

JavaScript 简介

了解 JavaScript

JavaScript 基础

了解 JavaScript

- 前言
- JavaScript 介绍
- 计算机组成

网页、网站和应用程序

- 网页：单独的一个页面。
- 网站：一系列相关的页面组合到一起。
- 应用程序：可以和用户产生交互，并实现某种功能。

前端三层

- | | | |
|--------------|-----|--------------|
| • HTML | 结构层 | 从语义的角度描述页面结构 |
| • CSS | 样式层 | 从美观的角度描述页面样式 |
| • JavaScript | 行为层 | 从交互的角度描述页面行为 |

JavaScript 的应用场景

JavaScript 发展到现在几乎无所不能，常见的应用场景有：

- 1. 网页特效
- 2. 服务端开发(Node.js)
- 3. 命令行工具(Node.js)
- 4. 桌面程序(Electron)
- 5. App(Cordova)
- 6. 控制硬件-物联网(Ruff)
- 7. 游戏开发(cocos2d-js)
-

JavaScript 基础

了解 JavaScript

- 前言
- **JavaScript 介绍**
- 计算机组成

JavaScript 是什么

- JavaScript 是一种运行在客户端 的脚本语言，最早是在 HTML（标准通用标记语言下的一个应用）网页上使用，用来给 HTML 网页增加动态功能。
- 浏览器就是一种运行 JavaScript 脚本语言的客户端，JavaScript 的解释器被称为 JavaScript 引擎，为浏览器的一部分。

JavaScript 简史

- 在 WEB 日益发展的同时，网页的大小和复杂性不断增加，受制于网速的限制，为完成简单的表单验证而频繁地与服务器交换数据只会加重用户的负担，当时走在技术革新最前沿的 Netscape（网景）公司，决定着手开发一种客户端语言，用来处理这种简单的验证。
- 1995 年，就职于 Netscape 公司的 Brendan Eich，开始着手为即将于 1996 年 2 月发布的 Netscape Navigator 2 浏览器开发一种名为 LiveScript 的脚本语言。为了尽快完成 LiveScript 的开发，Netscape 与 Sun 公司建立了一个开发联盟。在 Netscape Navigator 2 正式发布前夕，Netscape 为了搭上媒体热炒 Java 的顺风车，临时把 LiveScript 改名为 JavaScript。
- 由于 JavaScript 1.0 获得的关注度越来越高，1996 年，微软就在其 Internet Explorer 3 中加入了名为 JScript 的 JavaScript 实现，这意味着有了两个不同的 JavaScript 版本，导致 JavaScript 没有一个标准化的语法和特性。
- 1997 年，以 JavaScript 1.1 为蓝本的建议被提交给了欧洲计算机制造商协会（ECMA, European Computer Manufacturers Association）。该协会指定 39 号技术委员会（TC39, Technical Committee #39）负责“标准化一种通用、跨平台、供应商中立的脚本语言的语法和语义”。TC39 由来自 Netscape、Sun、微软、Borland 及其他关注脚本语言发展的公司的程序员组成，他们经过数月的努力完成了 ECMA-262 标准，定义一种名为 ECMAScript 的新脚本语言。

