IS IT ME OR PYTHON MEMORY MANAGEMENT?

ME: YULIIA BARABASH

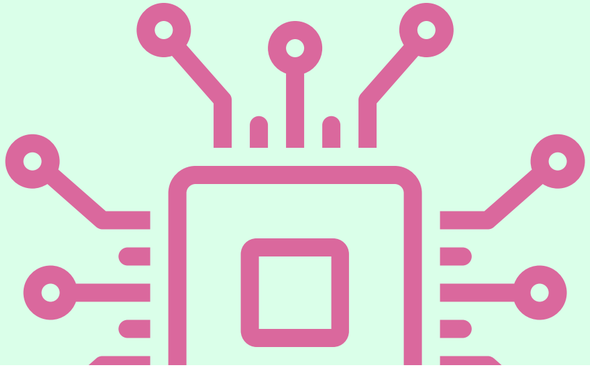
I AM CLOUD ENGINEER AT NORDCLOUD

ME: YULIIA BARABASH

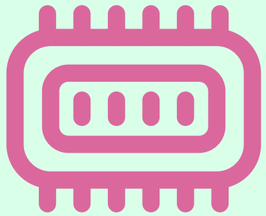
I AM CLOUD ENGINEER AT NORDCLOUD

Nordcloud
an IBM Company

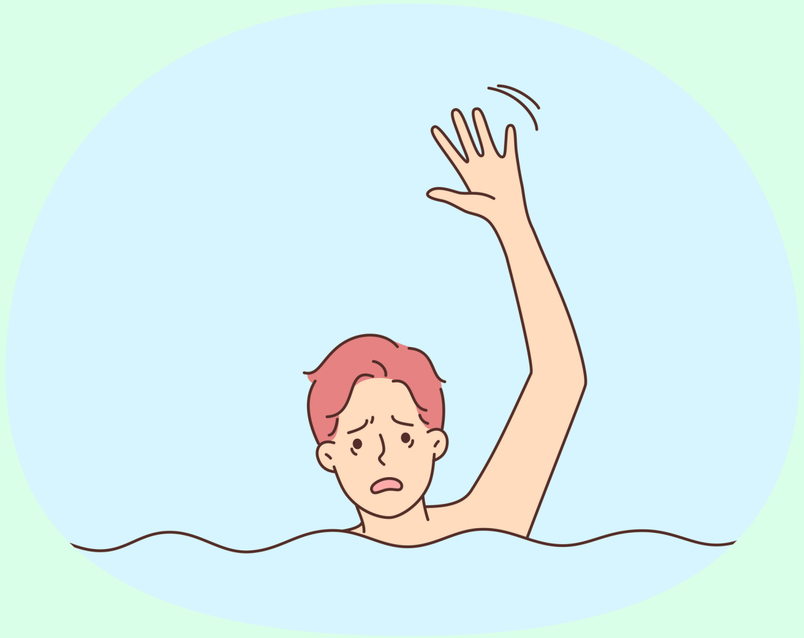
GOOD AT
UNDERSTANDING AND REMEBERING

