Strings and String Builder

String.

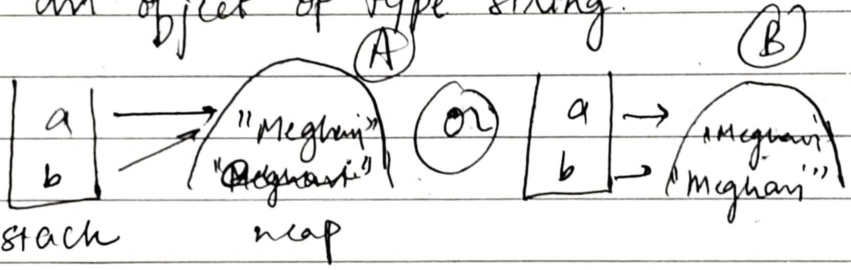
As the name suggests it is a collection or sequence of characters-

"Meghavi" → This is a string and it is of a String Datatype

String name = "Meghavi Jadav"
↳ class of java Datatype ↘ refrence variable ↳ object

Every string you create is an object of type String.

String a = "Meghavi"
String b = "Meghavi"

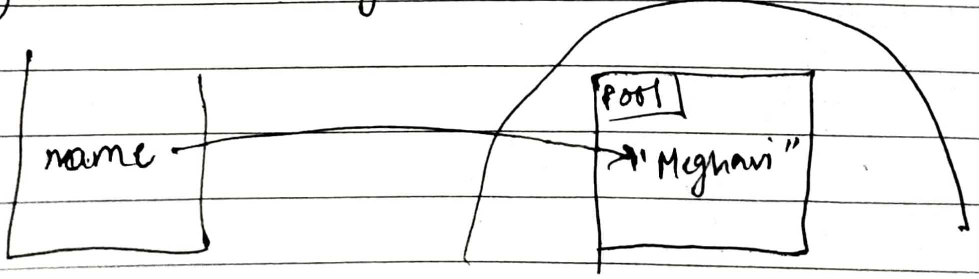


To understand this we need to learn two concepts
1. String Pool
2. Immutability

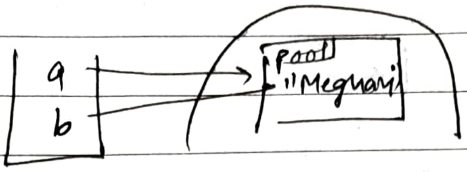
String Pool

It is a seperate memory structure inside the heap.

String name = "Meghavi"



What is the use case? Why separate pool?
→ All the similar values are not recreated in the pool

So its more like 

Benefits - Makes the program more optimised

Now, usually if change in "Meghavi" is made using ref. variable 'a' then it will reflect in 'b' as well.

↳ But this won't happen because of Immutability.
Strings are immutable in Java
↳ Why so? Because of security

```
String a = "Meghavi";
String b = "Meghavi";

sout (a);        ← you have not changed but created a new object
a = "Jadav"
sout (a);
sout (b);
```

Output -
~~"Meghavi"~~ Meghavi
Jadav
Meghavi

Even when you change 'a', 'b' shall remain the same

Why for security reasons you can't change?

If 4 people are named 'Meghavi' and ~~son~~ person 1 tries to change their name to 'Vedanti' then all the names of the rest 3 people shall change to "Vedanti"
WHICH WE DO NOT WANT

Comparison of Strings.

1. == method $\rightarrow$ comparator

false

a $\rightarrow$ "Meghani"

b $\rightarrow$ "Meghani"

true

$$\begin{tikzcd}
a \arrow[r] & \text{"Meghani"} \\
b \arrow[ru] &
\end{tikzcd}$$

How to creat a new object with ~~diff~~ the same name

String a = new String("Meghani");
String b = new String("Meghani");

↳ created outside the pool in the heap.

2. • equals method

This will check if the values are same or not? irrespective of whether there are two strings with the same object