ECMAScript 发展过程

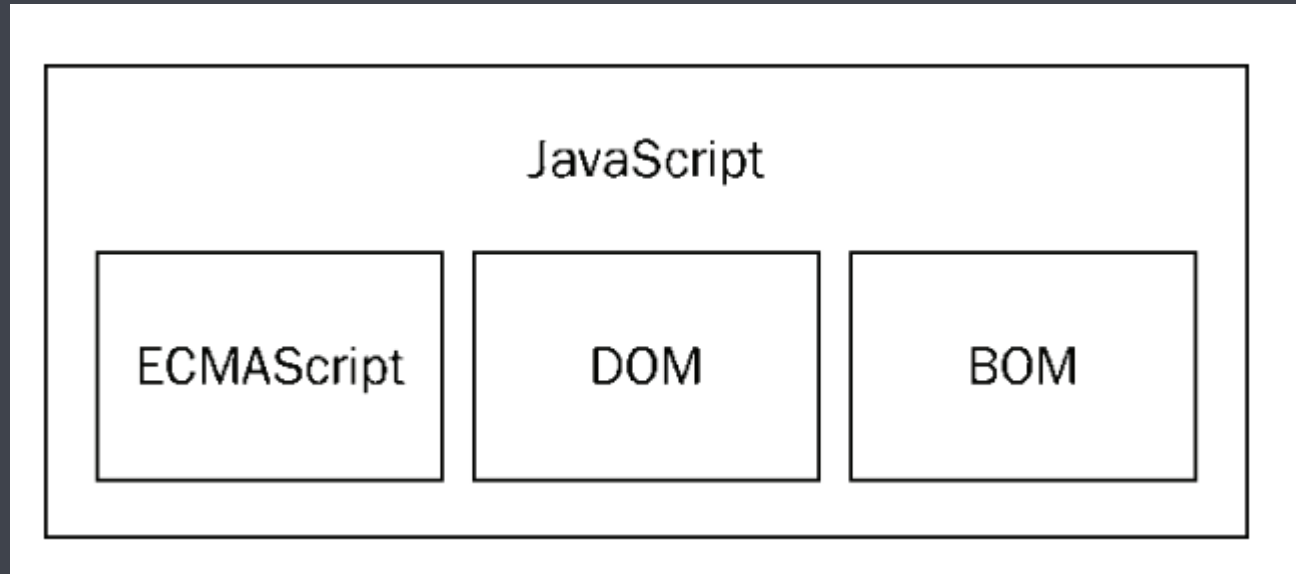
版本	时间	说明
ECMAScript 1	1997.06	
ECMAScript 2	1998.06	
ECMAScript 3	1999.12	成为 JavaScript 的通行标准，得到了广泛支持。
ECMAScript 4	2007.10	ECMAScript 4.0 版草案发布，对 3.0 版做了大幅升级。草案发布后，由于 4.0 版的目标过于激进，各方对于是否通过这个标准，发生了严重分歧。以 Yahoo、Microsoft、Google 为首的大公司，反对 JavaScript 的大幅升级，主张小幅改动；以 JavaScript 创造者 Brendan Eich 为首的 Mozilla 公司，则坚持当前的草案。
	2008.07	由于各方分歧太大，争论过于激进，ECMA 开会决定，中止 ECMAScript 4.0 的开发，将其中涉及现有功能改善的一小部分，发布为 ECMAScript 3.1，而将其他激进的设想扩大范围，放入以后的版本，由于会议的气氛，该版本的项目代号起名为 Harmony（和谐）。会后不久，ECMAScript 3.1 就改名为 ECMAScript 5。
ECMAScript 5	2009.12	ECMAScript 5.0 版正式发布。Harmony 项目则一分为二，一些较为可行的设想定名为 JavaScript.next 继续开发，后来演变成 ECMAScript 6；一些不是很成熟的设想，则被视为 JavaScript.next.next，在更远的将来再考虑推出。
ECMAScript 5.1	2011.06	成为 ISO 国际标准（ISO/IEC 16262:2011）
ECMAScript 6/ECMAScript 2015	2015.06.17	ECMAScript 6 发布正式版本，并且更名为“ECMAScript 2015”。

ECMAScript 的应用

- Javascript, JScript, ActionScript 等脚本语言都是基于 ECMAScript 标准实现的。
- 在 JavaScript, JScript 和 ActionScript 中声明变量, 操作数组等语法完全一样, 因为它们都是 ECMAScript。但是在操作浏览器对象等方面又有各自独特的方法, 这些都是各自语言的扩展。

