新起点电脑教程

第 4 章

JavaScript 脚本语言

本章要点

- JavaScript 简介
- 数据类型
- 表达式和运算符
- JavaScript 循环语句
- JavaScript 函数

本章主要内容

JavaScript 是一种脚本技术，网页通过脚本程序可以实现数据的传输和动态交互。本章将简要介绍 JavaScript 技术的基本知识，包括基本语法和常用特效技术，并通过演示实例来介绍各个知识点的使用方法，为读者步入本书后面知识的学习奠定基础。

4.1 JavaScript 简介

JavaScript 是一种基于对象(Object)和事件驱动(Event Driven)并具有安全性能的脚本语言。其目的是与 HTML 超文本标记语言、Java 脚本语言(Java 小程序)相互结合，实现 Web 页面中链接多个对象，并与 Web 客户交互的效果，从而实现客户端应用程序的开发。

↑扫码看视频

4.1.1 运行环境

运行 JavaScript 所需的最低软件环境如下。

- Windows 95/98 或 Windows NT。
- Netscape Navigator x.0 或 Internet Explorer x.0。
- 用于编辑 HTML 文档的字符编辑器(WS、WPS、Notepad、WordPad 等)或 HTML 文档编辑器。

运行 JavaScript 所需的最低硬件环境如下。

- 基本内存 32MB。
- CRT 至少需要 256 种颜色，分辨率在 640 像素×480 像素以上。

4.1.2 JavaScript 的格式

使用 JavaScript 的语法格式如下。

```
<Script Language ="JavaScript">
JavaScript 脚本代码 1
JavaScript 脚本代码 2
……
</Script>
```

4.1.3 一个典型的 JavaScript 文件

下面通过一个具体实例来认识 JavaScript 文件的基本结构。

实例 4-1：演示 JavaScript 在页面中的作用。
源文件路径：daima\4\4-1

实例文件 1.html 的具体实现代码如下。

```
<html>
<head>
```

```
<Script Language ="JavaScript">
// JavaScript 开始
alert("这是第一个 JavaScript 例子!");              //提示语句
alert("欢迎你进入 JavaScript 世界!");              //提示语句
alert("今后我们将共同学习 JavaScript 知识! ");     //提示语句
</Script>
</Head>
</Html>
```

在上述实例代码中，<Script Language="JavaScript"></Script>之间的部分是 JavaScript 脚本语句。执行后的效果如图 4-1 所示。

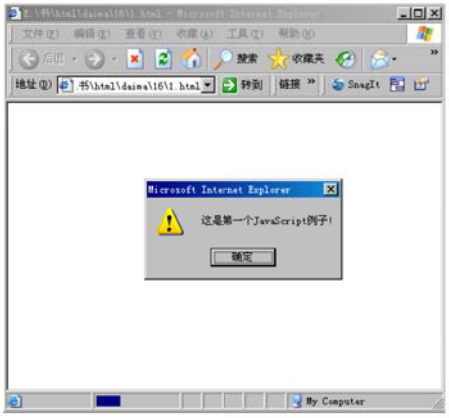


图 4-1　执行效果

知识精讲

上述实例文件是 HTML 文档，为标准的 HTML 格式。而在现实应用中，JavaScript 脚本程序将被专门编写并保存为.js 格式文件。当 Web 页面需要这些脚本程序时，只需通过<script src="文件名"></script>调用即可。

4.2　数 据 类 型

JavaScript 脚本语言同其他语言一样，有其自身的基本数据类型、表达式和算术运算符以及程序的基本框架结构。JavaScript 通过数据类型来处理数字和文字，通过变量提供存放信息的地方，通过表达式完成较复杂的信息处理。本节将简要介绍 JavaScript 数据类型的基本知识，为读者步入本书后面知识的学习奠定基础。