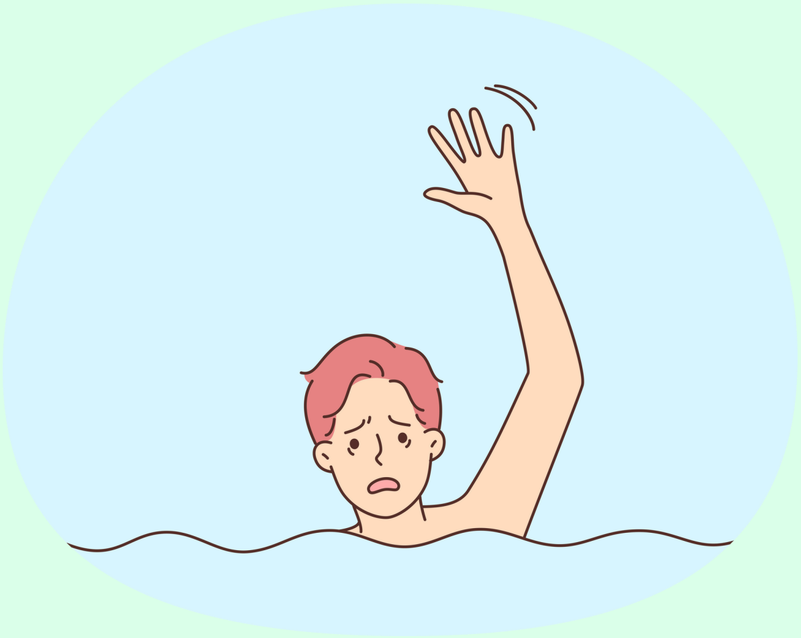


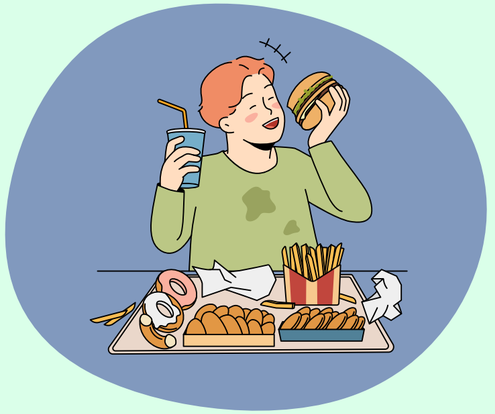
LET'S TALK ABOUT PERFOMANCE



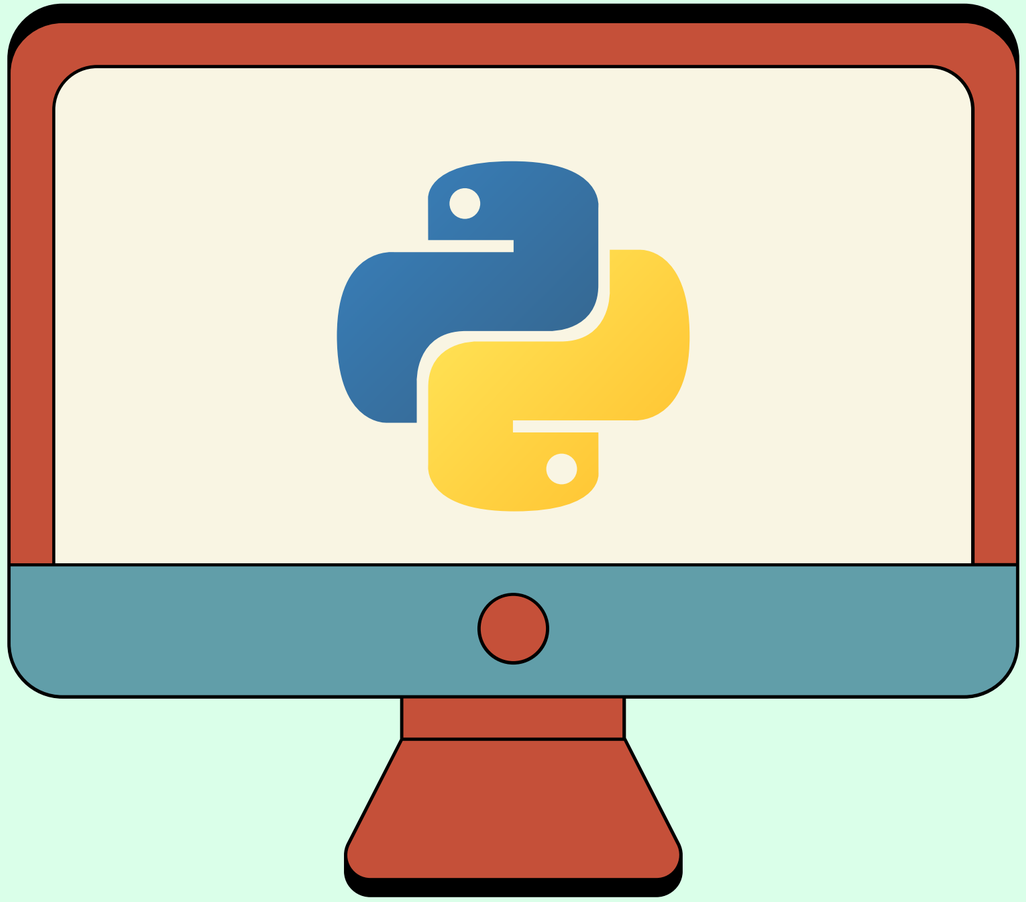
IT IS BOB
AND
HE WANTS TO BE
AN ATHLEAT





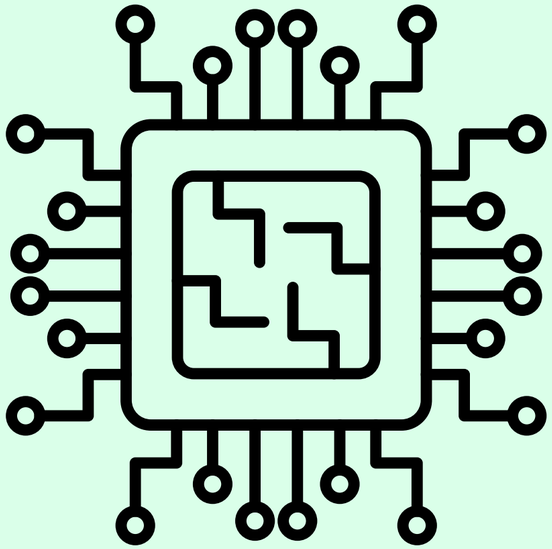


I AM NOT ALONE



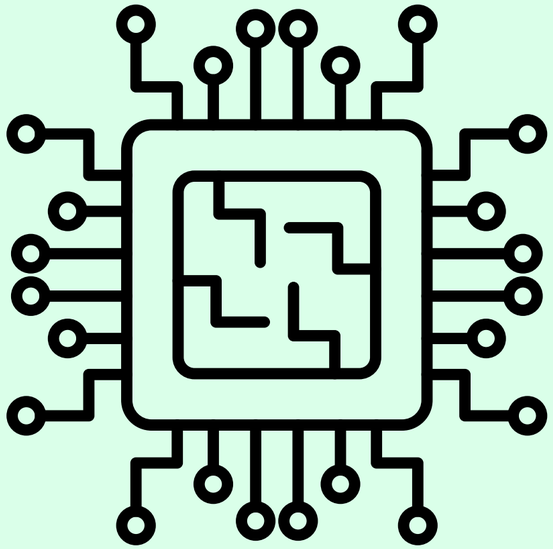
COMPONENTS OF HARDWARE:

COMPONENTS OF HARDWARE:

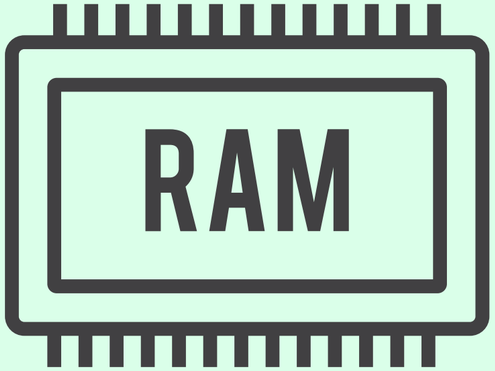


COMPUTING UNITS

COMPONENTS OF HARDWARE:

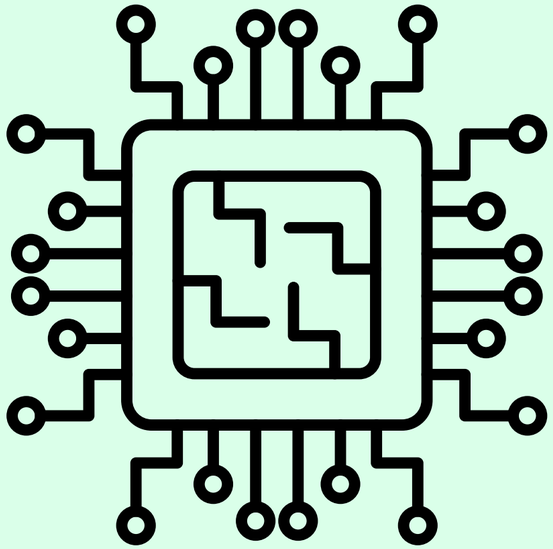


COMPUTING UNITS

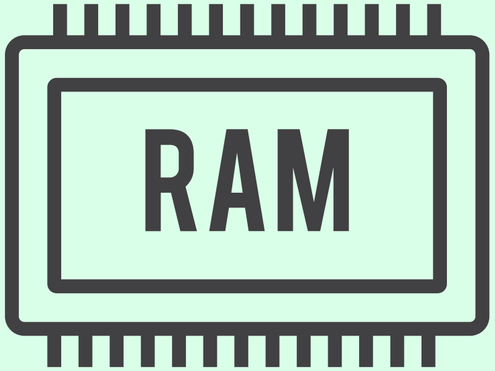


MEMORY UNITS

COMPONENTS OF HARDWARE:



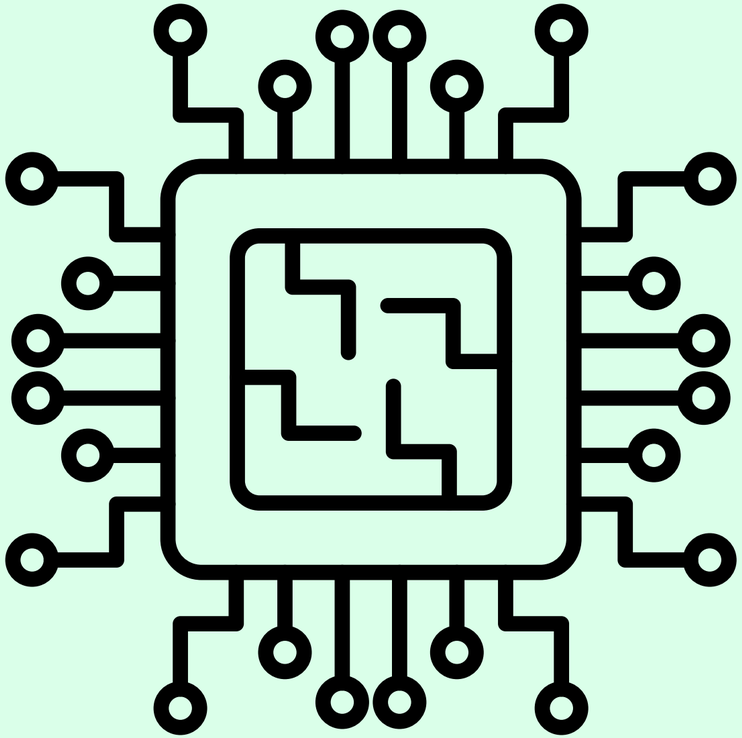
COMPUTING UNITS



MEMORY UNITS

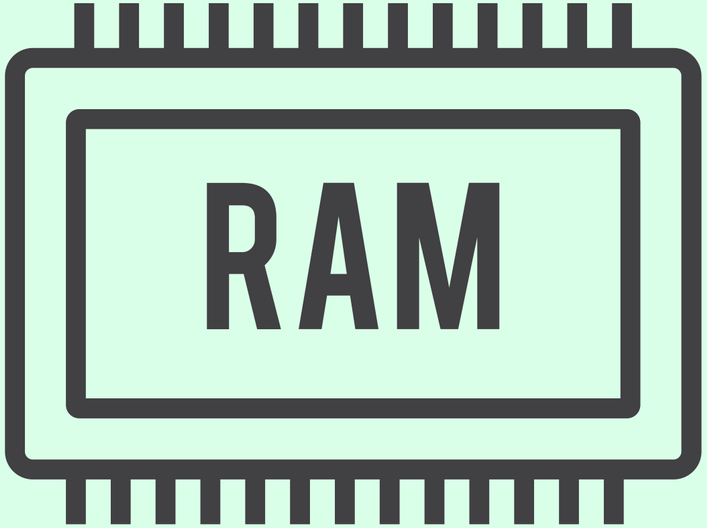
COMPUTING UNITS:

- responsible for execution of operations
- influence by:
  - number of machine code instructions
  - execution speed of instructions
- types:
  - CPU
  - GPU



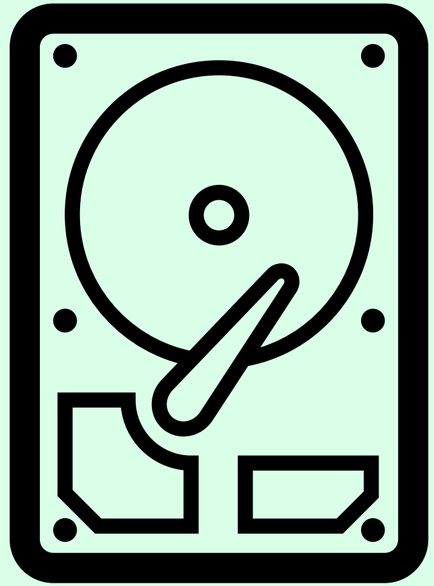
MEMORY UNITS:

- responsible for storing data and information
- influence by:
  - latency and architecture of memory units
- main examples:
  - Hard Disk Drive(HDD)
  - Solid State Drive(SSD)
  - Cache L1/L2
  - Random Access Memory (RAM)



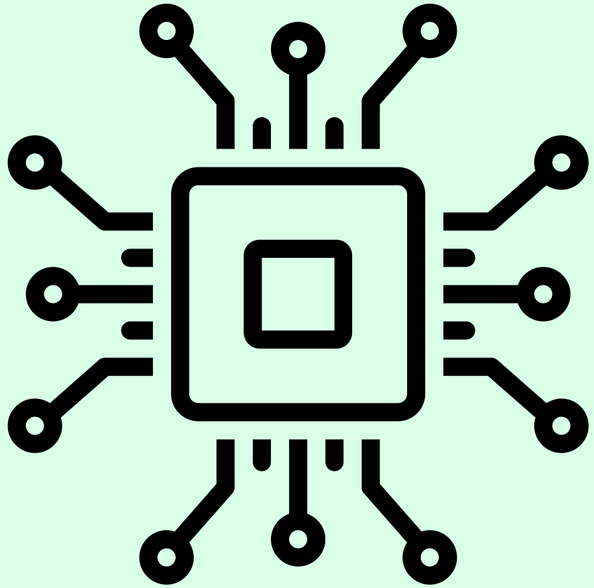
HARD DISK DRIVE(HDD)/SOLID-STATE DRIVE(SSD):

- long-term storage
- responsible for store the files and data needed for execution
- SSD has better performance the HDD
- lower speed write/read operations compare to RAM



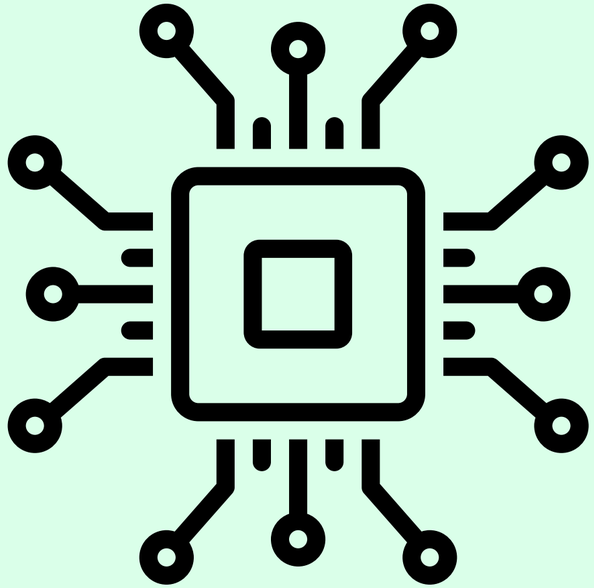
RANDOM ACCESS MEMORY (RAM):

