

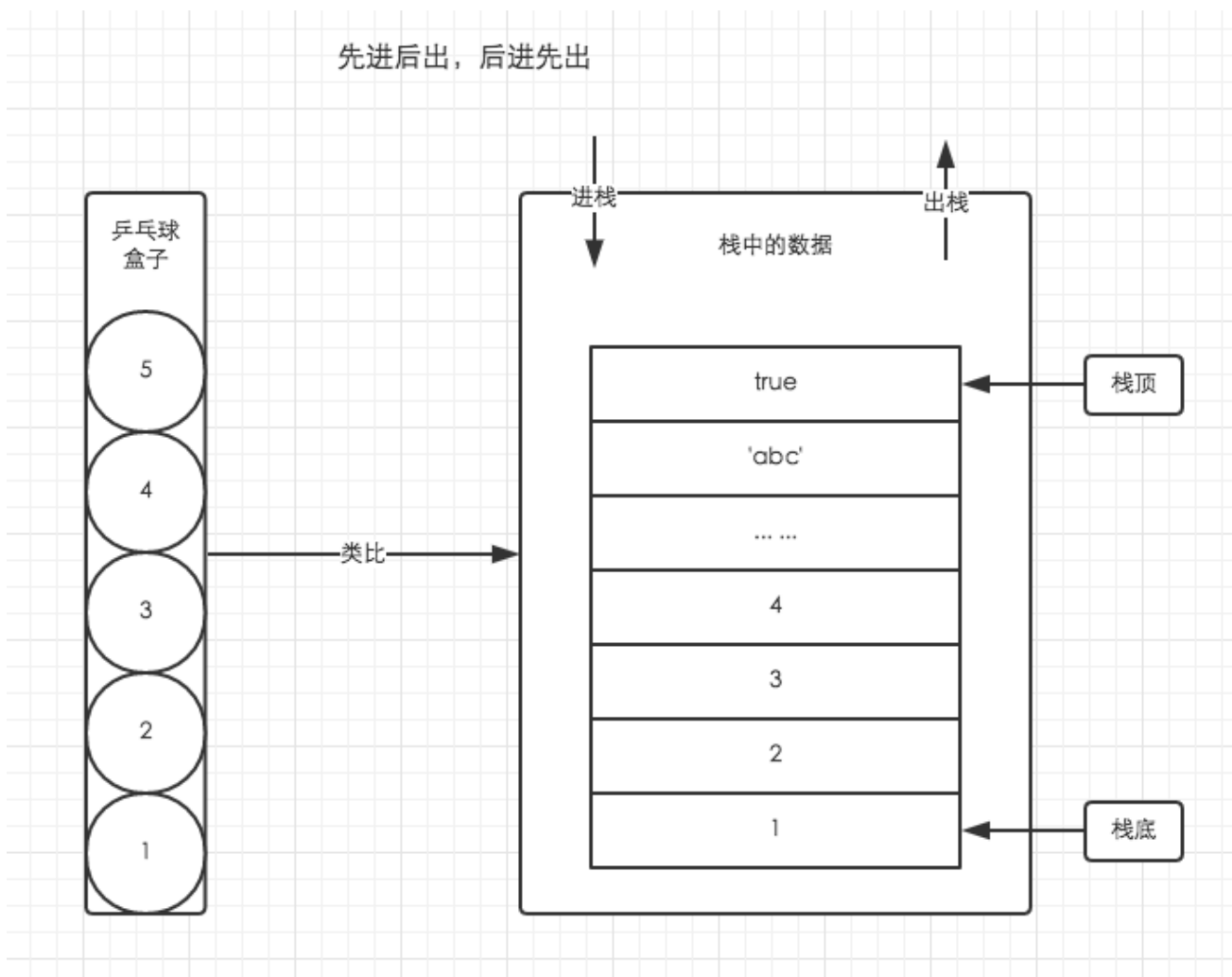
JavaScript 內存空間圖解

資料結構

stack 棧

整齊

乒乓球盒



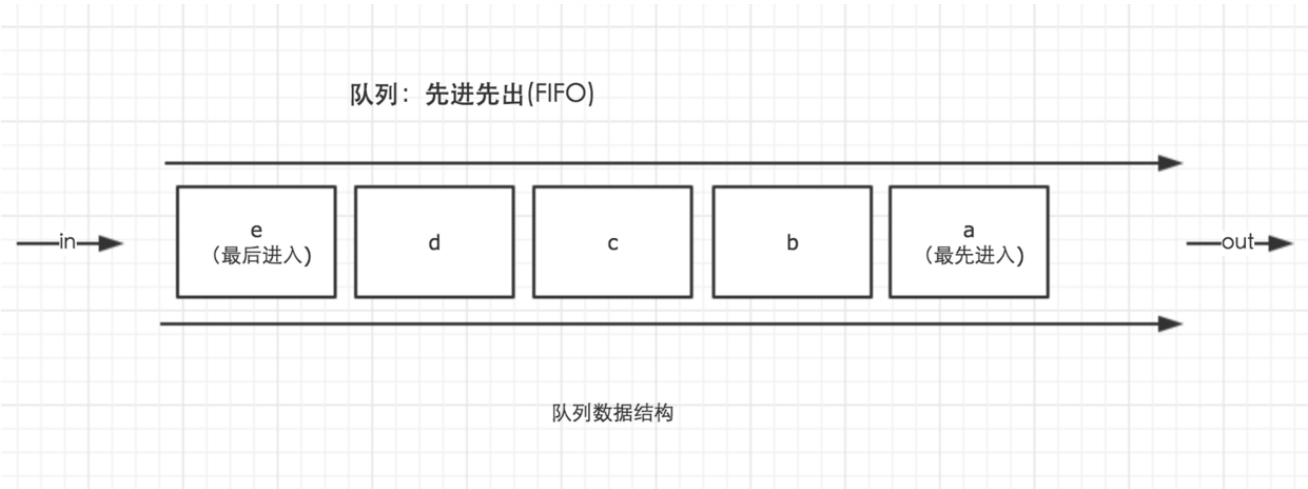
heap 堆

不整齊

書架與書

書雖然也整齊的存放在書架上，但是我們只要知道書的名字，我們就可以很方便的取出我們想要的書，而不用像從乒乓球盒子裡取乒乓一樣，非得將上面的所有乒乓球拿出來才能取到中間的某一個乒乓球。好比在 JSON 格式的數據中，我們存儲的key-value是可以無序的，因為順序的不同並不影響我們的使用，我們只需要關心書的名字。

queue 隊列



比較

棧（stack）的尋址速度要快於堆（heap）。

	結構	存放方式及大小	分配釋放者	存放數據
堆（heap）	沒有結構	任意存放，大小未知	程序員 / 由 OS 回收	運行時數據區、對象、自定義類型等
棧（stack）	有結構	按次序存放，每個區塊大小確定	操作系統	函數的參數值、局部變量的值等

JavaScript 的 2 種數據類型

JavaScript 的數據類型分為兩大類：

- 1. 基本類型：Undefined、Null、Boolean、Number 和 String
- 2. 引用類型：Object、Array、Function 等，變量實際保存的是一個指針，指向另一個位置

	內存分配	數據大小	訪問方式	本質
基本類型	棧 (stack)	確定	按值訪問	數據段
