

数组是JavaScript最常见的概念之一，它为我们提供了处理数据的许多可能性。您可以在编程开始之初就了解它，在本文中，我想向您展示一些您可能不知道并且可能非常实用的技巧。有助于编码！让我们开始吧。

1. 数组去重

这是一个非常流行的关于JavaScript数组的采访问题，数组去重。这里有一个快速简单的解决方案，可以使用一个新的Set()。我想向您展示两种可能的方法，一种是使用.from()方法，另一种是使用spread操作符(...).

```
var fruits = ["banana", "apple", "orange", "watermelon", "apple", "orange", "grape", "apple"];

// First method
var uniqueFruits = Array.from(new Set(fruits));
console.log(uniqueFruits);
// returns ["banana", "apple", "orange", "watermelon", "grape"]

// Second method
var uniqueFruits2 = [...new Set(fruits)];
console.log(uniqueFruits2);
// returns ["banana", "apple", "orange", "watermelon", "grape"]
```

2. 替换数组中的特定值

有时在创建代码时需要替换数组中的特定值，有一种很好的简单的方法可以做到这一点，我们可以使用.splice(start、valueToRemove、valueToAdd)，并将所有三个参数传递给它，这些参数可以指定我们希望从哪里开始修改、希望修改多少值和新值。

```
var fruits = ["banana", "apple", "orange", "watermelon", "apple", "orange", "grape", "apple"];
fruits.splice(0, 2, "potato", "tomato");
console.log(fruits);
// returns ["potato", "tomato", "orange", "watermelon", "apple", "orange", "grape", "apple"]
```

3. 没有map()的映射数组

也许每个人都知道数组的map()方法，但是有一个不同的解决方案，它可以用来获得类似的效果和非常干净的代码。我们可以使用.from()方法。

```

var friends = [
  { name: "John", age: 22 },
  { name: "Peter", age: 23 },
  { name: "Mark", age: 24 },
  { name: "Maria", age: 22 },
  { name: "Monica", age: 21 },
  { name: "Martha", age: 19 },
]

var friendsNames = Array.from(friends, ({name}) => name);
console.log(friendsNames);
// returns ["John", "Peter", "Mark", "Maria", "Monica", "Martha"]

```

4. 空数组

您是否有一个满是元素的数组，但是您需要出于任何目的对其进行清理，并且您不想逐个删除项？很容易就可以在一行代码中完成。要清空一个数组，您需要将数组的长度设置为0，就是这样！

```

var fruits = ["banana", "apple", "orange", "watermelon", "apple", "orange", "grape",
"apple"];

fruits.length = 0;
console.log(fruits);
// returns []

```

5. 将数组转换为对象

我们有一个数组，但出于某种目的，我们需要一个对象来处理这些数据，而将数组转换为对象的最快方法是使用众所周知的spread运算符(...)。

```

var fruits = ["banana", "apple", "orange", "watermelon"];
var fruitsObj = { ...fruits };
console.log(fruitsObj);
// returns {0: "banana", 1: "apple", 2: "orange", 3: "watermelon", 4: "apple", 5:
"orange", 6: "grape", 7: "apple"}

```

6. 用数据填充数组

在某些情况下，当我们创建一个数组时，我们希望用一些数据来填充它，或者我们需要一个具有相同值的数组，在这种情况下，.fill()方法提供了一个简单明了的解决方案。

```

var newArray = new Array(10).fill("1");
console.log(newArray);
// returns ["1", "1", "1", "1", "1", "1", "1", "1", "1", "1"]

```

7. 合并数组

您知道如何不使用.concat()方法将数组合并到一个数组中吗?有一种简单的方法可以用一行代码将任意数量的数组合并。正如您可能已经意识到的，spread操作符(...)在处理数组时非常有用，在本例中也是如此。

```
var fruits = ["apple", "banana", "orange"];
var meat = ["poultry", "beef", "fish"];
var vegetables = ["potato", "tomato", "cucumber"];
var food = [...fruits, ...meat, ...vegetables];
console.log(food);
// ["apple", "banana", "orange", "poultry", "beef", "fish", "potato", "tomato", "cucumber"]
```

8. 求两个数组的交集

这也是Javascript面试中最受欢迎的题目之一，因为它考察了你是否可以使用数组方法以及你的逻辑是什么。为了找到两个数组的交集，我们将使用本文前面展示的方法之一，以确保数组中的值不重复，并使用.filter方法和.include方法。最后，将得到两个数组的交集。例：

```
var numOne = [0, 2, 4, 6, 8, 8];
var numTwo = [1, 2, 3, 4, 5, 6];
var duplicatedValues = [...new Set(numOne)].filter(item => numTwo.includes(item));
console.log(duplicatedValues);
// returns [2, 4, 6]
```

9. 从数组中删除假值

首先，让我们定义假值。在Javascript中，假值是false, 0, "", null, NaN, undefined。现在我们可以来看看如何从数组中删除这类值。为此，我们将使用.filter()方法。

```
var mixedArr = [0, "blue", "", NaN, 9, true, undefined, "white", false];
var trueArr = mixedArr.filter(Boolean);
console.log(trueArr);
// returns ["blue", 9, true, "white"]
```

10. 从数组中获取随机值

有时我们需要从数组中随机选择一个值。要以一种简单、快速、简短的方式创建它，并保持代码整洁，我们可以根据数组长度获得一个随机索引号。让我们看看代码：

```
var colors = ["blue", "white", "green", "navy", "pink", "purple", "orange", "yellow", "black", "brown"];
var randomColor = colors[Math.floor(Math.random() * (colors.length))];
```

11. 数组反转

当我们需要反转我们的数组时，没有必要通过复杂的循环和函数来创建它，有一个简单的数组方法可以为我们做所有的事情，只需一行代码，我们就可以使我们的数组反转。让我们检查一下：

```
var colors = ["blue", "white", "green", "navy", "pink", "purple", "orange", "yellow",  
"black", "brown"];  
var reversedColors = colors.reverse();  
console.log(reversedColors);  
// returns ["brown", "black", "yellow", "orange", "purple", "pink", "navy", "green",  
"white", "blue"]
```

12. .lastIndexOf()方法

在JavaScript中，有一个有趣的方法，它允许查找给定元素的最后一次出现的索引。例如，如果我们的数组有重复的值，我们可以找到它最后一次出现的位置。让我们看看代码示例：

```
var nums = [1, 5, 2, 6, 3, 5, 2, 3, 6, 5, 2, 7];  
var lastIndex = nums.lastIndexOf(5);  
console.log(lastIndex);  
// returns 9
```

13. 将数组中的所有值相加

这个也是面试中经常被问到的问题，将数组中的所有值相加；它可以在一行代码中使用.reduce方法来解决。让我们看看代码：

```
var nums = [1, 5, 2, 6];  
var sum = nums.reduce((x, y) => x + y);  
console.log(sum);  
// returns 14
```