

```
sv<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
```

BOM简称浏览器对象模型 browser object model

window对象:

所有浏览器都支持window对象 它表示浏览器的窗口

所有的javascript全局对象 函数以及变量均自动成为 window对象的成员

全局变量是window对象的属性

全局函数是window对象的方法<br/>

```
<!--
```

```
window.open() #打开新窗口
```

```
window.close() #关闭当前窗口
```

```
-->
```

```
<button onclick="window.open('home')">打开新窗口</button><br/>
```

```
<button onclick="window.close()">关闭当前窗口</button><br/>
```

```
<!--
```

window.location对象用于获取当前页面的地址URL，并把浏览器重定向到新页面

location.hostname 返回web主机的域名

location.pathname 返回当前页面的路径和文件名

location.port 返回web主机的端口

location.protocol 返回所使用的web协议 http https

location.href 返回当前页面的url

```
-->
```

```
<script>
```

```
  document.write(location.hostname + "<br/>");
```

```
  document.write(location.pathname+ "<br/>");
```

```
  document.write(location.port+ "<br/>");
```

```
  document.write(location.protocol+ "<br/>");
```

```
  document.write(location.href+ "<br/>")
```

```
</script>
```

<!--

window.history对象包含浏览器的历史

history.back() 与浏览器点击后退按钮相同 返回历史列表中的前一个URL

history.forward() 与浏览器中点击向前按钮相同 返回历史浏览列表中的下一个URL

-->

<button onclick="window.history.back()">后</button><br/>

<button onclick="window.history.forward()">前</button><br/>

<!--

window.navigator 对象包含有关访问者浏览器的信息

注意:

来自navigator对象的信息具有误导性 不应该被用于检测用户浏览器信息

因为 navigator数据可被浏览器使用者更改

-->

<script>

document.write(navigator.appCodeName);

document.write(navigator.appName);

document.write(navigator.appVersion);

document.write(navigator.cookieEnabled);

document.write(navigator.platform);

document.write(navigator.userAgent);

</script>

<!--

javaScript弹窗

可以在javascript中创建三种的消息框 警告 确认 提示

-->

<script>

//警告框 经常用于确保用户可以得到某些信息

//window.alert("123")

//确认框 用户验证是否接受用户操作 当确认卡弹出的时候 点击确认或取消来确定

//用户操作 当点击确认时候 返回true 点击取消返回false

```
var r = window.confirm("确定吗? ");
```

```
if (r == true){  
    document.write("确认!")  
}else{  
    document.write("取消")  
}
```

//提示框

// 提示框经常用于提示用户在进入页面输入某个值 当提示框出现后 用户需要输入

// 某个值 然后点击确认或取消才能继续操作， 如果用户点击确认 返回值为输入的

// 值 如果点击取消 那么返回值为 null

```
var p = window.prompt("请输入你的名字");
```

```
if (p != null){  
    document.write("你输入的是: " + p)  
}else{  
    document.write("啥都没输入")  
}
```

</script>

javascript计时器

通过JS 可以做到在一个设定的时间间隔之后执行代码 而不是在函数被调用后立即执行，

两个关键的计时器方法

setInterval() 间隔指定的毫秒数不停的执行指定代码

setTimeout () 在指定的毫秒数后执行一次代码

```
<script>
```

```
    // window.setTimeout(function(){alert ("timetout")}, 5000);  
    // window.setInterval(function(){alert("hello!!")}, 1000)
```

```
</script>
```

javascript cookie

Cookie用于存储WEB页面的用户信息, 用于记录一些信息, 用于存储在访问网站时候产生的一些数据, 如保持用户登录状态等

cookie 以键值对的形式存储 如下 username=liushuo

```
<script>
```

```
    //使用javascript创建cookie
```

```
    document.cookie = "name=liushuo";  
    document.cookie = "age=20";
```

```
    //获取cookie
```

```
    var z = document.cookie;  
    document.write(z)
```

```
</script>
```

```
</body>
```

```
</html>
```