# JavaScript

# 上机实训指导手册

## 上机实训1：打开和关闭页面，将弹出不同的窗口信息。

### 【上机目的】

1．掌握页面加载时的执行函数；

2．掌握页面关闭时的执行函数。

### 【上机要求】

本例是一个简单的JavaScript程序，主要用来说明如何编写JavaScript程序以及在HTML中如何使用。本例主要实现的功能为：当页面打开时，显示“尊敬的客户，欢迎光临您光临本网站”窗口，关闭页面时弹出窗口“欢迎下次光临！”。

### 【上机指导】

步骤1：新建HTML文档，输入以下代码。

<!DOCTYPE html>

<html>

<head>

<title>第一个Javascript程序</title>

</head>

<body>

</body>

</html>

步骤2：保存HTML文件，选择相应的保存位置，文件名为welcome.html。

步骤3：在HTML文档的head部分，键入如下代码。

<script>

<script>

//页面加载时执行的函数

function showEnter(){

alert("尊敬的客户，欢迎光临您光临本网站");

}

//页面关闭时执行的函数

function showLeave(){

alert("欢迎下次光临！");

}

//页面加载事件触发时调用函数

window.onload=showEnter;

//页面关闭载事件触发时调用函数

window.onbeforeunload=showLeave;

</script>

步骤4：保存网页，浏览最终效果。程序效果，如图1-1和1-2所示。

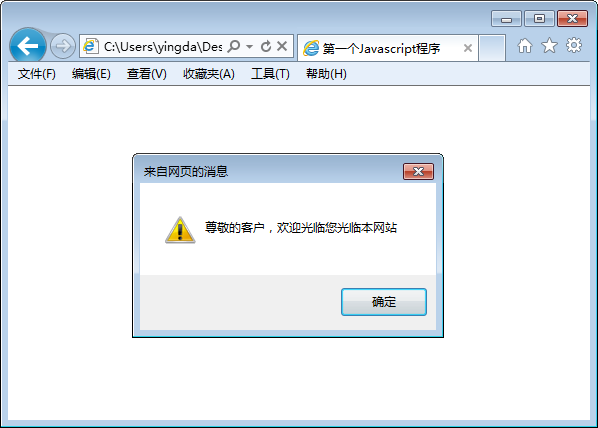
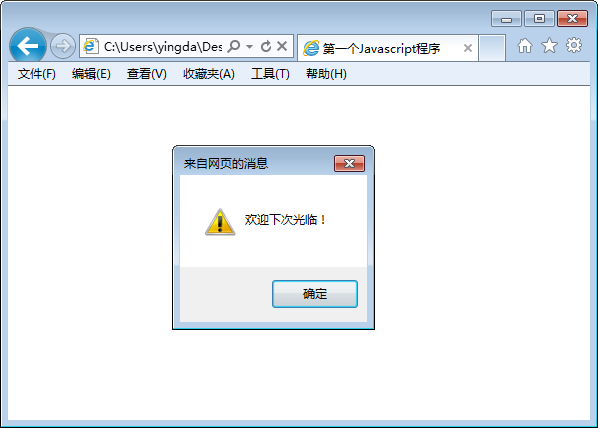
 

图1-1 页面加载时效果 图1-2 页面关闭时效果

## 上机实训2：调用外部JavaScript文件

### 【上机目的】

1．掌握调用外部JavaScript文件的方法；

2．理解在代码中添加JavaScript代码和调用外部JavaScript文件的区别。

### 【上机要求】

调用JavaScript文件，输出内容为“hello”。在HTML文件中调用该脚本文件，其调用方法如下所示。

<script src=外部脚本文件路径>

</script>

### 【上机指导】

步骤1：新建JavaScript文件

新建文件为hello.js，并在文件中键入如下代码。

// JavaScript Document

//向页面输入问候语

document.write("hello");

步骤2：新建HTML文件

按照以前的方法创建HTML文件，并保存。注意，为了能够保证示例的正常运行，请将该文件与hello.js保存于同一位置处。在HTML文件中，键入加粗部分所示代码。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title> JavaScript嵌入HTML5文件</title>

