链滴

JavaScript - 数组 for 循环，reduce 及方法链剖析

作者：Vanessa

原文链接：https://ld246.com/article/1594599696957

来源网站：链滴

2020-07-13

描述

对数组的 for 循环，reduce 和方法链进行不同纬度的比对。

循环

说明

- for 循环中 for...in、for...of 和 forEach 的不同点
- 由于函数式编程的流行，目前使用率有所下降
- 可以对迭代进行控制，如跳过元素或提前 return
- 返回的结果数组需要在循环外预先声明
- 使用 Array.prototype.push() 或扩展运算符 ... 来进行元素的添加
- O(N) 复杂度，每一个元素只能迭代一次

代码

```
const files = [ 'foo.txt ', '.bar', '    ', 'baz.foo' ];
let filePaths = [];

for (let file of files) {
  const fileName = file.trim();
  if(fileName) {
    const filePath = `~/cool_app/${fileName}`;
    filePaths.push(filePath);
  }
}

console.log(filePaths); // ["~/cool_app/foo.txt", "~/cool_app/.bar", "~/cool_app/baz.foo"]
```

reduce

说明

- 在 Array.prototype.reduce() 中，将空数组做为初始值的
- 随着函数式编程的流行，如今也有更多的开发者开始使用
- 对迭代的控制较弱，不能跳过元素或提前 return
- 如果需要的话，可以和其他方法形成链
- 使用 Array.prototype.push() 或扩展运算符 ... 来进行元素的添加
- O(N) 复杂度，每一个元素只能迭代一次

代码

```
const files = [ 'foo.txt ', '.bar', '    ', 'baz.foo' ];
const filePaths = files.reduce((acc, file) => {
  const fileName = file.trim();
  if(fileName) {
    const filePath = `~/cool_app/${fileName}`;
    acc.push(filePath);
  }
  return acc;
}, []);

console.log(filePaths); // ["~/cool_app/foo.txt", "~/cool_app/.bar", "~/cool_app/baz.foo"]
```

方法链

说明

- 使用 Array.prototype.map() 和 Array.prototype.filter()
- 随着函数式编程的流行，如今也有更多的开发者开始使用
- 对迭代的控制较弱，不能跳过元素或提前 return
- 声明式，更加容易阅读和重构，链可以根据需要不断添加
- 不需要使用 Array.prototype.push() 或扩展运算符 ...
- O(cN) 复杂度, c 表示每个元素迭代的次数，也就是链的长度

代码

```
const files = [ 'foo.txt ', '.bar', '    ', 'baz.foo' ];
const filePaths = files
  .map(file => file.trim())
  .filter(Boolean)
  .map(fileName => `~/cool_app/${fileName}`);

console.log(filePaths); // ["~/cool_app/foo.txt", "~/cool_app/.bar", "~/cool_app/baz.foo"]
```

返回总目录

每天 30 秒系列之 JavaScript 代码