How to master coding

Important things to Master coding

① Learn Basics

② Practice Everyday, Incrementally work on harder problems

③ Debug

④ Run, make planned mistakes, compile & Debug

⑤ Do mini fun projects

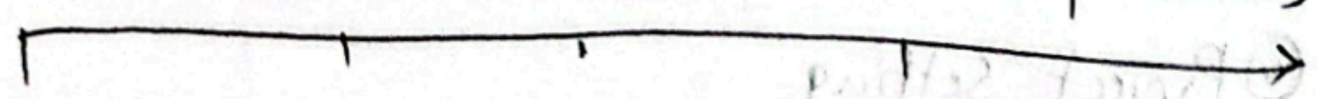
⑥ Learn from others / GitHUB

⑦ Logic Building

→ understanding about computes-fundamentals, computer compiler, softare coding Role

→ Gradually increase the difficulty levels of problem you are choosing to Solve like Very Basic to Advanced let say addition of 2 Numbers to Graphs.

Incrementally work on Harder problems



Practice Problem Solving Everyday in computer

③ Debugging

The Person who makes the debugging He will be a Good Engineer.

can solve Hard problems easily using debugging.

Debugging is an important skill of computer Science & coding.

used to fix bugs

④ Run your code and make some planned mistakes on your code Removing any " or ; or commenting any line or Renaming any variable name etc.

make planned mistakes & let see what happens to the code

Debug & fix them

⑥ Do some mini projects on learned Topics

⑥ Learn from others / Github

as an engineer have to be ready to learn life long in the coding journey

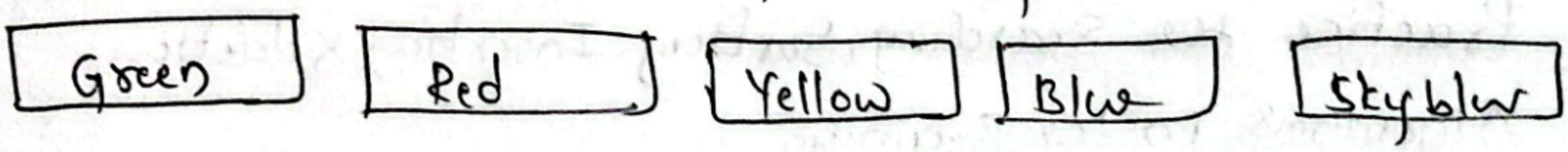
Have to see others code, like How others Built amazing software with the code, learn from their code

⑦ Logic Building

To improve logic building in coding we must have ability to write algorithms in form of code & Application of the algorithms

Divide & Conquerer Method

lets understand with a simple example



Search for color Blue

→ step by step process for search color blue

① Traverse one item at a time

② Compare the color = Blue

③ If true return found → True

Else go to search on next element

④ Repeat till. traversal complete

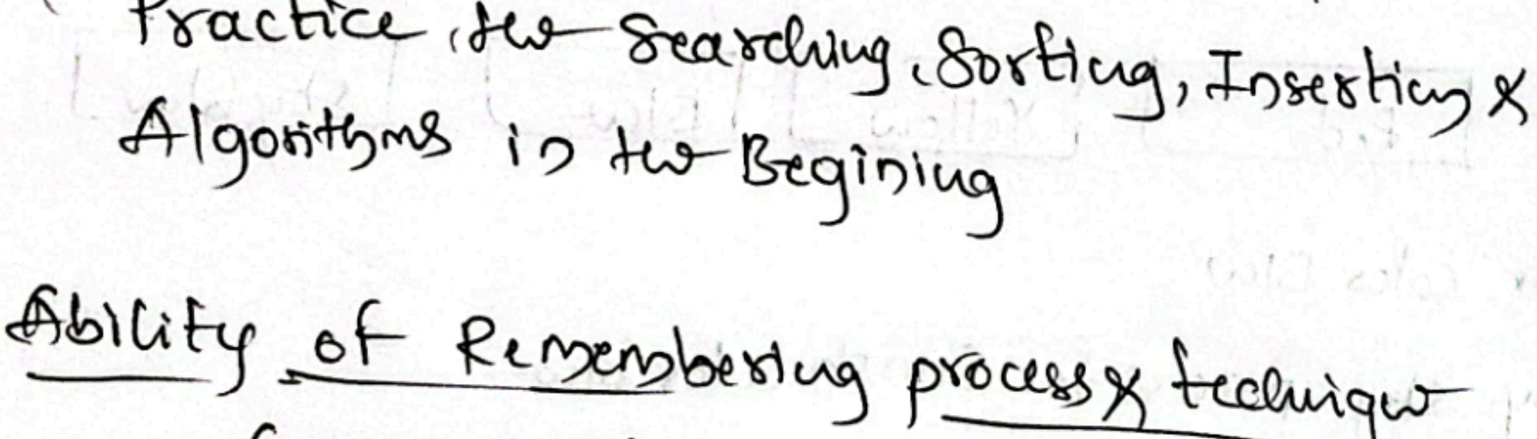
Spudo Code:

① Iterate (looping) over color list/Array

② Check if item is given
   ↳ use of "if statement"

③ Otherwise continue to next element.

Translate turs Spudo Code to Actual code by seeing the algorithms and understanding Requirments.



Logic Building:

finding Highest, lowest element, Comparisons
Practice the Searching, Sorting, Insertion & deletion
Algorithms in the Begining

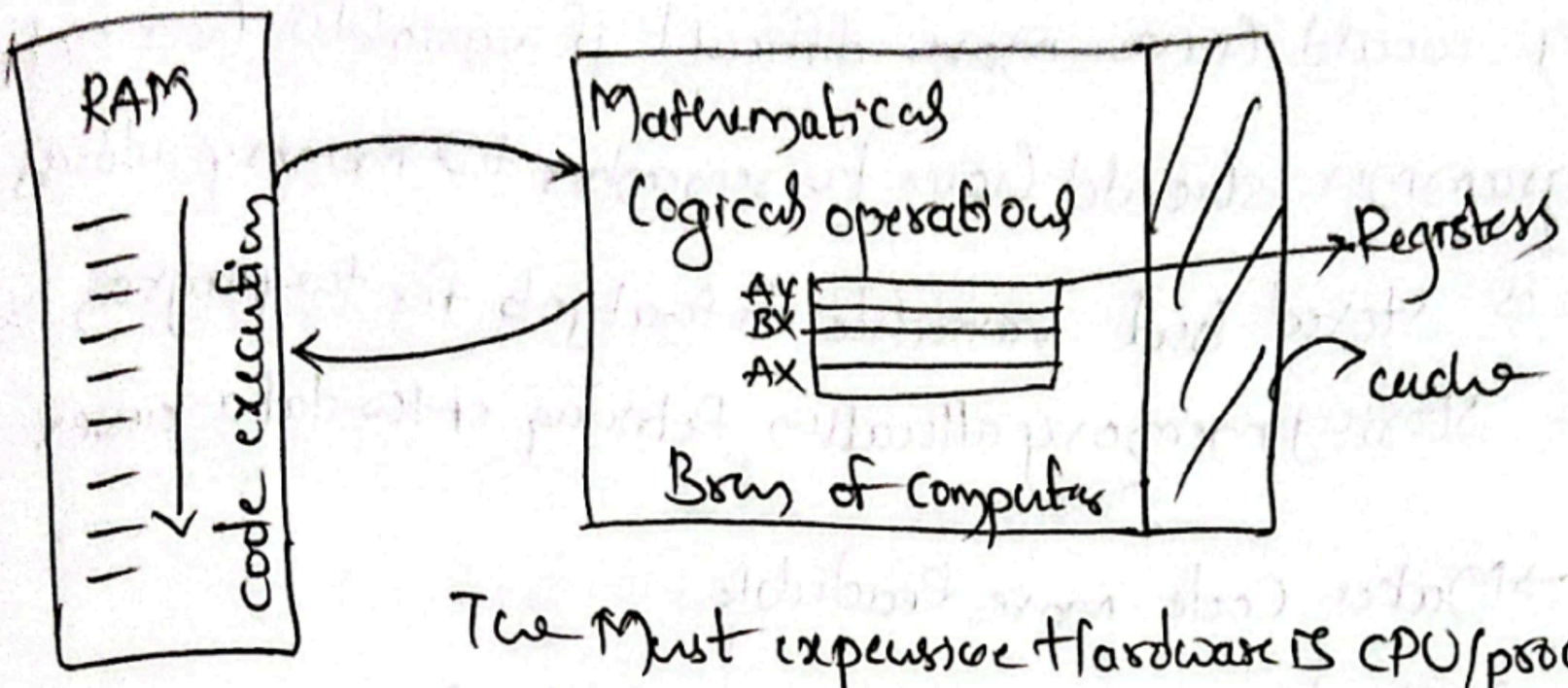
Ability of Remembering process & technique
Learn about the other datastructure