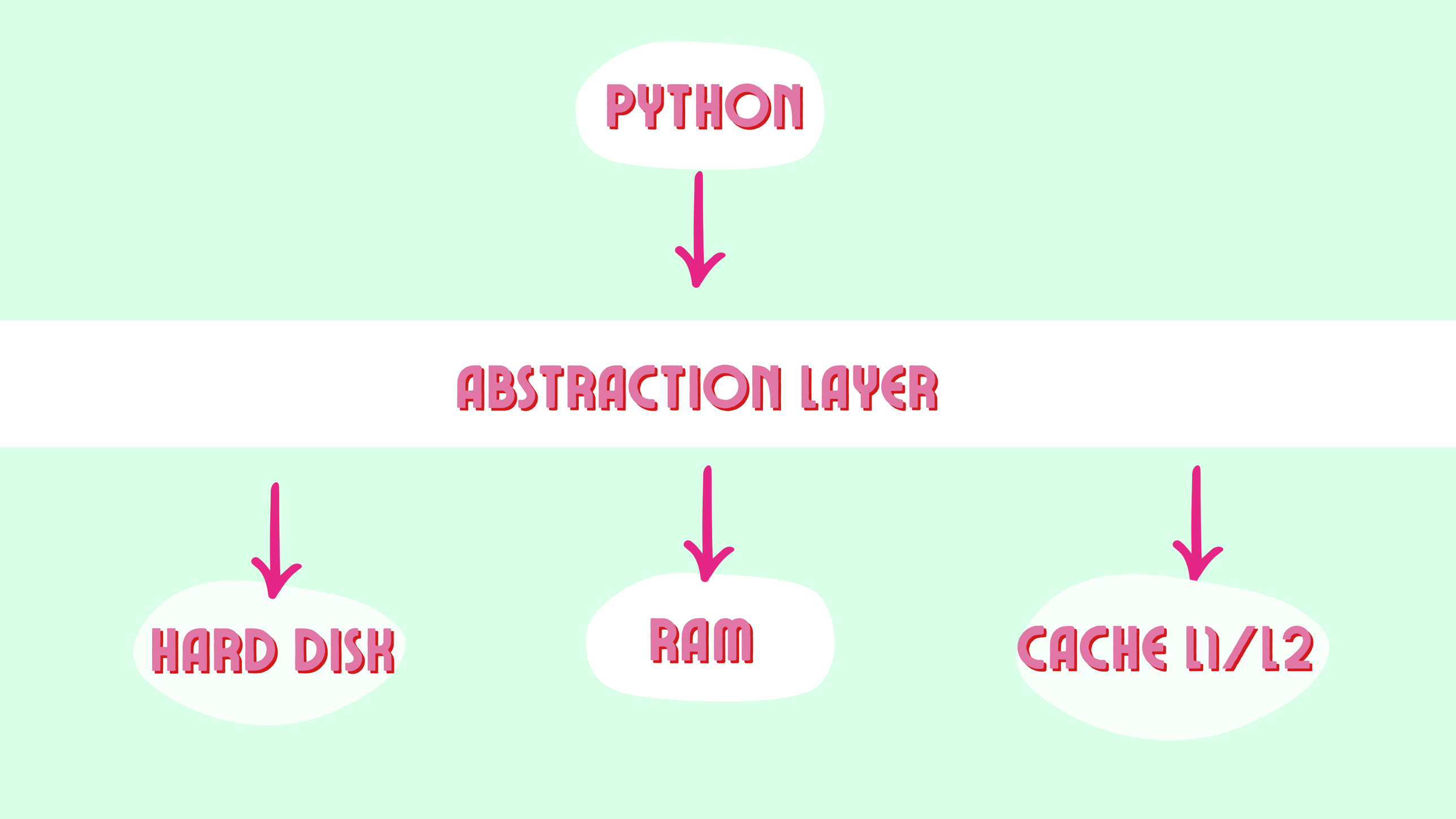
- short-term memory
- stores data and objects that are currently being used by program
- additionally it stores python private heap space



CACHE L1/L2:

- store most frequently used data by CPU
- has fastest and smallest type of memory
- example:
    - store most often executed instructions

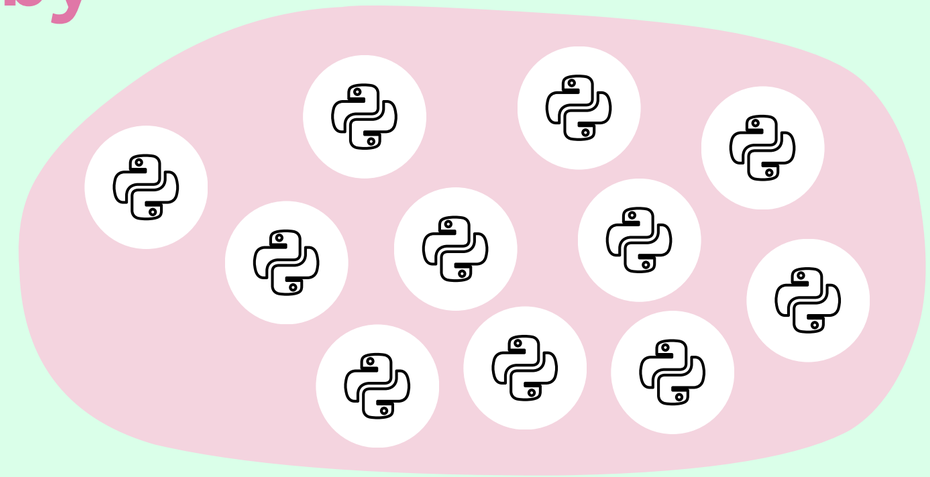



PYTHON
ABSTRACTION LAYER
HARD DISK
RAM
CACHE L1/L2


PYTHON
ABSTRACTION LAYER
RAM

A PYTHON PRIVATE HEAP SPACE:

- part of memory that is managed by Python
- everything is object in python
- store such as:
  - list, set, dictionaries
  - code
  - function
- is managed by memory manager




