



计算机类专业“互联网+”创新型精品教材

# JavaScript+jQuery 开发实战

主 编 薛慧丽 刘志方

JavaScript+jQuery 开发实战

主 编 薛慧丽 刘志方



北京出版集团  
北京出版社



北京出版集团  
北京出版社

图书在版编目 (CIP) 数据

JavaScript+jQuery 开发实战 / 薛慧丽, 刘志方主编.  
北京: 北京出版社, 2025.1. -- ISBN 978-7-200-  
-19220-9

I . TP312.8; TP393.092.2

中国国家版本馆 CIP 数据核字第 20251G6Q28 号

JavaScript+jQuery 开发实战

JavaScript+jQuery KAIFA SHIZHAN

---

主 编: 薛慧丽 刘志方

出 版: 北京出版集团

北 京 出 版 社

地 址: 北京北三环中路 6 号

邮 编: 100120

网 址: [www.bph.com.cn](http://www.bph.com.cn)

总 发 行: 北京出版集团

经 销: 新华书店

印 刷: 定州启航印刷有限公司

版 印 次: 2025 年 1 月第 1 版 2025 年 1 月第 1 次印刷

成品尺寸: 185 毫米 × 260 毫米

印 张: 14.5

字 数: 326 千字

书 号: ISBN 978-7-200-19220-9

定 价: 46.00 元

教材意见建议接收方式: 010-58572341 邮箱: [jiaocai@bphg.com.cn](mailto:jiaocai@bphg.com.cn)

如有印装质量问题, 由本社负责调换

质量监督电话: 010-82685218 010-58572341 010-58572393



## 模块一 初识 JavaScript ..... 1

任务一 JavaScript 的认识 ..... 1

任务二 JavaScript 的基本应用 ..... 6



## 模块二 JavaScript 基础 ..... 14

任务一 标识符和变量 ..... 14

任务二 流程控制 ..... 24

任务三 数组 ..... 32



## 模块三 函数 ..... 37

任务一 初识函数 ..... 37

任务二 函数进阶 ..... 45



## 模块四 对象 ..... 54

任务一 初识对象 ..... 54

任务二 内置对象 ..... 61



## 模块五 DOM ..... 75

任务一 DOM 概述 ..... 75

任务二 DOM 操作 ..... 78

任务三 节点获取与操作 ..... 90

任务四 获取和设置属性 ..... 99



## 模块六 事件处理..... 104

- 任务一 事件基础 ..... 104
- 任务二 事件对象 ..... 109
- 任务三 常用事件 ..... 114



## 模块七 BOM..... 122

- 任务一 window 对象 ..... 122
- 任务二 定时器 ..... 138
- 任务三 JavaScript 执行机制 ..... 143



## 模块八 jQuery ..... 151

- 任务一 获取元素和 DOM 操作 ..... 151
- 任务二 事件处理 ..... 159
- 任务三 动画效果 ..... 165



## 模块九 JavaScript 面向对象 ..... 177

- 任务一 ES6 面向对象语法 ..... 177
- 任务二 构造函数与原型对象 ..... 185



## 模块十 正则表达式..... 194

- 任务一 认识正则表达式 ..... 194
- 任务二 注册页面表单验证 ..... 208



## 模块十一 项目案例：传统文化网..... 219

## 模块一

# 初识 JavaScript

JavaScript 是一种基于对象、事件驱动的代码语言，不仅安全可靠，更以其动态交互性为网页赋予了鲜活的生命力。与 HTML、Java 代码语言一起实现在一个 Web 页面中链接多个对象，与 Web 客户交互，从而可以开发客户端的应用程序等。

## 任务一

## JavaScript 的认识

### 任务描述

小明已经掌握了 HTML 和 CSS 相关技术，已经能够设计出各种漂亮的静态网页。小明想让网页具有良好的交互性，他尚未理解动态交互式网页与“JavaScript”之间的紧密联系。本任务我们将和小明一起学习 JavaScript 的主要功能、历史发展、核心特点、重要组成。