JavaScript 的组成

- JavaScript 是由 ECMAScript, DOM 和 BOM 三者组成的。



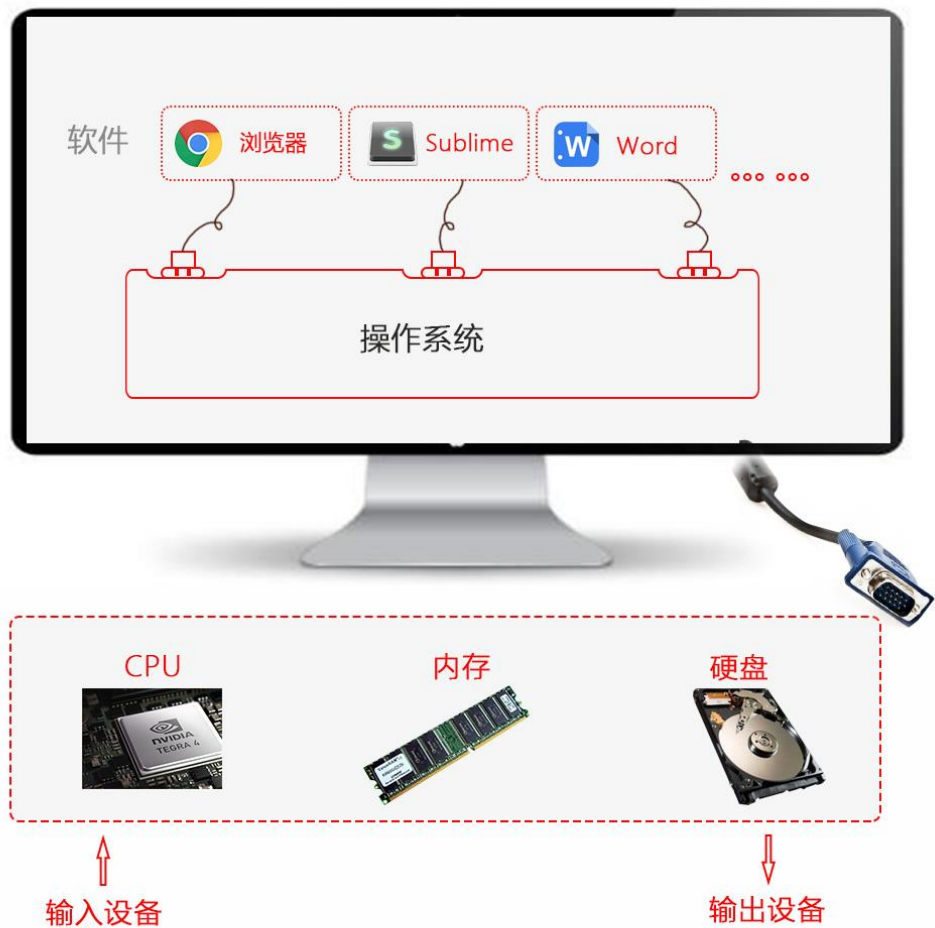
JavaScript 基础

了解 JavaScript

- 前言
- JavaScript 介绍
- **计算机组成**

计算机组成

计算机软硬件关系示意图



软件

- 应用软件：浏览器(Chrome/IE/Firefox)、QQ、Sublime、Word
- 系统软件：Windows、Linux、mac OS



硬件

- 三大件：CPU、内存、硬盘 -- 主板
- 输入设备：鼠标、键盘、手写板、摄像头等
- 输出设备：显示器、打印机、投影仪等



JavaScript 书写语法

掌握 JavaScript 的基本语法

JavaScript 书写语法

掌握 JavaScript 的基本语法

- 书写位置
- 注释
- alert 语句
- prompt 语句
- console 控制台

JavaScript 的书写位置

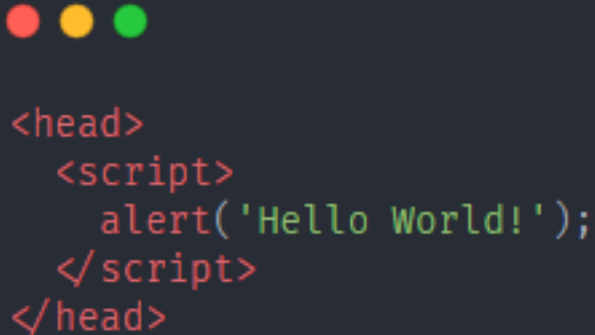
- 写在行内



```
<input type="button" value="按钮" onclick="alert('Hello World')" />
```

JavaScript 的书写位置


- 写在 html 中的 `<script>` 标签中



```
<head>
  <script>
    alert('Hello World!');
