

- 3
- 前端博客
- 前端头条
- 前端俱乐部
- 前端花名册
- 排行榜
- 关注微博

# JavaScript 面试中常见算法问题详解

2017-03-08 • by 王下邀月熊 • 3

编者注：一些常用的JavaScript算法，下文提到的很多问题从算法角度并不一定那么困难，不过用 JavaScript 内置的 API 来完成还是需要一番考量的。

原文：<https://zhuanlan.zhihu.com/p/25308541>

## JavaScript Specification

### 阐述下 JavaScript 中的变量提升

所谓提升，顾名思义即是 JavaScript 会将所有的声明提升到当前作用域的顶部。这也就意味着我们可以在某个变量声明前就使用该变量，不过虽然 JavaScript 会将声明提升到顶部，但是并不会执行真的初始化过程。

### 阐述下 use strict; 的作用

use strict; 顾名思义也就是 JavaScript 会在所谓严格模式下执行，其一个主要的优势在于能够强制开发者避免使用未声明的变量。对于老版本的浏览器或者执行引擎则会自动忽略该指令。

```
// Example of strict mode
"use strict";

catchThemAll();
function catchThemAll() {
  x = 3.14; // Error will be thrown
  return x * x;
}
```

### 解释下什么是 Event Bubbling 以及如何避免

Event Bubbling 即指某个事件不仅会触发当前元素，还会以嵌套顺序传递到父元素中。直观而言就是对于某个子元素的点击事件同样会被父元素的点击事件处理器捕获。避免 Event Bubbling 的方式可以使用event.stopPropagation() 或者 IE 9 以下使用event.cancelBubble。

### == 与 === 的区别是什么

=== 也就是所谓的严格比较，关键的区别在于=== 会同时比较类型与值，而不是仅比较值。

```
// Example of comparators
0 == false; // true
0 === false; // false

2 == '2'; // true
2 === '2'; // false
```

### 解释下 null 与 undefined 的区别

JavaScript 中，null 是一个可以被分配的值，设置为 null 的变量意味着其无值。而 undefined 则代表着某个变量虽然声明了但是尚未进行过任何赋值。

### 解释下 Prototypal Inheritance 与 Classical Inheritance 的区别

在类继承中，类是不可变的，不同的语言中对于多继承的支持也不一样，有些语言中还支持接口、final、abstract 的概念。而原型继承则更为灵活，原型本身是可以可变的，并且对象可能继承自多个原型。

## 数组

### 找出整型数组中乘积最大的三个数

#### 文章目录

- 阐述下 JavaScript 中的变量提升
- 阐述下 use strict; 的作用
- 解释下什么是 Event Bubbling 以及如何避免
- == 与 === 的区别是什么
- 解释下 null 与 undefined 的区别
- 解释下 Prototypal Inheritance 与 Classical Inheritance 的区别
- 找出整型数组中乘积最大的三个数
- 寻找连续数组中的缺失数
- 数组去重
- 数组中元素最大差值计算
- 数组中元素乘积
- 数组交集
- 颠倒字符串
- 乱序同字母字符串
- 会问字符串
- 使用两个栈实现入队与出队
- 判断大括号是否闭合
- 二进制转换
- 二分搜索
- 判断是否为 2 的指数值