



链滴

# reduce 函数的用法

作者: [zhujie](#)

原文链接: <https://ld246.com/article/1589419582637>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

这是一个考察面试者对reduce函数用途的js面试题。下面我们看一下reduce函数的函数介绍：

reduce() 方法接收一个函数作为累加器，数组中的每个值（从左到右）开始缩减，最终计算为一个值。reduce() 对于空数组是不会执行回调函数的。参数如下：

参数	描述										
<code>function(total, currentValue, index, arr)</code>	必需。用于执行每个数组元素的函数。 函数参数: <table border="1"><thead><tr><th>参数</th><th>描述</th></tr></thead><tbody><tr><td><code>total</code></td><td>必需。初始值, 或者计算结束后的返回值。</td></tr><tr><td><code>currentValue</code></td><td>必需。当前元素</td></tr><tr><td><code>currentIndex</code></td><td>可选。当前元素的索引</td></tr><tr><td><code>arr</code></td><td>可选。当前元素所属的数组对象。</td></tr></tbody></table>	参数	描述	<code>total</code>	必需。初始值, 或者计算结束后的返回值。	<code>currentValue</code>	必需。当前元素	<code>currentIndex</code>	可选。当前元素的索引	<code>arr</code>	可选。当前元素所属的数组对象。
参数	描述										
<code>total</code>	必需。初始值, 或者计算结束后的返回值。										
<code>currentValue</code>	必需。当前元素										
<code>currentIndex</code>	可选。当前元素的索引										
<code>arr</code>	可选。当前元素所属的数组对象。										
<code>initialValue</code>	可选。传递给函数的初始值										

可以看到这是一个求和，或者统计有关的函数，那么可以得出以下用途。

基础数据：

```
const baseData = [
  {age: 10, name: '小明'},
  {age: 15, name: '小黄'},
  {age: 18, name: '小王'},
  {age: 18, name: '小白'},
  {age: 21, name: '小陈'}
]
```

## 1. 数组求和

求出数组中所有年龄的和：

```
// 方法一
baseData.reduce((total, currentValue) => total + currentValue.age, 0);
// 方法二
baseData.reduce((previousValue, currentValue, index) => {
  console.log(index)
  if(typeof previousValue === 'number') {
    return previousValue + currentValue.age
  } else {
    return previousValue.age + currentValue.age
  }
})
```

运行截图：



上面求和有2种写法，我们重点看一下第二种写法，可以看到第二种方式没有传第二个参数，这就导致previousValue变量可能是返回结果的和，也可能是函数的第一项，所以需要特殊判断一下。而且遍历开始的index也是从1开始的。2种方式来看，第一种更简便，第二种会少一次循环。

## 2. 数组去重

去掉数组中年龄重复的数据

```
let resultArr = baseData.reduce((result, currentValue, index) => {
  if (!result.find(item => item.age === currentValue.age)) {
    result.push(currentValue)
  }
  return result
}, [])
console.log(resultArr, 'result')
```

## 3. 统计个数

找出年龄小于18岁的数据

```
baseData.reduce((result, currentValue) => currentValue.age < 18 ? ++result : result, 0); // 2
```

## 4. 多维数组转换为一维数组

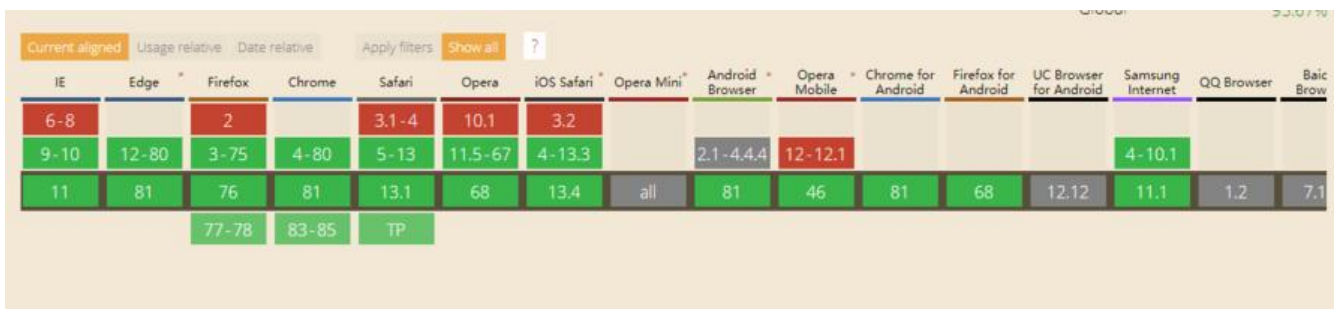
```
let arr = [[0, 1], [2, 3], [4,[5,6,7]]]
const newArr = function(arr){
  return arr.reduce((pre,cur)=>pre.concat(Array.isArray(cur)?newArr(cur):cur),[])
}
console.log(newArr(arr)); //[0, 1, 2, 3, 4, 5, 6, 7]
```

## 5. 统计一个数组中每个元素出现的次数

```
const arraySum = (arr, val) => arr.reduce((acc, cur) => {
  return cur == val ? acc + 1 : acc + 0
}, 0);
```

```
let arr = [ 0, 1, 3, 0, 2, 0, 2, 3 ]
console.log(arraySum(arr, 0)) // 数组arr中 0 元素出现的次数为3
```

## 兼容性



IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baic Brow
6-8		2		3.1-4	10.1	3.2									
9-10	12-80	3-75	4-80	5-13	11.5-67	4-13.3		2.1-4.4.4	12-12.1				4-10.1		
11	81	76	81	13.1	68	13.4	all	81	46	81	68	12.12	11.1	1.2	7.1
		77-78	83-85	TP											

可以看出除了特别老的浏览器，其他都支持了，放心使用。

## 总结

可以看到，`reduce`函数应用在数组求和，数组去重，统计，数组转换等方面。熟练掌握能减少你的代

量。使代码变更更简洁，提高开发效率。

---

线上体验地址: [地址](#)

所有的源码都可以在我的仓库地址: [地址](#)

学习如逆水行舟，不进则退，前端技术飞速发展，如果每天不坚持学习，就会跟不上，我会陪着大家每天坚持推送博文，跟大家一同进步，希望大家能关注我，第一时间收到最新文章。