  </script>
</head>
```

JavaScript 的书写位置

- 写在外部 js 文件中，在页面引入



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

- 注意：引用外部 js 文件的 `<script>` 标签中不可以写 JavaScript 的代码，自定义的 JavaScript 代码，需要书写在另一个新的 `<script>` 标签中。

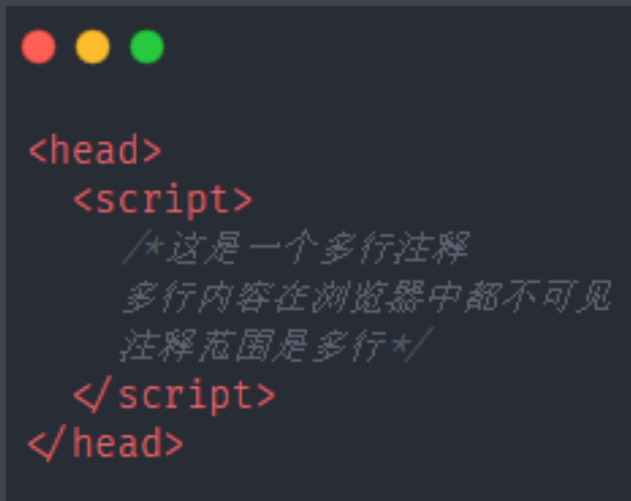
JavaScript 书写语法

掌握 JavaScript 的基本语法

- 书写位置
- **注释**
- alert 语句
- prompt 语句
- console 控制台

JavaScript 的注释

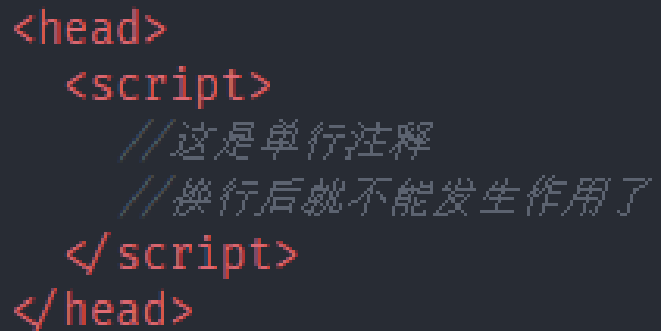
- js 注释分为两种：块级注释和单行注释。
- 块级注释：又叫做多行注释，作用范围是选中的多行，写作 `/**/`。



```
<head>
  <script>
    /*这是一个多行注释
    多行内容在浏览器中都不可见
    注释范围是多行*/
  </script>
</head>
```

JavaScript 的注释

- 单行注释：作用范围只有符号后面的一行，写作 `//`。



```
<head>
  <script>
    //这是单行注释
    //换行后就不能发生作用了
  </script>
</head>
```

