链滴

Array - zipWith (advanced)

作者：Vanessa
原文链接：https://ld246.com/article/1562298684138
来源网站：链滴

2019-07-05

描述

创建一个数组，其元素应按照原始数组中的位置进行分组。具体如何分组、是否应该被合并应以提供函数的返回值作为最终值。

提示

- 检测提供的最后一个参数是否为函数
- 使用 `Math.max()` 获取参数中最长数组的长度
- 创建一个长度为第二步结果的数组作为返回值
- 使用带有迭代功能的 `Array.from()` 对被分组的元素创建一个数组
- 如果参数中的数组长度不一样时，请使用 `undefined` 替代找不到的值
- 对元素进行每一次分组时调用提供的函数进行分组 `(...group)`

代码

```
const zipWith = (...array) => {
  const fn = typeof array[array.length - 1] === 'function' ? array.pop() : undefined;
  return Array.from(
    { length: Math.max(...array.map(a => a.length)) },
    (_, i) => (fn ? fn(...array.map(a => a[i])) : array.map(a => a[i]))
  );
};
```

示例

对传入的数组进行纵向相加：

```
zipWith([1, 2], [10, 20], [100, 200], (a, b, c) => a + b + c); // [111,222]
```

对传入的数组进行纵向相加，如纵向位为空则使用指定字符进行替换：

```
zipWith(
  [1, 2, 3],
  [10, 20],
  [100, 200],
  (a, b, c) => (a != null ? a : 'a') + (b != null ? b : 'b') + (c != null ? c : 'c')
); // [111, 222, '3bc']
```

返回总目录

每天 30 秒系列之 JavaScript 代码