

第1章 JavaScript简介



- 什么是JavaScript
- JavaScript的特性
- 编写第一个JavaScript程序



目录

1.1

什么是JavaScript

[点击查看本小节知识架构](#)

1.2

JavaScript的特性

[点击查看本小节知识架构](#)

1.3

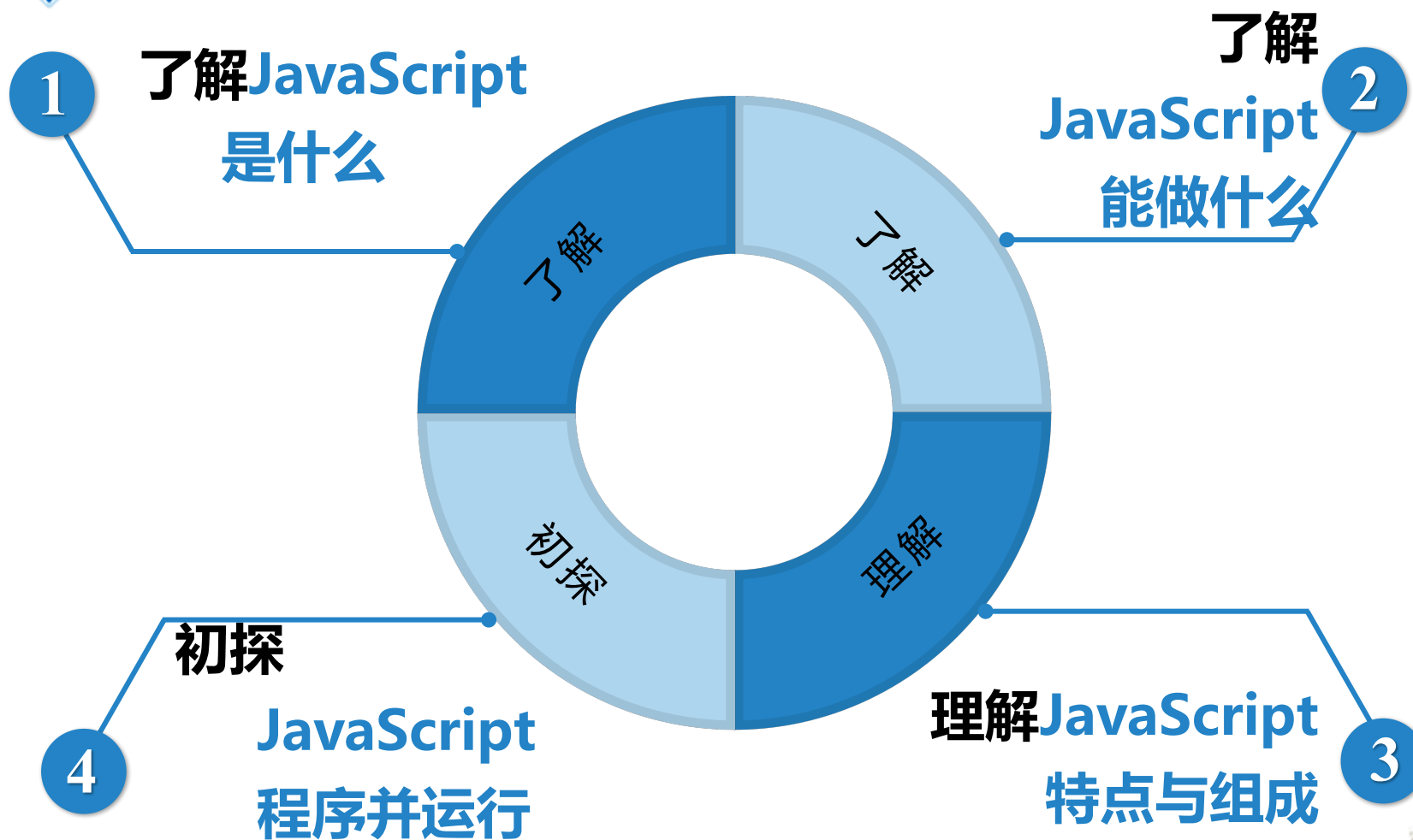
编写第一个JS程序

[点击查看本小节知识架构](#)