↑扫码看视频

4.2.1 数据类型概述

在 JavaScript 中有如下 4 种基本数据类型。

- 数值类型：包括常用的整数和实数。
- 字符串类型：用双引号或单引号括起来的字符或数值。
- 布尔型：使用 True 或 False 表示的值。
- 空值。

在 JavaScript 的基本类型中的数据可以是常量，也可以是变量。由于 JavaScript 采用弱类型的形式，因而一个数据的变量或常量不必首先作声明，而是在使用或赋值时确定其数据类型，当然也可以先声明数据的类型。

4.2.2 JavaScript 常量

常量是一种固定不变的数据类型，程序中的常量一旦定义数值，则一直保持固定的数值直至程序模块结束。JavaScript 中主要包括如下类型的常量。

1. 整型常量

JavaScript 的整型常量通常又称为字面常量，它是不能改变的数据。其整型常量可以使用十六进制、八进制和十进制数值来表示。

2. 实型常量

实型常量由整数部分和小数部分共同表示，如 12.32 和 193.98 等。也可以使用科学或标准方法表示，如 5E7 和 4e5 等。

3. 布尔常量

布尔常量只有 True 或 False 两种状态，其主要功能是用来说明或代表一种对象的状态或标志，以说明操作流程。

智慧锦囊

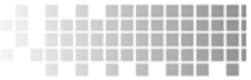
JavaScript 中的布尔常量与 C++中的不同。C++可以用 1 或 0 表示其状态，而 JavaScript 只能用 True 或 False 来表示其状态。

4. 字符型常量

字符型常量即使用单引号(')或双引号(")括起来的一个或几个字符。例如，"This is a book of JavaScript"、"3245"和"ewrt234234"等。

5. 空值

JavaScript 中只有一个空值 null，表示什么也没有。如果试图引用没有定义的变量，则将返回一个 null 值。



6．特殊字符

同C语言一样，JavaScript 中同样有以反斜杠符号(/)开头的不可显示的特殊字符。通常称为控制字符，可以作为脚本代码的注释。

4.2.3　JavaScript 变量

变量是一种可以变化的数据类型，变量一经定义可以根据程序的需要而代表不同的数值。变量的主要功能是存取数据和提供存放信息的容器。在使用变量时必须明确变量的命名、类型、声明和其作用域。在下面的内容中，将对这些知识进行简要介绍。

1．变量的命名

JavaScript 中的变量命名与其他编程语言相比，主要应遵循如下两点。

- 必须是一个有效的变量，即变量以字母开头，后面可以出现数字如 test1、text2 等。除下划线(_)作为连字符外，变量名称不能有空格、+、-、,等其他符号。
- 不能使用 JavaScript 中的关键字作为变量。

在 JavaScript 中定义了 40 多个关键字，这些关键字不能作为变量的名称，而只能在 JavaScript 的内部使用。例如，var、int、double、true 不能作为变量的名称。同时在对变量命名时，最好把变量的意义与其代表的意思对应起来，以免出现错误。

2．变量的类型

JavaScript 中的变量声明通常使用命令 var，使用此命令的语法格式如下。

```
var 变量名="变量值";
```

在上述格式中定义了一个变量名，并同时赋予了变量的值。

在 JavaScript 中，变量可以不作声明，在使用时再根据数据的类型来确定变量的类型。例如下面的一段代码。

```
x=100
y="125"
xy= True
cost=19.5
```

其中，x 为整数，y 为字符串，xy 为布尔型，cost 为实型。

3．变量的声明和作用域

JavaScript 的变量可以在使用前先作声明并赋值。JavaScript 是采用动态编译的，而动态编译的缺点是不易发现代码中的错误，特别是在变量命名方面。而对变量进行声明后，可以及时发现代码中的错误。

在 JavaScript 中有全局变量和局部变量。全局变量定义在所有函数体之外，其作用范围是整个函数；而局部变量定义在函数体之内，只对该函数是可见的，而对其他函数则是不可见的。

4.3 表达式和运算符

在 JavaScript 应用程序中，通常使用表达式和运算符来实现对数据的处理。本节将简要介绍 JavaScript 表达式和运算符的基本知识，并通过具体实例来实现。