Variable & Data types

One of the fundamentals is "Variables & data types"

Why? Need of Variables & Data types

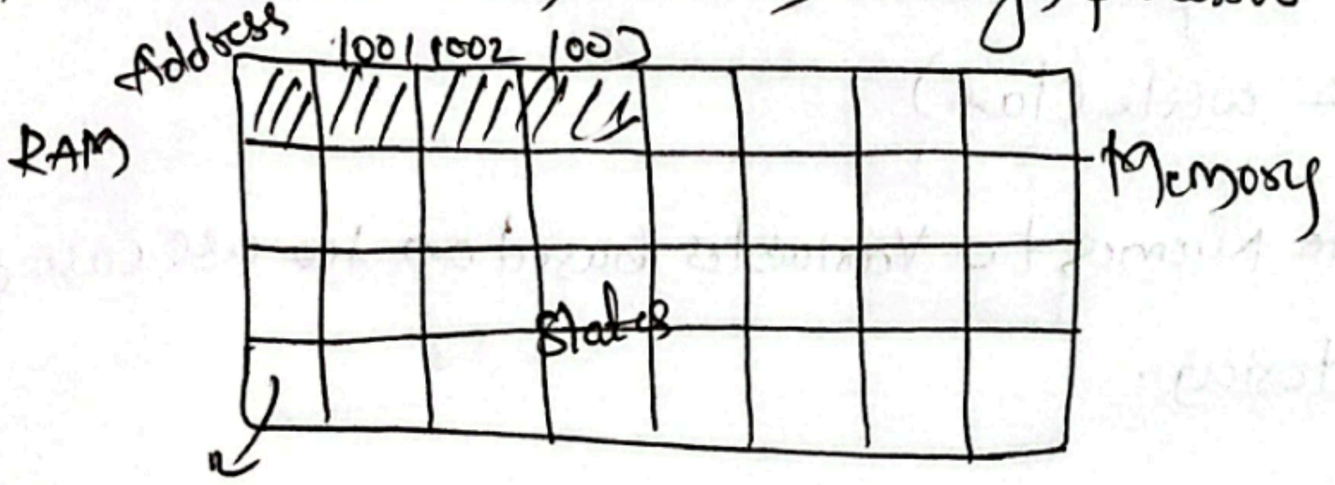
Variables



The Most expensive Hardware is CPU/processor so we needed RAM for cost cutting.

we load software to the RAM

we Access software from the RAM through processor



Byte

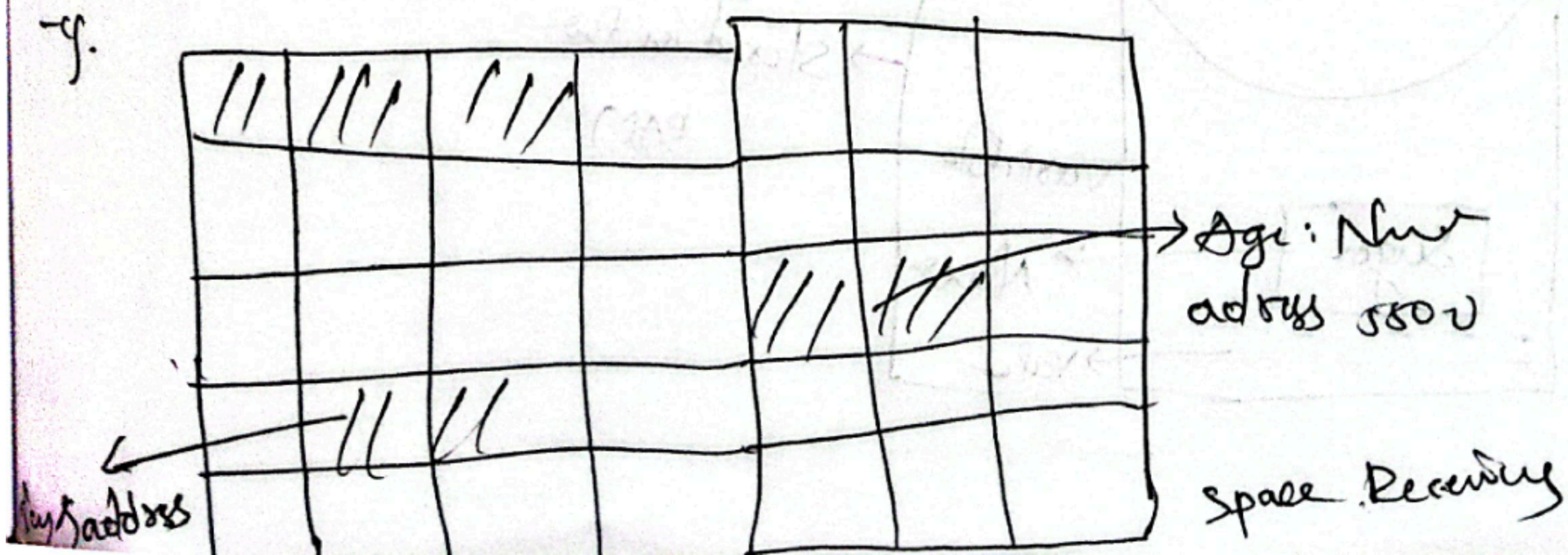
Operating System allocate memory based on the software Requirement.

