链滴

对比 Mutable 和 Immutable 及 Mutating 和 Non-Mutating

作者：Vanessa

原文链接：https://ld246.com/article/1550210424716

来源网站：链滴

2019-02-15

回答

基本解释如下：

- Mutable：对象主题可以被修改
- Immutable：对象一旦被创建就不能被修改
- Mutating：修改对象主题的方法
- Non-Mutating：不会使原始对象发生变化的修改方法

在 JavaScript 中，对象是可被修改的，但原始值却是不能被修改的。这意味着可以通过执行某些操作修改对象的原始引用，但对原始值执行任何操作都不能修改他的初始值。

所有的 String.prototype 方法都不能对初始值产生任何影响，他们都只会返回一个新的字符串。相而言，Array.prototype 的某些方法也不会修改初始数组的引用，只会产生一个新的数组，但是某些法却能产生变化。

```
const myString = "hello!"
// 返回一个新数组，并不会修改初始值
myString.replace("!", "") // "hello"

const originalArray = [1, 2, 3]
// 原始数组被修改
originalArray.push(4) // [1, 2, 3, 4]
// 返回一个新数组，不会修改原始数组
originalArray.concat(4) // [1, 2, 3, 4, 4]
```

加分回答

- 数组中 Mutating 的方法如：copyWithin、fill、pop、push、reverse、shift、sort、splice、unshift
- 数组中 Non-Mutating 的方法如：map、slice、concat、filter
- Immutable.js

返回总目录

每天 30 秒