

「前端进阶」JS中的栈内存堆内存



云中桥 Lv5

2019年06月25日 00:32 · 阅读 23185

+ 关注

 520

 59

 收藏



你知道的越多，你不知道的越多

👍 520

💬 59

★ 收藏

引言

JS的内存空间分为栈(stack)、堆(heap)、池(一般也会归类为栈中)。

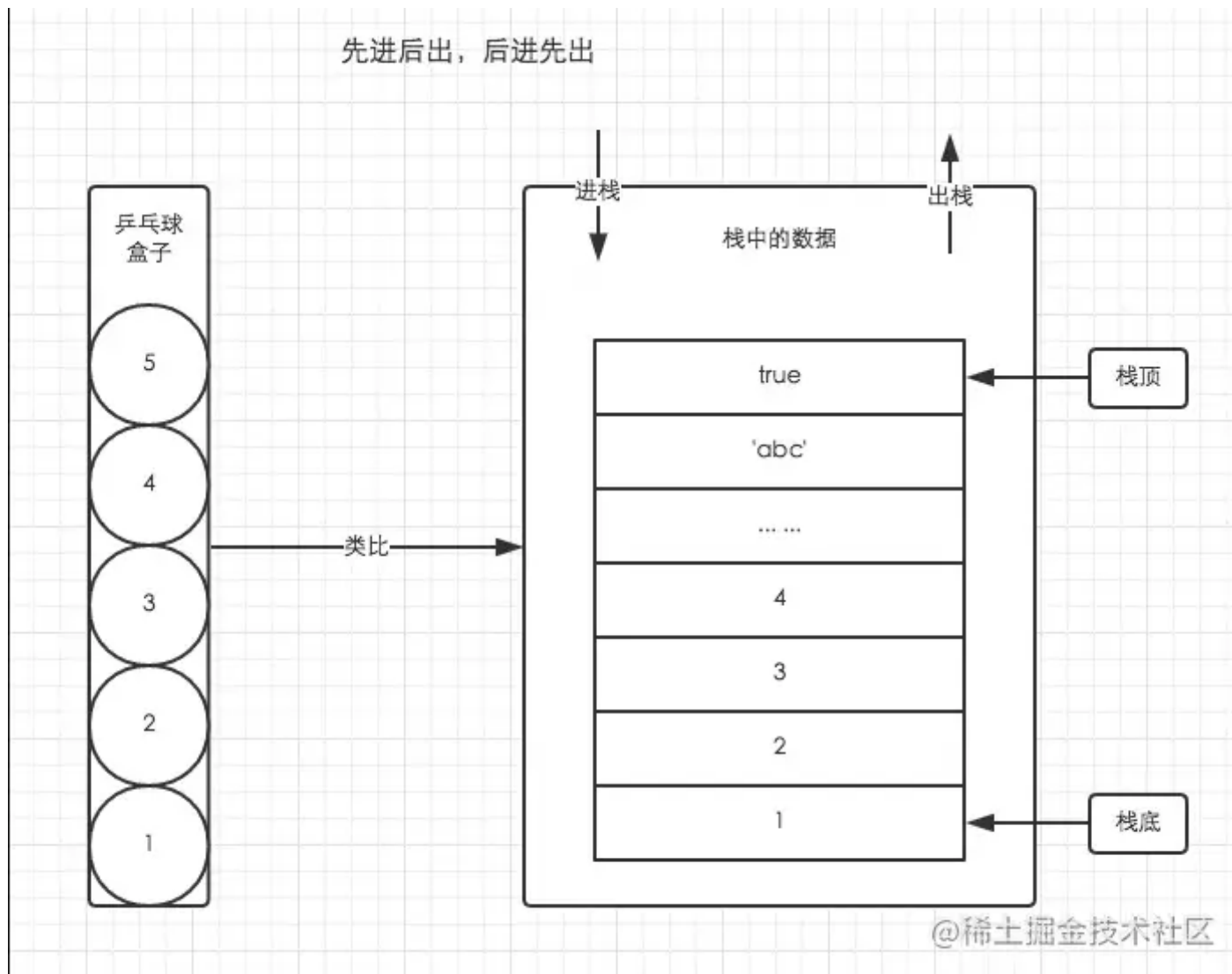
其中栈存放变量，堆存放复杂对象，池存放常量，所以也叫常量池。

栈数据结构

栈是一种特殊的列表，栈内的元素只能通过列表的一端访问，这一端称为栈顶。栈被称为是一种后入先出（LIFO，last-in-first-out）的数据结构。由于栈具有后入先出的特点，所以任何不在栈顶的元素都无法访问。为了得到栈底的元素，必须先拿掉上面的元素。