### 任务目标

1. 能够说明 JavaScript 的主要功能，如用户交互、动态效果等。
2. 会简述 JavaScript 的历史发展。
3. 能够概括 JavaScript 的核心特点。
4. 会解释 JavaScript 的组成，包括 ECMAScript、DOM 和 BOM。
5. 能在学习 JavaScript 的道路上勇往直前，无惧挑战，不断追求卓越。

### 任务实施

#### 一、JavaScript 的主要功能

JavaScript 是一种广泛用于网页开发的编程语言，能够增强网页的交互性和动态效

果。它基于 ECMAScript 标准，支持面向对象编程，并具有事件驱动和动态类型的特点。JavaScript 通过 DOM 操作 HTML 文档，利用 BOM 与浏览器交互。它不仅能在浏览器中运行，还能在服务器端（如 Node.js）进行应用开发，是前端开发的核心技术之一。

对于制作一个网页而言，HTML、CSS 和 JavaScript 分别代表了结构、样式和行为，结构是网页的骨架，样式是网页的外观，行为是网页的交互逻辑，如表 1-1 所示。

表 1-1 HTML、CSS 和 JavaScript 的作用

| 语言         | 作用 | 说明           |
|------------|----|--------------|
| HTML       | 结构 | 决定网页的结构和内容   |
| CSS        | 样式 | 决定网页呈现给用户的模样 |
| JavaScript | 行为 | 从交互的角度提升用户体验 |

JavaScript 内嵌于 HTML 网页中，通过浏览器内置的 JavaScript 引擎直接编译，把一个原本只用来显示的页面，转变成支持用户交互的页面程序。下面通过一些示例来展示 JavaScript 能够制作的页面效果，如图 1-1 所示。



图 1-1 JavaScript 在网页中的应用

在项目开发中，页面中的许多常见的交互效果都可以利用 JavaScript 来实现。JavaScript 可以使网页的互动性更强、用户体验更好。

二、JavaScript 的历史发展

在 20 世纪 90 年代中期，互联网正处于蓬勃发展的黄金时期，Web 页面如雨后春笋般涌现，通过浏览器上网的用户数量激增。然而，那时的浏览器在处理简单的表单验证时，却不得不依赖于服务器端的烦琐交互，这无疑给用户带来了极大的不便。面对这一问题，技术革新的先驱 Netscape 公司开始深思，寻求一种创新的解决方案。

Netscape 公司敏锐地意识到, Navigator 浏览器急需一种轻量级的代码语言, 能够直接嵌入网页, 实现实时的浏览器行为控制。在网速缓慢且上网成本高昂的时代背景下, 许多操作在服务器端执行显得尤为低效。例如, 用户遗漏了“用户名”字段而匆忙单击“发送”, 待数据到达服务器时才发现问题, 这样, 不仅费时费力, 更增加了用户的挫败感。因此, Netscape 公司决定开发一种浏览器代码语言, 它能够在用户提交数据之前, 直接在网页端进行简单的验证和提示, 如“请填写用户名”。这种代码语言的设计理念是功能实用、语法简洁, 易于学习和部署, 从而极大地提升了用户体验和效率。

基于这些设想, 1995 年, Netscape 公司雇佣程序员 Brendan Eich 开发出了这种网页代码语言的第一版, 其最初的名字叫 LiveScript, 但为了和当时非常盛行的 Java 搭上关系, LiveScript 改名为 JavaScript。自此, JavaScript 正式问世。后来随着互联网应用越来越广泛, JavaScript 也不断演化。发展至今, 已经从一个简单的表单验证器成为一门强大而复杂的语言。

众多 Web 浏览器对 JavaScript 的支持很不一致, 相同的语言特性在不同的浏览器中会有所差异, 这种差异对开发者影响极大, 开发者在开发时不得不为不同的浏览器编写不同的代码, 这种难堪的局面一直持续到 JavaScript 标准的制定。1997 年发布了 ECMA-262 语言规范, 将 JavaScript 语言标准化并重命名为 ECMAScript, 现在各种浏览器都以该规范为标准。语言和系统接口标准化后可以大大减轻开发人员的负担, 不用为不同的语言特性或接口编写不同的代码, 这也增强了软件的可移植性。

