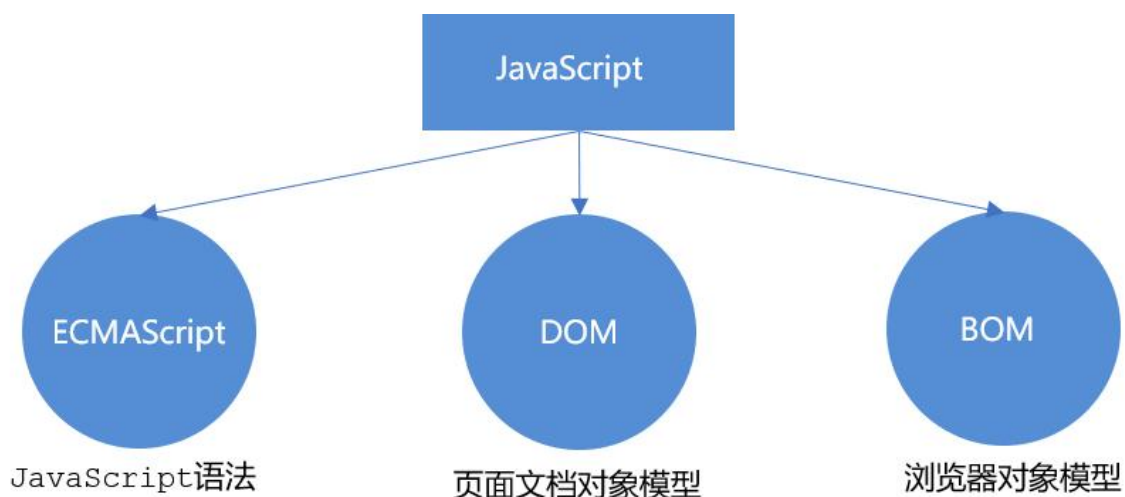


什么是JavaScript?

- JavaScript 是世界上最流行的语言之一，是一种运行在客户端的脚本语言（Script 是脚本的意思）。
- 脚本语言：不需要编译，运行过程中由 js 解释器(js 引擎) 逐行来进行解释并执行。
- 现在也可以基于 Node.js 技术进行服务器端编程。



JS 组成

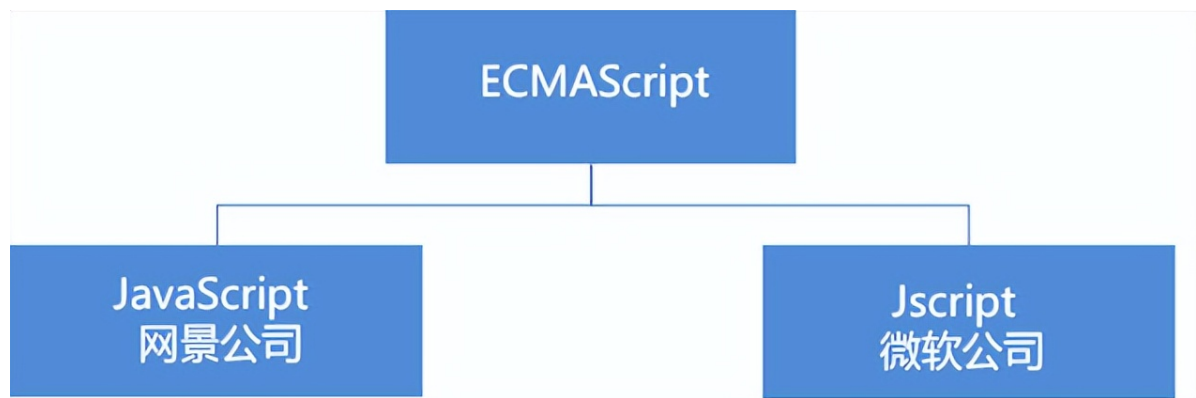


ECMAScript

ECMAScript 是由 ECMA 国际（原欧洲计算机制造商协会）进行标准化的一门编程语言，这种语言在万维网上应用广泛，它往往被称为 JavaScript 或 JScript，但实际上后两者是 ECMAScript 语言的实现和扩展。

公众号：黑猫编程

网址：<https://noi.hioier.co>



ECMAScript: 规定了JS的编程语法和基础核心知识, 是所有浏览器厂商共同遵守的一套JS语法工业标准。

DOM: 文档对象模型

文档对象模型 (DocumentObject Model, 简称DOM), 是W3C组织推荐的处理可扩展标记语言的标准编程接口。通过 DOM 提供的接口可以对页面上的各种元素进行操作 (大小、位置、颜色等)

