Types of Languages :-

\# At the very basic level computer translates any language into 0's and 1's.

① Procedural Language

↳ Specifies a series of well structured order of Statements, functions and commands to complete a task.

Ex → Java, Python, C, C++ etc.

② Functional Language

→ Block of Code

↳ Writing a program in pure functions
Also follow first class functions.

Ex→ Python

functions in the language is treated like any other Variables.

③ Object oriented Language

↳ Code + data = Object
Developed to make it easier to develop, debug and maintain.

Eg → Java, Python, C++

Static v/s Dynamic Languages

① Static Languages

→ have to specify the type
int a = 10 ✓
int a = "Yash" ✗

→ Compile Time Error
→ More Control

② Dynamic Languages

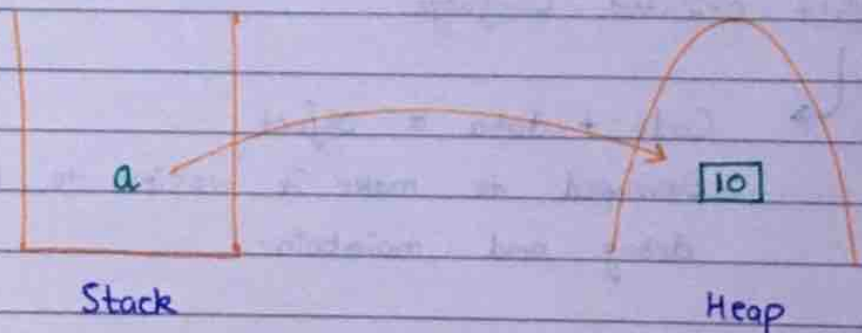→ Do not have to specify the type
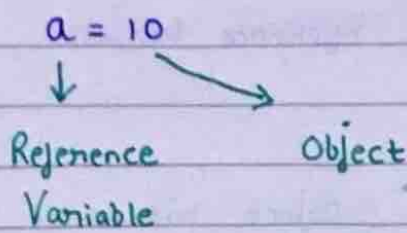a = 10 ✓
a = "Yash" ✓

→ Runtime Error
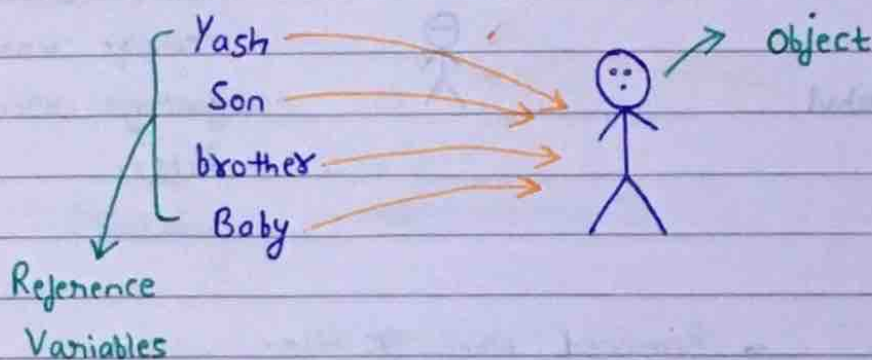→ Saves time

Stack and Heap memory

a = 10

Stack   Heap

Stack points towards heap.

a = 10   # Objects have a type
↓   Can be defined using
Reference   Object   Classes.
Variable

Yash ──────→
Son ──────→
brother ──────→   ──→ Object
Baby ──────→

Reference
Variables

if any change was made via Yash reference
Variable , original object will be changed

and since all of the reference Variables are pointing
towards the same object , change will be
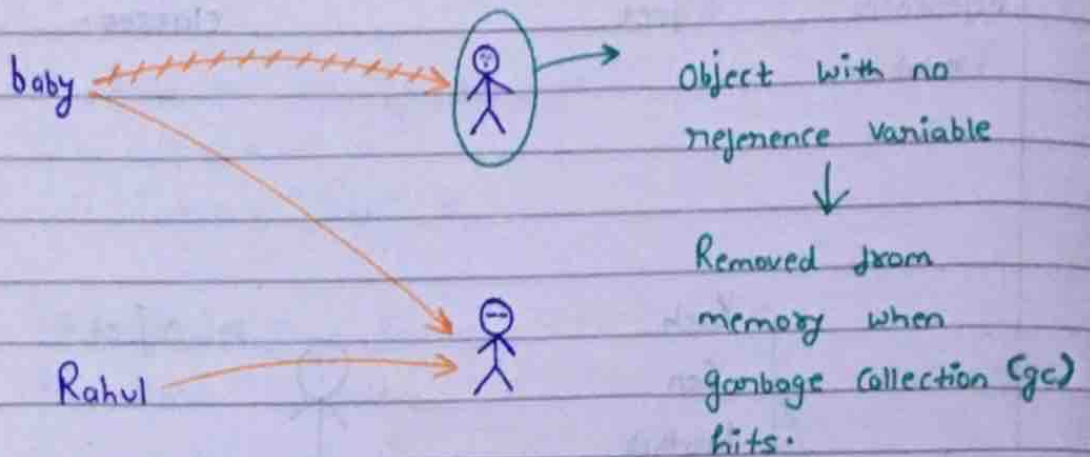Visible to all the Variables .

Ex →   a ─────────────→
                                      [1,3,5,9]
               B ─────────────→
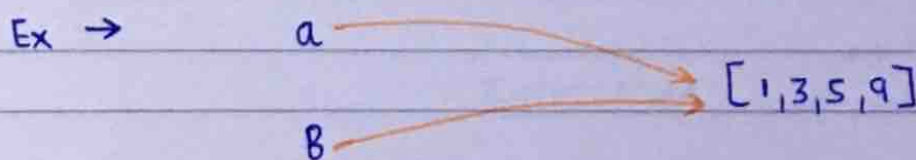
if,
  a [0] = 99
then,
     B = [99,3,5,9]

Garbage Collection

↳ It hits when there is no reference variable.

baby ++++++++++++→ 👤 → Object with no reference variable

↓

Removed from memory when garbage collection (gc) hits.

Rahul ————→ 👤

a = ⑩ ——→ Removed when gc hits.

a = 30