### 三、JavaScript 的核心特点

#### 1. 解释型代码语言

JavaScript 是一种解释型代码语言, 与 C、C++ 等语言需要先编译再运行不同, 使用 JavaScript 编写的代码不需要预先编译, 可以直接在浏览器或运行的环境中执行。

#### 2. 基于对象

JavaScript 是一种基于对象的代码语言, 使用 JavaScript 不仅可以创建新的对象, 也可以操作使用已有的对象。这种特性使得 JavaScript 具有高度的灵活性和可定制性。

#### 3. 弱类型

JavaScript 是一种弱类型的编程语言, 对使用的数据类型没有严格的要求。例如, 一个变量可以被初始化为任意类型, 并且可以随时改变这个变量的类型。

#### 4. 动态性

JavaScript 是一种动态性很强的语言, 它支持在运行时改变代码的结构和行为。JavaScript 也是一种事件驱动的代码语言, 可以通过用户的行为(如鼠标单击、键盘输入等)触发相应的事件响应。

#### 5. 跨平台性

JavaScript 不依赖于特定的操作系统, 仅需要浏览器的支持即可运行。因此, 一个 JavaScript 代码在编写完成后可以在任意支持 JavaScript 的浏览器上运行, 实现了跨平台性。

## 6. 简单易学

JavaScript 的语法相对简单，入门门槛较低。它采用了类似于 C 语言的语法，对于有其他编程经验的开发者来说比较容易上手。同时，JavaScript 的学习资源丰富，社区活跃，可以快速获取到大量学习资料和帮助。

## 7. 实时性好

JavaScript 作为前端代码语言，能够在客户端执行，因此能够实时地响应用户的交互操作，为用户提供更加流畅的体验。这种实时性使得 JavaScript 在开发交互性强、用户体验要求高的 Web 应用时非常有优势。

## 8. 拥有丰富的框架和库

JavaScript 拥有众多优异的框架和库，如 React、Angular、Vue 等，这些框架和库大大简化了 Web 开发的流程，提高了开发效率，使得开发者能够更专注于业务逻辑而非底层细节。

## 9. 支持异步编程

JavaScript 支持异步编程，这使得它非常适合处理涉及网络请求和 I/O 操作的应用场景。通过异步编程，JavaScript 能够更高效地处理大量并发请求，不会造成程序阻塞，提升了应用的响应速度和性能。

## 10. 生态系统繁荣

JavaScript 生态系统庞大繁荣，拥有大量的第三方库和工具，这些资源为开发者提供了各种各样的解决方案，不仅加速了开发进程，还促进了代码的复用和优化。

综上所述，JavaScript 是一种功能强大且灵活多变的编程语言，它以其解释型、基于对象、弱类型、动态性、跨平台、简单易学等特点，在 Web 开发领域占据了举足轻重的地位。JavaScript 的实时响应和事件驱动机制，使其能够实时响应用户的交互操作，提供丰富的用户体验。因此，JavaScript 不仅是 Web 开发的核心技术之一，也是未来前端开发的重要发展方向。

## 四、JavaScript 的重要组成

JavaScript 是由 ECMAScript、DOM、BOM 三部分组成的，如图 1-2 所示。

### 1. ECMAScript

ECMAScript 是 JavaScript 的核心，它定义了语言的基础语法、类型、语句、关键字、保留字、操作符、对象等。

ECMAScript 标准由 Ecma International

(欧洲计算机制造商协会) 进行标准化，并定期发布新版本，如 2015 年 6 月发布的 ES6/ES2015，带来了巨大的革新，包括箭头函数、模板字符串、类、模块、解构赋值、for...of 循环、let/const 块级作用域、Promise、Map/Set、Proxy 和 Reflect 等。以后的版本在 ES6 的基础上新增部分功能。