学习目标





1.1 什么是JavaScript



[返回目录](#)

1.1.1

● JavaScript的诞生

1.1.2

● JavaScript与Java的关系

1.1.3

● 为什么学习JavaScript





1.1 什么是JavaScript

1.1.1 JavaScript的诞生

- 1991年8月6日，来自欧洲核子研究中心的科学家Tim Berners-Lee，启动了世界上第一个可以正式访问的网站（<http://info.cern.ch/>），从此人类进入互联网时代。
- 但当时的浏览器属于国家专利，一般用于军事方面。一家来自美国的计算机服务公司，即网景通信公司（Netscape Communications Corporation），在1994年12月发布了网景浏览器(Netscape Navigator)1.0正式版，成为当时最热门的商业浏览器，占据着市场的主要份额。
- 网景浏览器1.0正式版只能用来浏览网页，不具备与访问者互动的能力，如单击、滑动、验证等功能，因此网景公司迫切需要一种网页脚本语言，以实现浏览器与网页的互动。





1.1 什么是JavaScript

1.1.1 JavaScript的诞生

- 在1995年，网景公司聘请程序员Brendan Eich（如图所示）开发这种网页脚本语言。由于当时时间紧、任务重，Brendan Eich只用10天就设计完成了这种语言的第一版。由于设计时间太短，语言的一些细节在当时还不够严谨，但JS语言的雏形就此诞生。





1.1 什么是JavaScript

1.1.2 JavaScript与Java的关系

- 网景公司的这种浏览器脚本语言，最初名字叫做Mocha，1995年9月改为LiveScript。同年12月，网景公司与Sun公司（即Java语言的发明者和所有者）达成协议，后者允许将这种语言叫做JavaScript。这样，网景公司可以借助Java语言的声势，而Sun公司则将自身的影响力扩展到浏览器。所以说JavaScript语言本身与Java语言并没有任何关系，只是互相炒作而已。就像雷锋和雷峰塔的关系一样。JavaScript与Java无关系示意图，如图所示。

JavaScript



≠

Java





1.1 什么是JavaScript

1.1.2 JavaScript与Java的关系

- 1995年12月4日，网景公司与Sun公司联合发布了JavaScript语言。
- 1996年3月，网景浏览器(Netscape Navigator) 2.0浏览器正式内置JavaScript脚本语言。





1.1 什么是JavaScript

1.1.3 为什么学习JavaScript

- JavaScript语言既适合作为学习编程的入门语言，也适合用于日常开发的工作语言。它是目前最有希望、前途最光明的计算机语言之一。其特点如图所示。



- (1) 简单性**
- 相对于其他编程语言，JavaScript语法更加简单，适合初学者学习。JavaScript的很多语法都是借鉴大家熟悉的Java或C++的语法，这对于学习JavaScript具有非常大的帮助。
- 除了语言本身的简单以外，JavaScript开发环境也是相当的简单，只需要文本编辑器就可以编写JavaScript程序。它不需要对开发环境做过多的配置，快速入门是这种语言的最大特点。





1.1 什么是JavaScript

1.1.3 为什么学习JavaScript

- **(2) 安全性**
- JavaScript语言主要用来完成浏览器与访问者之间的交互效果，并不会涉及到数据存储服务器、网络文档修改或删除等功能。例如，用户的账号、密码、支付等功能的实现，均不涉及，因此没有安全性问题。从另外一个角度来说，JavaScript本身没有操作数据的功能，所以说学习JavaScript这门语言本身就是“非常安全”的。
- **(3) 广泛性**
- JavaScript可以应用于被浏览器解析的Web端、可以作为后端语言使用、还可以用于构建移动端 APP等。此外，JavaScript 还可以用来构建桌面应用，世界上最流行的2D游戏引擎之一 Cocos2d 和最流行的3D游戏引擎之一Unity3D可以由JavaScript来开发。





1.1 什么是JavaScript

1.1.3 为什么学习JavaScript

- **(4) 开放性**
- JavaScript属于客户端脚本语言，可以在浏览器中直接查看到其源代码，对于学习和借鉴都很有帮助，可以说JavaScript这门语言具有完全的开放性。
- 在大型互联网公司的不断推广下，JavaScript生态圈也在不断的完善之中，各种类库、API接口层出不穷，从而使其生态环境蒸蒸日上。使用各种现成的框架或类库，可以快速地搭建大型应用，同时其开放的环境使得这门语言成为当下最流行的编程语言之一。





