Python

Memory Management

Memory management

Reference counting works by counting the number of times an object is referenced by other objects in the system. When references to an object are removed, the reference count for an object is decremented. When the reference count becomes zero, the object is deallocated.

For example, Let’s suppose there are two or more variables that have the same value, so, what Python virtual machine does is, rather than creating another object of the same value in the private heap, it actually makes the second variable point to that originally existing value in the private heap. Therefore, in the case of classes, having a number of references may occupy a large amount of space in the memory, in such a case referencing counting is highly beneficial to preserve the memory to be available for other objects

Memory Parts

There are two parts of memory:

- Stack memory
- Heap memory

The methods/method calls and the references are stored in **stack memory** and all the values objects are stored in a **private heap**.