<script src=”hello.js”>

</script>

</head>

<body>

</body>

</html>

程序的运行结果，如图1-3所示。

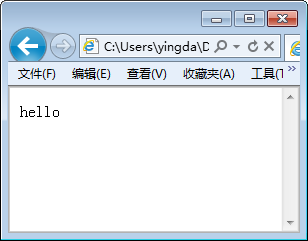


图1-3 页面关闭时效果

## 上机实训3：理解变量的作用范围

### 【上机目的】

1．掌握全局变量的含义；

2．理解局部变量的含义。

### 【上机要求】

创建了名为carname的变量，并向其赋值“Volvo”，然后把它放入id="demo"的HTML段落中。

### 【上机指导】

新建网页文件，代码如下：

<!DOCTYPE html>

<html>

<body>

<p>点击这里来创建变量，并显示结果。</p>

<button onclick="myFunction()">点击这里</button>

<p id="demo"></p>

<script type="text/javascript">

function myFunction()

{

var carname="Volvo";

document.getElementById("demo").innerHTML=carname;

}

</script>

</body>

</html>

浏览效果如图1-4所示。单击其中的【点击这里】按钮，可以看到两个元素发生了变化。如图1-5所示。

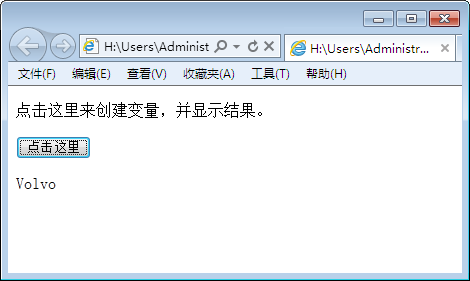
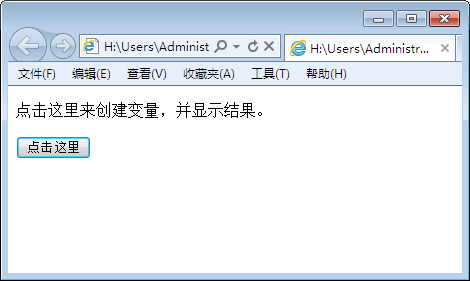


图1-4 程序运行结果 图1-5 程序运行结果

## 上机实训4：使用typeof 运算符

### 【上机目的】

1．掌握typeof 运算符的使用方法；

2．理解typeof运算符调用变量或值的返回值。

### 【上机要求】

使用typeof 运算符，返回常见的变量或值的返回值。

### 【上机指导】

新建网页文件，代码如下：

<!DOCTYPE html>

<html>

<body>

<script type="text/javascript">

typeof(1);

typeof(NaN);

typeof(Number.MIN\_VALUE);

typeof(Infinity);

typeof("123");

typeof(true);

typeof(window);

typeof(document);

typeof(null);

typeof(eval);

typeof(Date);

typeof(sss);

typeof(undefined);

document.write ("typeof(1): "+typeof(1)+"<br>");

document.write ("typeof(NaN): "+typeof(NaN)+"<br>");

document.write ("typeof(Number.MIN\_VALUE): "+typeof(Number.MIN\_VALUE)+"<br>")

document.write ("typeof(Infinity): "+typeof(Infinity)+"<br>")

document.write ("typeof(\"123\"): "+typeof("123")+"<br>")

document.write ("typeof(true): "+typeof(true)+"<br>")

document.write ("typeof(window): "+typeof(window)+"<br>")

document.write ("typeof(document): "+typeof(document)+"<br>")

document.write ("typeof(null): "+typeof(null)+"<br>")

document.write ("typeof(eval): "+typeof(eval)+"<br>")

document.write ("typeof(Date): "+typeof(Date)+"<br>")

document.write ("typeof(sss): "+typeof(sss)+"<br>")

document.write ("typeof(undefined): "+typeof(undefined)+"<br>")

</script>

</body>

</html>

浏览效果如图1-6所示。

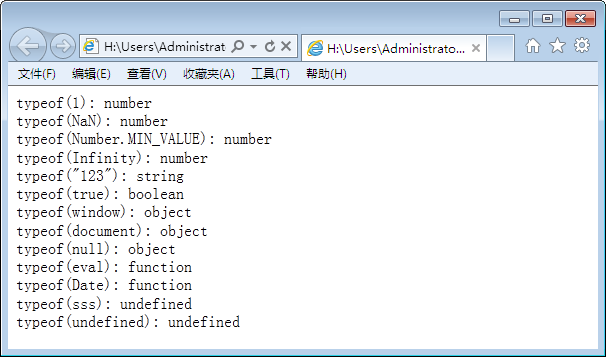


图1-6 程序运行结果

## 上机实训5：局部变量和全局变量的优先级

### 【上机目的】

1．理解局部变量的优先级；

2．理解全局变量的优先级。

### 【上机要求】

创建一个网页，分析局部变量和全局变量的优先级

### 【上机指导】

新建网页文件，代码如下：

<!DOCTYPE html>

<html>

<head>

<title>变量的优先级</title>

<body>

<script language="javascript">

<!--

var scope="全局变量";//声明一个全局变量

function checkscope()