BOM: 浏览器对象模型

浏览器对象模型(Browser Object Model, 简称BOM) 是指浏览器对象模型, 它提供了独立于内容的、可以与浏览器窗口进行互动的对象结构。通过BOM可以操作浏览器窗口, 比如弹出框、控制浏览器跳转、获取分辨率等。

定义常量和变量

- `const`用来定义常量。
- `let`和`var`用来定义变量, 推荐使用`let`, 作用域更加严格。



版权图片

```
1  <!-- 行内引入 -->
2  <!-- <button onclick="alert('xxx')">点击我</button> -->
3  <!-- 内部引入 -->
4  <script>
5      var b;
6      b = 3;
7      var b = 4;
8      console.log(b);
9      var a = 1;
10     console.log(a);
11     console.log('hello world.');
```

// console.log('hello world.');

```
12
13 </script>
14 <!DOCTYPE html>
15 <html lang="en">
16
17 <head>
18     <meta charset="UTF-8">
19     <meta http-equiv="X-UA-Compatible" content="IE=edge">
20     <meta name="viewport" content="width=device-width, initial-scale=1.0">
21     <title>Document</title>
22     <script>
23         // let a = 'cat';
24         // console.log(a);
25     </script>
26 </head>
27
28 <body>
29     <script>
30         let a;
31         a = 1;
32         // let a = 2;
```

```

33     console.log(a);
34
35     const PI = 3.1415926;
36     // PI = 3.15;
37     console.log(PI, typeof PI);
38 </script>
39
40 <!-- 外部引入 -->
41 <script type="text/javascript" src="./static/js/test01.js">
42     // console.log('xxx');
43 </script>
44
45 <!-- 模块引入 -->
46 <script type="module">
47     import { info, add } from "./static/js/test02.js";
48     console.log(info['name']);
49     console.log(add(1, 2));
50 </script>
51 </body>
52
53 </html>

```

数据类型

JS 把数据类型分为两类：

- 简单数据类型 (Number,String,Boolean,Undefined,Null)
- 复杂数据类型 (object)
- typeof 查看数据类型

简单数据类型	说明	默认值
Number	数字型，包含 整型值和浮点型值，如 21、0.21	0
Boolean	布尔值类型，如 true、false，等价于 1 和 0	false
String	字符串类型，如 "张三" 注意咱们js 里面，字符串都带引号	""
Undefined	let a; 声明了变量 a 但是没有给值，此时 a = undefined	undefined
Null	let a = null; 声明了变量 a 为空值	null

```

1 <script>
2     // let name = prompt('请输入你的名字');
3     // console.log(name, typeof name);
4     console.log(Number.MAX_VALUE);
5     console.log(Number.MIN_VALUE);
6
7     console.log(Infinity);
8     console.log(-Infinity);
9
10    console.log(NaN);
11
12    console.log(isNaN(123));
13    console.log(isNaN('hello'));
14
15    console.log('10' == 10);

```

```
16 console.log('10' === 10);
17 </script>
```

输入输出

方法	说明	归属
alert(msg)	浏览器弹出警示框	浏览器
console.log(msg)	浏览器控制台打印输出信息	浏览器
prompt(info)	浏览器弹出输入框，用户可以输入	浏览器

```
1  const readline = require('readline');
2
3  const rl = readline.createInterface({
4    input: process.stdin,
5    output: process.stdout
6  });
7
8  let n = -1, buf = '';
9
10 rl.on('line', line => {
11   if (n < 0) n = parseInt(line.trim());
12   else {
13     buf = line.split(' ').map(x => {
14       return parseInt(x);
15     });
16     rl.close();
17   }
18 });
19
20 rl.on('close', () => {
21   let s = '';
22   buf.sort();
23
24   for (let i = 0; i < n; i++) s += buf[i].toString() + ' ';
25
26   console.log(s);
27
28   process.exit(0);
29 });
```

转义字符

转义符	解释说明
\n	换行符, n 是 newline 的意思
\\	斜杠 \
\t	tab 缩进
\b	空格, b 是 blank 的意思

```

1 let name = 'cat', age = 28;
2 let info = `My name is ${name}, I\'m ${age} years old.`;
3
4 console.log(info);
5 console.log(`My name is ${name}, I\'m ${age} years old.`);

```

比较运算符

运算符名称	说明	案例	结果
<	小于号	1 < 2	true
>	大于号	1 > 2	false
>=	大于等于号 (大于或者等于)	2 >= 2	true
<=	小于等于号 (小于或者等于)	3 <= 2	false
==	判等号 (会转型)	37 == 37	true
!=	不等号	37 != 37	false
=== !==	全等 要求值和 数据类型都一致	37 === '37'	false