在这里，为方便理解，通过类比乒乓球盒子来分析栈的存取方式。





这种乒乓球的存放方式与栈中存取数据的方式如出一辙。处于盒子中最顶层的乒乓球 5，它一定是最后被放进去，但可以最先被使用。而我们想要使用底层的乒乓球 1，就必须将上面的 4 个乒乓球取出来，让乒乓球 1 处于盒子顶层。这就是栈空间先进后出，后进先出的特点。

👍 520

💬 59

★ 收藏

堆数据结构

堆是一种经过排序的树形数据结构，每个结点都有一个值。通常我们所说的堆的数据结构，是指二叉堆。堆的特点是根结点的值最小（或最大），且根结点的两个子树也是一个堆。由于堆的这个特性，常用来实现优先队列，堆的存取是随意，这就如同我们在图书馆的书架上取书，虽然书的摆放是有顺序的，但是我们想取任意一本时不必像栈一样，先取出前面所有的书，我们只需要关心书的名字。

变量类型与内存的关系

基本数据类型

基本数据类型共有6种：

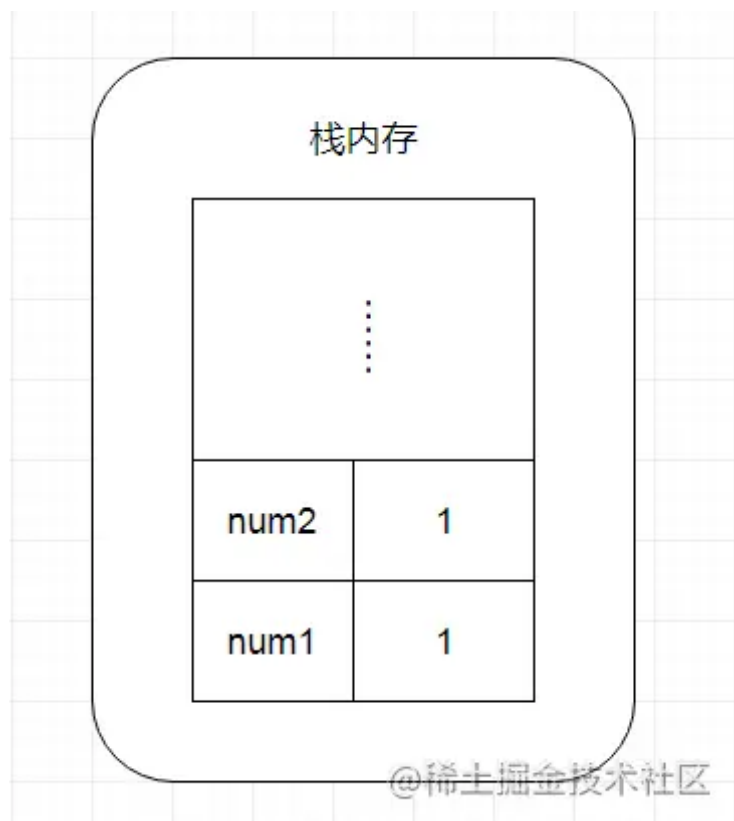
1. Sting
2. Number
3. Boolean
4. null
5. undefined
6. Symbol

