JS数组专题1️⃣ — 数组扁平化

三巫 Lv2
2018年08月06日 01:36 · 阅读 7361

一、什么是数组扁平化

1. 扁平化，顾名思义就是减少复杂性装饰，使其事物本身更简洁、简单，突出主题。
2. 数组扁平化，对着上面意思套也知道了，就是将一个复杂的嵌套多层的数组，一层一层的转化为层级较少或者只有一层的数组。

Ps: `flatten` 可以使数组扁平化，效果就会如下：

```
const arr = [1, [2, [3, 4]]];
console.log(flatten(arr)); // [1, 2, 3, 4]
```


> 从中可以看出，使用 `flatten` 处理后的数组只有一层，下面我们来试着实现一下。

2.1 普通递归

- 这是最容易想到的方法，简单，清晰！

复制代码

```
/* ES6 */
const flatten = (arr) => {
  let result = [];
  arr.forEach((item, i, arr) => {
    if (Array.isArray(item)) {
      result = result.concat(flatten(item));
    } else {
      result.push(arr[i])
    }
  })
  return result;
};

const arr = [1, [2, [3, 4]]];
console.log(flatten(arr));
```

复制代码

```
/* ES5 */
function flatten(arr) {
  var result = [];
  for (var i = 0, len = arr.length; i < len; i++) {
    if (Array.isArray(arr[i])) {
      result = result.concat(flatten(arr[i]))
```

```
      result.push(arr[i])
    }
  }
  return result;
}

const arr = [1, [2, [3, 4]]];
console.log(flatten(arr));
```

2.2 `toString()`

- 该方法是利用 `toString` 把数组变成以逗号分隔的字符串，然后遍历数组把每一项再变回原来的类型。

先来看下 `toString` 是怎么把数组变成字符串的

复制代码

```
[1, [2, 3, [4]]].toString()
// "1,2,3,4"
```

完整的展示

复制代码

```
/* ES6 */
const flatten = (arr) => arr.toString().split(',').map((item) => +item);

const arr = [1, [2, [3, 4]]];
console.log(flatten(arr));
```

复制代码

```
/* ES5 */
function flatten(arr) {
  return arr.toString().split(',').map(function(item){
    return +item;
  });
}

const arr = [1, [2, [3, 4]]];
console.log(flatten(arr));
```

> 这种方法使用的场景却非常有限，必须数组中元素全部都是 `Number`。 也可以全部都为 `String`，具体实现大家自己体会。

2.3 `[].concat.apply` + `some`

- 利用 `arr.some` 判断当数组中还有数组的话，循环调用 `flatten` 扁平函数(利用 `[].concat.apply` 扁平), 用 `concat` 连接，最终返回 `arr`;

复制代码

```
/* ES6 */
const flatten = (arr) => {
  while (arr.some(item => Array.isArray(item))){
    arr = [].concat.apply([], arr);
  }
  return arr;
}
```

复制代码

```
/* ES5 */
/**
* 封装Array.some
* @param {function} callback    - 回调函数
* @param {any}      currentThis - 回调函数中this指向
*/
Array.prototype.some = function (callback, currentThis){
  let context = this;
  let flag = false;
  currentThis = currentThis || this;
  for (var i = 0, len = context.length; i < len; i++) {
    const res = callback.call(currentThis, context[i], i, context);
    if (res) {
      flag = true;
    } else if (!flag) {
      flag = false;
    }
  }
  return flag;
}

function flatten(arr){
  while(arr.some(item => Array.isArray(item))){
    arr = [].concat.apply([], arr);
  }
  return arr;
}
```

2.4 `reduce`

- `reduce` 本身就是一个迭代循环器，通常用于累加，所以根据这一特点有以下：

复制代码

```
function flatten(arr){
  return arr.reduce(function(prev, cur){
    return prev.concat(Array.isArray(cur) ? flatten(cur) : cur)
  }, [])
}

const arr = [1, [2, [3, 4]]];
console.log(flatten(arr));
```

2.5 ES6 中的 解构运算符 `...`

- `...` 每次只能展开最外层的数组，被 `[].concat` 后， `arr` 就扁平化一次。

复制代码

```
function flatten(arr){
  while(arr.some(item => Array.isArray(item))){
    arr = [].concat(...arr);
  }
  return arr;
}
```

番外篇将给大家讲解 `lodash` 中 `flatten` 的实现源码，感谢大家阅读！

分类： 前端　　标签： JavaScript 前端

评论

热门评论

不是发哥 Lv1 前端 @ 搞倒闭好多家了 3年前

为什么第一种flatten属于es6的做法，const、let换成var不行么?

