JavaScript 简介

* [JS 教程](http://www.w3school.com.cn/js/index.asp)
* [JS 实现](http://www.w3school.com.cn/js/js_howto.asp)

**JavaScript 是世界上最流行的编程语言。**

**这门语言可用于 HTML 和 web，更可广泛用于服务器、PC、笔记本电脑、平板电脑和智能手机等设备。**

**JavaScript 是脚本语言**

JavaScript 是一种轻量级的编程语言。

JavaScript 是可插入 HTML 页面的编程代码。

JavaScript 插入 HTML 页面后，可由所有的现代浏览器执行。

JavaScript 很容易学习。

**您将学到什么**

下面是您将在本教程中学到的主要内容。

**JavaScript：写入 HTML 输出**

**实例**

document.write("<h1>This is a heading</h1>");

document.write("<p>This is a paragraph</p>");

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_intro_document_write)

**提示：**您只能在 HTML 输出中使用 document.write。如果您在文档加载后使用该方法，会覆盖整个文档。

**JavaScript：对事件作出反应**

**实例**

<button type="button" onclick="alert('Welcome!')">点击这里</button>

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_intro_alert)

alert() 函数在 JavaScript 中并不常用，但它对于代码测试非常方便。

onclick 事件只是您即将在本教程中学到的众多事件之一。

**JavaScript：改变 HTML 内容**

使用 JavaScript 来处理 HTML 内容是非常强大的功能。

**实例**

x=document.getElementById("demo") //查找元素

x.innerHTML="Hello JavaScript"; //改变内容

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_intro_inner_html)

您会经常看到 document.getElementByID("*some id*")。这个方法是 HTML DOM 中定义的。

DOM（文档对象模型）是用以访问 HTML 元素的正式 W3C 标准。

您将在本教程的多个章节中学到有关 HTML DOM 的知识。

**JavaScript：改变 HTML 图像**

本例会动态地改变 HTML <image> 的来源 (src)：

The Light bulb

点击灯泡就可以打开或关闭这盏灯

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_lightbulb)

JavaScript 能够改变任意 HTML 元素的大多数属性，而不仅仅是图片。

**JavaScript：改变 HTML 样式**

改变 HTML 元素的样式，属于改变 HTML 属性的变种。

**实例**

x=document.getElementById("demo") //找到元素

x.style.color="#ff0000"; //改变样式

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_intro_style)

**JavaScript：验证输入**

JavaScript 常用于验证用户的输入。

**实例**

if isNaN(x) {alert("Not Numeric")};

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_intro_validate)

**您知道吗？**

**提示：**JavaScript 与 Java 是两种完全不同的语言，无论在概念还是设计上。

Java（由 Sun 发明）是更复杂的编程语言。

ECMA-262 是 JavaScript 标准的官方名称。

JavaScript 由 Brendan Eich 发明。它于 1995 年出现在 Netscape 中（该浏览器已停止更新），并于 1997 年被 ECMA（一个标准协会）采纳。

# JavaScript 使用

* [JS 简介](http://www.w3school.com.cn/js/js_intro.asp)
* [JS 输出](http://www.w3school.com.cn/js/js_whereto.asp)

**HTML 中的脚本必须位于 <script> 与 </script> 标签之间。**

**脚本可被放置在 HTML 页面的 <body> 和 <head> 部分中。**

## <script> 标签

如需在 HTML 页面中插入 JavaScript，请使用 <script> 标签。

<script> 和 </script> 会告诉 JavaScript 在何处开始和结束。

<script> 和 </script> 之间的代码行包含了 JavaScript：

<script>

alert("My First JavaScript");

</script>

您无需理解上面的代码。只需明白，浏览器会解释并执行位于 <script> 和 </script> 之间的 JavaScript。

那些老旧的实例可能会在 <script> 标签中使用 type="text/javascript"。现在已经不必这样做了。JavaScript 是所有现代浏览器以及 HTML5 中的默认脚本语言。

## <body> 中的 JavaScript

在本例中，JavaScript 会在页面加载时向 HTML 的 <body> 写文本：

### 实例

<!DOCTYPE html>

<html>

<body>

.

.

<script>

document.write("<h1>This is a heading</h1>");

document.write("<p>This is a paragraph</p>");

</script>

.

.

</body>

</html>

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_intro_document_write)

## JavaScript 函数和事件

上面例子中的 JavaScript 语句，会在页面加载时执行。

通常，我们需要在某个事件发生时执行代码，比如当用户点击按钮时。

如果我们把 JavaScript 代码放入函数中，就可以在事件发生时调用该函数。

您将在稍后的章节学到更多有关 JavaScript 函数和事件的知识。

## <head> 或 <body> 中的 JavaScript

您可以在 HTML 文档中放入不限数量的脚本。

脚本可位于 HTML 的 <body> 或 <head> 部分中，或者同时存在于两个部分中。

通常的做法是把函数放入 <head> 部分中，或者放在页面底部。这样就可以把它们安置到同一处位置，不会干扰页面的内容。

## <head> 中的 JavaScript 函数

在本例中，我们把一个 JavaScript 函数放置到 HTML 页面的 <head> 部分。

该函数会在点击按钮时被调用：

### 实例

<!DOCTYPE html>

<html>

<head>

<script>

function myFunction()

{

document.getElementById("demo").innerHTML="My First JavaScript Function";

}

</script>

</head>

<body>

<h1>My Web Page</h1>

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

<button type="button" onclick="myFunction()">Try it</button>

</body>

</html>

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_whereto_head)

## <body> 中的 JavaScript 函数

在本例中，我们把一个 JavaScript 函数放置到 HTML 页面的 <body> 部分。

该函数会在点击按钮时被调用：

### 实例

<!DOCTYPE html>

<html>

<body>

<h1>My Web Page</h1>

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

<button type="button" onclick="myFunction()">Try it</button>

<script>

function myFunction()