基本数据类型保存在栈内存中，因为基本数据类型占用空间小、大小固定，通过按值来访问，属于被频繁使用的数据。

为了更好的搞懂基本数据类型变量与栈内存，我们结合以下例子与图解进行理解：



```
let num1 = 1;  
let num2 = 1;
```



PS: 需要注意的是闭包中的基本数据类型变量不保存在栈内存中，而是保存在堆内存中。这个问题，我们后文再说。

引用数据类型

👍 520

💬 59

★ 收藏

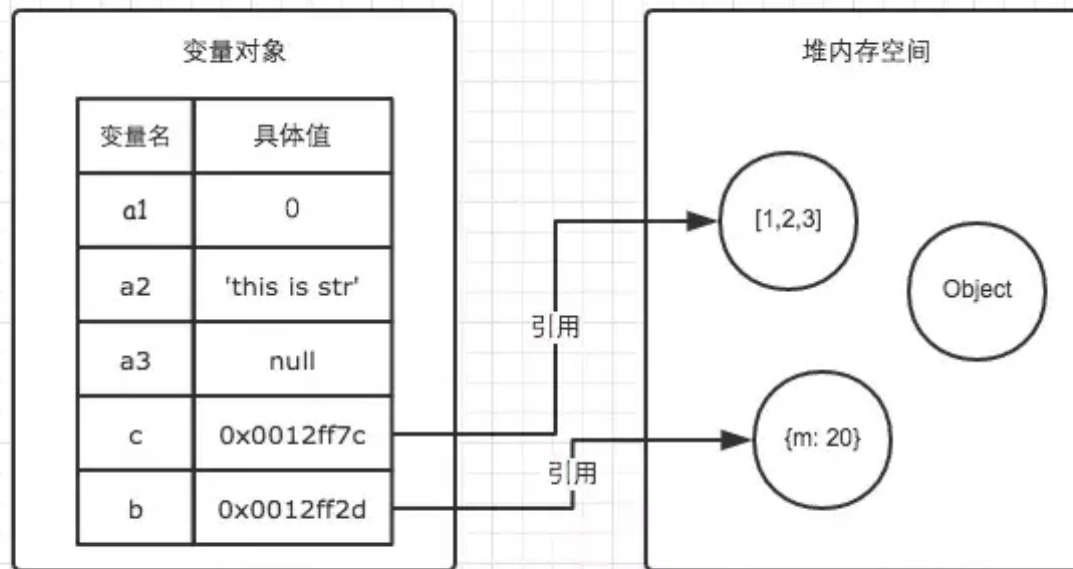
引用数据类型存储在堆内存中，因为引用数据类型占据空间大、大小不固定。如果存储在栈中，将会影响程序运行的性能；引用数据类型在栈中存储了指针，该指针指向堆中该实体的起始地址。当解释器寻找引用值时，会首先检索其在栈中的地址，取得地址后从堆中获得实体

为了更好的搞懂变量对象与堆内存，我们结合以下例子与图解进行理解。

```
// 基本数据类型-栈内存
let a1 = 0;
// 基本数据类型-栈内存
let a2 = 'this is string';
// 基本数据类型-栈内存
let a3 = null;

// 对象的指针存放在栈内存中，指针指向的对象存放在堆内存中
let b = { m: 20 };
// 数组的指针存放在栈内存中，指针指向的数组存放在堆内存中
let c = [1, 2, 3];
```

[复制代码](#)