1 4

三巫 Lv2 （作者） 3年前

第一种的ES6做法中使用了箭头函数、forEach等ES6语法，而对于const、let而言，如你所说，在这里没有特殊意义，只是针对ES6中做了相应的变化而已，所以可以换成var。

点赞 回复

Lpeanut 3年前

forEach是es5语法

点赞 回复

查看更多回复

全部评论 13

最新 最热

117 13 收藏

function formatArr(p, r = []) {
let res = r;
for (let i = 0; i < p.length; i++) {
if (Array.isArray(p[i])) {
formatArr([...p[i]], res);
} else {…

展开

点赞 回复

ttttttttt君 1年前

可以用generator
function * flat (tree) {
for (const iterator of tree) {
if (Array.isArray(iterator)) {
yield * flat(iterator) // 这里也是一个被代理的generator函数
} else {…

展开

点赞 回复

松鼠桂鱼 Lv2 undefined @ undefined 3年前

Array.prototype.flat() 可以介绍一下

1 1

117 13 收藏

看了mdn这个方法还没放出来，不过可以试着封装下

点赞 回复

doudou0818 3年前

2.1 直接用...就可以吧console.log([1, ...[2, ...[3, 4]]])

1 1

Lpeanut 前端工程师 @ 专业七星... 3年前

2.3的some方法es5就有，不用重新定义吧。

1 1

不是发哥 Lv1 前端 @ 搞倒闭好多家了 3年前

为什么第一种flatten属于es6的做法，const、let换成var不行么？

1 4

三巫 Lv2 （作者） 3年前

第一种的ES6做法中使用了箭头函数、forEach等ES6语法，而对于const、let而言，如你所说，在这里没有特殊意义，只是针对ES6中做了相应的变化而已，所以可以换成var。

点赞 回复

Lpeanut 3年前

117 13 收藏

点赞 回复

查看更多回复

相关推荐

张吉成 | 4月前 | JavaScript

树结构、扁平化数组相互转换

1922 42 3

风雨踏梦行 | 8月前 | JavaScript · 前端

Js实现扁平化数据结构和tree转换 --每天进步一点点

5012 55 4

HZ_YZ | 4月前 | JavaScript · 前端

js使用reduce实现扁平化数组转换为树形数据

403 2 1

dreamer_sen | 9月前 | 面试

【JavaScript】数组扁平化的六种方式

2291 32 评论

117 13 收藏

摸鱼的春哥 | 2月前 | 前端 · JavaScript

2022，前端的天🌧要怎么变？

6.8w 659 250

樊衍 | 6月前 | 面试

前端面试手撕 对象的扁平化与反扁平化

817 8 评论

HighClassLickDog | 4月前 | 前端 · JavaScript

因为懒，我把公司的后管定制成了低代码中台

6.4w 459 171

小只前端攻城狮 | 9月前 | 面试 · 前端

🔥🔥JS如何实现数组扁平化？不同的方法有什么区别？

1070 40 10

Ned | 5月前 | JavaScript · 前端

JavaScript | 让数组扁平化的三个方法！

1469 37 3

弹铁蛋同学 | 2年前 | 前端

面试官连环追问：数组拍平（扁平化） flat 方法实现

想染上学习瘾 | 4天前 | 前端

js之对象扁平化和数组扁平化

164 6 评论

粥里有勺糖 | 10月前 | 面试 · 前端

经典面试题-数组扁平化&展开多级数组

1019 14 评论

前端阿飞 | 4月前 | 前端 · JavaScript

10个常见的前端手写功能，你全都会吗?

8.8w 2220 178

大帅老猿 | 10月前 | 前端 · JavaScript

产品经理：你能不能用div给我画条龙?

10.8w 2906 576

大帅老猿 | 9月前 | 前端 · JavaScript · HTML

2天赚了4个W，手把手教你用Threejs搭建一个Web3D汽车展厅！

5.9w 1420 240

前端小智 | 2年前 | JavaScript · ECMAScript 6

13 个 JS 数组精简技巧，一起来看看

CUGGZ | 5月前 | 前端 · JavaScript · 程序员

33个非常实用的JavaScript一行代码，建议收藏！

7.3w 1729 59

前端胖头鱼 | 5月前 | JavaScript · Vue.js · 前端

就因为JSON.stringify，我的年终奖差点打水漂了

7.6w 1064 233

---

nanfeiyan | 2年前 | JavaScript

reduce实现filter,map 数组扁平化等

3923 52 7

EarlEcho | 1年前 | 算法

【算法】JS 实现对象的扁平化

5562 22 3

117 13 收藏