[返回目录](#)

1.2 JavaScript的特性

1.2.1

● 语言特性

1.2.2

● JavaScript与ECMAScript

1.2.3

● 与HTML和CSS的关系

1.2.4

● JavaScript的组成

1.2.5

● JavaScript版本





1.2 JavaScript的特性

1.2.1 语言特性

- 每种编程语言都有自己的语言特性，只有了解语言的独到之处，才能更好地理解这门语言，JavaScript语言特性如图所示。



- **1.解释型**
- 编译型语言在计算机运行代码前，先把代码翻译成计算机可以理解的文件，如Java、C++等属于编译型语言；而解释型语言则不同，解释型语言的程序不需要在运行前编译，在运行程序时才编译，如JavaScript、PHP等属于解释型语言。





1.2 JavaScript的特性

1.2.1 语言特性

- 解释型语言的优点是可移植性较好，只要有解释环境，可在不同的操作系统上运行。代码修改后即可运行，无须编译，上手方便、快速。但缺点是需要解释环境，运行起来比编译型语言慢，占用资源多，代码效率低。
- **2. 弱类型**
- 弱类型语言是相对强类型语言而言的。在强类型语言中，变量类型有多种，如int、char、float、boolean等，不同的类型相互转换有时需要强制转换。而JavaScript只有一种类型var，当其为变量赋值时会自动判断类型并进行转换，因此JavaScript是弱类型语言。
- 弱类型语言的优点是易于学习、语言表达简单易懂、代码更优雅、开发周期更短、更加偏向逻辑设计。缺点是程序可靠性差、调试烦琐、变量不规范、性能低下。





1.2 JavaScript的特性

1.2.1 语言特性

- **3. 动态性**
 - 动态性语言指在变量定义时不一定进行赋值操作，只需在使用时作赋值操作即可。这种方式使得代码更灵活、方便。在JavaScript中有多处用到动态性，如获取元素、原型等。
- **4. 事件驱动**
 - JavaScript可以直接对用户或客户输入做出响应，无须经过web程序。它对用户的响应以事件驱动的方式进行，即由某种操作动作引起相应的事件响应，如单击、移动窗口、选择菜单等。
- **5. 跨平台**
 - JavaScript依赖于浏览器本身，与操作环境无关。只要计算机能运行浏览器，且浏览器支持JavaScript，即可正确执行，从而实现“编写一次，走遍天下”





1.2 JavaScript的特性

1.2.2 JavaScript与ECMAScript关系

- 为了与微软公司竞争，1996年11月，网景公司决定将 JavaScript 提交给国际标准化组织ECMA，希望这门语言能够成为国际标准。次年，ECMA 发布262号标准文件（ECMA-262）的第一版，规定了浏览器脚本语言标准，并将此语言称为ECMAScript，版本为1.0版。
- 标准开始就是针对 JavaScript 语言而制定，但名称为ECMA JavaScript而不是JavaScript有两个原因。一是商标，Java 是 Sun 公司的商标，根据授权协议，只有网景公司可以合法地使用 JavaScript 这个名字，且 JavaScript 本身也已经被网景公司注册为商标；二是体现这门语言的制定者是 ECMA，而不是网景，这有利于保证JavaScript语言的开放性和中立性。因此，ECMAScript 和 JavaScript 的关系为前者是后者的规范，后者是前者的实现。

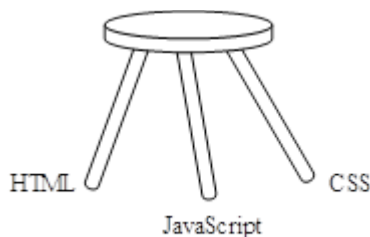




1.2 JavaScript的特性

1.2.3 与HTML和CSS的关系

- JavaScript语言主要用来完成浏览器与访问者之间的交互效果，需要和HTML、CSS配合使用，三者就像板凳的三条腿，缺一不可。Web前端三大核心技术的关系，如图所示。如果想要对JavaScript语言有更深入的理解，需要对HTML和CSS有一定的认知。



