



flitenwall

专家

码龄6年

暂无认证

145

原创

2万+

周排名

1万+

总排名

56万+

访问



等级

4151

积分

1236

粉丝

858

获赞

233

评论

3459

收藏



私信

关注

搜博主文章



热门文章

超详细SPSS主成分分析计算指标权重

(一) 165075

超详细SPSS主成分分析计算指标权重

(二: 权重计算及极差法标准化) 146262

利用Excel计算耦合度及耦合协调度 67635

DIV+CSS网页布局实例 20742

JavaScript实现冒泡排序算法 17882

分类专栏



2022轩灵大前端知识...

1篇

(翻译) Weakmap详解

翻译

flitenwall



于 2020-12-11 16:58:31 发布



2187



收藏 7

版权

分类专栏:

技术翻译

JavaScript

文章标签:

Weakmap

弱引用

javascript

es6



技术翻译

同时被 2 个专栏收录

2 订阅

4 篇文章

订阅专栏

先看一个例子






```
1 let obj = { name: 'toto' }
2
3 // { name: 'toto' } 这个对象能够被读取到, 因为obj这个变量名有对它的引用
4
5 // 将引用覆盖掉
6 obj = null
7
8 // 这个对象将会被从内存中移除, 因为我们已经失去了对它所有的引用
```

再来看另外一个例子

```
1 let obj = { name: 'toto' }
2 let arr = [ obj ]
3
4 obj = null
```

在这个例子中, 对象 `{name: 'toto'}` 不会被从内存中移除, 因为数组arr保存了对它的引用

强引用和弱引用之间有什么区别呢?

	一些分享	2篇
	python3开发博客	5篇
	node	2篇
	python	49篇
	面试	1篇



最新评论

超详细SPSS主成分分析计算指标权重 (...
bonycow: 同问。看了其他老师的做法，应该就是主成分的初始特征值

利用Excel计算耦合度及耦合协调度
asdzxc_81: 就是算错了，呵呵

超详细SPSS主成分分析计算指标权重 (...
breakthrough:): 想问一下如果数据不是数字的话 该怎么做呢？

JavaScript实现冒泡排序算法
m0_61527887: 是七次排序吧

利用Excel计算耦合度及耦合协调度
number_f: g(y)是需要剩下的那个系统的数据，再进行计算权重然后计算的。博主只...

您愿意向朋友推荐“博客详情页”吗？



强烈不推荐



不推荐



一般般



推荐



强烈推荐

最新文章

事实上，javascript中的大多数变量都保存着对一个对象的强引用。比如上面这个数组保存着对对象 `{name:'toto'}` 的强引用

如果一个变量保存着对一个对象的强引用，那么这个对象将不会被垃圾回收，但是如果一个变量只保存着对这个对象的弱引用，那么这个对象将会被垃圾回收

一些变量类型在对象上有一个弱引用，这就是 `Weakmap` 的情况

Weakmap

`weakmap` 是一个额外的数据存储，它可以让我们从外部（第三方库）扩展或者封装一个对象，而不需要进行垃圾回收的推断，或者能够智能的创建一个缓存函数。

不用担心看不明白，在比较 `map` 和 `weakmap` 之前我将解释并展示它的含义。

Map和Weakmap的比较

使用map，对象会占用内存，可能不会被垃圾回收。Map对一个对象是强引用

```
1 let obj = { name: 'toto' }
2 let mapObj = new Map()
3 mapObj.set(obj, 'any value')
4
5 obj = null
6 mapObj.size() // 1
```

`Weakmap` 则是完全不同的，它不会阻止关键对象的垃圾回收

第一条规则，`weakmap` 只接受 `object` 作为 `key`，第二条规则是它只保存对对象的弱引用。

```
1 let obj = { name: 'toto' }
2 let weakmapObj = new WeakMap()
3 weakmapObj.set(obj, 'any value')
4
5 obj = null
6 weakmapObj.size() // 0
```

2021年的最后一天，我发布了2022年的第一个视频

2022年轩灵大前端知识体系王牌打造计划

《左耳听风-高效学习篇》阅读笔记

2021年 18篇 2020年 81篇
2019年 50篇

目录

强引用和弱引用之间有什么区别呢？

Weakmap

Map和Weakmap的比较

何时使用Weakmap？

原文链接

对象被垃圾回收器删除，因为weakmap在对象{ name: 'toto' }上只有弱引用，而这个对象已经没有强引用了。(只有变量obj有保持引用)

何时使用Weakmap？

正如你所看到的，Weakmap 可以用在任何地方

缓存器函数

```
1  const cache = new WeakMap()
2
3  const process = function (obj) {
4    // 如果输入的值不在缓存器中
5    if (!cache.has(obj)) {
6      // 想象一个函数需要很大的内存或者资源
7      // 当输入相同时，我们不想重复执行bigOperation函数
8      const result = bigOperation(obj)
9      // 所以此时执行一次函数并将它的结果存入缓存中
10     cache.set(obj, result)
11   }
12   return cache.get(obj)
13 }
14
15 let obj = { /* any object */ }
16 // 第一次我们没有这个输入作为缓存，所以在第二次的时候我们才不需要执行这个函数，
17 const firstResult = process(obj)
18 // 只需要从缓存中取出结果
19 const secondeResult = process(obj)
20 // 源对象将被从weakmap中移除
21 obj = null
```

使用 map ，这个缓存器函数应该将obj对象保存在内存中。

但这将导致内存泄漏！

当我们对一个不再使用的对象保持引用的时候将会造成内存泄漏，所以如果你不再使用对象，请删除它的任何变量引用。

使用 `weakmap` 时我们不应该使用 `.keys()` / `.values()` / `.entries()`，因为我们不知道何时垃圾回收器会移除这个对象。