@稀土掘金技术社区

因此当我们要访问堆内存中的引用数据类型时，实际上我们首先是从变量中获取了该对象的地址指针，然后再从堆内存中取得我们需要的数据。

从内存角度来看变量复制

基本数据类型的复制

复制代码

👍 520

💬 59

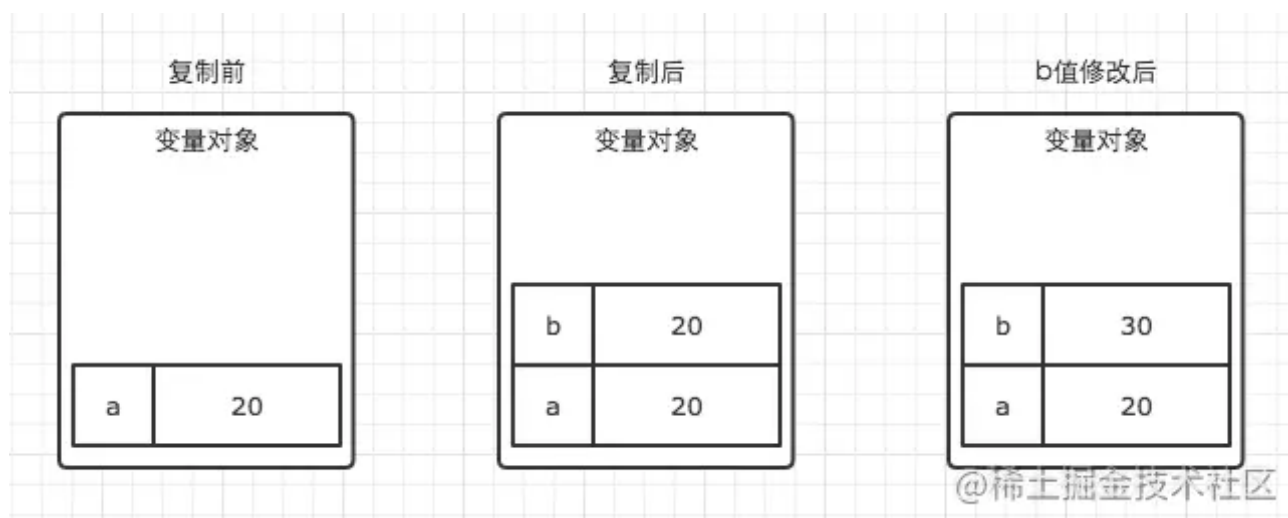
★ 收藏


```
b = 30;  
console.log(a); // 此时a的值是多少，是30？还是20？
```

答案是：20

在这个例子中，a、b 都是基本类型，它们的值是存储在栈内存中的，a、b 分别有各自独立的栈空间，所以修改了 b 的值以后，a 的值并不会发生变化。

从下图可以清晰的看到变量是如何复制并修改的。



引用数据类型的复制

```
let m = { a: 10, b: 20 };
```

复制代码

👍 520

💬 59

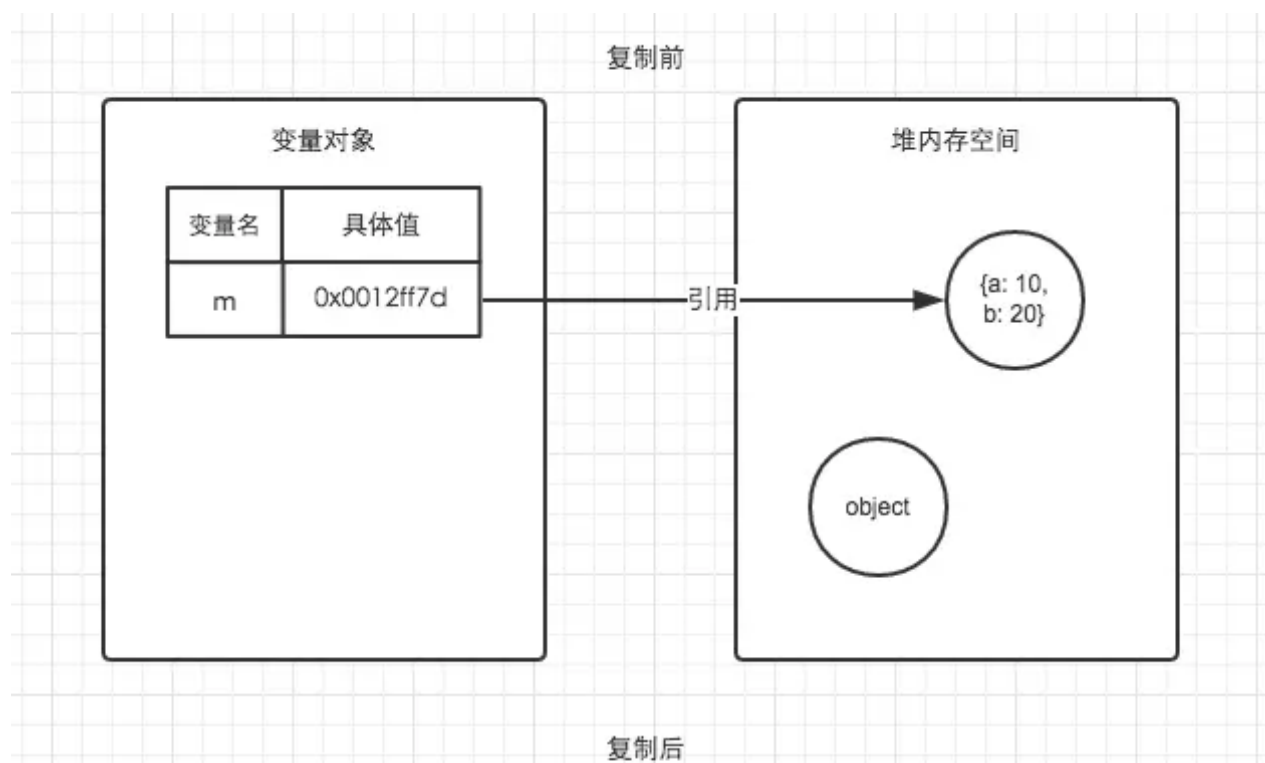
★ 收藏

```
n.a = 15;  
console.log(m.a) //此时m.a的值是多少，是10？还是15？
```