↑扫码看视频

4.3.1 JavaScript 表达式

定义完变量后，就可以对其进行赋值、改变和计算等一系列处理，而这些过程通常由表达式来完成。由上述描述可以看出，JavaScript 表达式是变量、常量、布尔和运算符的集合。所以，表达式可以分为算术表述式、字符串表达式、赋值表达式以及布尔表达式等。

4.3.2 JavaScript 运算符

运算符是能够完成某种操作的一系列符号。在 JavaScript 中常用的运算符有如下几种：算术运算符、比较运算符和布尔逻辑运算符。

JavaScript 中的运算符使用方式有双目运算符和单目运算符两种。其中，双目运算符具体使用的语法格式如下。

```
操作数 1 运算符 操作数 2
```

由上述格式可以看出，双目运算符由两个操作数和一个运算符组成。例如，50+40 和"This"+"that"等。而单目运算符只需一个操作数，并且其运算符可在前或在后。

1. 算术运算符

JavaScript 中的算术运算符有单目运算符和双目运算符两种。JavaScript 中常用的双目运算符如表 4-1 所示。

表 4-1 常用的双目运算符列表

元 素	描 述	元 素	描 述
+	表示加	-	表示减
*	表示乘	/	表示除
|	表示按位或	&	表示按位与
<<	表示左移	>>	表示右移
>>>	表示零填充	%	表示取模

JavaScript 中常用的单目运算符如表 4-2 所示。

表 4-2　常用的单目运算符列表

元　素	描　述	元　素	描　述
-	表示取反	~	表示取补
++	表示递加 1	--	表示递减 1

2．比较运算符

JavaScript 中比较运算符的基本操作过程是，首先对它的操作对象进行比较，然后返回一个 true 或 false 值来表示比较结果。

JavaScript 中常用的比较运算符如表 4-3 所示。

表 4-3　常用的比较运算符列表

元　素	描　述	元　素	描　述
<	表示小于	>	表示大于
<=	表示小于等于	>=	表示大于等于
=	表示等于	!=	表示不等于

3．布尔逻辑运算符

JavaScript 中常用的布尔逻辑运算符如表 4-4 所示。

表 4-4　常用的布尔逻辑运算符列表

元　素	描　述	元　素	描　述
!	表示取反	&=	表示取与之后赋值
&	表示逻辑与	|=	表示取或之后赋值
|	表示逻辑或	^=	表示取异或之后赋值
^	表示逻辑异或	?:	表示三目操作符
||	表示或	==	表示等于
|=	表示不等于		

使用三目操作符的语法格式如下。

```
操作数? 结果 1：结果 2
```

如果操作数的结果为真，则表述式的结果为“结果 1”，否则为“结果 2”。

实例 4-2：用 JavaScript 在页面中实现跑马灯效果。

源文件路径：daima\4\4-2

文件 2.html 是一个测试页面，功能是调用文件 2.js 中定义的特殊效果，其主要代码如下。

```
<html xmlns="http://www.w3.org/1999/xhtml">
……
<style type="text/css">
<!--
body {
    background-color: #666666;                    /* 设置页面背景颜色 */
}
-->
</style>
<Script src ="2.js"></Script>                     <!--调用脚本程序-->
</head>
<body>
</body>
</html>
```

文件 2.js 的功能是定义页面的特效样式，实现页面跑马灯显示的效果。其主要实现代码如下。

```
var msg="这是一个跑马灯效果的 JavaScript 文档";
var interval = 100;                               //开始定义变量
var spacelen = 120;
var space10=" ";
var seq=0;
function Scroll() {                               //定义一个滚动函数
len = msg.length;                                 //提示语句长度
window.status = msg.substring(0, seq+1);          //窗体状态
seq++;
if ( seq >= len ) {
seq = spacelen;
window.setTimeout("Scroll2();", interval );       //根据长度设置时间
}
else
window.setTimeout("Scroll();", interval );        //根据长度设置时间
}
function Scroll2() {
var out="";
for (i=1; i<=spacelen/space10.length; i++) out +=
space10;
out = out + msg;                                  //输出设置
len=out.length;                                   //获取长度
window.status=out.substring(seq, len);
seq++;
if ( seq >= len ) { seq = 0; };
window.setTimeout("Scroll2();", interval );       //设置时间
}
Scroll();
```

