Your Ultimate Guide To Landing Top AI roles

2.3.3

Memory layout of a Program

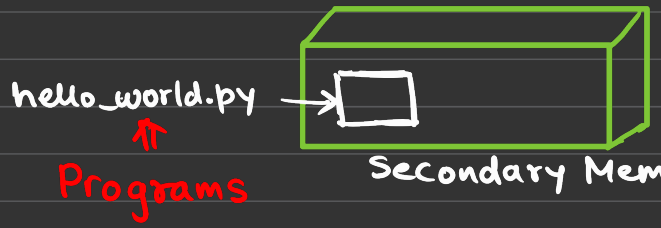
→ Since Python is a very high level programming language, it abstracts details of how a program is stored in memory during execution.

→ But understanding Memory layout is crucial to understand Recursion in DSA. And without Recursion, DSA is just like food without salt.

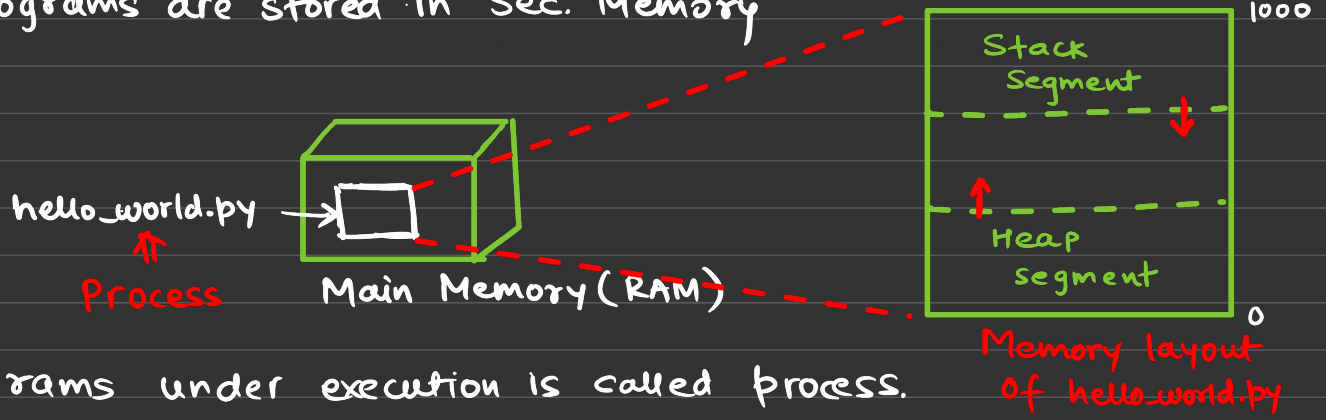
→ when you write a python program and save it as hello_world.py the program file is stored in Hard Disk. (Secondary Memory)

→ when you execute your program using cmd Python hello_world.py,

the program loads into main memory (RAM) and executes using CPU.



↳ Programs are stored in sec. Memory



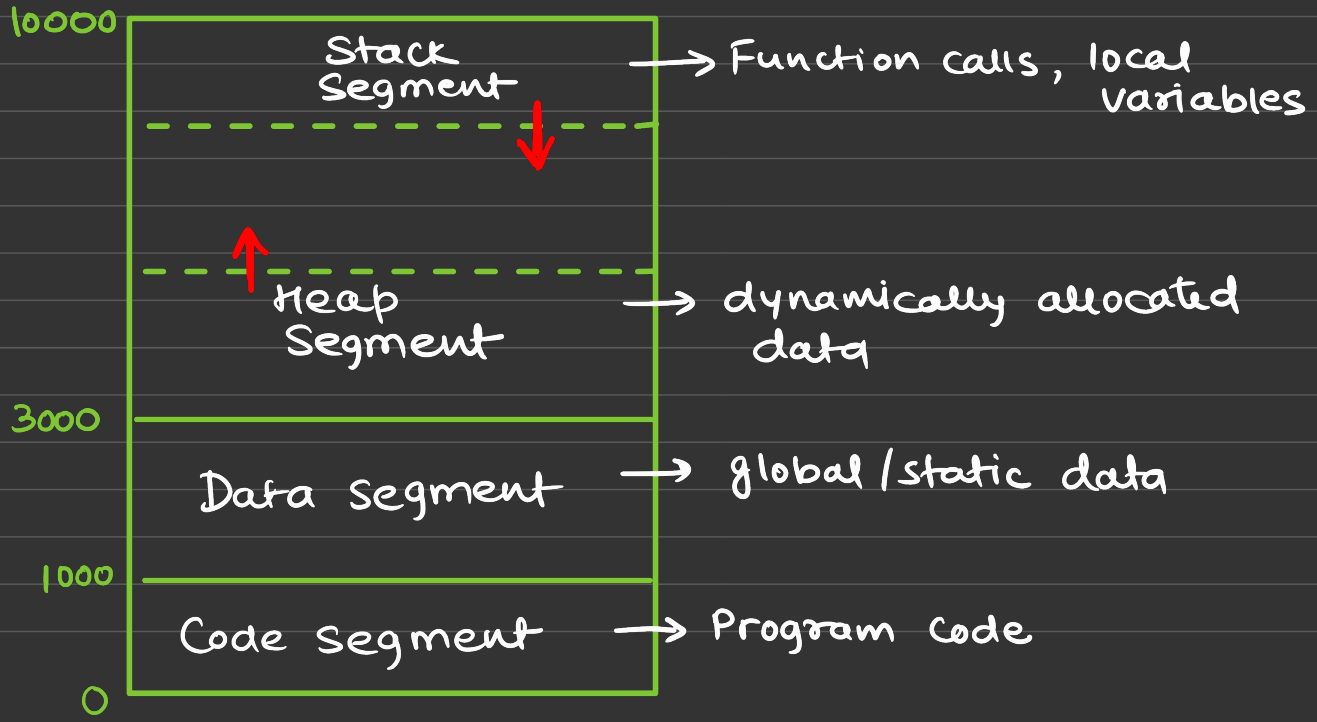
↳ Programs under execution is called process.

