# 对象内存分配与回收

## 对象有现在新生代Eden上分配

大多数情况下， 对象优先在新生代Eden区域中分配。当Eden区域没有足够的空间进行分配时，JVM将触发一次Minor GC（新生代GC）。

**大对象直接进入老年代**

所谓大对象是指需要大量连续内存空间的对象。JVM提供了一个-XX:PretenureSizeThreshold参数，令大于这个值得对象直接在老年代中分配。

**长期存活的对象将进入老年代**

JVM采用分代收集的思想管理内存，内存回收时就必须能识别哪些对象该放到新生代，哪些该放到老年代。为了做到这一点，JVM为每个对象定义了一个对象年龄Age，没经过一次新生代GC后仍然存活的对象，其年龄Age会增加1，当年龄到了一定程度（默认是15）时，将会被晋升到老年代中。对象晋升老年代的年龄限定值，可通过-XX:MaxTenuringThreshold来设置。

**Minor GC和Full GC的区别**

Minor GC（新生代GC）：指发生在新生代的垃圾收集动作，因为对象太多都具备朝生夕灭特性，所以Minor GC非常频繁，回收速度也比较快。

Major GC（老年代GC）：指发生在老年代中的GC，他执行的时候，经常会伴随至少一次的Minor GC，所以总体称为Full GC。Major GC的速度一般会比Minor GC慢10倍以上。