上述代码通过设置变量和函数，并结合窗体和载入提示语句的长度实现了跑马灯效果。执行效果如图 4-2 所示。

图 4-2　执行效果

4.4　JavaScript 循环语句

JavaScript 程序是由若干语句组成的，循环语句是编写程序的指令。JavaScript 提供了完整的基本编程语句，本节将简要介绍常用的 JavaScript 循环语句知识，为读者步入本书后面知识的学习奠定基础。

↑扫码看视频

4.4.1　if 条件语句

if 条件语句的功能是，根据系统用户的输入值作出不同的反应提示。例如，可以编写一段特定程序实现对不同输入文本的反应。使用 if 条件语句的语法格式如下。

```
if(表述式)
语句段 1;
…
else
语句段 2;
…
```

if...else 语句是 JavaScript 中最基本的控制语句，通过它可以改变语句的执行顺序。在其表达式中必须使用关系语句来实现判断，并且是作为一个布尔值来估算的。若 if 后的语句有多行，则必须使用花括号将其括起来。

另外，通过 if 条件语句可以实现条件的嵌套处理。使用嵌套 if 语句的语法格式如下。

```
if(布尔值)语句 1;
else(布尔值)语句 2;
else if(布尔值)语句 3;
…
else 语句 4;
```

在上述格式中，每一级的布尔表达式都会被计算。若为真，则执行其相应的语句；若为假，则执行 else 后的语句。

实例 4-3：根据用户输入的字符显示提示。
源文件路径：daima\4\4-3

实例文件 3.html 的主要代码如下。

```
<html xmlns="http://www.w3.org/1999/xhtml">
……
<style type="text/css">
<!--
body {
    background-color: #666666;
}
-->
</style>
<Script Language ="JavaScript">
var monkey_love = prompt("你喜欢吗？","敲入是或否。");      //判断语句
if (monkey_love == "是") then                              //如果值为是
{
alert("谢谢！很高兴您能来这儿！请往下读吧！");             //根据长度设置时间
}
</Script>
</head>
<body>
</body>
</html>
```

执行上述实例代码后，首先显示一个提问语句，当输入字符“是”并单击“确定”按钮后将显示 alert 中的提示。执行效果如图 4-3 所示。

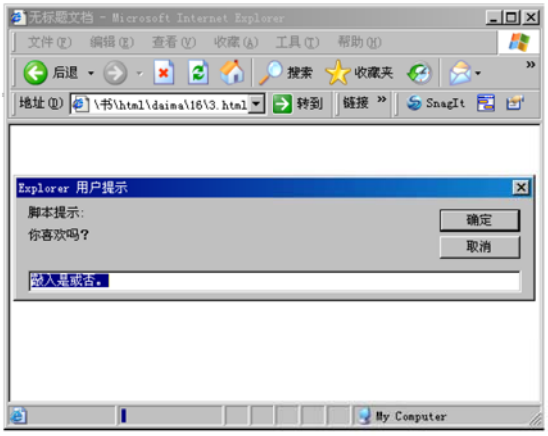


图 4-3　执行效果

4.4.2 for 循环语句

for 循环语句的功能是实现条件循环，当条件成立时执行特定语句集，否则将跳出循环。使用 for 循环语句的语法格式如下。

```
for(初始化;条件;增量)
语句集;
```

其中，“条件”是用于判别循环停止时的条件。若条件满足，则执行循环体，否则将跳出。“增量”用来定义循环控制变量在每次循环时按什么方式变化。三个主要语句之间，必须使用分号分隔。

实例 4-4：根据定义变量的长度在页面中显示指定数量的文本。

源文件路径：daima\4\4-4

实例文件 5.html 的主要代码如下。

```
<script language="JavaScript">
var a_line="";                                          //初始变量为空
var width="100";                                        //定义变量大小
for (loop=0; loop < width; loop++)
{
 a_line = a_line + "看我的数量";                        //变量变化
}
</script>
</head>
<body>
<body>
<script language="JavaScript">
document.write (a_line);                                //输出变量
</script>
```

