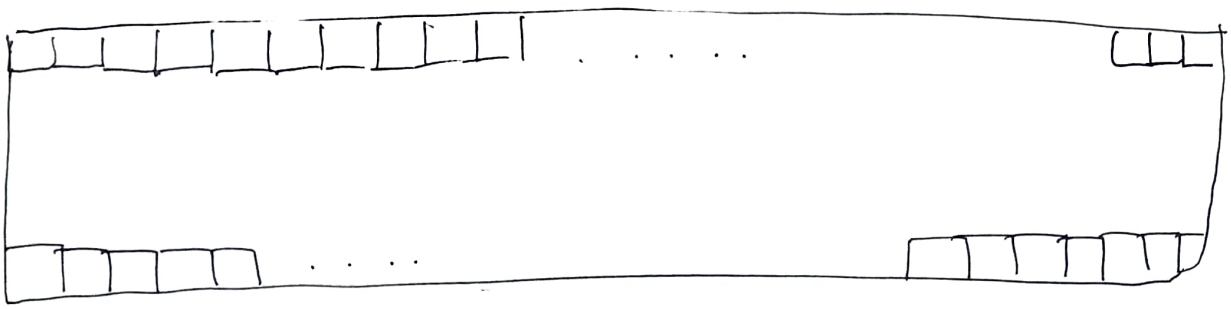
Notes → Class #1.

* So last time there was this thing called the MEMORY. And you said we store programs there. Is that all?

Not quite. Actually... memory is like a large city with thousands of (actually millions) tiny houses. Like this



And every house has an ADDRESS. For a computer memory this address is like a house number (like 1001, 10003, etc.)

Now, each of these houses can store just ONE number. Any number but just one of them.
But what if we want to store bigger things →
We combine the houses together and give each house a Name.

Like so:

→ These are the addresses of the combined houses.

house No 1 → This is the name of the house.

Now, we can store whatever we want in the memory houses!

* So, the names of the houses are called VARIABLES

And whatever we store in those houses are called VALUES.

* Variables and Values! Huh! What kind of values are we talking about?

↳ Numbers or numeric values eg. 1, 1001, 3429

↳ Strings (which is a collection of characters* and numbers)

characters can be 'a' to 'z' or 'A' to 'Z' or any such thing.

eg. "hello", "this is a string", "129".

* So you can store any value in any variable?

Maybe. But you should not. You can't keep an elephant in your house. I mean you can... but you shouldn't. Right? Elephants belong in the jungle.

The way to think about this is as below →

When you say

X = Y

(left side) (right side)

This should be a variable name

This should be a value (or another variable name)

And once you say this, the house size gets fixed.

So suppose you say

tomato = 127

This becomes a NUMERIC Variable because This is a NUMERIC Value.

If, instead, you say

brinjal = "i am not superman"

This becomes a STRING variable because This is a STRING value

---

* But what is this '=' sign? What does it mean?

It is called an ASSIGNMENT. (uh-syne-ment) It means that you are telling the computer to store whatever is on the right-hand side of "=" to the house whose name is on the left-hand side.

So :-

potato = 13

means

"Store 13 in the house with the name 'potato'"

* Enough! Now show me a program:-

Here it goes →

```
X = 127
Y = 13
Z = X + 1
W = Y * Z
PRINT (W)
```

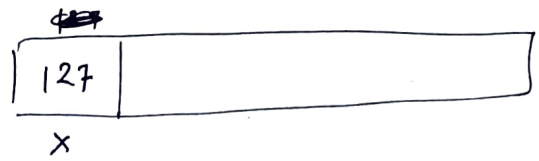
This is the program

And this is what it does.

So when this RUNs on a Computer, this is what happens.

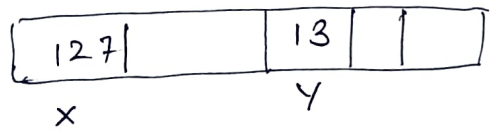
Step 1. X = 127

The processor puts 127 in the house with the name X.



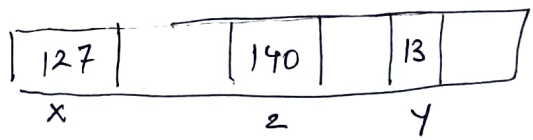
Step 2. Y = 13

Puts the value 13 in the house with the name Y



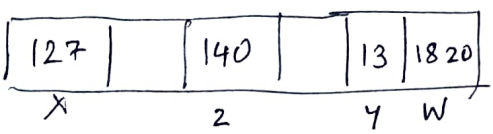
Step 3. Z = X + 1

- Computer processor GETS the value in house X.
- Adds 1 to it
- And puts the answer in the house named Z.



Step 4. W = Y * Z

- Gets the value in house Y
- Gets the value in house Z
- Multiplies them together
- Stores the answer in house W



Step 5. PRINT(W)

This reads the value in house with the name W and prints it on the screen. i.e. shows it to us.

* So can I do this +, -, *, / with variables? I only did this with 2 numbers right?

- +, -, * (multiply), / (divide) are called operators.
- operators can be used with values as well as variables.
- e.g. 2 + 3 is fine.

4 - 1 is also fine

X = 10
Y = X + 5 is also fine

X = 20
Y = 30
Z = X * Y is also fine.

But if
X = "hello"
Y = 10
Z = X + Y is Not fine.

* So you mean there are some rules?

- Actually, you can keep these rules in your mind →
  - +, -, *, / can be used with numbers and numeric variables freely.
  - + can be used with strings also.
    e.g. "hello" + "world" will mean "hello world"
  - * can be used with a string on the left and a number (whole number only) on the right.
    e.g. "hello" * 3 will mean "hellohellohello"

Summary.

- Variables are names of places in the memory.
  - ↳ They are used because
    a. It's hard to remember addresses all the time.
    b. Sometimes we combine smaller houses to create more space and we can give it a name.
       = ↳ the larger house
- Values are numeric (numbers) or strings.
- Variables store values. Values are stored in variables.
- Assignment (=) does the job of storing values into variables.
- Programs move values between different variables doing interesting things (like +, -, *, /) in between.
- Operators like (+, -, *, /) can be used on both values and variables.