最后一个例子

动态无泄漏内存的访问计数器

```
1 // 访问计数器
2 let visitsCountMap = new WeakMap()
3
4 // 增加访问计数
5 function countUser(user) {
6   const count = visitsCountMap.get(user) || 0
7   visitsCountMap.set(user, count + 1)
8 }
9
10 let toto = { name: "toto" }
11
12 countUser(toto) // 计算访问次数
13
14 // 将toto对象从内存中移除
15 toto = null
```

原文链接

[Weakmap in javascript](#)

你不知道的 `WeakMap`

高先生的猫 447

相信很多读者对 ES6 引入的 `Map` 已经不陌生了，其中的一部分读者可能也听说过 `WeakMap`。既生 `Map` 何生 `W...`

ES6: `Map()` 与 `WeakMap()`

weixin_43856797的博客 2531

`Map()`、`WeakMap()` 其实与 `Set()`、`WeakSet()` 的方法与特性大致相同，这里只是来说一下他们的不同 关于`Set(...`

评论 3 您还未登录，请先 [登录](#) 后发表或查看评论

你不知道的 WeakMap_高先生的猫的博客

3-23

let map = new WeakMap(); map.set(sem, "全栈修仙之路"); sem = null; // 覆盖引用 2.2 WeakMap 与垃圾回收 WeakM...

map与weakmap_ES6新增的Map和WeakMap 又是什么玩意?非...

3-30

WeakMap的基本用法 WeakMap结构的使用方式和Map结构一样: let wm = new WeakMap(); 两者都是使用new来...

WeakHashMap的神话

lijiecong的专栏 1447

http://www.iteye.com/topic/587995

在广大的Java界, 关于WeakHash...

ES6新增的Map和WeakMap 又是什么玩意? 非常详细的解释

qq_43623970的博客 247

上一篇文章讲了set和weakSet,这节咱就讲Map和weakMap是什么?这两篇文章并没有什么联系,主要知识用法类似...

Map与WeakMap的区别与使用_李小茶的博客_weakmap和map区别

3-23

var map2 = new Map([[nie, kth], [kth, nie]]) console.log(map2.get(nie)); // {name: 'Keith'} console.log(map2.get(kth)); /...

es6 (基础十四) Map与WeakMap

u011705725的博客 365

一、Map 1.map的作用: key可以是任意的内容 2.使用 长度:size 删除:delete clear 增:set 查:get 传统的对象...

Map与WeakMap的区别与使用

hh3167253066的博客 846

我们常用的对象Object, 是由key:value集合组成的, 但key只能是字符串, 有很大的使用限制。当我们需要其他类...

【ES6基础】Map与WeakMap

前端达人 9993

开篇ES6里除了增加了Set (集合) 类型外 (笔者在这篇文章《Set与WeakSet》有过介绍), 今天的这篇文章笔...

WeakMap详解

jiaoqi6132的博客 282

带键的集合 一个Map对象就是一个简单的键值对映射集合, 可以按照数据插入时的顺序遍历所有的元素。 for...of...

ES6-WeakMap与Map的区别, Set与WeakSet的区别

知其白, 守其黑; 和其光, 同其尘。 601

什么是WeakMap WeakMap结构与Map结构类似, 也是用于生成键值对的集合。 // WeakMap 可以使用 set 方法...

Map与WeakMap

SilentLand的博客 77

Map与WeakMap

细说WeakMap

github_38618068的博客 91

先看一段简单的代码: <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>二维数组持...

es6新增数据结构Map()与WeakMap() 最新发布

qq_45284875的博客 231

const obj1 = { name: 'qxl'} const obj2 = { name: 'hello'} const str = 'hello world' const mapObj = new Map() mapO...

【数据类型】Set、Map、WeakSet、WeakMap 对比

进击的阿彬 54

Set MDN: Set 对象允许你存储任何类型的唯一值，无论是原始值或者是对象引用。let a = new Set(); undefined ...

Javascript ES6 WeakMap详解

luoqiang0831的专栏 545

前段时间在看技术文章时看到 WeakMap 很陌生 感觉有点out了 然后了解了下MDN上说 WeakMap 对象是一组键/...

WeakMap的学习与应用场景

weixin_38382659的博客 1479

WeakMap 是什么？ WeakMap 与 Map 类似，也是生成 键值对的组合，但是有区别： 1.WeakMap 只能接受对象...

es6 WeakMap

歪脖先生的博客 329

WeakMap 含义 WeakMap结构与Map结构类似，也是用于生成键值对的集合。 // WeakMap 可以使用 set 方法添...

Map和WeakMap Set和WeakSet详解，属性和方法（MDN）

furfur-jiang的博客 370

这里写目录标题MapObjects 和 maps 的比较属性方法WeakMap属性方法Set属性方法WeakSet方法判等 Map Ma...

Map和WeakMap(笔记)

旅行者的博客 47

{ // 添加元素 let map = new Map() map.set([1,2,3],'number') console.log(map) let map2 = new Map([['name','Jack'...

Map和WeakMap的方法和区别

ta_huang的博客 714

Map和WeakMap 使用new Map()语法进行声明， Map键的类型可以使用任意对象作为键（字符串，object类型）...

WeakMap WeakSet的用途

weixin_30764137的博客 146

1. WeakMap WeakSet有一个特性，就是加入其中的元素不会算入引用计数，所以当其他地方没有对对象的引用...

“相关推荐”对你有帮助么？



非常没帮助



没帮助



一般



有帮助



非常有帮助

©2022 CSDN 皮肤主题：黑客帝国 设计师：我叫白小胖 返回首页

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 ©1999-2022北京创新乐知网络技术有限公司 版权与免责声明 版权申诉 出版物许可证 营业执照