- 那么，三者是如何进行分工的呢？下面用一个盖房子的例子来描述三者之间的关系，首先需要把房子的地基和骨架搭建好，即良好的结构（HTML）。然后给房子刷上油漆和添加窗户，对房子样式进行美化（CSS）。最后给房子添加电梯和地暖，与住户进行一些行为上的交互（JavaScript），这样房子才算搭建完毕。





1.2 JavaScript的特性

1.2.3 与HTML和CSS的关系

- 总而言之，首先通过HTML搭建网页的结构，然后用CSS设置网页的样式，最后通过JavaScript添加网页的交互效果。从而完成整个前端的开发过程。具体分工如下：
HTML负责 结构、CSS负责 样式、JavaScript 负责行为，根据W3C组织规定的Web标准，应该尽可能让三者进行分离式开发，最后再整合到一起，从而实现最终的效果。

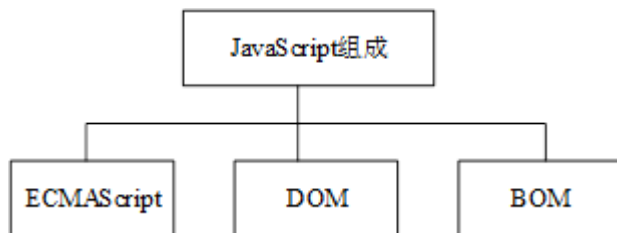




1.2 JavaScript的特性

1.2.4 JavaScript的组成

- JavaScript的组成包含三大部分，分别为ECMAScript、DOM和BOM。
JavaScript组成，如图所示。



- 1.ECMAScript**
- ECMAScript是JavaScript语言的规范，是JavaScript的核心内容，它描述了语言的基本语法和数据类型等。ECMAScript是一套标准，定义了JavaScript编码方式与语言特性。

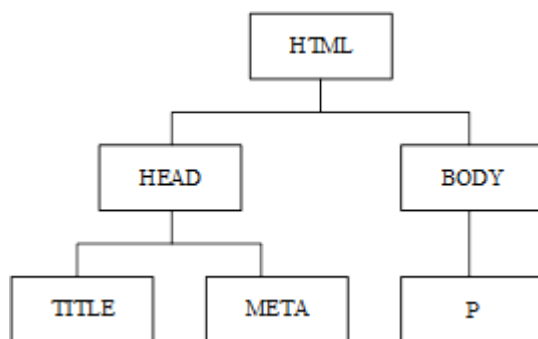




1.2 JavaScript的特性

1.2.4 JavaScript的组成

- **2.DOM**
- 文档对象模型（ Document Object Model ，简称DOM ）是W3C组织推荐的处理可扩展标记语言（ HTML或XML ）的标准编程接口（ API ）。网页上组织页面（ 或文档 ）的对象被组织在一个树形结构中，通过DOM操作的方式可以让页面跟JavaScript进行通信或交互。DOM树展示，如图所示。