{

document.getElementById("demo").innerHTML="My First JavaScript Function";

}

</script>

</body>

</html>

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_whereto_body)

**提示：**我们把 JavaScript 放到了页面代码的底部，这样就可以确保在 <p> 元素创建之后再执行脚本。

## 外部的 JavaScript

也可以把脚本保存到外部文件中。外部文件通常包含被多个网页使用的代码。

外部 JavaScript 文件的文件扩展名是 .js。

如需使用外部文件，请在 <script> 标签的 "src" 属性中设置该 .js 文件：

### 实例

<!DOCTYPE html>

<html>

<body>

<script src="myScript.js"></script>

</body>

</html>

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_externalexample)

在 <head> 或 <body> 中引用脚本文件都是可以的。实际运行效果与您在 <script> 标签中编写脚本完全一致。

**提示：**外部脚本不能包含 <script> 标签。

# avaScript 语句

* [JS 输出](http://www.w3school.com.cn/js/js_whereto.asp)
* [JS 注释](http://www.w3school.com.cn/js/js_comments.asp)

## JavaScript 语句

JavaScript 语句向浏览器发出的命令。语句的作用是告诉浏览器该做什么。

下面的 JavaScript 语句向 id="demo" 的 HTML 元素输出文本 "Hello World"：

document.getElementById("demo").innerHTML="Hello World";

## 分号 ;

分号用于分隔 JavaScript 语句。

通常我们在每条可执行的语句结尾添加分号。

使用分号的另一用处是在一行中编写多条语句。

**提示：**您也可能看到不带有分号的案例。

在 JavaScript 中，用分号来结束语句是可选的。

## JavaScript 代码

JavaScript 代码（或者只有 JavaScript）是 JavaScript 语句的序列。

浏览器会按照编写顺序来执行每条语句。

本例将操作两个 HTML 元素：

### 实例

document.getElementById("demo").innerHTML="Hello World";

document.getElementById("myDIV").innerHTML="How are you?";

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_statements)

## JavaScript 代码块

JavaScript 语句通过代码块的形式进行组合。

块由左花括号开始，由右花括号结束。

块的作用是使语句序列一起执行。

JavaScript 函数是将语句组合在块中的典型例子。

下面的例子将运行可操作两个 HTML 元素的函数：

### 实例

function myFunction()

{

document.getElementById("demo").innerHTML="Hello World";

document.getElementById("myDIV").innerHTML="How are you?";

}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_blocks)

您将在稍后的章节学到更多有关函数的知识。

## JavaScript 对大小写敏感。

JavaScript 对大小写是敏感的。

当编写 JavaScript 语句时，请留意是否关闭大小写切换键。

函数 getElementById 与 getElementbyID 是不同的。

同样，变量 myVariable 与 MyVariable 也是不同的。

## 空格

JavaScript 会忽略多余的空格。您可以向脚本添加空格，来提高其可读性。下面的两行代码是等效的：

var name="Hello";

var name = "Hello";

## 对代码行进行折行

您可以在文本字符串中使用反斜杠对代码行进行换行。下面的例子会正确地显示：

document.write("Hello \

World!");

不过，您不能像这样折行：

document.write \

("Hello World!");

## 您知道吗？

**提示：**JavaScript 是脚本语言。浏览器会在读取代码时，逐行地执行脚本代码。而对于传统编程来说，会在执行前对所有代码进行编译。

# JavaScript 注释

* [JS 语句](http://www.w3school.com.cn/js/js_statements.asp)
* [JS 变量](http://www.w3school.com.cn/js/js_variables.asp)

**JavaScript 注释可用于提高代码的可读性。**

## JavaScript 注释

JavaScript 不会执行注释。

我们可以添加注释来对 JavaScript 进行解释，或者提高代码的可读性。

单行注释以 // 开头。

### 例子

下面的例子使用单行注释来解释代码：

// 输出标题：

document.getElementById("myH1").innerHTML="Welcome to my Homepage";

// 输出段落：

document.getElementById("myP").innerHTML="This is my first paragraph.";

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_comments1)

## JavaScript 多行注释

多行注释以 /\* 开始，以 \*/ 结尾。

下面的例子使用多行注释来解释代码：

### 例子

/\*

下面的这些代码会输出

一个标题和一个段落

并将代表主页的开始

\*/

document.getElementById("myH1").innerHTML="Welcome to my Homepage";

document.getElementById("myP").innerHTML="This is my first paragraph.";

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_comments2)

## 使用注释来阻止执行

### 例子 1

在下面的例子中，注释用于阻止其中一条代码行的执行（可用于调试）：

//document.getElementById("myH1").innerHTML="Welcome to my Homepage";

document.getElementById("myP").innerHTML="This is my first paragraph.";

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_comments3)

### 例子 2

在下面的例子中，注释用于阻止代码块的执行（可用于调试）：

/\*

document.getElementById("myH1").innerHTML="Welcome to my Homepage";

document.getElementById("myP").innerHTML="This is my first paragraph.";

\*/

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_comments4)

## 在行末使用注释

在下面的例子中，我们把注释放到代码行的结尾处：

### 例子

var x=5; // 声明 x 并把 5 赋值给它

var y=x+2; // 声明 y 并把 x+2 赋值给它

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=js_comments5)

* [JS 语句](http://www.w3school.com.cn/js/js_statements.asp)
* [JS 变量](http://www.w3school.com.cn/js/js_variables.asp)

## 课外阅读

JavaScript 高级教程： [ECMAScript 语法](http://www.w3school.com.cn/js/pro_js_syntax.asp)