**JS MILESTONE 7**

Garbage Collection

● Mark-and-Sweep

● Reference Counting

● Performance

● Managing Memory

1. Garbage collection:

* Javascript, automatically free up memory that is no longer in use.
* When objects are created dynamically, they occupy memory in heap.
* Even if a variable pointing to object is deleted, only reference is deleted. The object stays in memory itself.
* If we keep creating objects without freeing memory, it can lead to out of memory error.
* To prevent this, **garbage collection** automatically releases memory for objects that are no longer referenced/ unreachable in program.
* Garbage collector deallocates memory for unreferenced objects, ensuring available space for new objects to be allocated
* It tracks reference of object in memory and removes objects that are no longer referenced
  + Mark – sweep algorithm:
    - If the object is alive it is reachable from ‘root’.
    - Root refers to non-heap references
    - 1st marksweep finds non-heap references.
    - After this pointer-chasing is done, where we follow any references in any heap objects we encounter
    - As we go through the objects, we mark them.
    - Sweeping phase - Then while iterating, the objects which are not marked, are removed from the program
    - Initially it marks, identifying reachable objects, and clearing nonreachable objects from the heap.
    - The disadvantage of mark-sweep algorithm is fragmentation.
    - The reachable objects get fragmented with many small unused memory regions between them.
* Reference counting:
  + - Each object maintains a reference count, which indicates how many references point to it.
    - When an object's reference count drops to zero, it is immediately removed from memory because it is no longer reachable.
    - But this is not possible in circular reference.
    - Even if its trace is removed from the root, reference will not be erased. So it can’t be removed from the root.
    - To overcome this mark-sweep algorithm is used.