1.2 JavaScript的特性

1.2.4 JavaScript的组成

- 通俗地说，DOM就是获取或设置HTML和CSS代码的功能实现。下面通过示例可以看出HTML页面的每个部分都是一个节点的衍生物，具体示例如下：

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <title>demo</title>
6 </head>
7 <body>
8     <p>hello JavaScript!</p>
9 </body>
10 </html>
```

- DOM 通过创建树来表示文档中的所有标签结构（即节点），从而使开发者对文档的内容和结构具有空前的控制力。用 DOM API可以轻松地删除、添加和替换节点。





1.2 JavaScript的特性

1.2.4 JavaScript的组成

- **3.BOM**
- 浏览器对象模型（ Browser Object Model，简称BOM）是对浏览器窗口进行访问和操作的功能接口。例如，弹出新的浏览器窗口、获取浏览器信息等。注意，BOM是作为JavaScript的一部分而不是作为W3C组织的标准，每款浏览器都有自己的实现方式，这会导致BOM代码的兼容性不如ECMAScript和DOM代码的兼容性。





1.2 JavaScript的特性

1.2.5 JavaScript版本

- JavaScript的版本主要指核心部分ECMAScript的版本。1998年和1999年，分别发布了ECMAScript2.0和ECMAScript3.0。ECMAScript3.0版是一个巨大的成功，在业界得到广泛支持，成为通行标准，为JavaScript 语言的基本语法奠定了基础。直到目前，初学者开始学习 JavaScript，还是在学3.0版的语法。
- 2000年，ECMAScript 4.0 开始酝酿。但是由于此版本太过激进，对 3.0进行彻底升级，导致标准委员会的一些成员不愿意接受。因此，中止 ECMAScript 4.0 的开发，将其中涉及现有功能改善的一小部分，临时发布为 ECMAScript 3.1，而将其其他激进的设想扩大范围，放入以后的版本中。





1.2 JavaScript的特性

1.2.5 JavaScript版本

- 2009年12月，ECMAScript3.1正式改名为 ECMAScript5.0（ ECMAScript3.0和ECMAScript5.0语法基本相同，并保持兼容模式），并正式发布。2011年6月，ECMAScript 5.1 版本发布，它是当前最为稳定的一个版本。
- 2015年6月，ECMAScript 6.0 正式通过，成为国际标准，此版本对JavaScript语言有了较大的改进，提供了很多特性和新功能。
- 本教材主要以ECMAScript5.1版本作为核心进行讲解，同时也会专门对ECMAScript6.0进行讲解。





1.3 编写第一个JS程序



[返回目录](#)

1.3.1

● JavaScript编辑工具

1.3.2

● JavaScript在HTML的引用方式

1.3.3

● 注释、空格、分号

1.3.4

● JavaScript调试控制台

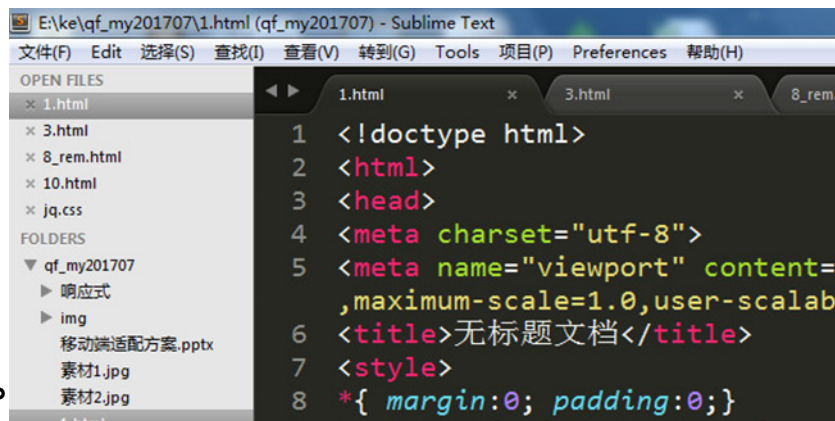




1.3 编写第一个JS程序

1.3.1 JavaScript编辑工具

- 俗话说，工欲善其事，必先利其器。在进行JavaScript开发时，首先需要选择适合的编辑工具。常见的网页编码工具有以下几种。
- Dreamweaver，它是Adobe公司出品。
- Sublime Text，它是程序员Jon Skinner开发的。
- WebStorm，它是JetBrains公司出品。
- HBuilder，它是DCloud公司出品。
- 推荐使用Sublime Text，它是一款强大的网页编辑器，且对JS支持情况非常好，强大的插件功能，可适应各种类库的编码操作。如图所示。





1.3 编写第一个JS程序

1.3.2 JavaScript在HTML中的引用方式

- CSS在HTML中有三种引入方式，JavaScript与CSS在HTML引入方式完全相同，如下所示。
- **1. 行间引入方式（行内式）**
- 行间引入方式是通过HTML标签的属性进行操作，一般不推荐使用，因为它违背了Web标准中规定的结构、样式、行为三者分离开发的原则，接下来通过案例演示行间JavaScript展示效果，具体如例所示。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>编写第一个 JS 程序</title>
6 </head>
7 <body>
8 <div onclick="alert('Hello JavaScript!')">点击</div>
9 </body>
10 </html>
```





