



链滴

Memoization 是什么?

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1550152136095>

来源网站: [链滴](#)

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

2019-02-14

回答

Memoization 是用来缓存函数调用的输出结果，以便减少后续再次调用时的运算，进而加快运算速的一种优化技术。Memoization 在再次调用有相同输入的另一函数时将直接返回缓存的该函数的输出结果，但第一次的计算当然是必不可少的。

JavaScript 对此的一个基本实现如下：

```
const memoize = fn => {
  const cache = new Map()
  return value => {
    const cachedResult = cache.get(value)
    if (cachedResult !== undefined) return cachedResult
    const result = fn(value)
    cache.set(value, result)
    return result
  }
}
```

加分回答

• 上述实现中，即使函数有多个参数也只会返回一个一元函数。以下是 underscore 的实现：

```
_.memoize = function(func, hasher) {
  var memoize = function(key) {
    var cache = memoize.cache;
    var address = '' + (hasher ? hasher.apply(this, arguments) : key);
    if (!_.has(cache, address)) cache[address] = func.apply(this, arguments);
    return cache[address];
  };
  memoize.cache = {};
  return memoize;
};
```

- Memoization 的第一个函数调用通常会比平常慢，因为他含有检查缓存结果是否存在及在结果返回之前设置为缓存的运算。
- Memoization 可以提高后续函数再次调用的性能，但这仍需要该函数进行过一次调用后才起作用。
- 函数为纯函数时，方才可以使用 Memoization。

返回总目录

每天 30 秒