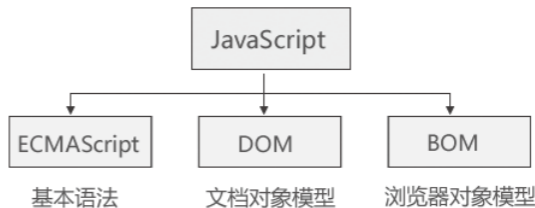


图 1-2 JavaScript 的组成部分

## 2. DOM

DOM 是 Document Object Model 的简称, 是 HTML 和 XML 文档的编程接口。它提供了对文档的结构化的表述, 并定义了一种可以从程序中对该结构进行访问的方式, 从而改变文档的结构、样式和内容。JavaScript 通过 DOM API 可以动态地修改 HTML 内容和结构 (如大小、位置、颜色等), 以及响应用户的操作。



JavaScript 的认识

## 3. BOM

BOM 是 Browser Object Model 的简称, 提供了独立于内容与浏览器窗口进行交互的对象。例如, 通过 BOM 可以控制浏览器窗口的大小、位置以及导航等。

### 思考与练习

#### 一、填空题

1. JavaScript 起源于 Netscape 公司, 最初被命名为 LiveScript, 后来被改名为 JavaScript, 并在 \_\_\_\_\_ 年由 ECMA (欧洲计算机制造商协会) 进行了标准化。
2. JavaScript 的一个显著特点是它的 \_\_\_\_\_, 这意味着 JavaScript 代码可以直接在用户的浏览器中执行, 而无须通过服务器。
3. JavaScript 主要由 ECMAScript、DOM (文档对象模型) 和 \_\_\_\_\_ 三部分组成, 这三者共同构成了 JavaScript 的核心技术栈。
4. 在 JavaScript 中, \_\_\_\_\_ 是代码的基本单元, 它用于执行特定的任务或操作。

#### 二、选择题

1. JavaScript 主要用于 ( )。  
A. 服务器端编程                      B. 桌面应用开发  
C. 网页交互和动态内容              D. 移动应用开发
2. 关于 JavaScript 的历史, 以下说法正确的是 ( )。  
A. JavaScript 最初是由 Microsoft 公司开发的  
B. JavaScript 的命名与 Java 语言有直接的关联  
C. JavaScript 的标准化是由 W3C (万维网联盟) 完成的  
D. JavaScript 的最初版本是在 1995 年发布的
3. JavaScript 的 ( ) 使得它成为前端开发的重要工具。  
A. 面向对象              B. 跨平台性              C. 静态类型              D. 编译执行
4. JavaScript 的组成中, ( ) 用于操作 HTML 文档的结构、样式和内容。  
A. ECMAScript                      B. BOM (浏览器对象模型)  
C. DOM (文档对象模型)              D. AJAX (异步 JavaScript 和 XML)

## 任务二

## JavaScript 的基本应用

## 任务描述

小明已经了解了 JavaScript 的基本组成、作用和特点，并意识到它对于提升网页交互性的重要性。然而，小明对如何在 JavaScript 中实现输入和输出操作还不是很清晰。本任务我们将和小明一起学习 JavaScript 中的输出语句。

## 任务目标

1. 会使用三种 JavaScript 引入 HTML 的方法。
2. 会使用 JavaScript 的常用输入 / 输出语句。
3. 遵守实训室操作规范，养成良好的职业素养。

## 任务实施

JavaScript 代码编辑器有很多选择，每个都有其独特的特点和优势。可以选择常用的专业前端开发工具，如 Visual Studio Code (VS Code)、HBuilder、Sublime Text 等，这些专业的开发工具内部集成了 JavaScript 代码的开发环境，使用起来简单方便。