Target Machine RAM capacity is unknown, so the variable came into picture to develope software without thinking about target Machine

RAM's capacity

programmers dont need to remember the address where the data is stored in the memory, He can Just remember the Name given to a adder

Yes, Variable is Nothing But a Name Given to adders in the memory.



Why Variables exists?

Coding would have more difficult if variables have not used because programmer should have to remember the memory address where data is stored but variable do that job for the program which makes data storing, memory allocation Retriving of the data easier.

① Simplicity → Makes Code more Readable

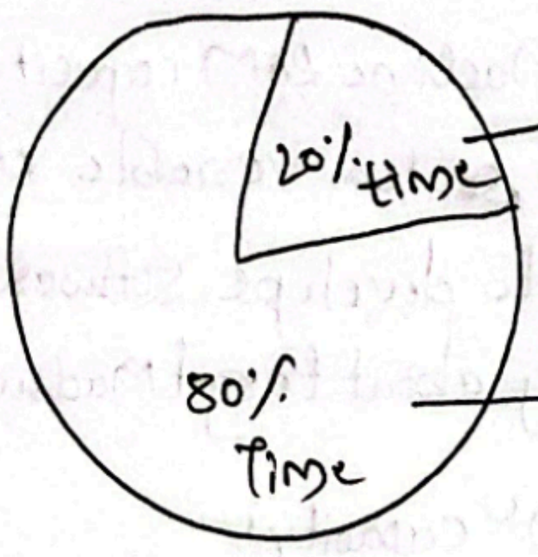
② Flexibility of allocation Dynamic memory allocation

Rules to declare Variable Names

① we cannot use keywords

ex: int for while class)

② Give appropriate Names to Variables based on the use case and the data it is storing.