PYTHON
ABSTRACTION LAYER
RAM


PYTHON
MEMORY MANAGER
RAM

MEMORY MANAGER:

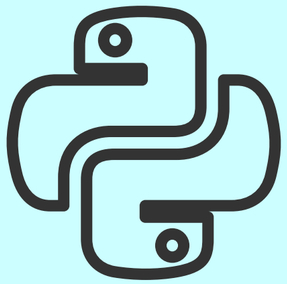
- responsible for managing memory inside private heap space
- has functionality to allocate and free space depends on state of application
- garbage collector is a part of memory manager

RECAP:

- Performance depends on execution environment
- Main components:
  - computing units
  - memory units
  - communication between them
- Types of memory
  - Hard Disk Drive (HDD)/Solid-State Drive (SSD)
  - Random Access Memory (RAM)
  - Cache L1/L2
- All objects are stored in private heap space
- Private heap space is manage by memory manager

STOP

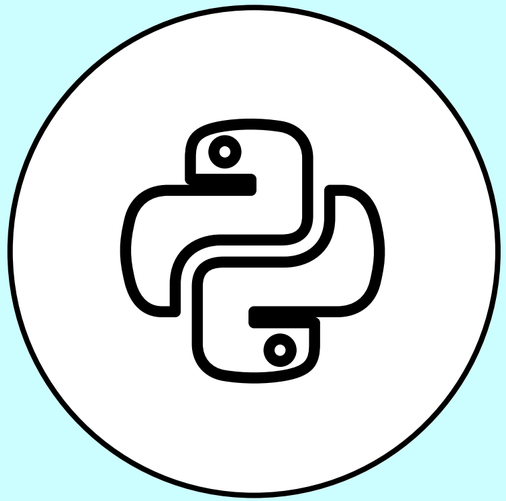
BUT HOW MEMORY MANAGER KNOWS WHAT TO DO



INSIDE PYTHON

EVERYTHING IS OBJECT:

- representations for:
  - code
  - function
  - none
  - integers
  - so on



OBJECTS IN PYTHON:

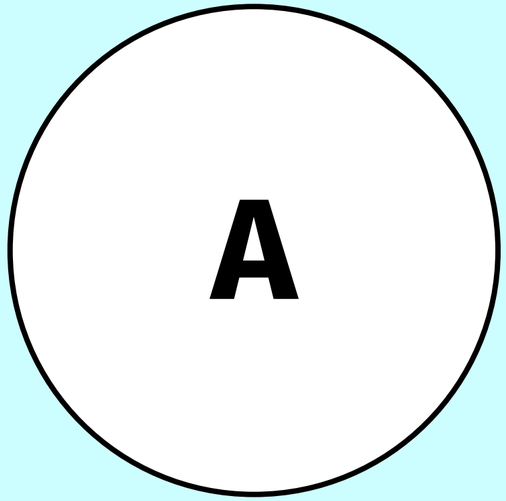
- Everything is an object inside python.
- When we create integer, Python create object inside private heap space

A = 3

VALUE	3
TYPE	INT
REF COUNT	1

VARIABLES IN PYTHON:

- references to the actual object in memory
- think of them as names or labels that point to the actual object in memory



EXAMPLE:

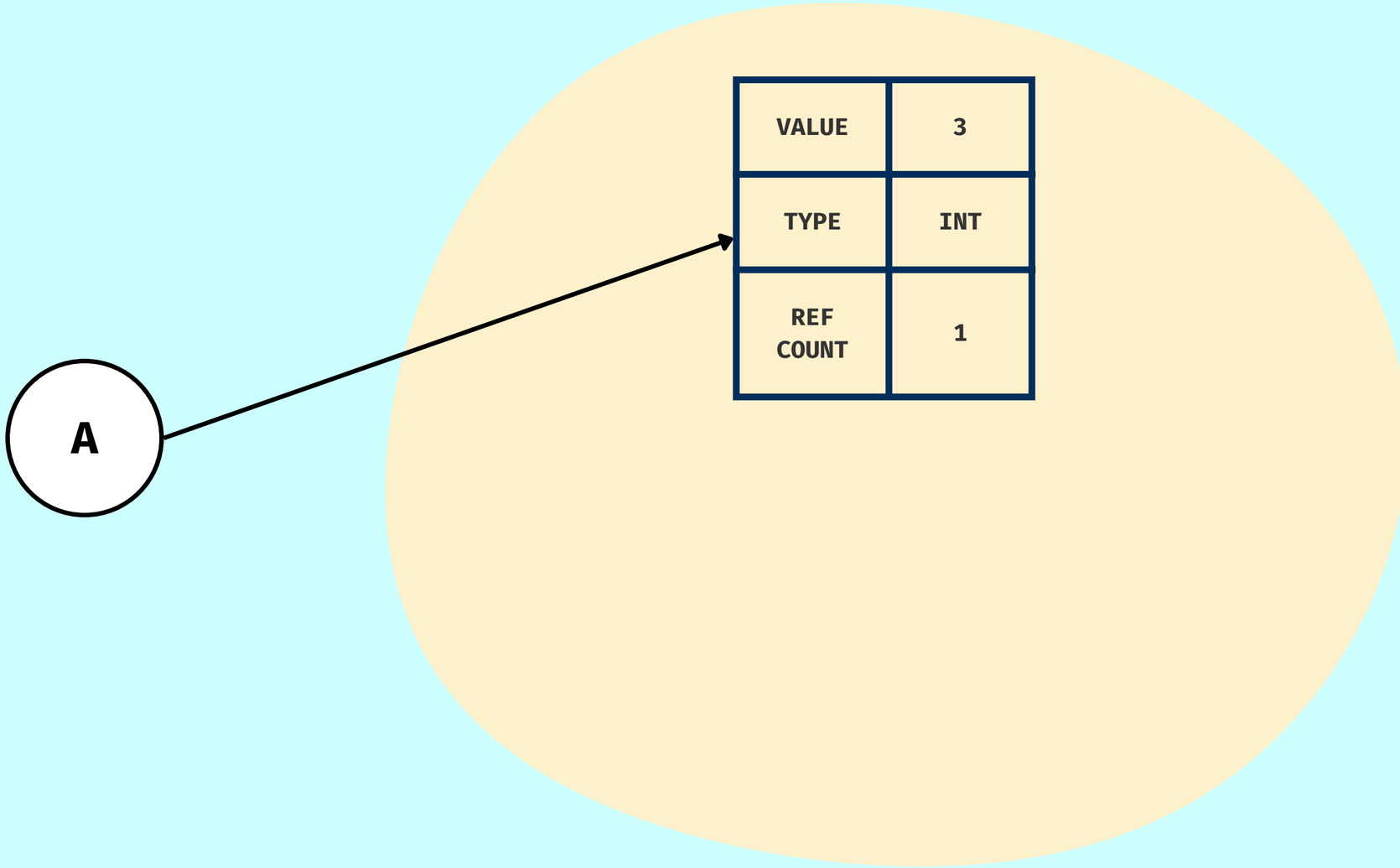
a = 3

a = 6

EXAMPLE:

a = 3

a = 6

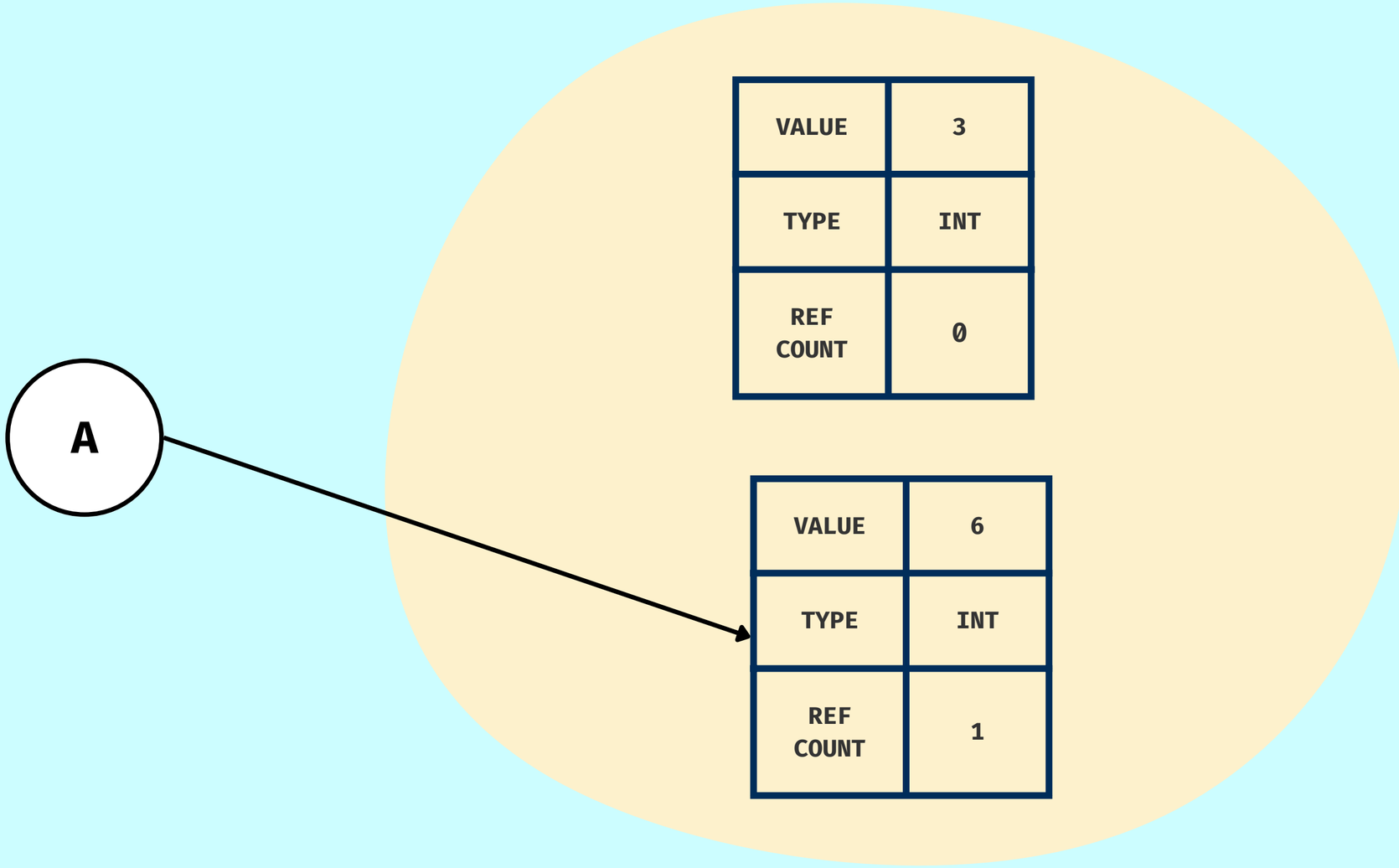


private heap

EXAMPLE:

a = 3

a = 6



private heap

STOP

BUT WHAT ACCTUALLY HAPPENDS TO THE OLD OBJECT

OBJECTS IN PYTHON:

VALUE	3
TYPE	INT
REF COUNT	1

strategies for managing objects:

- reference counting
- garbage collection

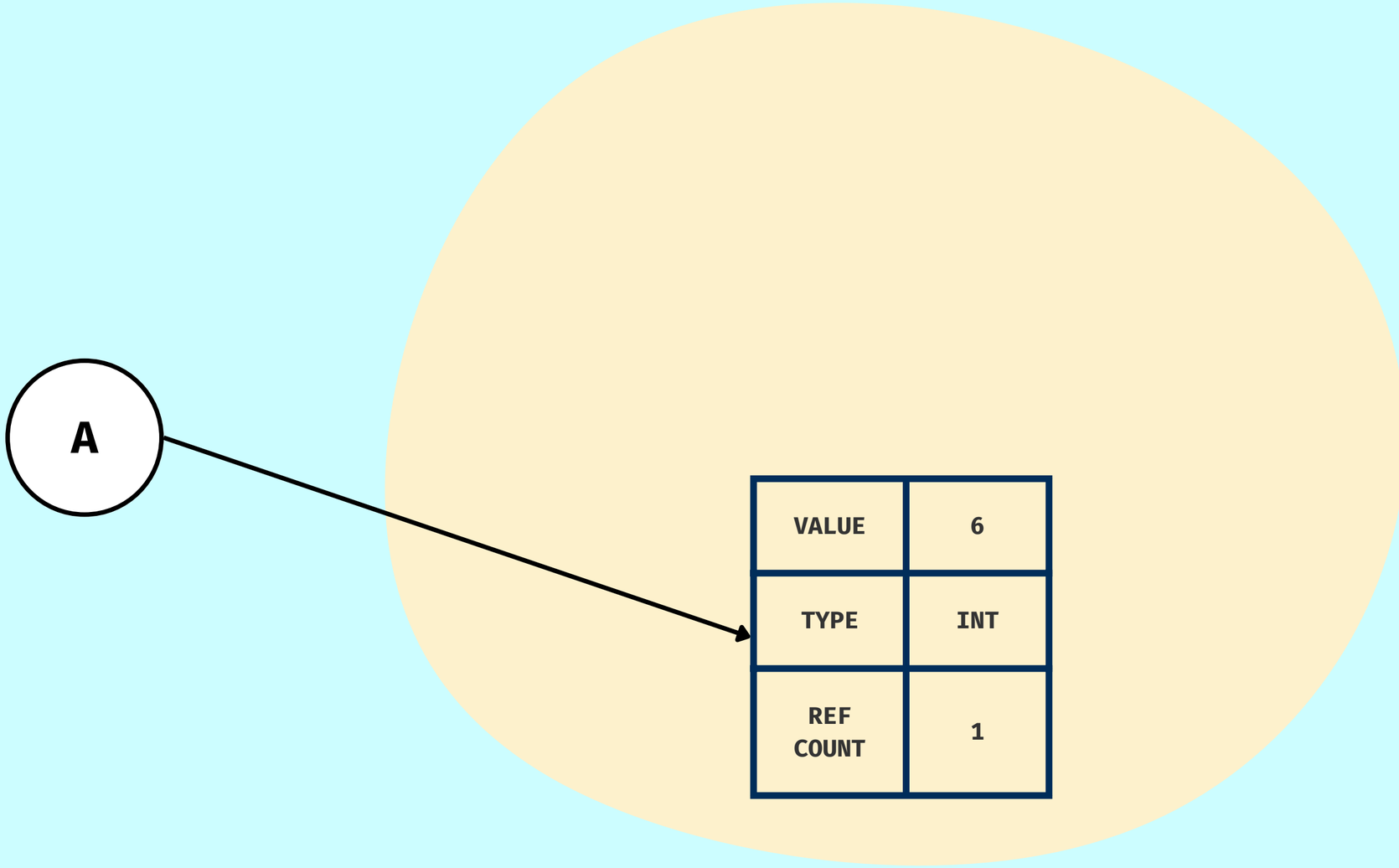
GARBAGE COLLECTOR:

- responsible for collection memory that no longer in used
- can be trigger manually and by threshold
- has module gc

EXAMPLE:

a = 3

a = 6



private heap

RECAP:

- think about about execution environment
- everything is object inside Python
- objects are stored in private heap space
- memory manager is responsible for private heap space
- memory manager relies:
  - reference counting
  - garbage collection

REFERENCES:

- Memory Management in Python
- CPython Internals: Your Guide to the Python 3
- High Performance Python

QUESTIONS