{

var scope="局部变量";//声明一个同名的局部变量

document.write(scope);//使用的是局部变量，而不是全局变量

}

checkscope();//调用函数，输出结果

//-->

</script>

</body>

</head>

</html>

浏览效果如图1-7所示。

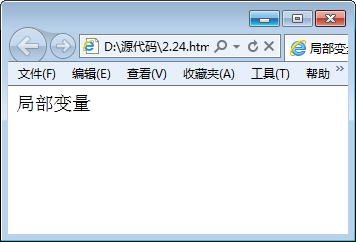


图1-7 程序运行结果

JavaScript没有块级作用域，函数中的所有变量无论是在哪里声明的，在整个函数中都有意义。

javascript 无块级作用域，代码如下：

<!DOCTYPE html>

<html>

<head>

<title>变量的优先级</title>

<body>

<script language="javascript">

<!--

var scope="全局变量";//声明一个全局变量

function checkscope()

{

alert(scope); //调用局部变量，将显示“undefined”而不是“局部变量

var scope="局部变量";//声明一个同名的局部变量

alert(scope);//使用的是局部变量，将显示“局部变量”

}

checkscope();//调用函数，输出结果

//-->

</script>

</body>

</head>

</html>

程序运行结果如图1-8所示。单击【确定】按钮，弹出结果如图1-9所示。

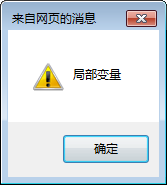
 

图1-8 程序运行结果 图1-9 程序运行结果

## 上机实训6：综合应用if…else语句

### 【上机目的】

1．掌握if…else语句的使用方法；

2．理解计算时间差的方法

### 【上机要求】

本实例就通过JavaScript实现在页面中显示距离2019年元旦的天数。

### 【上机指导】

步骤1：定义JavaScript的函数，实现判断系统当前时间与2019年元旦相距的天数，代码如下。

function countdown(title,Intime,divId){

var online= new Date(Intime); //根据参数定义时间对象

var now = new Date(); //定义当前系统时间对象

var leave = online.getTime() - now.getTime(); //计算时间差

var day = Math.floor(leave / (1000 \* 60 \* 60 \* 24))+1;

if (day > 1){

if(document.all){

divId.innerHTML="<b>——距"+ title+"还有"+day +"天！</b>"; //页面显示信息

}

}else{

if (day == 1) {

if(document.all){

divId.innerHTML="<b>——明天就是"+title+"啦!</b>";

}

}else{

if (day == 0) {divId.innerHTML="<b>今天就是"+title+"呀！</b>";

}else{

if(document.all){

divId.innerHTML="<b>——唉呀！"+title+"已经过了！</b>";

}

}

}

}

}

步骤2：在页面中定义相关表格，用于显示当前时间距离2019年元旦的天数。

<table width="350" height="450" border="0" align="center" cellpadding="0" cellspacing="0">

<tr>

<td valign="bottom" ><table width="346" height="418" border="0" cellpadding="0" cellspacing="0">

<tr>

<td width="76"> </td>

<td width="270">

<div id="countDown">

<b>——</b></div>

<script language="javascript">

countdown("2019年元旦","1/1/2019",countDown); <!--调用JavaScript函数-->

</script>

</td>

</tr>

</table></td>

</tr>

</table>

步骤3：运行相关程序，即可得出最终的效果。如图1-10所示。

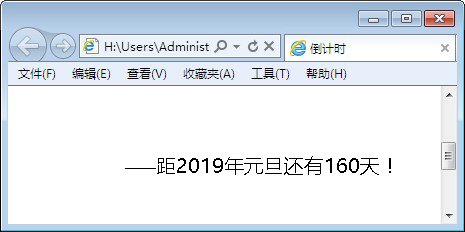


图1-10 程序运行结果