



链滴

解释递归及其使用场景

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1550632462419>

来源网站: [链滴](#)

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

回答

递归描述了用自身相似的方法去重复事物的过程。在 JavaScript 中，递归可以理解为函数重复调用自直到满足某个基本条件时。递归需要基本条件来打断他的循环，否则函数就会不停的调用自身。在处包含未知嵌套深度的数据结构时，递归非常有用。

例如，数据库返回一些平面数组结构的评论给你，他们在数据库中是通过父子 id 来进行关联的，但要你在界面上为其展示为嵌套的树形结构。每条评论要么是顶层的评论（没有父节点）要么是回复父点的评论。评论可以是回复的回复的回复。。。我们在此之前不知道评论到底存在几级评论。这时候需要递归的帮助。

```
const nest = (items, id = null, link = "parent_id") =>
  items
  .filter(item => item[link] === id)
  .map(item => ({ ...item, children: nest(items, item.id) })))
```

```
const comments = [
  { id: 1, parent_id: null, text: "First reply to post." },
  { id: 2, parent_id: 1, text: "First reply to comment #1." },
  { id: 3, parent_id: 1, text: "Second reply to comment #1." },
  { id: 4, parent_id: 3, text: "First reply to comment #3." },
  { id: 5, parent_id: 4, text: "First reply to comment #4." },
  { id: 6, parent_id: null, text: "Second reply to post." }
]
```

```
nest(comments) // [{id: 1..., childred: [...]}, ...]
```

在上面的示例中，打断自身的基本条件为 `filter()` 返回一个空数组。链上 `map()` 并不会调用包含递归用的回调函数，因此可以打破循环。

加分回答

- 在处理包含未知嵌套深度的数据结构时，递归非常有用。
- 递归必须要有一个基本条件来打断他的循环，否则他就会不停的调用自身。
- 使用递归函数的优点是逻辑简单清晰，缺点是过深的调用会导致栈溢出。

返回总目录

每天 30 秒