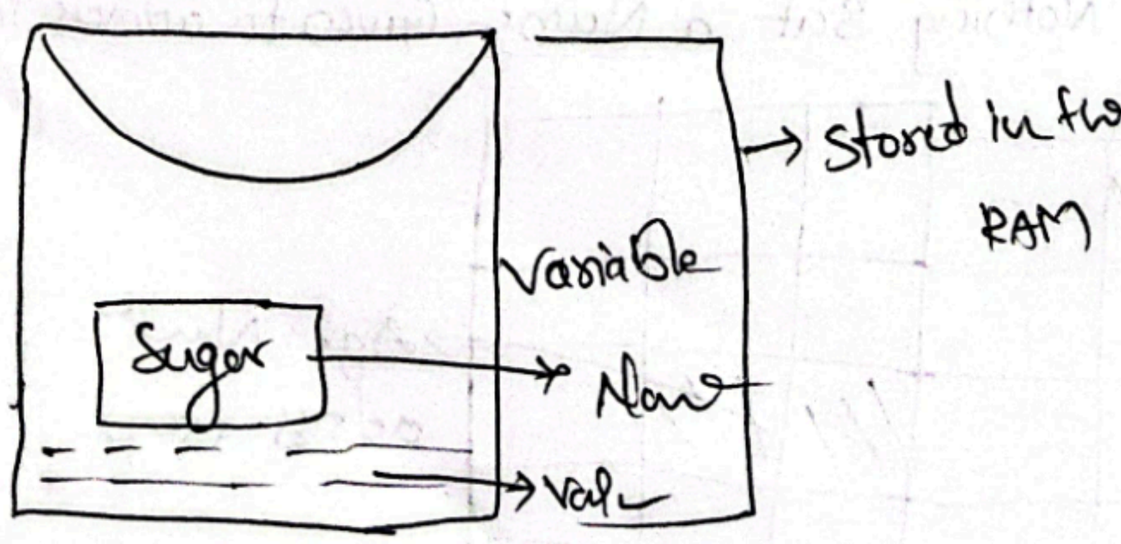


time
→ 20% - writing the code

→ 80% Time for understanding Reading the code written by others

So keep the variable names simple Readable

It is for the Humans. we have to make the code more Readable understandable with the help of variable



→ Stored in the RAM

DATA TYPES

Why need of data types?