答案是：15

在这个例子中，m、n都是引用类型，栈内存中存放地址指向堆内存中的对象，引用类型的复制会为新的变量自动分配一个新的值保存在变量中，但只是引用类型的一个地址指针而已，实际指向的是同一个对象，所以修改 n.a 的值后，相应的 m.a 也就发生了改变。

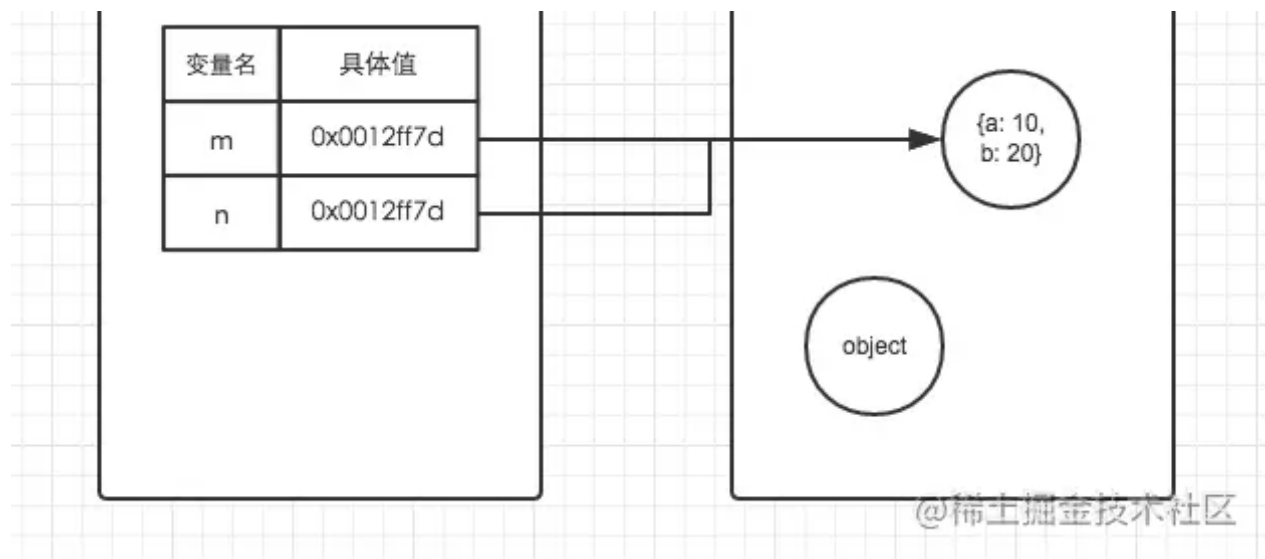
从下图可以清晰的看到变量是如何复制并修改的。



👍 520

💬 59

★ 收藏



栈内存和堆内存的优缺点

在JS中，基本数据类型变量大小固定，并且操作简单容易，所以把它们放入栈中存储。引用类型变量大小不固定，所以把它们分配给堆中，让他们申请空间的时候自己确定大小，这样把它们分开存储能够使得程序运行起来占用的内存最小。

栈内存由于它的特点，所以它的系统效率较高。堆内存需要分配空间和地址，还要把地址存到栈中，所以效率低于栈。

栈内存和堆内存的垃圾回收

栈内存中变量一般在它的当前执行环境结束就会被销毁被垃圾回收制回收，而堆内存中的变量则不会，因为不确定其他的地方是不是还有一些对它的引用。堆内存中的变量只有在所有对它的引用都结束的时候才会被回收。



520



59



收藏

[关于垃圾回收详细内容参见另一篇文章\(JS中的内存管理\)](#)

