



链滴

根据食谱和已有食材计算可烹饪的最大次数

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1546833887125>

来源网站: [链滴](#)

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

2019-01-07

题目

```
/**
 请按要求实现 `batches(recipe, available)` 函数：
 1. 含有 2 个参数，第一个为食谱对象，第二个为已有食材对象
 2. 返回可烹饪的最大次数
 3. 以下代码执行时，需返回正确结果且运行过程中无异常
*/

// return 0
batches(
  { milk: 100, butter: 50, flour: 5 },
  { milk: 132, butter: 48, flour: 51 }
)

batches(
  { milk: 100, flour: 4, sugar: 10, butter: 5 },
  { milk: 1288, flour: 9, sugar: 95 }
)

// return 1
batches(
  { milk: 100, butter: 50, cheese: 10 },
  { milk: 198, butter: 52, cheese: 10 }
)

// return 2
batches(
  { milk: 2, sugar: 40, butter: 20 },
  { milk: 5, sugar: 120, butter: 500 }
)
```

回答

```
const batches = (recipe, available) =>
  Math.floor(
    Math.min(...Object.keys(recipe).map(k => available[k] / recipe[k] || 0))
  )
```

加分回答

• 扩展运算符：在函数调用 / 数组构造时，将数组表达式或者 string 在语法层面展开；在构造简单对象时，将对象表达式按 key-value 的方式展开。如：

```
const spread = (x, y, z) => {
  console.log(x, y, z) // 1, 2, 'h'
}
const spread2 = (...arg) => {
  console.log(arg) // [1, 2, "h", "e", "l", "l", "o", "w"]
}
spread(...[1, 2, ...'hellow']);
```

```
spread2([1, 2, ...'hellow']);  
console.log({ ...{ foo: 'bar', x: 42 }, ...{ foo: 'baz', y: 13 } }); // { foo: "baz", x: 42, y: 13 }
```

- `Object.keys()` 会返回对象自身可枚举属性值的数组。如：

```
console.log(Object.keys( ['a', 'b', 'c'])); // ['0', '1', '2']  
console.log(Object.keys({ 0: 'a', 1: 'b', 2: 'c' })); // ['0', '1', '2']
```

- `Array.prototype.map()` 创建一个新数组，返回回调函数对原有数据的处理值。如：

```
console.log([1, 4, 9, 16].map(x => x * 2)); // [2, 8, 18, 32]
```

- 用 `||` 或 `&&` 可替代简单的 `if ... else ...`
- `Math.floor()`：返回小于或等于一个给定数字的最大整数，即下取证
- `Math.min()`：返回零个或多个数值的最小值

返回总目录

每天 30 秒