## 一、JavaScript 代码的引用

编写 JavaScript 代码时，有 3 种引用位置，分别是行内式、嵌入式和外链式。

## 1. 行内式

行内式是直接在 HTML 标签的属性中编写 JavaScript 代码，下面通过具体操作步骤进行演示。

创建一个简单的 HTML 页面，将文档命名为 example1-2-1.html。具体代码如下。

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-
6       scale=1.0" />
7     <title>Document</title>
8   </head>
9 </html>
```

在上述代码中，第 4 行声明了网页的编码为 UTF-8，帮助浏览器正确识别网页的编码。在声明编码后，还需要确保文件本身的编码也是 UTF-8。目前大多数代码编辑

器新建的文件编码默认都是 UTF-8。另外，Windows 记事本默认的编码是 ANSI，在记事本中编写的网页容易出现乱码，因此在操作时应杜绝使用记事本编写代码文件。

(1) 使用行内式编写 JavaScript 代码，实现单击按钮“我是行内式”后，弹出一个警告框提示“行内式”，示例代码如下。

```
<body>
  <input type="button" value="我是行内式" onclick="alert('行内式')" />
</body>
```

在上述代码中，写在 onclick 属性里的代码就是 JavaScript 代码。需要说明的是，行内式只有在临时测试或者特殊情况下使用。

(2) 通过浏览器访问 example1-2-1.html，单击“我是行内式”按钮，运行结果如图 1-3 所示。



图 1-3 单击“我是行内式”按钮显示

## 2. 嵌入式

使用 <script>...</script> 标签将代码嵌入 HTML 文档中。在学习 JavaScript 时，代码测试方便，是最常使用的方式。

下面通过具体操作步骤进行演示。

(1) 创建 example1-2-2.html，示例代码如下。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script type="text/javascript">
    alert(“嵌入式”);
  </script>
  <script></script>
</body>
</html>
```

在上述代码中，其末尾的分号“；”表示该语句结束，后面可以编写下一条语句。<script>...</script> 标签还有一个 type 属性，在 HTML 中该属性的默认值为“text/javascript”，因此在编写时可以省略 type 属性。

(2) 通过浏览器访问 example1-2-2.html，页面打开后，就会自动弹出一个警告框，提示信息为“嵌入式”。如图 1-4 所示。



图 1-4 嵌入式

### 3. 外链式

外链式是将 JavaScript 写入扩展名为“.js”的文件内，然后通过<script>标签的src属性将“.js”文件嵌入 HTML 文档中。外链式适合 JavaScript 代码较多的情况。

外链式有利于 HTML 页面代码结构化，把大段的 JavaScript 代码独立到 HTML 页面之外，既美观，也方便文件级别的代码复用。需要注意的是，外链式的<script>标签内不可以编写 JavaScript 代码。

下面我们通过具体操作步骤进行演示。

(1) 创建 example1-2-3.html，示例代码如下。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="test.js"></script>
</head>
<body>

</body>
</html>
```

(2) 创建 test.js 文件，在文件中编写 JavaScript 代码，如下所示。

```
alert("外链式");
```

(3) 通过浏览器访问 example1-2-3.html，页面打开后，就会自动弹出一个警告框，提示信息为“外链式”。



图 1-5 外链式

## 二、常用输入/输出语句

在 JavaScript 中常用的输出语句有：警告对话框 `alert()`、控制台输出 `console.log()`、输出语句 `document.write()`、提示对话框 `prompt()`。

### 1. 警告对话框 `alert()`

`alert()` 方法会创建一个警告对话框，确保用户可以看到某些信息。如注册网站时弹出的提示用户输入信息。在前面的案例中已经演示过，利用 `alert()` 可以很方便地输出一个结果，经常用于测试程序。

`alert()` 方法的基本语法格式如下：

```
alert ('提示信息');
```

### 2. 控制台输出 `console.log()`

`console.log()` 函数用于在浏览器的控制台中输出信息。可以将一个或多个参数输出到控制台中，这对于调试代码和查看变量值非常有用。

`console.log()` 的基本语法格式如下：

```
console.log(message);
```

“message”是必须项，用于在控制台上输出要显示的信息。可以是字符、表达式等。

创建 `example1-2-4.html`，示例代码如下。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log('你好');
  </script>
</body>
</html>
```

