



链滴

Array - pullBy (advanced)

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1559038031885>

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

基于给定的迭代函数和对象获取出需要过滤值，最终根据需要过滤出的值来修改原始数组。

提示

- 检查提供给函数中的最后一个参数是否为函数，如果不是函数的话返回 `undefined`
- 使用 `Array.prototype.map()` 让指定的所有数组中的对象依次传入迭代函数 `fn` 中，从而获取需过滤出的值
- 使用 `Array.prototype.filter()` 和 `Array.prototype.includes()` 过滤出不需要的值
- 通过使用 `Array.prototype.length = 0` 将原始数组的长度重置为 0，从而改变其传入的数组
- 使用 `Array.prototype.push()` 为原始数组重新填入过滤后的值

代码

```
const pullBy = (arr, ...args) => {
  const length = args.length;
  let fn = length > 1 ? args[length - 1] : undefined;
  fn = typeof fn === 'function' ? (args.pop(), fn) : undefined;
  let argState = (Array.isArray(args[0]) ? args[0] : args).map(val => fn(val));
  let pulled = arr.filter((v, i) => !argState.includes(fn(v)));
  arr.length = 0;
  pulled.forEach(v => arr.push(v));
};
```

示例

移除原始数组中 `x` 不等于指定数组中的 `x` 的元素：

```
var myArray = [{ x: 1 }, { x: 2 }, { x: 3 }, { x: 1 }];
pullBy(myArray, [{ x: 1 }, { x: 3 }], o => o.x); // myArray = [{ x: 2 }]
```

返回总目录

[每天 30 秒系列之 JavaScript 代码](#)