JavaScript 的注释

- VScode 快捷键:
- 单行注释: 注释和解开注释都是 `ctrl+ /`
- 多行注释: `ctrl+shift+ /`

JavaScript 书写语法

掌握 JavaScript 的基本语法

- 书写位置
- 注释
- **alert 语句**
- prompt 语句
- console 控制台

为了方便了解 JS 的书写语法，以及后面知识的学习，我们需要先学习几条输出、输入语句。

alert() 语句

- alert: 警示、警告。
- 作用: 在浏览器中弹出一个警示框, 警示框的警示内容可以人为自定义。
- alert 语句是一个 js 内置好的功能 (函数、方法), 要想实现功能必须在 alert 关键字后面加小括号执行, 自定义的内容需要传递给小括号内的参数, 输出时参数位置的内容会出现在弹框位置。



```
<head>
  <script>
    alert("这是一个警示框");
  </script>
</head>
```

语法

- 语法1: alert 语句作为一个函数, 如果要执行, 必须在后面紧跟着添加小括号。
- 语法2: alert 语句作为一个函数, 小括号内部可以传递参数, 根据数据类型不同, 有不同的语法要求, 例如文字内容 (字符串) 必须写在一对引号内部。
- 语法3: 如果需要添加一对引号, 要么是单引号, 要么是双引号, 不能一单一双。
- 语法4: 所有的有特殊功能的符号必须是英文字符。
- 语法5: 语句后面的分号必要性。
- 语法6: js 对换行、缩进、空格也不敏感。
- 语法7: 如果 js 中没有特殊控制结构, 代码都是从上往下、从左往右进行加载。

JavaScript 书写语法

掌握 JavaScript 的基本语法

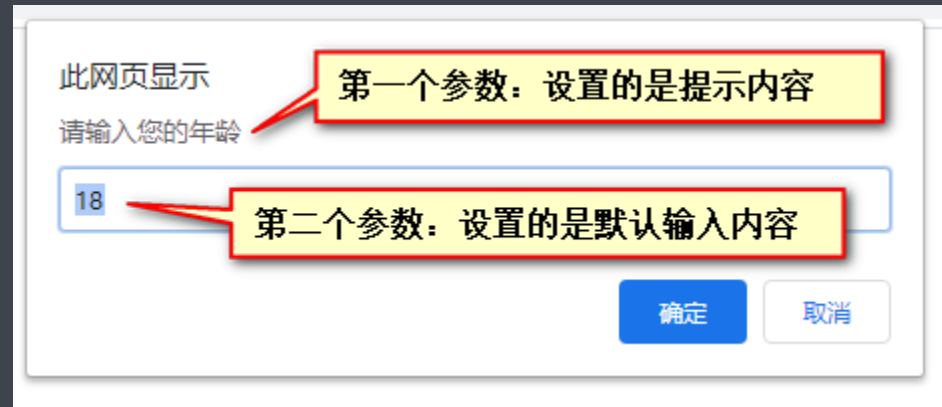
- 书写位置
- 注释
- alert 语句
- **prompt 语句**
- console 控制台

prompt() 语句

- prompt: 提示。
- 作用: 弹出一个对话框, 内部有一个提示语句以及一个输入框, 可以在输入框中根据提示任意输入内容。
- prompt 语句也是 js 内置的一个功能, 必须加小括号执行, 有两个参数可以进行传递, 每个参数一般都是字符串类型, 必须加引号, 两个参数中间用逗号分隔, 引号和逗号都必须是英文输入法状态。