在上述代码中，首先定义变量 a_line 的初始值为空值，变量 width 的值为 100；然后使用 for 循环语句设置 loop=0 并持续加 1 直到 loop<width；最后在 a_line 上加 width 次文本“看我的数量”，并在页面上输出 width 个文本“看我的数量”。执行效果如图 4-4 所示。

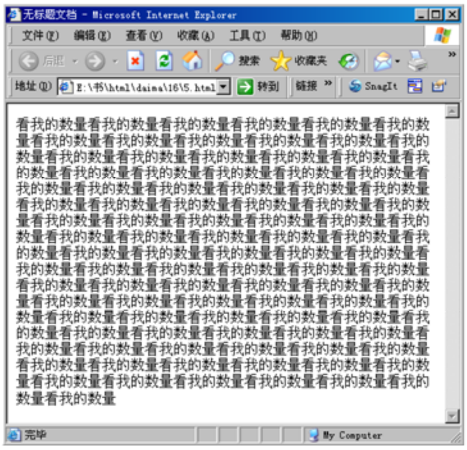


图 4-4 执行效果

4.4.3 while 循环语句

while 循环语句与 for 循环语句一样，当条件为真时则重复循环，否则将退出循环。使用 while 循环语句的语法格式如下。

```
while(条件)
语句集;
```

实例 4-5：根据用户输入的数值在页面中显示指定数量的文本。

源文件路径：daima\4\4-5

实例文件 6.html 的主要代码如下。

```
<script language="JavaScript">
var width = prompt("想显示几个 x 呀?","5");          //提示对话框
var a_line="";                                       //变量赋值
var loop=0;
while (loop < width)
{
 a_line = a_line + "看我的数量";                    //变量处理
 loop=loop+1;
}
</script>
</head>
<body>
<script language="JavaScript">
document.write (a_line);                             //变量显示
</script>
```

在上述代码中，首先定义了变量 a_line 的初始值为空值，变量 loop 值为空；然后，使用 while 循环语句设置当 loop 小于所请求的 width 时，在 a_line 上加入一次文本“看我的数量”，并在循环值上加 1；最后，在页面上输出请求个数的文本“看我的数量”。执行效果分别如图 4-5 和图 4-6 所示。

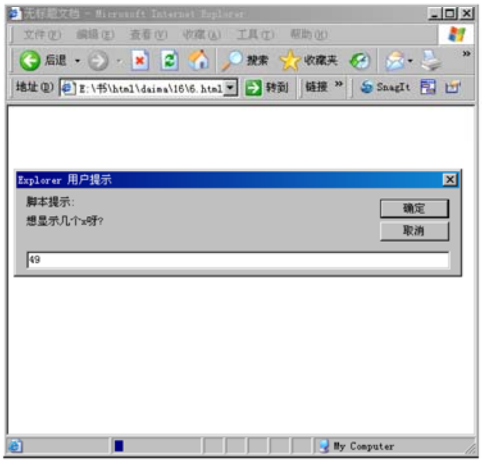


图 4-5　输入显示文本个数为 49

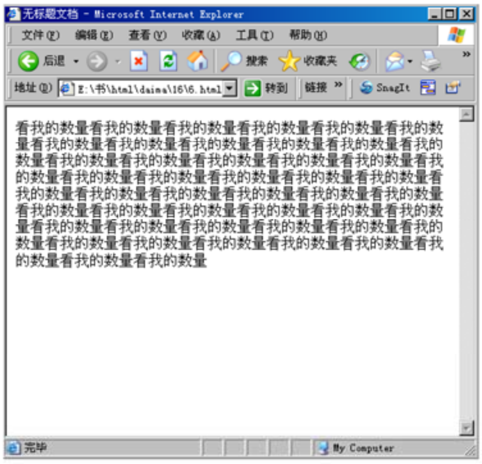


图 4-6　输出 49 个“看我的数量”文本

4.4.4　do…while 循环语句

在 JavaScript 中，do…while 的中文解释是“执行……当……继续执行”。在“执行(do)”后面跟随命令语句，在“当(while)”后面跟随一组判断表达式。如果判断表达式的结果为真，则执行后面的程序代码。