使用 Chrome 浏览器打开 `example1-2-4.html`，按 F12 键（或在网页空白区域单击鼠标右键，在弹出的菜单中选择“检查”）启动开发者工具，然后切换到 Console 控制台选项卡，如图 1-6 所示。

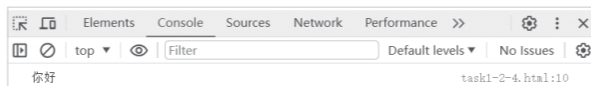


图 1-6 浏览器控制台

从图 1-6 中可以看出，控制台显示了输出结果“你好”，右边的“task1-2-4.html:10”表示输出的代码来自 example1-2-4.html 文件中的第 10 行。

### 3. 输出语句 document.write()

document.write() 方法用于在 HTML 文档页面中输出内容，document.write() 方法的基本语法格式如下：

```
document.write('输出内容');
```

创建 example1-2-5.html，示例代码如下。

```
<script>
    document.write('<b>我爱我的祖国——中华人民共和国</b>');
</script>
```

通过浏览器测试，运行结果如图 1-7 所示。



图 1-7 在网页中输出加粗文本

### 4. 提示对话框 prompt()

prompt() 方法会弹出一个对话框，用于收集用户关于特定问题的反馈信息，提示对话框具有返回值。prompt() 的基本语法格式如下：

```
prompt ('提示信息', '输入框的默认信息');
```

该方法的返回值可以被引用或存储到变量中。

创建 example1-2-6.html，示例代码如下。

```
<script>
    prompt ('请输入你喜欢的课程:', '前端开发');
</script>
```

运行效果如图 1-8 所示。可以在提示对话框中输入自己喜欢的课程，默认喜欢课程为“前端开发”。



图 1-8 在网页中提示对话框

【案例】制作一个简单的个人简介生成器：该生成器将引导用户输入个人信息，并在网页上展示个性化的简介。提示：使用 `prompt` 函数收集用户输入；使用 `document.write` 在网页上输出内容。

分析：

① 使用 `prompt` 函数提示用户输入以下信息：姓名；年龄；来自的城市；兴趣爱好。

② 输出展示：利用 `document.write` 将用户输入的信息组织成一段个人简介，并展示在网页上。

打开前端开发工具 VS Code，首先新建“Chapter01”项目；接着在该项目中新建一个名为“task1-2-1.html”的 HTML 文件，输入如下代码：



在浏览器显示  
个人信息

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <script>
// 1. 获取用户输入
    let fullName = prompt("请输入您的姓名：");
    let age = prompt("请输入您的年龄：");
    let city = prompt("请输入您来自的城市：");
    let hobby = prompt("请输入您的兴趣爱好：");

