



链滴

Array - permutations (advanced)

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1558663216200>

来源网站: [链滴](#)

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

描述

⚠️warning 警告：该方法的执行时间和数组中的每一个元素呈指数增长。任何超过 8 到 10 个元素的数组将会导致你的浏览器由于尝试解决所有不同的组合而被挂起。

为数组中的元素生成所有的排列组合。

提示

- 使用递归
- 依次指定数组中的每一个元素，然后使用剩下的元素来创建对应部分的所有排列
- 使用 `Array.prototype.map()` 来连接指定元素和上一步骤中的排列组合结果
- 使用 `Array.prototype.reduce()` 来组合每一次递归的结果
- 基本的退出条件为数组的长度等于 `2` 或者 `1`

代码

```
const permutations = arr => {  
  if (arr.length <= 2) return arr.length === 2 ? [arr, [arr[1], arr[0]]] : arr;  
  return arr.reduce(  
    (acc, item, i) =>  
      acc.concat(  
        permutations([...arr.slice(0, i), ...arr.slice(i + 1)]).map(val => [item, ...val])  
      ),  
    []  
  );  
};
```

示例

返回所有排列组合：

```
permutations([1, 33, 5]); // [[ 1, 33, 5 ], [ 1, 5, 33 ], [ 33, 1, 5 ], [ 33, 5, 1 ], [ 5, 1, 33 ], [ 5, 33, 1 ]]
```

返回总目录

[每天 30 秒系列之 JavaScript 代码](#)