闭包与堆内存

闭包中的变量并不保存在栈内存中，而是保存在堆内存中。这也就解释了函数调用之后为什么闭包还能引用到函数内的变量。

我们先来看什么是闭包：

```
function A() {  
  let a = 1;  
  function B() {  
    console.log(a);  
  }  
  return B;  
}  
let res = A();
```

[复制代码](#)

函数 A 返回了一个函数 B，并且函数 B 中使用了函数 A 的变量，函数 B 就被称为闭包。

函数 A 弹出调用栈后，函数 A 中的变量这时候是存储在堆上的，所以函数B依旧能引用到函数A中的变量。现在的 JS 引擎可以通过逃逸分析辨别出哪些变量需要存储在堆上，哪些需要存储在栈上。

系列文章推荐

 520

 59

 收藏

- [「前端进阶」彻底弄懂函数柯里化](#)
- [「前端进阶」JS中的内存管理](#)
- [「前端进阶」数组乱序](#)

参考

- [JavaScript高级程序设计](#)
- [内存空间详细图解](#)

写在最后

- 文中如有错误，欢迎在评论区指正，如果这篇文章帮到了你，欢迎 [点赞](#) 和 [关注](#)
- 本文同步首发与[github](#)，可在[github](#)中找到更多精品文章，欢迎 [Watch](#) & [Star](#) ★
- 后续文章参见：[计划](#)

欢迎关注微信公众号 [【前端小黑屋】](#)，每周1-3篇精品优质文章推送，助你走上进阶之旅





分类:

前端

标签:

JavaScript



520

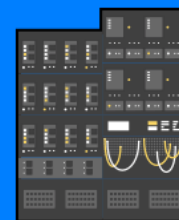


59



收藏

找对——
属于你的
技术圈子



回复进群加入
掘金
微信交流群



评论

输入评论 (Enter换行, Ctrl + Enter发送)

热门评论

 520

 59

 收藏

你是从哪了解到"基本数据类型保存在栈内存中，引用数据类型保存在堆内存中"的？有依据么？人云亦云？

👍 7 💬 10

云中桥 Lv5 (作者)

2年前

《javascript高级程序设计》4.4 小结，有总结

👍 1 💬 回复

云中桥 Lv5 (作者)

2年前

我在文章的参考中，已经写明了

👍 点赞 💬 回复

查看更多回复 ▾

我在曾经眺望彼岸 Lv2 前端 @ 网易

2年前

有个问题：既然栈是后进先出的，如果我取最底层的变量，是不是要拿掉前面所有的变量才能取出来，也就是取出最先声明的变量成本是最大的？

👍 点赞 💬 6

Gavin_zijef Lv1

2年前

变量存在变量提升

👍 点赞 💬 回复

云中桥 Lv5 (作者)

2年前

👍 520

💬 59

☆ 收藏

越浅的变量（如全局变量）是不能轻易将数据从堆栈中取出的

👍 6 💬 回复

查看更多回复 ▾

全部评论 59

🕒 最新

🔥 最热



xiaosa66

3月前

不全对 可以看下这个：JavaScript中变量存储在堆中还是栈中？ - 六耳的回答 - 知乎

🔗 www.zhihu.com。供参考

👍 点赞 💬 回复



无言非影 Lv2 前端开发工程师 @ 深圳

3月前

学习学习

👍 点赞 💬 回复



AGANAIZIJI Lv1 前端

6月前

到底存在哪的，🔗 zhuanlan.zhihu.com。

👍 点赞 💬 回复



努努

8月前

👍 520

💬 59

☆ 收藏

👍 点赞 1



yangcrazy30 Lv2

1月前

读取和清除是两个不同的概念，读取是基于词法环境的，清除值得是垃圾回收，这里的后进先出指的是垃圾回收

👍 点赞 回复



米苏 J Lv1 猎豹移动

11月前

闭包那块，函数A执行形成的私有作用域中的变量，被放在堆内存缓存起来的才是闭包吧 浏览器里展示的scope 中 闭包也是 { a:1 }

👍 点赞 回复



TaRiz Lv1 bytedance

12月前

有个疑问，原生类型存放在栈里面，如果取到相应的变量呢

👍 点赞 回复



一溪之石 Lv2 前端 @ 自由前端er

1年前

关于 这个堆内存 一般所指二叉堆，我表示不太理解，我们定义的对象都有一个指针在栈里面，而且每个堆内存空间相互独立，为什么会有二叉堆描述的父子关系，能通过指针直接映射，为何要递归查询？还望赐教