→ whenever we talk about memory in general, it means Main Memory.

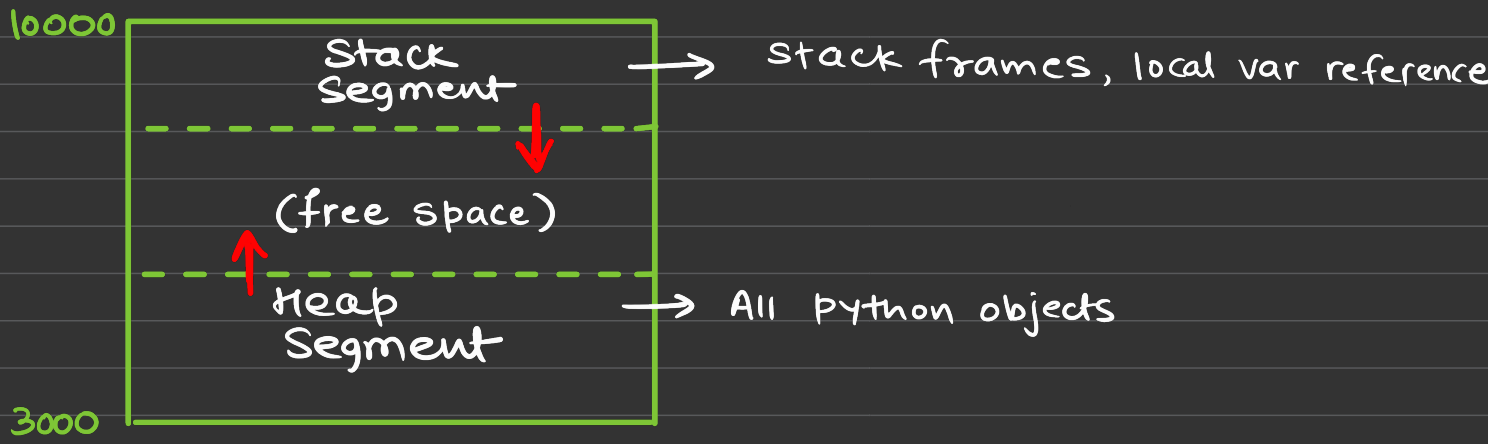
General Memory layout

→ To execute any program (Python/Java/c++), we need to load it in the main memory.

→ Let's see how the ~~programs~~ process looks like inside the Main memory.



→ Let's try to decode memory layout in Python with examples.