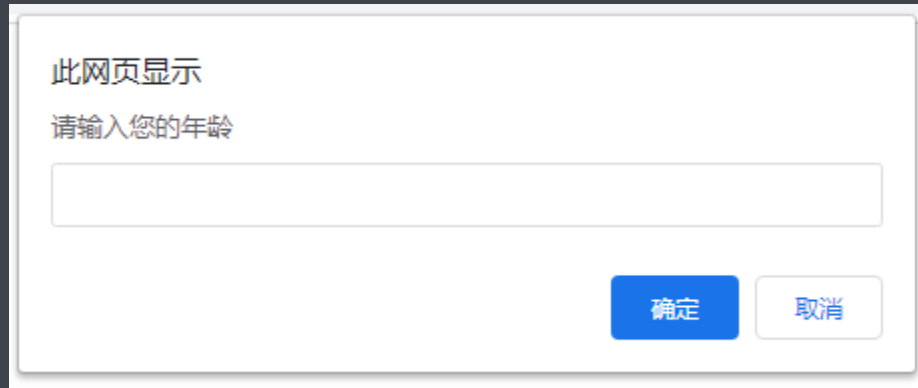
prompt() 语句

- `prompt("请输入您的年龄","18");`



prompt() 语句

- 第二个参数位置的值，可以删除并重新输入新的内容，甚至代码中可以不写第二个参数。
- `prompt("请输入您的年龄");`



此网页显示

请输入您的年龄

确定 取消

JavaScript 书写语法

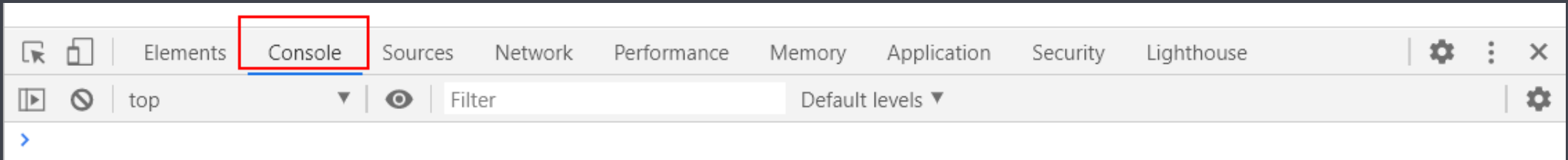
掌握 JavaScript 的基本语法

- 书写位置
- 注释
- alert 语句
- prompt 语句
- **console 控制台**

console 控制台

- 在大部分浏览器中，都有一个控制台，内部可以查看 HTML、css 代码，甚至调试代码错误。
- 浏览器中右键点击审查元素（检查），都可以打开控制台。
- 快捷键： F12.
- 控制台中有一个 console（控制台） 的选项面板，在 js 中有非常重要的作用。