1.3 编写第一个JS程序

1.3.2 JavaScript在HTML中的引用方式

- **2.内部引入方式**
- 内部引入方式是通过<script>标签的方式进行设置，类似于CSS中的<style>标签引入方式，接下来通过案例演示内部JavaScript展示效果，具体如例所示。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>编写第一个 JS 程序</title>
6 </head>
7 <body>
8 <script>
9     // 弹出窗口
10    alert('Hi JavaScript!');
11 </script>
12 </body>
13 </html>
```

- <script>标签可以写在HTML结构中的任意位置，比如<head>中或<body>中。需要注意，alert方式会阻止后续代码的执行，因此，当不单击弹窗的“确定”按钮时，会一直显示加载的状态，当单击了“确定”按钮后，页面加载完成。





1.3 编写第一个JS程序

1.3.2 JavaScript在HTML中的引用方式

- **3. 外部引入方式**

- 外部引入方式，也是通过<script>标签的方式进行设置，但引入的是一个外部js文件，通过src属性引入链接的地址。类似于CSS中的<link>标签引入方式，具体示例如下：

```
<script src="js/hello.js"></script>
```

- 执行代码和展示效果都与内部引入方式类似，只是采用一个外部的JavaScript文件，这样可以在多个页面中共享同一段JavaScript代码。





1.3 编写第一个JS程序

1.3.3 注释、空格、分号

- JavaScript中的注释分为单行代码注释和多行代码注释两种，下面将分别介绍这两种注释方式。

- (1) 单行代码注释，通过//的方式，具体示例代码如下：

```
<script>
    //alert('Hi JavaScript!');
</script>
```

- (2) 多行代码注释，通过/* */的方式，具体示例代码如下：

```
<script>
    /*alert('Hi JavaScript!');
    alert('Hello JavaScript!');*/
</script>
```

- JavaScript语法对空格没有限制，两个语法之间可以通过多个空格隔开，需要注意，不能对一个完整的语法进行空格分隔，否则将会报错。具体示例代码如下。

```
<script>
    alert(      'Hi JavaScript!'      ); // ✓
    alert      ('Hi JavaScript!');     // ✓
    ale        rt('Hi JavaScript!');   // ✕
</script>
```





1.3 编写第一个JS程序

1.3.3 注释、空格、分号

- 一般在语句结束后，需要添加分号，表示此行代码已结束，但并不是强制的，养成好的编程习惯很重要。





1.3 编写第一个JS程序

1.3.4 JavaScript调试控制台

- 一般情况下，可以利用alert()方式打印JavaScript计算后的值，从而进行程序的测试。但alert()方式会阻止后续代码的执行，连续测试会很不方便。JavaScript中提供了另外一种测试方式，即console.log()方式，此方式与alert()类似，可以打印出相关的值，但二者的区别在于console.log()方法不会阻止后续代码的运行，因此，console.log()方式适合做连续的输出操作。
- 如果想要查看console.log()方式打印的数据值，需要通过浏览器自带的调试控制台进行打印输出。例如Chrome浏览器，按F12键即可打开调试控制台。





1.3 编写第一个JS程序

1.3.4 JavaScript调试控制台

- 接下来通过案例演示Chrome浏览器调试控制台，具体如例所示。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>编写第一个 JS 程序</title>
6 </head>
7 <body>
8 <script>
9     //在 Chrome 调试控制台打印相关信息
10    console.log('Hi JavaScript!');
11 </script>
12 </body>
13 </html>
```

- 首先编写案例，然后在Chrome浏览器中运行当前案例，运行后再按下F12键，打开Chrome浏览器内置的调试工具，然后选择Console选项，即可以查看到案例中console.log()语法中的代码信息。





1.3 编写第一个JS程序

1.3.4 JavaScript调试控制台

- 如果尝试添加多个console.log()代码输出，可以看到同时会执行多条语句。但是如果尝试添加多个alert()代码输出时，只会弹出一个提示框，只有当单击提示框的“确认”按钮后，才会执行下一次的弹窗提示。





本章小结

- 通过本章的介绍，希望读者能够初步了解什么是JavaScript及学习JavaScript的必要性。重点掌握JavaScript的特性和基本组成模式，编写并运行第一个JavaScript程序，通过浏览器查看运行效果。JavaScript环境搭建非常简单，编辑代码也非常方便。只要对HTML和CSS有一定的基础，学习JavaScript是非常轻松的一件事情。



THANK YOU