* Content of Heap Segment

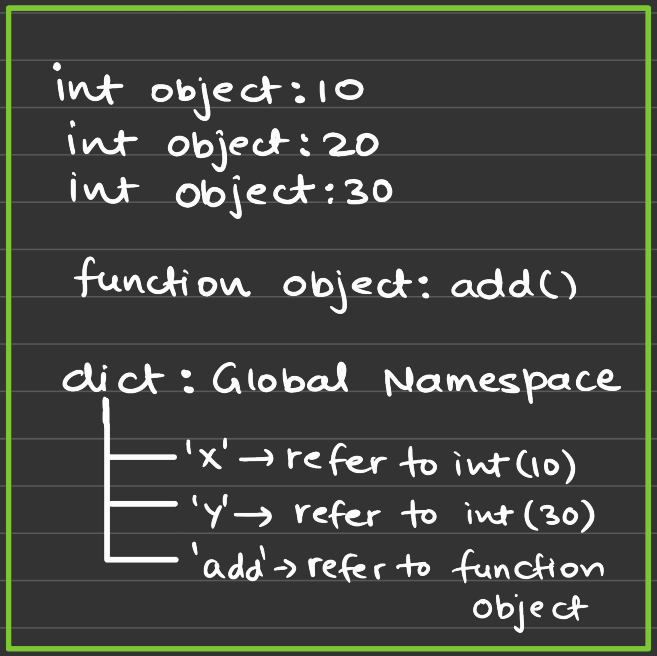
① All python objects : integers, strings, lists, functions etc

② All namespaces : globals() and locals() are dicts

③ Objects stay here as long as there are references to them.

Ex:-

```
x = 10
def add(a, b):
    result = a+b
    return result
y = add(x, 20)
```



NOTE

① The global namespace itself is a dict object in the heap

② All object it references are also in heap.

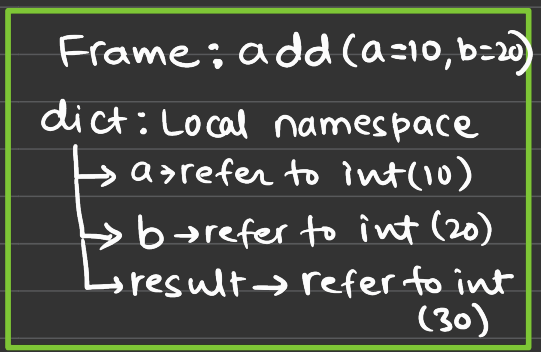
* Content of stack segment

① Call stack Frame : one per function call

② Each frame holds - Local Variables, Argument Passed and return address.

Ex:-

```
x = 10
def add(a, b):
    result = a+b
    return result
y = add(x, 20)
```



Stack

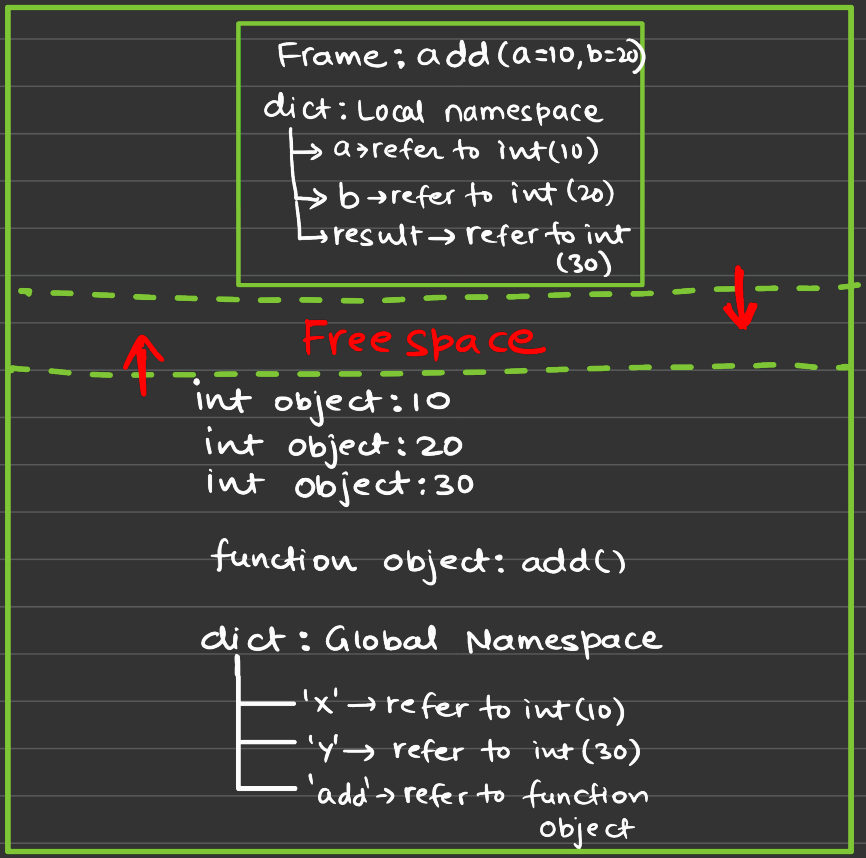
NOTE

① Programs in Python are stored as object of type code in the heap.

② Code objects represent byte-compiled executable Python code (ByteCode)

Ex:-

```
x = 10
def add(a, b):
    result = a+b
    return result
y = add(x, 20)
```




Like
Subscribe