chrome 浏览器的控制台

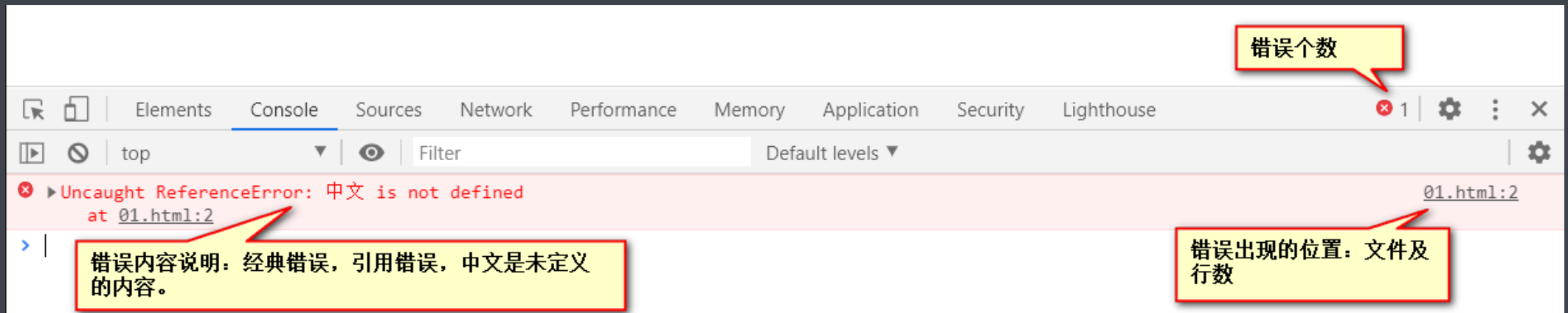


IE 浏览器的控制台



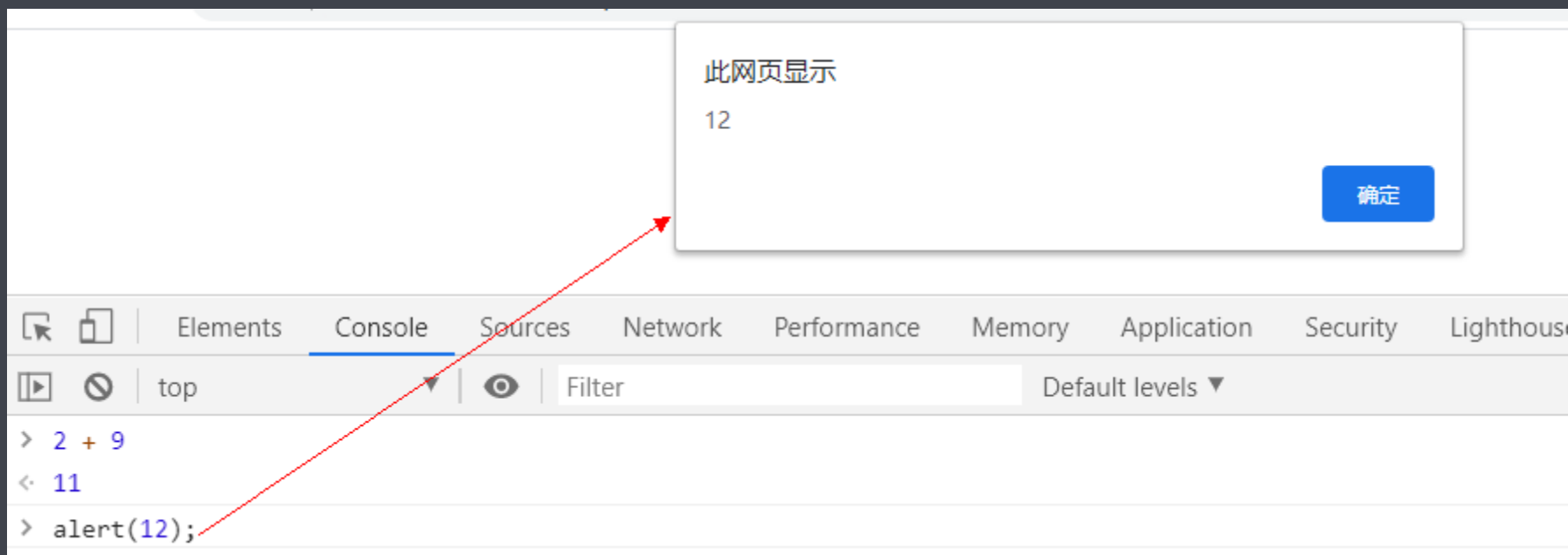
控制台功能

- 作用一：帮助程序员调试程序中出现的 bug。可以提示错误的个数、错误所在的行数、错误类型。



控制台功能

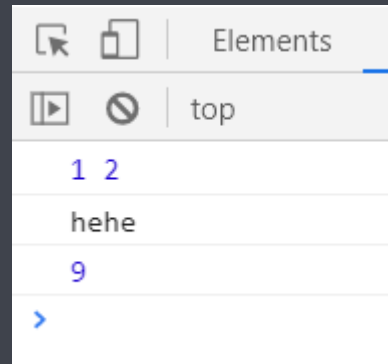
- 作用二：可以在控制台中直接书写一些语句，进行执行。



控制台功能

- 作用三：js 有一个内置对象 console，内部封装了大量的属性和方法（函数），可以通过给 console 对象打点调用方法名()、属性名方式去执行。有一个叫做 log（日志）方法，可以实现参数位置自定义内容在控制台输出。

```
console.log(1,2);  
console.log("hehe");  
console.log(4+5);
```



拉勾教育

— 互联网人实战大学 —



下载「拉勾教育App」
获取更多内容