// 2. 输出个人简介
    document.write("个人简介：<br/>");
    document.write(
      "大家好，我叫" +
        fullName +
        "，今年" +
        age +
        "岁，来自美丽的" +
        city +
        "。我的兴趣爱好是" +
        hobby +
        "，希望能和大家成为朋友。"
    );
    </script>
  </body>
</html>
```

在浏览器运行后，根据提示分别输入：孙悟空、18、北京、踢球，则浏览器显示如图 1-9 所示。

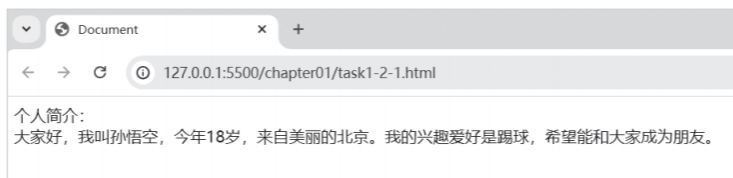


图 1-9 在网页中显示个人简介

## 知识链接

### 1. 控制台的使用

JavaScript 控制台（通常指的是浏览器的开发者工具中的 Console 面板）是开发人员在网页开发和调试时的重要工具。

在大多数浏览器中，你可以通过以下方式打开开发者工具（包括 Console 面板）：

- ① 右键单击页面上的任意位置，选择“检查”或“审查元素”（取决于浏览器）。
- ② 使用快捷键：在 Windows 上通常是 F12 或 Ctrl+Shift+I，在 macOS 上通常是 Cmd+Opt +I。

③ 在地址栏前或标签页右键单击，选择“检查”或“审查元素”。

在浏览器的控制台中可以直接输入 JavaScript 代码来执行，这为 JavaScript 初学者提供了很大的便利。打开控制台后，会看到一个闪烁的光标，此时可以输入代码，按回车键执行。图 1-10 演示了直接在控制台中输入代码执行的效果。

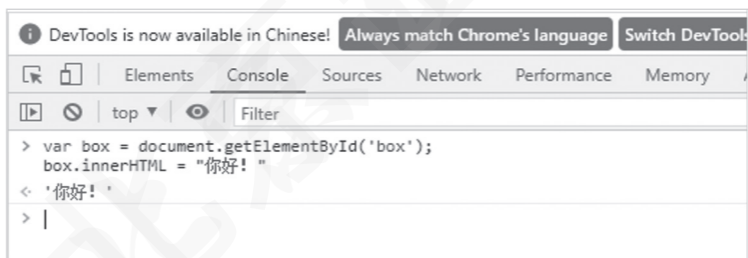


图 1-10 控制台输入

在图 1-10，代码前面的“>”图标表示该行代码是用户输入的信息，下一行的“<”图标表示控制台输出的结果，用于显示用户输入的表达式值。

在控制台中还可以用“Ctrl+鼠标滚轮”放大或缩小，用快捷键【Shift+Enter】在输入的代码中进行换行。

### 2. 注释

在 JavaScript 开发过程中，使用注释是为了便于代码的可读性，它在程序解析时会被 JavaScript 解释器忽略。JavaScript 支持单行注释和多行注释，具体示例如下。

① 单行注释“//”

```
document.write('<b>这是加粗文本</b>');
```

```
//输出加粗的文字：这是加粗文本
```

上述示例中，“//”和后面的“输出一句话”是一个单行注释。以“//”开始，到该行结束或 JavaScript 标记结束之前的内容都是注释。

## ② 多行注释 “/\* \*/”

```
/* document.write('<b>这是加粗文本</b>'); */
```

上述示例中，“/\*”和“\*/”之间的内容为多行注释，多行注释以“/\*”开始，以“\*/”结束。同时，多行注释中可以嵌套单行注释，但不能再嵌套多行注释。

在 Visual Studio Code 编辑器中，可以使用快捷键对当前选中的行添加注释或取消注释。单行注释使用快捷键“Ctrl+/", 多行注释使用快捷键“Shift+Alt+A”。

## 思考与练习

### 一、填空题

1. JavaScript 代码中单行注释以 \_\_\_\_\_ 开始。
2. document.write('人生在实现中国梦的奋进追逐中展现出勇敢奔跑的英姿'); 的输出结果是 \_\_\_\_\_。

### 二、选择题

1. 下列选项中不属于 ECMAScript 6 保留关键字的是 ( )。  
A. delete            B. this            C. static            D. new
2. JavaScript 为代码添加多行注释的语法为 ( )。  
A. <!-- -->    B. //            C. /\* \*/            D. @

### 三、实训题

用 JavaScript 代码，将唐代著名浪漫主义诗人李白的《望庐山瀑布》这首诗输出到网页，输出效果如图 1-11 所示：

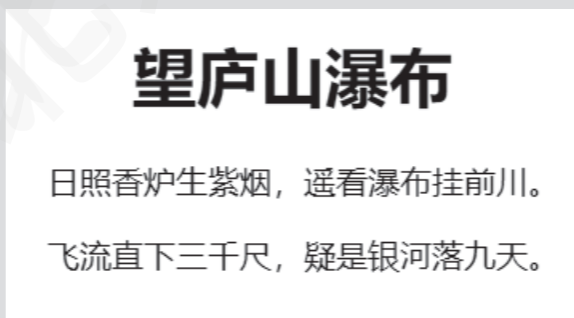


图 1-11 浏览器显示效果