使用 do…while 循环语句的语法格式如下。

```
do {
<程序语句区>
}
while(<逻辑判断表达式>)
```

4.4.5　break 控制语句

break 控制语句的功能是终止循环结构的执行，通常将 break 放在循环语句的后面。使用 break 语句的语法格式如下。

```
循环语句
break
```

例如下面的一段代码。

```
<script>
a=new array(5,4,3,2,1);                    //数组初始值
sum=0                                      //变量初始值
for(i=0,i<a.length;++i)                    //小于数组长度则变量递增
{
if (i==3 ) break;                          //变量为 3 则停止
sum+=a[i]
}
</script>
```

在上述代码中，for 语句在 i 等于 0、1、2、3 时执行。当 i 等于 3 时，if 条件为真，执行 break 语句，使 for 语句立刻终止，停止执行后面的循环代码。

4.4.6　switch 循环语句

switch 的中文解释是“切换”，其功能是根据不同的变量值来执行对应的程序代码。如果判断表达式的结果为真，则执行后面程序代码。使用 switch 语句的语法格式如下。

```
switch(<变量>){
case<特定数值 1>:程序语句区;
break;
case<特定数值 2>:程序语句区;
break;
…
case<特定数值 n>:程序语句区;
break;
default         :程序语句区;
}
```

其中，default 语句是可以省略的。省略后，当所有的 case 都不符合条件时，便退出 switch 语句。

4.5 JavaScript 函数

函数为程序设计人员提供了一个功能强大的处理功能。通常在进行一个复杂的程序设计时，总是根据所要完成的功能，将程序划分为一些相对独立的部分，每部分编写一个函数。从而，使各部分充分独立，任务单一，程序清晰，易懂、易读、易维护。

↑扫码看视频

4.5.1 JavaScript 函数的构成

JavaScript 函数由如下 5 部分构成。

- 关键字：function。
- 函数或变量。
- 函数的参数：用小括号“( )”括起来，如果有多个，则用逗号“,”分开。
- 函数的内容：通常由一些表达式构成，外面用大括号“ { } ”括起来。
- 关键字：return。

其中，参数和 return 不是构成函数的必要条件。

实例 4-6：通过函数在页面上输出指定的文本。

源文件路径：daima\4\4-6

实例文件 9.html 的具体实现代码如下。

```
<html>
……
<style type="text/css">
<!--
body {
    background-color: #9966CC;
}
-->
</style>
</head>
<body>
<Script>
function showname(name)  {
  return "我叫"+name;
 }
document.write(showname("aaa"));
</Script>
</body>
</html>
```

在上述代码中定义了一个名为 showname 的函数，name 是函数的参数变量，然后通过 document 在页面上显示输出结果。执行效果如图 4-7 所示。

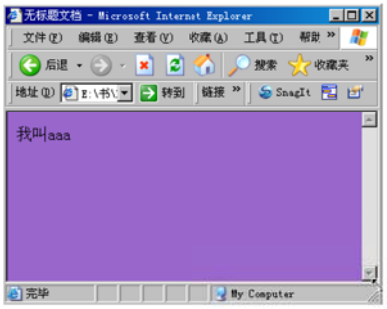


图 4-7　执行效果

4.5.2　JavaScript 常用函数

在 JavaScript 中有如下几类常用的函数。

- 编码函数：编码函数即函数 escape()，功能是将字符串中的非文字和数字字符转换成 ASCII 值。
- 译码函数：译码函数即函数 unescape()，和编码函数完全相反，功能是将 ASCII 字符转换成一般数字。
- 求值函数：求值函数 eval()有两个功能，一是进行字符串的运算处理，二是用来指出操作对象。
- 数值判断函数：数值判断函数 isNan()的功能是判断自变量参数是不是数值。
- 转整数函数：转整数函数 parseInt()的功能是将不同进制的数值转换成以十进制表示的整数值。使用 parseInt()具体的语法格式如下。

```
parseInt(字符串[,底数])
```

