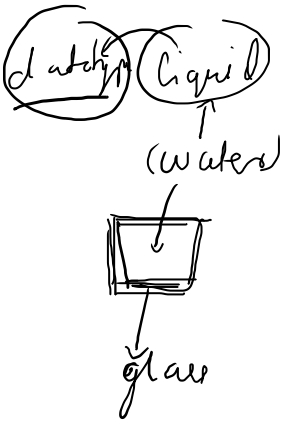
# Day 4




Revision of Day 3.

1. Variables
2. Datatypes

=== Vessels

3. ~~Comment~~ → not executed by Complier
   - ↳ Human BCkgs can read program well
4. Input in function
5. Functions

Operators ( )

precedence

priority

Rules

3 # 2 ** 3

5 ** 3 = 125

$2^3 = 8 + 3 - 11$

+ add
- sub
* multiply
/ div

** (pow)

$2^3$ = 2 ** 3 (2³)

$2^3 = 2 * 2 * 2$

$3^3 = 3 * 3 * 3$

Mathematical operations

2 ** (5+2)

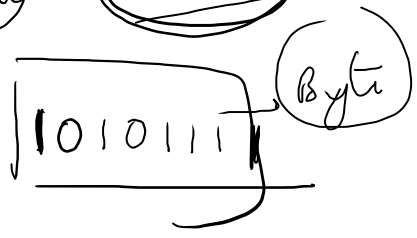
2 ** 7 = 128 $2^5 + 2x$

## logical operators

&& and | ! not | || or

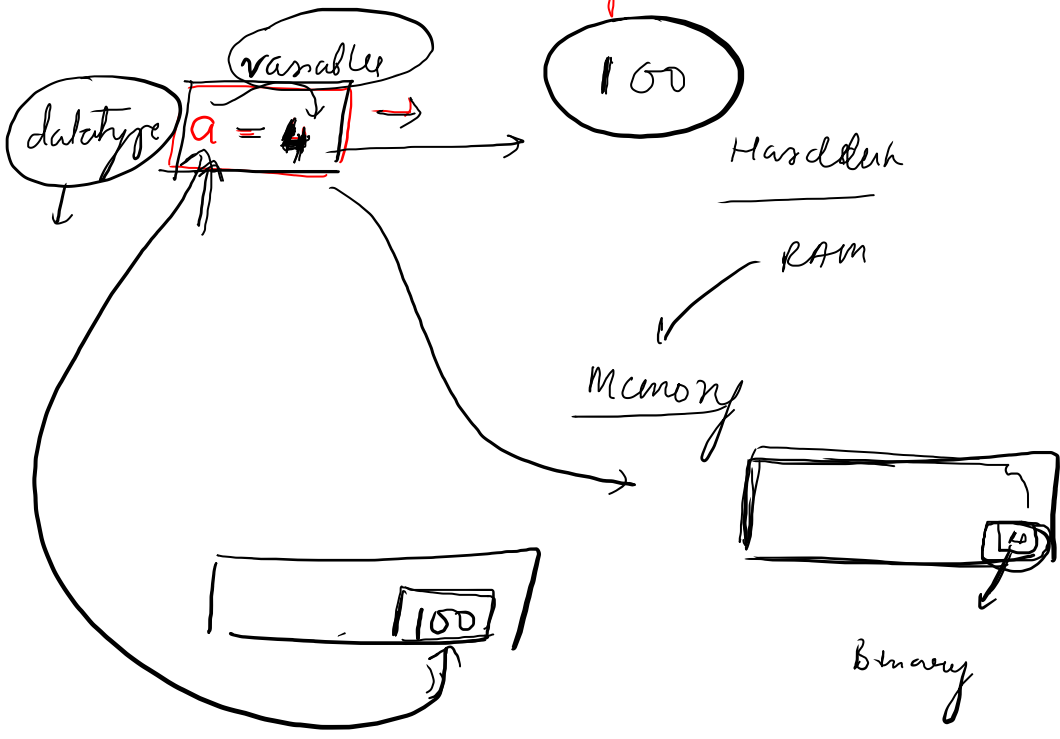
Bit → 1/0

Byte → 8 bits

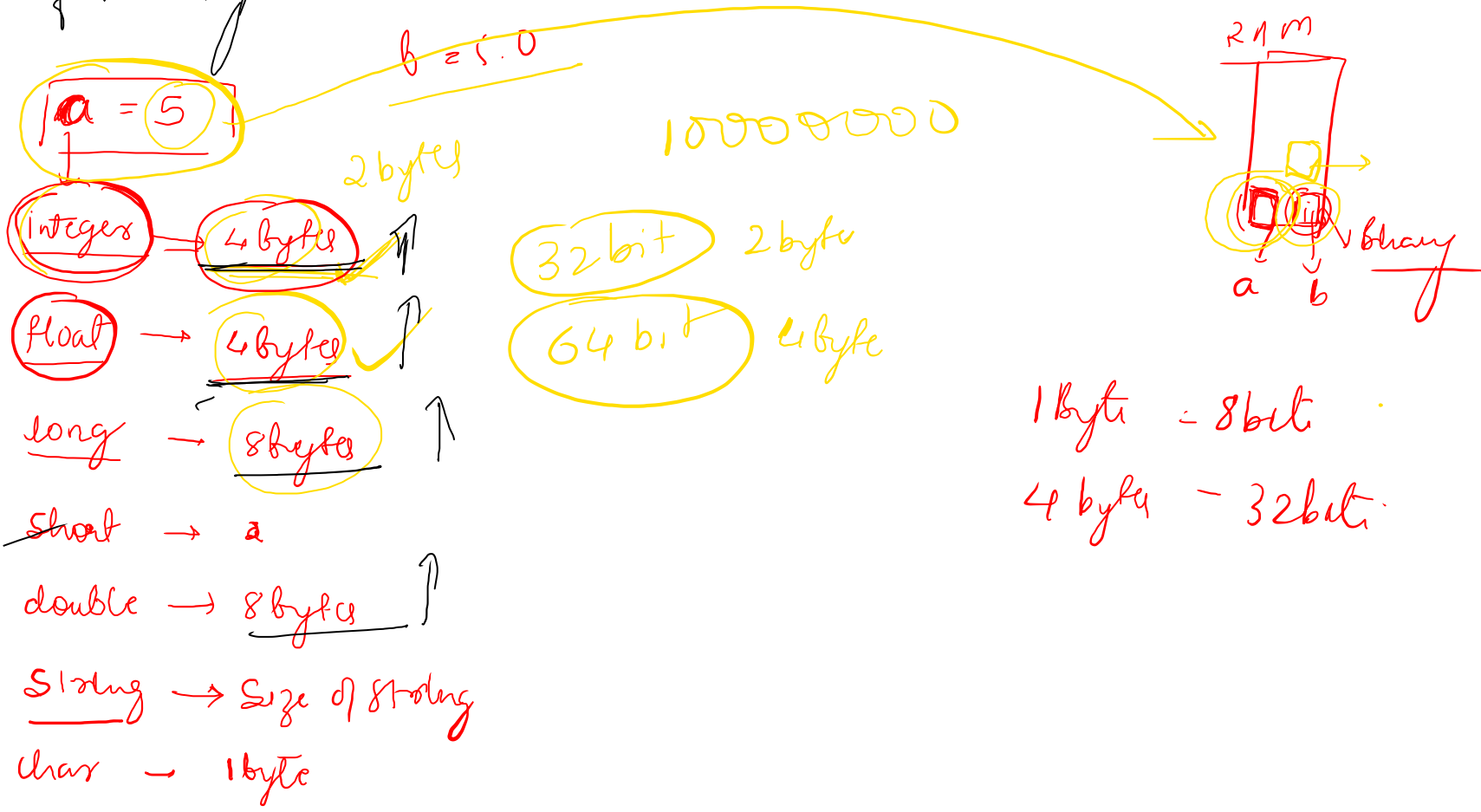



## Bitwise operators

↳ Bit manipulation



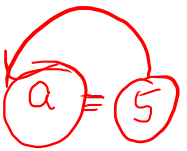
Every datatype will be allocated with different sizes of memory