引用類型	堆 (heap)	不確定	按引用訪問	對象

基礎數據類型, 變量對象

JavaScript 的執行上下文生成之後，會創建一個叫做變量對象的特殊對象，**JavaScript** 的基礎數據類型往往都會保存在變量對象中。

JavaScript 中有 5 中基礎數據類型，分別是 `Undefined`、`Null`、`Boolean`、`Number`、`String`。基礎數據類型都是按值訪問，因為我們可以直接操作保存在變量中的實際的值。

ES6 中新加了一種基礎數據類型 `Symbol`，可以先不用考慮他

引用數據類型, 堆內存

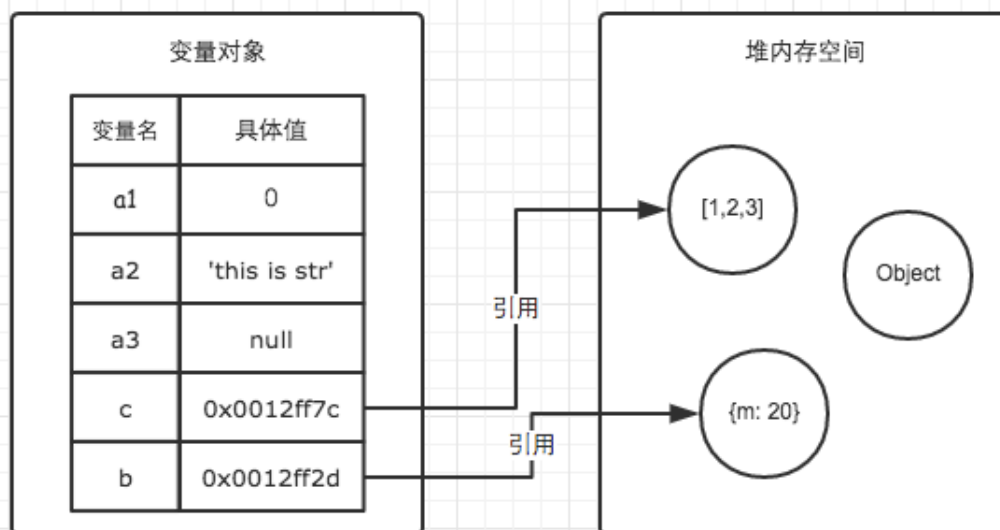
JS 的引用數據類型，比如數組 Array，它們值的大小是不固定的。引用數據類型的值是保存在堆內存中的對象。JavaScript 不允許直接訪問堆內存中的位置，因此我們不能直接操作對象的堆內存空間。

引用類型的值都是按引用訪問的。這裡的引用，我們可以理解為保存在變量對象中的一個地址，該地址與堆內存的實際值相關聯。

Example

```
var a1 = 0;    // 變量對象
var a2 = 'this is string'; // 變量對象
var a3 = null; // 變量對象

var b = { m: 20 }; // 變量b存在於變量對象中，{m: 20} 作為對象存在於堆內存中
var c = [1, 2, 3]; // 變量c存在於變量對象中，[1, 2, 3] 作為對象存在於堆內存中
```