通过上述格式可以将其他进制数值转换成为十进制数值。如果在执行过程中遇到非法字符，则立即停止执行，并返回已执行处理后的值。

- 转浮点函数：转浮点函数即函数 parseFloat()，功能是将指定字符串转换成浮点数值。如果在执行过程中遇到非法字符，则立即停止执行，并返回已执行处理后的值。

4.6　实践案例与上机指导

通过本章的学习，读者基本可以掌握 JavaScript 语言的知识。其实有关 JavaScript 语言的知识还有很多，这需要读者通过课外渠道来加深学习。下面通过练习操作，以达到巩固学习、拓展提高的目的。

↑扫码看视频

4.6.1 使用对象

JavaScript 中的对象是由属性(properties)和方法(methods)两个基本的元素构成的。其中，属性用来表示对象的特性，而方法用来实现具体的功能，方法通常与特定的函数相关联。

实例 4-7：从对象事件处理程序中调用函数。
源文件路径：daima\4\4-7

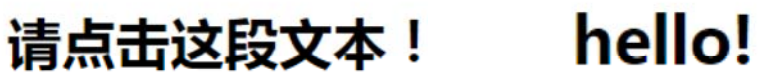
实例文件 dui.html 的主要实现代码如下。

```
<!DOCTYPE html>
<html>
<head>
<script>
function changetext(id)
{
id.innerHTML="hello!";
}
</script>
</head>
<body>

<h1 onclick="changetext(this)">请点击这段文本！</h1>

</body>
</html>
```

执行后的效果如图 4-8 所示。



初始效果　　点击后显示文本

图 4-8　执行效果

4.6.2 使用事件

在 JavaScript 中，通常将鼠标或热键的动作称为事件(Event)，而由鼠标或热键引发的一连串程序的动作，称为事件驱动(Event Driver)。而对事件进行处理的程序或函数，被称为事件处理程序(Event Handler)。

实例 4-8：将输入的文本转换为大写字母。
源文件路径：daima\4\4-8

本实例的实现文件为 da.html，具体实现代码如下。

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction()
```

```
{
var x=document.getElementById("fname");
x.value=x.value.toUpperCase();
}
</script>
</head>
<body>

请输入你的英文名：<input type="text" id="fname" onchange="myFunction()">
<p>当你离开输入框时，被触发的函数会把你输入的文本转换为大写字母。</p>

</body>
</html>
```

执行后发现在文本框中不能输入小写字母，例如输入小写字母 aaa 后的效果如图 4-9 所示。

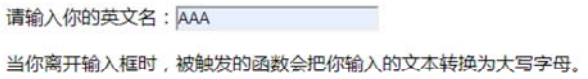


图 4-9　执行效果

思考与练习

本章详细讲解了 JavaScript 语言的知识，循序渐进地讲解了 JavaScript 简介、数据类型、表达式和运算符、JavaScript 循环语句、JavaScript 函数等知识。在讲解过程中，通过具体实例介绍了使用 JavaScript 的方法。通过本章的学习，读者应该熟悉使用流程控制语句的知识，掌握它们的使用方法和技巧。

1. 选择题

(1) 事件(　　)的功能是，响应用户单击表单中的 Reset 按钮。

A. Reset　　B. Resize　　C. Unload

(2) 属性(　　)的功能是设置状态工具栏的临时性信息。

A. status　　B. Parent　　C. defaultstatus

2. 判断对错

(1) 通过 JavaScript 中的窗口对象的多个方法可以实现信息的输出功能，主要有方法 window.alert()、document.write 和 document.writln()。　　(　　)

(2) for...in 是将一个已知对象的所有属性反复置给一个变量，使用计数器实现。　　(　　)

3. 上机练习

(1) 编写一个简单的 onmouseover-onmouseout 实例。

(2) 编写一个简单的 onmousedown-onmouseup 实例。