat()</u>	解析一个字符串并返回一个浮点数。	1	3
<u>parseInt()</u>	解析一个字符串并返回一个整数。	1	3
<u>String()</u>	把对象的值转换为字符串。	1	
<u>unescape()</u>	对由 <code>escape()</code> 编码的字符串进行解码。	1	3

index.html

```

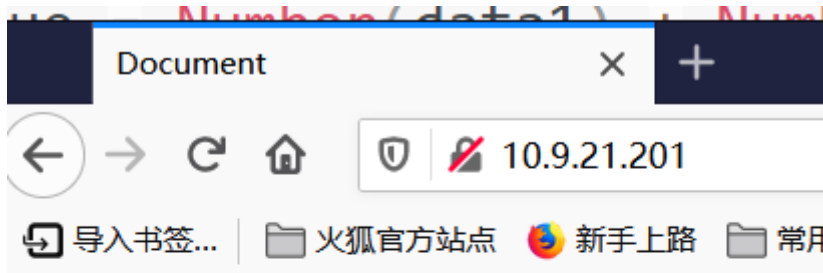
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <script type="text/javascript" src="09_js.js"></script>
7      <title>Document</title>
8  </head>
9
10 <body>
11     Data1:<input type="text" id="data1">
12     <br>
13     Data2:<input type="text" id="data2">
14     <br>
15     结果:<input type="text" id="res">
16     <br>
17     <input type="button" value="相加" onclick="my_calc(1)">
18     <input type="button" value="相减" onclick="my_calc(0)">
19 </body>
20

```

```
21 </html>
```

09_js.js

```
1 function my_calc(arg) {  
2  
3     //获取data1 data2的数据  
4     var data1 = document.getElementById("data1").value;  
5     var data2 = document.getElementById("data2").value;  
6  
7     //判断data1 data2必须是数值  
8     if (isNaN(data1) || isNaN(data2)) {  
9         alert("请输入数值");  
10        document.getElementById("data1").value = "";  
11        document.getElementById("data2").value = "";  
12    }  
13    else {  
14        if (arg == 1)//相加  
15        {  
16            document.getElementById("res").value = Number(data1) + Number(data2);  
17        }  
18        else if (arg == 0)//相减  
19        {  
20            document.getElementById("res").value = Number(data1) - Number(data2);  
21        }  
22    }  
23 }
```

Data1:

Data2:

结果: