

# JavaScript的基本语法

王红元 coderwhy

# 目录

## content



**1** JavaScript编写方式

**2** noscript元素的使用

**3** JavaScript注意事项

**4** JavaScript交互方式

**5** JavaScript语句和分号

**6** JavaScript注释方式

# JavaScript的编写方式

## ■ 位置一：HTML代码行内（不推荐）

```
<!-- 1. 百度一下 -->  
<a href="javascript:alert('百度一下')" onclick="alert('点击百度一下')">百度一下</a>
```

## ■ 位置二：script标签中

```
<a class="google" href="#">Google一下</a>  
<script>  
  const googleEl = document.querySelector(".google")  
  googleEl.onclick = function() {  
    alert("Google一下")  
  }  
</script>
```

## ■ 位置三：外部的script文件

□ 需要通过script元素的src属性来引入JavaScript文件

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

```
const bingEl = document.querySelector(".bing")  
bingEl.onclick = function() {  
  alert("Bing一下")  
}
```

# <noscript>元素

## ■ 如果运行的浏览器不支持JavaScript, 那么我们如何给用户更好的提示呢?

- 针对早期浏览器不支持 JavaScript 的问题, 需要一个页面**优雅降级的处理方案**;
- 最终, **<noscript> 元素**出现, 被用于给不支持 JavaScript 的浏览器提供替代内容;

## ■ 下面的情况下, 浏览器将显示包含在<noscript>中的内容:

- 浏览器不支持脚本;
- 浏览器对脚本的支持被关闭。

```
<body>  
  ...  
  <noscript>  
    <p>您的浏览器不支持或者关闭运行JavaScript</p>  
  </noscript>  
</body>
```



### 隐私设置和安全性



#### 网站设置

控制网站可以使用和显示什么信息 (如位置信息、摄像头、弹出式窗口及其他)



#### JavaScript

网站可以使用 JavaScript

#### 默认行为

网站会在您访问时自动采用此设置



<> 网站可以使用 JavaScript



不允许网站使用 JavaScript

# JavaScript编写的注意事项

## ■ 注意一: script元素不能写成单标签

- 在外联式引用js文件时, script标签中不可以写JavaScript代码, 并且script标签不能写成单标签;
- 即不能写成<script src= "index.js" />;

## ■ 注意二: 省略type属性

- 在以前的代码中, <script>标签中会使用 type= "text/javascript" ;
- 现在可不写这个代码了, 因为JavaScript 是所有现代浏览器以及 HTML5 中的默认脚本语言;

## ■ 注意三: 加载顺序

- 作为HTML文档内容的一部分, JavaScript默认遵循HTML文档的加载顺序, 即自上而下的加载顺序;
- 推荐将JavaScript代码和编写位置放在body子元素的最后一行;

## ■ 注意四: JavaScript代码严格区分大小写

- HTML元素和CSS属性不区分大小写, 但是在JavaScript中严格区分大小写;

## ■ 后续补充: script元素还有defer、async属性, 我们后续再详细讲解。



# JavaScript的交互方式

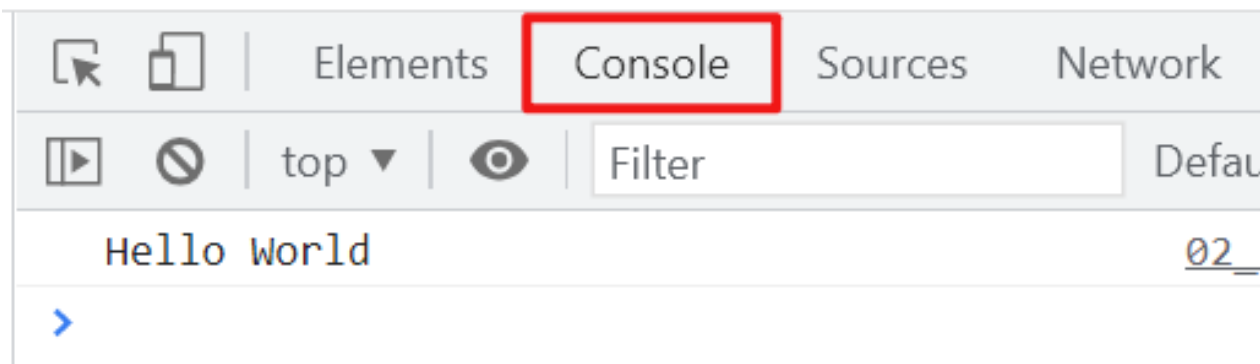
■ JavaScript有如下和用户交互的手段：

□ 最常见的是通过console.log, 目前大家掌握这个即可;

交互方法	方法说明	效果查看
alert	接受一个参数	弹窗查看
console.log	接受多个参数	在浏览器控制台查看
document.write	接受多个字符串	在浏览器页面查看
prompt	接受一个参数	在浏览器接受用户输入

# Chrome的调试工具

- 在前面我们利用Chrome的调试工具来调试了HTML、CSS，它也可以帮助我们来调试JavaScript。
- 当我们在JavaScript中通过console函数显示一些内容时，也可以使用Chrome浏览器来查看：



## ■ 另外补充几点：

- 1.如果在代码中出现了错误，那么可以在console中显示错误；
- 2.console中有一个 > 标志，它表示控制台的命令行
  - ✓ 在命令行中我们可以直接编写JavaScript代码，按下enter会执行代码；
  - ✓ 如果希望编写多行代码，可以按下shift+enter来进行换行编写；
- 3.在后续我们还会学习如何通过debug方式来调试、查看代码的执行流程；

# JavaScript语句和分号

■ **语句是向浏览器发出的指令**，通常表达一个**操作或者行为（Action）**。

- 语句英文是**Statements**；
- 比如我们前面编写的每一行代码都是一个语句，用于告知浏览器一条执行的命令；

```
alert("Hello World");  
alert("Hello Coderwhy");
```

■ **通常每条语句的后面我们会添加一个分号**，表示语句的结束：

- 分号的英文是**semicolon**
- 当存在换行符（line break）时，在大多数情况下可以**省略分号**；
- JavaScript 将**换行符理解成“隐式”的分号**；
- 这也被称之为**自动插入分号**（an automatic semicolon）；

■ **推荐：**

- 前期在对JavaScript语法**不熟悉**的情况**推荐添加分号**；
- 后期对JavaScript语法**熟练**的情况下，**任意**！



# JavaScript的注释

- 在HTML、CSS中我们都添加过注释，JavaScript也可以添加注释。
- JavaScript的注释主要分为三种：
  - 单行注释
  - 多行注释
  - 文档注释 (VSCode中需要在单独的JavaScript文件中编写才有效)

## ■ 详情见代码

```
// 单行注释  
  
/* 多行注释 */
```

```
/**  
 * 向某人打招呼  
 * @param {string} name 姓名  
 * @param {number} age 年龄  
 */  
function sayHello(name, age) {  
  
}  
  
sayHello("why", 18)
```

- 注意：JavaScript也不支持注释的嵌套

# VSCode插件和配置

## ■ 推荐一个VSCode的插件：（个人经常使用的）

- ES7+ React/Redux/React-Native snippets
- 这个插件是在react开发中会使用到的，但是我经常用到它里面的打印语句；

## ■ 另外再推荐一个插件：

- Bracket Pair Colorizer 2，但是该插件已经不再推荐使用；
- 因为VSCode已经内置了该功能，我们可以直接通过VSCode的配置来达到插件的效果；
- 如何配置呢？

```
"editor.bracketPairColorization.enabled": true,  
"editor.guides.bracketPairs": "active"
```