## Relational Operations


`==` (Equality) LHS == RHS

`=` (Assignment Operator)




`>`

`<`

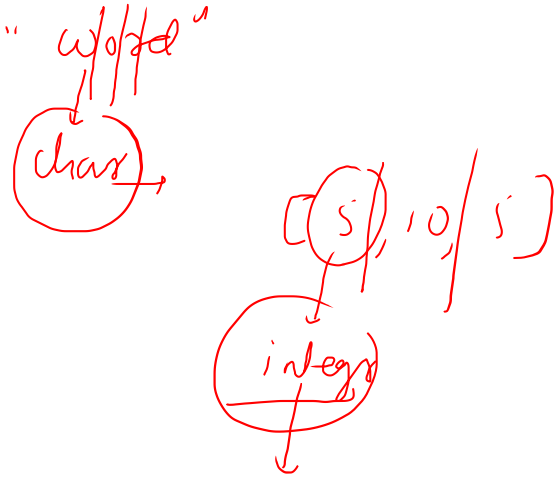
`>=`

`<=`

Datatypes

- Primitive
  - int, float, char, ...
- Non-primitive datatypes
  - List, Array, String, Stack, Queue

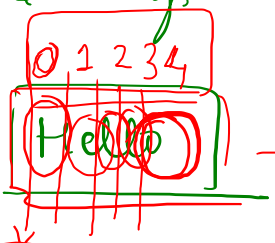
5.0 ✗




## How to write any program.

1. Think of a solution / logic of how to solve it without using program (Pseudo Code) → English lang
2. Ask yourself what concepts that i know in the programming language can be used to solve this
3. Write some code
4. Debug.

Strings → Non - primitive




→ accessed with indexes

→ starts from 0 → n - 1 → 4

(n - 1 : length of string)

char

→ len of string → 5

len() → find len of any string, list, array.

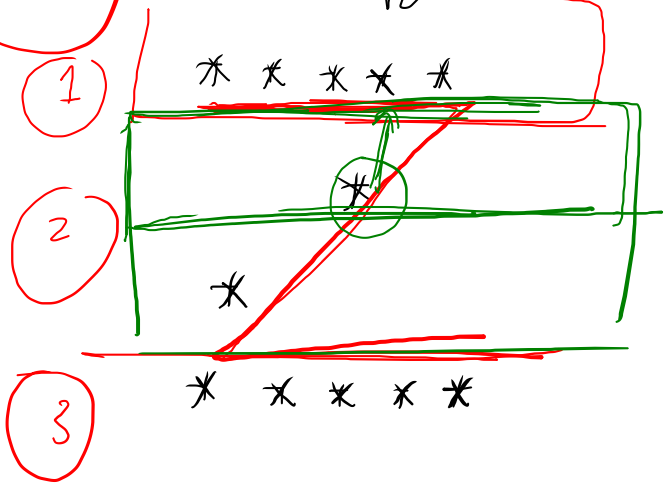
( Accessing of letters of string

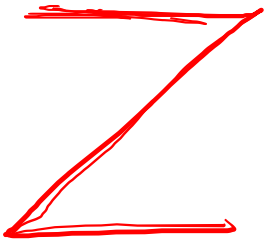
strname[index]

eg) name = "Sushil"

name[ 5 ]

# print Z in python



print()