👍 点赞 回复



黑灯瞎火_不速之客 前端

1年前

关于垃圾回收这块，引用计数清除和标记清楚的主要是堆内存吗

👍 520

💬 59

★ 收藏



我就不眨眼

8月前

同问

👍 点赞 💬 回复



yangcrazy30 Lv2 回复 我就不眨眼

1月前

是的

“同问”

👍 点赞 💬 回复



红花绿叶肉夹馍 前端 @ 电信

2年前

‘函数 B 就被称为闭包’——这个时候应该函数A是闭包吧

```
function A() {  
  let a = 1;  
  function B() {  
    console.log(a);  
  }...  
}
```

[展开](#)

👍 1 💬 回复



Barry焕 Lv1 前端切图仔

2年前

闭包是引用之前函数的变量所产生的一个变量对象

👍 520

💬 59

★ 收藏

kobe24 Lv1 前端软件工程师

2年前

闭包为啥能访问到函数内的变量，还是看的云里雾里的。能否解释得更详细一些呀 🤔

👍 点赞 💬 2

xmanlin Lv3

2年前

作用域链

👍 点赞 💬 回复

小善逸 Lv2

8月前

闭包是使用到的外层变量的集合，保存在堆中

👍 点赞 💬 回复

黄璇璇

2年前

基本数据类型那里 string打错了 😊

👍 点赞 💬 回复

Guokai Lv3 前端工程师 @ 阿里巴巴...

2年前

我总觉得闭包是一种现象，不是一个函数

👍 5 💬 回复

👍 520

💬 59

☆ 收藏

分析的很全面，感谢

👍 点赞 💬 回复

于先森 Lv1 前端工程师 @ 菜的睡不...

2年前

作者写的很棒 但是有两处可能有点疑问 一是引用数据类型中的变量对象栈中的存储顺序与之前所提不符，二是闭包时需要被外部引用后才会形成 光
光声明A函数并不会形成闭包 还需在外部const C = A() 不对处还往作者指出 谢谢。

👍 4 💬 回复

Gavin_zijef Lv1 前端

2年前

写得挺不错的，不过我看晚了

👍 点赞 💬 1

玄真君

2年前

晚了是什么意思

👍 点赞 💬 回复

我在曾经眺望彼岸 Lv2 前端 @ 网易

2年前

有个问题：既然栈是后进先出的，如果我取最底层的变量，是不是要拿掉前面所有的变量才能取出来，也就是取出最先声明的变量成本是最大的？

👍 点赞 💬 6

Gavin_zijef Lv1

2年前

👍 520

💬 59

☆ 收藏

👍 点赞 💬 回复



云中桥 Lv5 (作者)

2年前

后进先出，其中的先出，指的是垃圾回收，在编码的过程中，程序自上而下执行，嵌套越深的变量被出栈之后对程序的影响越小，嵌套越浅的变量（如全局变量）是不能轻易将数据从堆栈中取出的

👍 6 💬 回复

查看更多回复 ▾

郑鱼咚 Lv2 前端工程师 @ 一家不知...

2年前

最后一点关于闭包的地方，我有异义。闭包只是js在运行时的一种查找机制，和储存方式没什么关系。其他讲的都挺好。

👍 点赞 💬 3



Gavin_zijef Lv1

2年前

他说的是闭包中被引用的变量被保存在了堆内存中，

👍 点赞 💬 回复



artiely Lv1

2年前

那为什么大量使用闭包有内存泄漏的危险呢。 🤔

👍 点赞 💬 回复

查看更多回复 ▾

👍 520

💬 59

★ 收藏



子弈 Lv6 前端 @ Alibaba

2年前

其实只要学会底层语言，例如C，那么JavaScript高级封装的数据类型存储方式完全可以通过理解C指针从而变得更加透彻。

👍 2 💬 回复



purplezz Lv2 小小前端

2年前

讲的真的很详细,对我们初学者帮助很大 👍 👍

👍 点赞 💬 回复

查看全部 59 条回复 ▾

相关推荐

Gaby | 5月前 | JavaScript · 面试

🔥 连八股文都不懂还指望在前端混下去么

👁 18.1w 👍 4998 💬 241

Sunshine_Lin | 5月前 | 前端 · JavaScript · ECMAScript 6

「万字总结」熬夜总结50个JS的高级知识点，全都会你就是神!!!

