Variables in Java

- Variables are just a container for data . It holds a certain value . Variables can be declared as some certain types we will learn the types in the upcoming videos in great details.

- Give an analogy of drawers and keeping data there to use them in time of need by the program.

- They are simple a label put on the drawer, that is memory location, so that we can use the value on that memory location easily. Now it will be very hard to denote the memory by their actual name as they will be is certain binary digits with 0 and 1 inside the computer so we use variables.

- General structure of declaring any variable is:

  - **Type name = initial Value**

- Not going in too much detail about the types of variables, will be discussed as necessary.

- println() used to display variables and we can use **+** to combine texts and also variables together to display them on the screen, as we have already seen some before. But for numeric case, this sign acts as an operator, that is it will add the numbers together.

- Examples:

  ```java
  int x = 10;
  int y = 20;
  double a = 5.5;
  String name = "Ajwad Akil";
  System.out.println(x);
  System.out.println(y);
  System.out.println(a);
  System.out.println("My name is " + name);
  ```

  ```java
  int a = 10;
  int b = 20;
  int c = a + b;
  System.out.println(c);
  System.out.println(x + y);
  ```

- We also can declare a variable of the same type in this way:

  ```java
  int a, b, c;
  int x = 10, y = 100;
  System.out.println(x + y);
  ```

- An important note: **Naming Conventions of variables**

  - There are different Naming conventions of variables and it differs according to the programming languages. In java we use a system called **Camel Case** , shown by an example:

    ```java
    int thisIsReallyLong = 10;
    String thisIsEvenLongerThanBefore = "yes it is";
    int thisVariableIsRidiculouslyLong = 100;
    ```

- Also, when we declare a variable, try to declare the similar type of variables together, makes the code a lot cleaner. And also maintain the gap between the operators, this is extremely important for the codes to look beautiful. Not imposing this but if practices from the very beginning , then habit will be formed and the codes will look beautiful to everyone.
- Comments in java

```java
//This is comment in java
/** this is a multiline comment in java**/
int x = 10; //number of students in the class
int y = 5;  // number of classes
```

  - Documentation comments can be activated by this in intellij, this will be again reviewed during methods section of the tutorial.
  - Why comments, discuss in details the necessity and why it is a great idea and an absolute necessity to write comments for ourselves and also for others.