

柯里化：

笔记本： 1.有关印象笔记

创建时间： 2020/6/21 16:16

更新时间： 2020/7/19 16:08

作者： 1639079350@qq.com

标签： 4.柯里化

柯里化：

- 当函数有多个参数时，先传递一部分参数（该部分参数后面不变），然后返回一个新的函数接收剩余的参数，返回结果。
- 柯里化可以给一个函数传递较少的参数，得到一个已经记住了某些固定参数的新函数；
- 这是一种对函数参数的缓存；让函数变得更灵活，让函数的粒度更小；
- 可以把多元函数转变成一元函数，可以组合使用函数产生强大的功能；使用它的目的)

```
const { log } = console;
// 普通函数
function checkAge( min,age ) {
    return age >= min;
}
log(checkAge(18,20)); // true

// 柯里化函数 == 闭包
function justifyAge( min ) {
    return function (age) {
        return age >= min;
    }
}

// ES6柯里化函数
let justifyAge = min => ( age => age >= min );

let justifyAge18 = justifyAge(18);
let justifyAge20 = justifyAge(20);

log(justifyAge18(20)); // true
log(justifyAge18(24)); // true
```

lodash中通用的柯里化方法： **【_.curry(func)】**

1.功能：创建一个函数，该函数接收一个或者多个func参数，如果func所需要的参数都被提供，则执行func并返回执行结果，否则继续返回该函数并等待接收剩余的参数。

2.参数：需要柯里化的函数；【但是柯里化以后的传参是func的参数】

3.返回值：柯里化的函数；

```
const { log } = console;

function getSum(a,b,c) {
    return a + b + c;
}

const curried = _.curry(getSum);
log(curried(1,2,3)); // 6
log(curried(1,2)(3)); // 6
log(curried(1)(2,3)); // 6

// 函数参数只提供一部分
const match = _.curry(function(reg,str) {
    return str.match(reg);
});
const haveSpace = match(/\s+/g); // haveSpace缓存了reg，需要再传入参数str
log(haveSpace('hello world!')); // [" "]
log(haveSpace('helloworld!')); // null

// 函数参数只提供一部分
const filter = _.curry(function(func,array) {
    return array.filter(func);
}) // 传参func/arr
const findSpace = filter(haveSpace); // 缓存了func,传arr

log(filter(haveSpace,['hello world', 'john_Donne'])); // ["hello world"]
log((findSpace(['hello world', 'john_Donne']))); // ["hello world"]
```

模拟_.curry()方法原理：

- 1.根据_.curry的实现原理写自定义函数的实现形式；
- 2.根据实参和形参的个数分情况讨论实现形式；
- 2.形参的形式：伪数组转为数组：Array.from；

```
// 模拟实现 lodash 中的 curry 方法

function getSum (a, b, c) {
  return a + b + c
}

const curried = curry(getSum)

console.log(curried(1, 2, 3))
console.log(curried(1)(2, 3))
console.log(curried(1, 2)(3))

function curry (func) {
  return function curriedFn(...args) {
    // 判断实参和形参的个数
    if (args.length < func.length) {
      return function () {
        return curriedFn(...args.concat(Array.from(arguments)))
      }
    }
    return func(...args)
  }
}
```