👁 7.5w 👍 2175 💬 107

👍 520

💬 59

☆ 收藏

死磕 36 个 JS 手写题（搞懂后，提升真的大）

👁 10.3w 👍 3380 💬 185

大猪猪 | 1年前 | JavaScript

10分钟理解下js中的栈(stack)和堆(heap)

👁 2143 👍 5 💬 15

拜小白 | 2年前 | JavaScript

栈、堆、队列深入理解，面试无忧

👁 1.6w 👍 287 💬 22

儋周 | 3年前 | JavaScript · 前端

JavaScript 工作原理之一——引擎，运行时，调用堆栈(译)

👁 2808 👍 106 💬 2

sunshine小小倩 | 4年前 | JavaScript · 前端

this、apply、call、bind

👁 12.1w 👍 3146 💬 250

HollisChuang | 2年前 | Java

求你了，再问你Java内存模型的时候别再给我讲堆栈方法区了...

👁 1.4w 👍 337 💬 23

👍 520

💬 59

☆ 收藏

10个常见的前端手写功能，你全都会吗？

👁 8.8w 👍 2215 💬 178

程序员依扬 | 2年前 | 面试 · 前端

【1月最新】前端100问：能搞懂80%的请把简历给我

👁 52.7w 👍 9277 💬 308

hannie76327 | 9月前 | 前端

JS中的堆栈内存及闭包详解

👁 434 👍 3 💬 评论

老腰 | 2年前 | 前端 · JavaScript · 全栈

8年前端开发的知识点沉淀(不知道会多少字，一直写下去吧...)

👁 9.3w 👍 2645 💬 183

光光同学 | 3年前 | JavaScript

一次看懂Event Loop（彻底解决此类面试问题）

👁 8.7w 👍 1775 💬 133

老姚 | 2年前 | JavaScript · 前端

你未必知道的CSS知识点（第二季）

👁 6.4w 👍 2481 💬 196

👍 520

💬 59

☆ 收藏

7分钟理解JS的节流、防抖及使用场景

👁 11.5w 🍏 2431 💬 177

大帅老猿 | 10月前 | 前端 · JavaScript

产品经理：你能不能用div给我画条龙？

👁 10.8w 🍏 2903 💬 576

浪里行舟 | 3年前 | JavaScript · 前端

九种跨域方式实现原理（完整版）

👁 13.6w 🍏 2607 💬 107

荒山 | 2年前 | 前端 · 团队管理

if 我是前端团队 Leader，怎么制定前端协作规范？

👁 15.7w 🍏 4224 💬 227

敖丙 | 1年前 | 面试 · Java

面试官：怎么排查堆内存溢出呀？

👁 1.8w 🍏 246 💬 39

yck | 5月前 | 前端 · JavaScript · GitHub

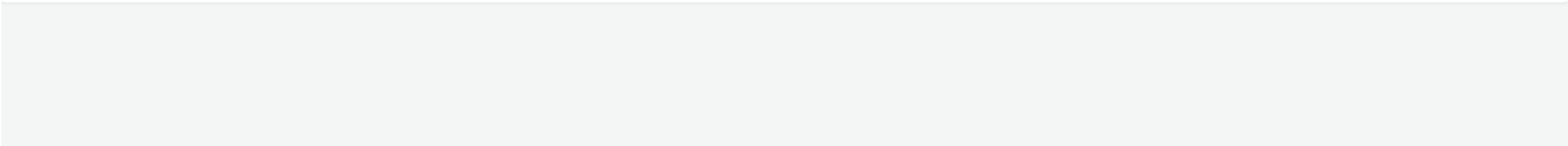
近 20 人爆肝数周，写给初中级前端的万字高级进阶指南

👁 10.0w 🍏 3505 💬 116

🍏 520

💬 59

☆ 收藏



 520

 59

 收藏