Age → 0 to 100

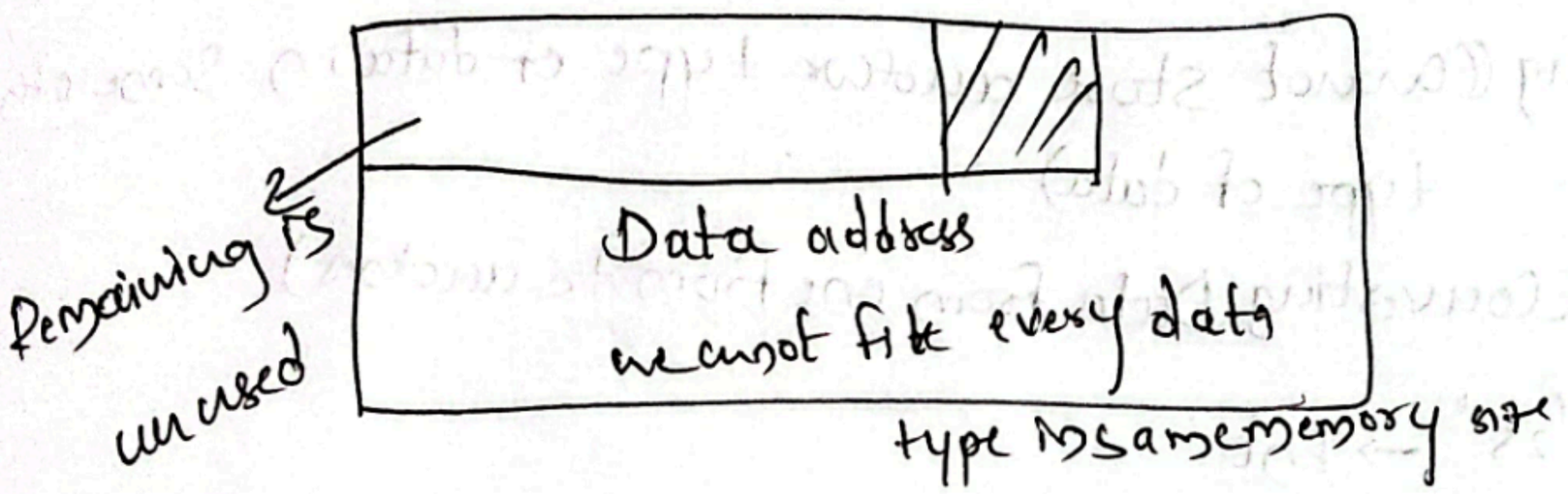
Standard → 1 to 12

Name → A to Z a to z Characters

Height → decimal value

Data types have fixed size

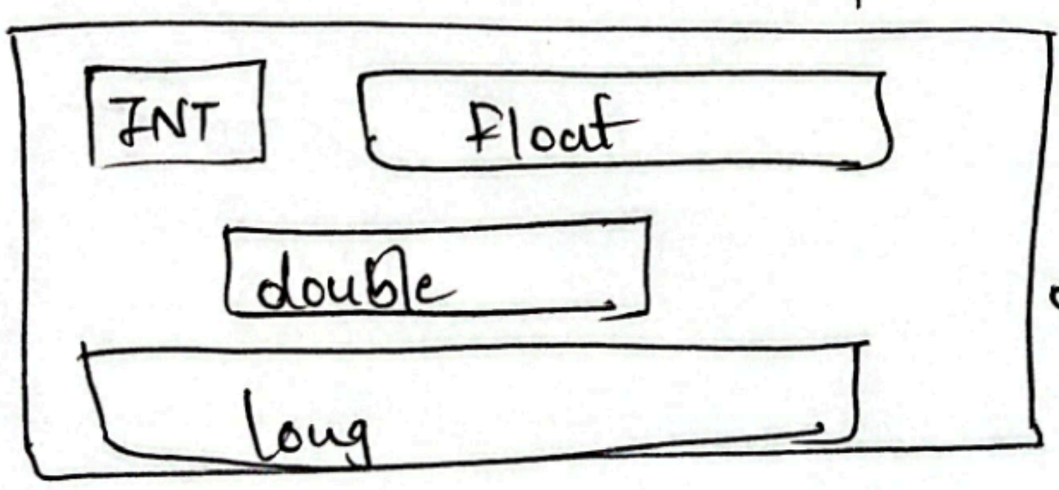
Why Data types have fixed size



Data overflow or Data loss (comparison)

To Avoid this

So Each data type have some particular size



Based on the data these are different Datatypes.

Types of Data

Data + Type

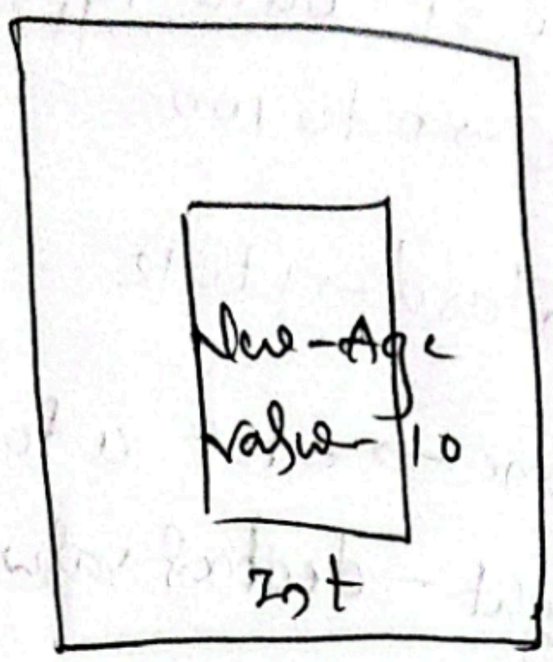
↓ ↓

Information kind of Information

int age = 10;

↓ ↓ ↓

Data type — Name of Variable — Value



Memory

① Data types are Introduced for efficient use of RAM space

② Type Checking (Cannot store another type of data in some other type of data)

③ Type Casting (Converting Data from one from to another)

"25" → Int

String

④ Data fetching Become easier

⑤ Reservation of memory to execute program.

Data Type	Min Value	Max Value	Number of Bits	Programming Languages Supported
bool	0 (false)	1 (true)	1	C, C++, C#, Java, Python
char	-128	127	8	C, C++, Java
unsigned char	0	255	8	C, C++
int8_t	-128	127	8	C, C++
uint8_t	0	255	8	C, C++
int16_t	-32,768	32,767	16	C, C++
uint16_t	0	65,535	16	C, C++
short	-32,768	32,767	16	C, C++, Java
unsigned short	0	65,535	16	C, C++
int	-2,147,483,648	2,147,483,647	32	C, C++, Java, Python
unsigned int	0	4,294,967,295	32	C, C++
int32_t	-2,147,483,648	2,147,483,647	32	C, C++
uint32_t	0	4,294,967,295	32	C, C++
float	~1.4E-45 (smallest positive value)	~3.4E+38 (largest positive value)	32	C, C++, Java, Python
long	-9,223,372,036,854,775,808	9,223,372,036,854,775,807	64	C, C++, Java
unsigned long	0	18,446,744,073,709,551,615	64	C, C++
int64_t	-9,223,372,036,854,775,808	9,223,372,036,854,775,807	64	C, C++
uint64_t	0	18,446,744,073,709,551,615	64	C, C++
double	~4.9E-324 (smallest positive value)	~1.8E+308 (largest positive value)	64	C, C++, Java, Python
long double	~3.4E-4932 (smallest positive value)	~1.1E+4932 (largest positive value)	80, 96, or 128	C, C++