垃圾回收

數據存放的規則：

棧（stack）：局部的、佔用空間確定的數據

- 每個函數 / 方法都會建立自己的內存棧，存儲函數參數和局部變量

堆（heap）：其他

- 所有的對象

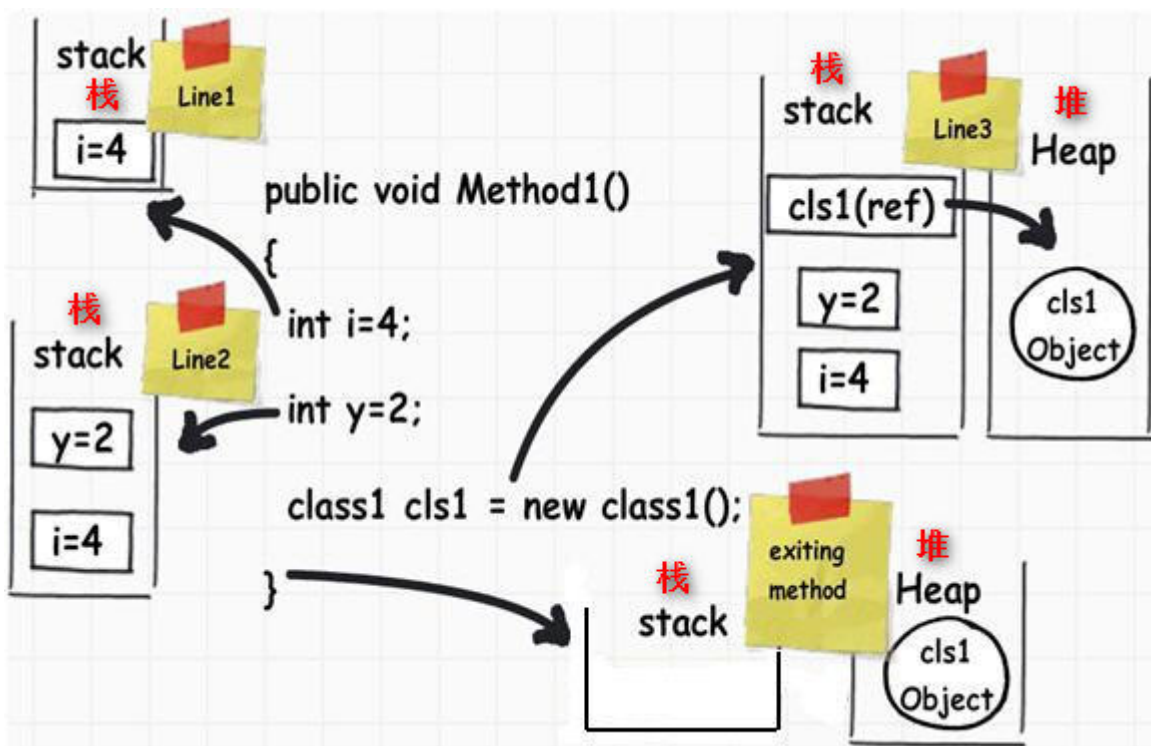
垃圾回收機制

- 隨著函數 / 方法的執行結束，這個函數 / 方法的內存棧也將自然銷毀
- 堆內存中的對象不會隨方法的結束而銷毀，即使方法結束後，這個對象還可能被另一個引用變量所引用（方法的參數傳遞時很常見），只有當一個對象沒有任何引用變量引用它時，系統的垃圾回收機制才會在核實的時候回收它。
- 因此，一般來說，內存洩漏都發生在堆（heap），即某些內存空間不再被使用了，卻因為種種原因，沒有被系統回收。

Example

```
public void Method1()
{
    int i=4;
    int y=2;
    class1 cls1 = new class1();
}
```

- 變量 i 和 y: 棧（stack）
 - 局部變量：只用在 Method1 區塊之內，不會用於區塊之外
 - 佔用的內存空間確定：整數
- 變量 cls1: 棧（stack）
 - 局部變量
 - 佔用的內存空間確定：指針，指向一個對象的實例
- 變量 cls1 的對象實例: 堆（heap）
 - 局部變量
 - 無法確知所佔用的內存空間大小



Method1 方法運行結束：

- 整個 stack 被清空，i、y 和 cls1 這三個變量消失，因為它們是局部變量，區塊一旦運行結束，就沒必要再存在了
- heap 之中的那個對象實例繼續存在，直到系統的垃圾清理機制（garbage collector）將這塊內存回收

資料來源

<http://www.jianshu.com/p/996671d4dcc4>

<http://www.jianshu.com/p/cd3fee40ef59>

<https://baoyuzhang.github.io/2017/07/03/%E3%80%90JS%E8%83%8C%E5%90%8E%E3%80%91JavaScript%E4%B8%AD%E7%9A%84%E5%A0%86%E6%A0%88/>