Garbage Collection

Computing

How automatic memory management works, and what it costs you.

The runtime periodically finds objects no longer reachable from the running program and reclaims their memory, so the programmer never calls free.

YOUNG GEN	OLD GEN	PAUSE
scanned often	rarely	ms to s

- Reachability, not reference counting alone, is the usual test.
- Generational GC: most objects die young, so scan the young ones often.
- A stop-the-world pause halts the program during collection.
- Reference counting frees immediately but cannot collect cycles.
- Rust avoids GC entirely with compile-time ownership rules.
- A 'leak' in a GC language is usually an unintended reference kept alive.

LOOK FOR IT

A cache with no eviction keeps every entry reachable, so nothing is ever collected. That is a leak in a GC language.

If latency matters, measure pause times, not just throughput. A high-throughput collector can still stall for a second.