\n → newline

\t → tab → tab space

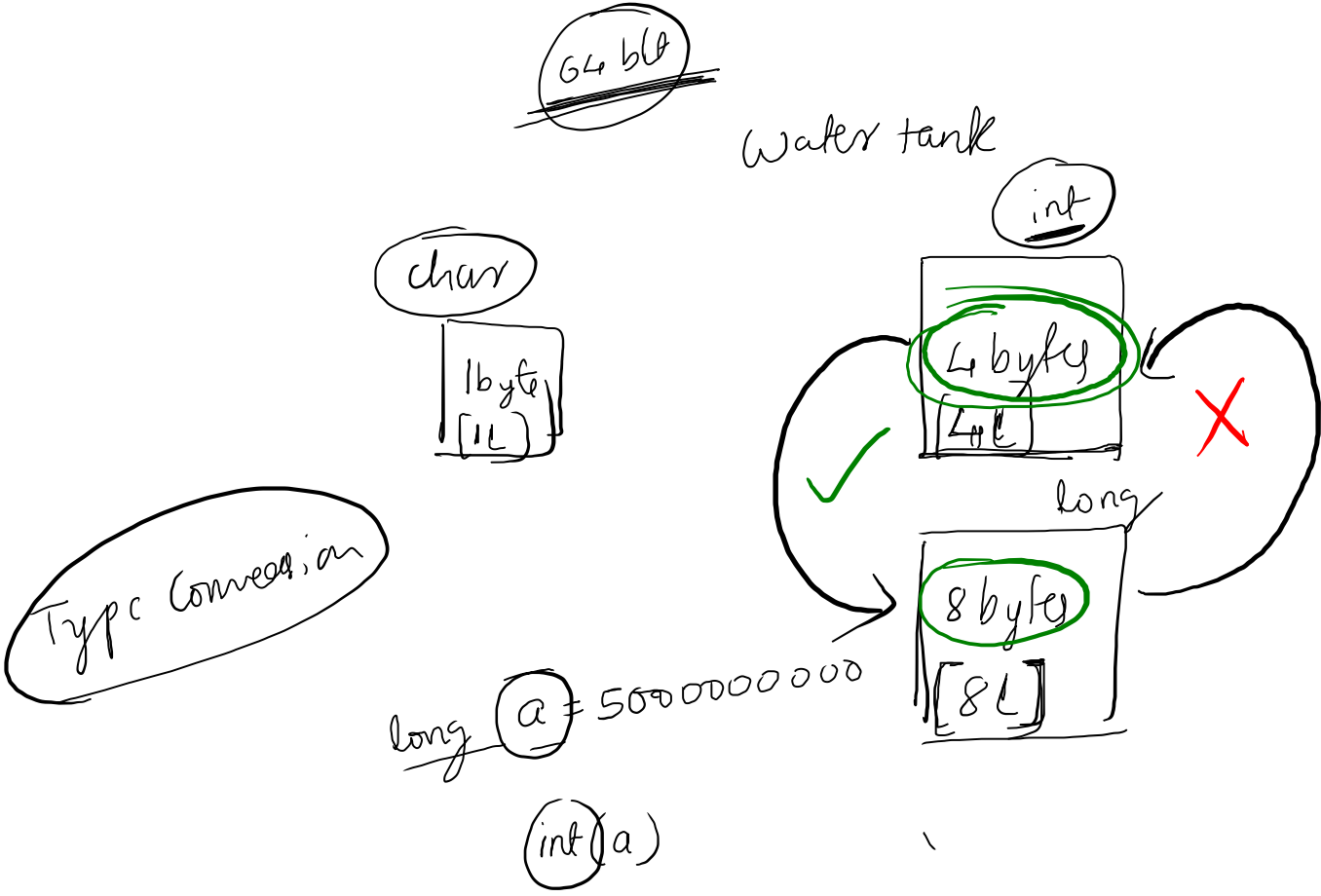
## String Contactination

adding 2 or more String to form 1 String.

**Type Conversion** → Converting 1 datatype into another datatype

- **Implicit** → compiler (auto)
- **Explicit** → manually programmer can do it.

auto Type Conversion ? lower datatype → higher datatype


64 bit
Water tank
int
char
1byte
[1L]
4 byte
[4L]
long
8 byte
[8L]
Type Conversion
long a = 5000000000
(int) a