

主要有三方面：**对象存储；垃圾回收；内存池；**

对象存储：

1. python中**万物皆对象**，对于每块对象，python会分配内存空间；
- 2.对于**整数和短小的字符等**，python会执行**缓存机制**，即将这些对象进行缓存，不会为相同的对象分配多个内存空间；（**内存驻留**）
- 3.容器对象，如列表、元组、字典等，存储的其他对象，**仅仅是其他对象的引用，即地址，并不是这些对象本身**

内存池：

机制：预先在内存中申请**一定数量的，大小相等的内存块留作备用**，当有新的内存需求时，就先从内存池中分配内存给这个需求，**不够了之后再申请新的内存。**

背景：在 C 中如果频繁的调用 malloc 时,是会产生性能问题的，**频繁的分配与释放小块的内存会产生内存碎片；**

python内存池管理：

1. 一套是针对小对象，就是大小小于**256kb**时，pymalloc会在内存池中申请内存空间；
2. 当**大于256kb**，则会直接执行 new/malloc 的行为来**申请新的内存空间。**

优势：能够**减少内存碎片，提升效率。**

垃圾回收：引用计数为主，标记清除和分代收集为辅；

引用计数：

机制：python会记录每个对象被引用个数，增加一个引用，计数加一，反之减一，个数为0时进行**释放对象内存；**

引用计数增加情况：

- 1.对象被**创建；**
- 2.对象被**引用；**
- 3.对象被**作为参数传递函数中；**（引用记数会+2，函数中有两个属性在引用该对象）
- 4.对象**存储到容器对象中**

引用计数减少情况：

- 1.对象**别名被销毁；**
- 2.对象的**别名被赋予其他对象**（a=1;a=2 对1的引用会减1）
- 3.一个对象**离开它的作用域**（a作为参数传递到**sys.getrefcount(a)**函数中，只在函数中起作用，一旦执行完毕就会销毁。）
- 4.对象**从容器对象中删除，或者容器对象被销毁**

标记清除

机制：基于**追踪回收技术**实现垃圾回收，主要处理的是一些**容器对象循环引用**问题；

原理：它分为两个阶段：第一阶段是**标记阶段**，把所有的活动对象打上标记，第二阶段是把那些**没有标记的对象**非活动对象进行**回收**。对象之间通过引用（指针）连在一起，**构成一个有向图**，对象构成这个有向

图的节点，而引用关系构成这个有向图的边。从根对象出发，沿着有向边遍历对象，可达的对象标记为活动对象，不可达的对象就是要被清除的非活动对象。根对象就是全局变量、调用栈、寄存器。

缺点：清除非活动的对象前它必须顺序扫描整个堆内存，哪怕只剩下小部分活动对象也要扫描所有对象（分代收集优化性能）。

分代收集

作用：典型的以空间换时间的技术，通过为对象划分代，可以减少标记-清除机制所带来的额外操作，提升对象循环引用问题的检测性能；

机制：Python将所有的对象分为0，1，2三代。所有的新建对象都是0代对象。当某一代对象经历过垃圾回收，依然存活，那么它就被归入下一代对象。垃圾回收启动时，一定会扫描所有的0代对象。如果0代经过一定次数垃圾回收，那么就启动对0代和1代的扫描清理。当1代也经历了一定次数的垃圾回收后，那么会启动对0，1，2，即对所有对象进行扫描。

循环引用的检测主要是收集所有容器对象后（字符串、数值对象是不可能造成循环引用问题），使用双向链表对这些对象进行引用，并记录引用个数，对每个容器对象，找到其正在引用的对象，并将这个被引用的对象引用计数减1，全部处理后，若引用计数为0，说明该对象是在循环引用存活下来的，进行销毁，由此得知，循环引用查找和销毁过程非常繁琐要分别处理每一个容器对象，分代收集可改善性能；