Day-15

Strings and StringBuilders

'a', 'b', '\n', '#' → Examples of characters

"Hi", "Hello", "How are you?" → Examples of strings

Even "a" is a string

Declaration & initialization

String str; String str1 = "I like rain!";

↓ String str2 = new String("but I get old");

In Java S is capital

To print ⇒ System.out.println(str1);

Taking i/p ⇒ String str = sc.next();

(or)

String str = sc.nextLine();

Eg: abc
_ def ghi

- String addition/concatenation

String a = "Hello!"
String b = "My name is khan";
String intro = a + " " + b;

- Some in-built functions

→ str.length()

→ Str.charAt()

→ Str.substring(starting Index, ending Index)

Eg. "Hello folks!"
0 1 2 3 4 5

str.substring(1,5)
↳ ello

→ str.split()

Eg: str="abc,def,ghi,jkl mnop";

Str.split(',');

↳ ["abc", "def", "ghi", "jklmnop"]

→ str.trim()

Eg: import java.util.Arrays;

```
class Solution {
    public static void main(String[] args) {
    String str = "Hello, I'm coming, Bye!";
     S.o.pln (str.length());
     S.o.pln (str.charAt(3));
     //substring
     S.o.pln (str.substring (7,17));
      S.o.pln (Str.substring(7));

      //split
      String arr[] = str.split (",");
       S.o.pln (Arrays.toString (arr));

       //contains
       S.o.pln (str.contains ("comming! "));
                S.o.pln (str.contains ("coming"));
       //trim
       String str2 = str.trim();
        S.o.pln (str);
        S.o.pln (str2);
```

O/P

```
29
l
o, I'm com
o, I'm coming, Bye!
[Hello, I'm coming,
  Bye!]
false
true
   Hello, I'm
 coming, Bye!
Hello, I'm coming,
        Bye!
```

String comparison

"abcd"

a) str1.equals(str2)

b) str1.equalsIgnoreCase(str2)

c) str1.compareTo(str2)

Int to string & string to Int

1) Integer.parseInt(str)

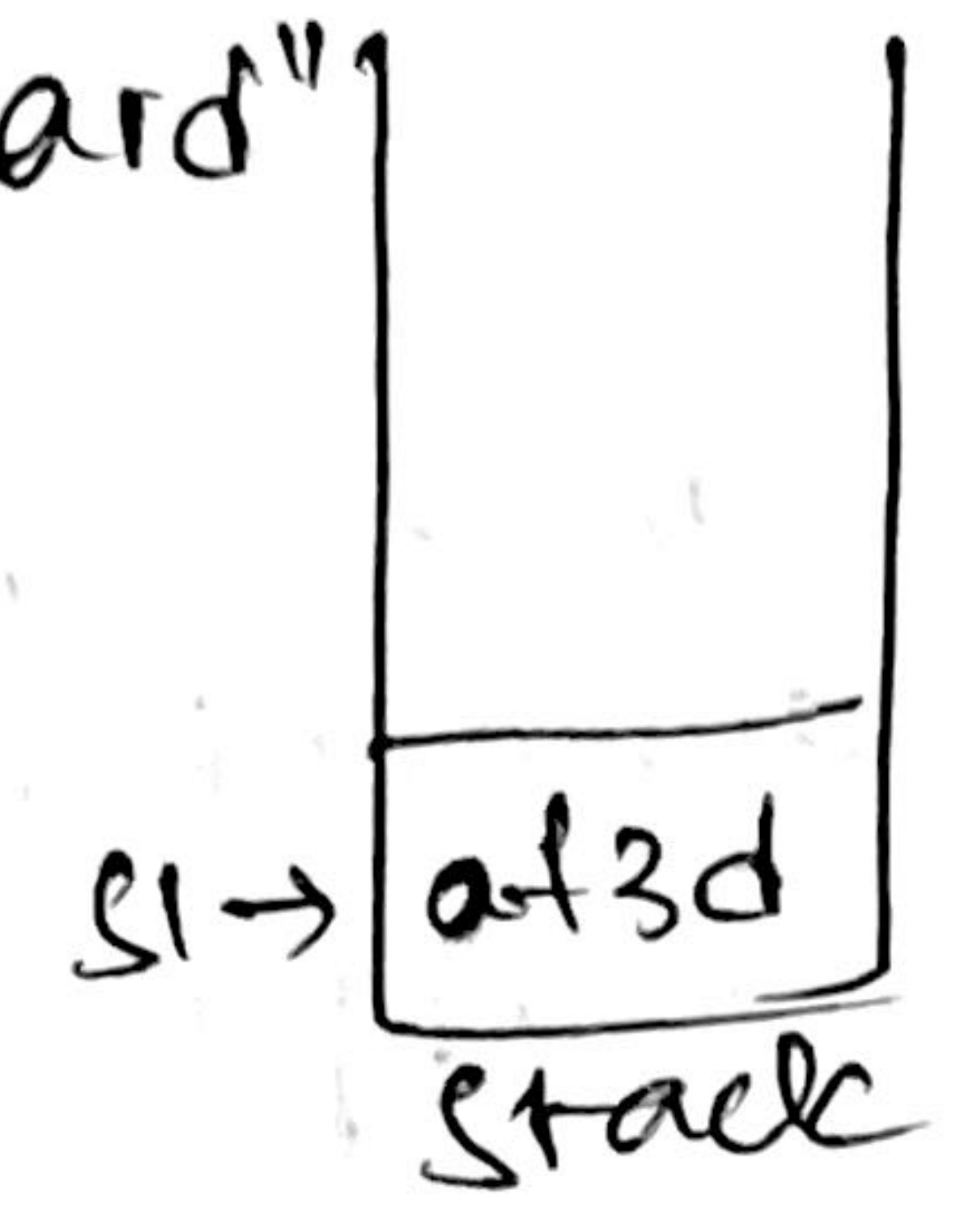
↳ convert string to int

2) Integer.toString()

↳ converts Int to string

About memory allocation
(Interning)

String s1 = "Learn Yard"



Strings are immutable

StringBuilder

String Builder str = new StringBuilder("abcdef");

Eg: SetCharAt

str.setCharAt(2, 'z');

Q) Check if a given string is a palindrome or not
↓
reads same when read forward & backward

Eg: racecar

```
import java.util.Scanner;
class ClassNd {
   public static void main(String[] args) {
     Scanner sc = new Scanner(System.in);
     String str = sc.next();
     int i=0, j=str.length()-1;
     String ans = "yes";
       while (i<=j){
       if(str.charAt(i) != str.charAt(j)){
          ans = "No";
          break;
       }
          i++;
          j--;
       } S.o.pln(ans);
     }
```

→ The main difference b/w 'String' and 'StringBuilder' in Java in their mutuability

1) Immutability Vs mutuability

- String ⇒

- Strings in Java are immutable, meaning once a 'String' object is created, its content cannot be changed.
- Any operation that appears to modify a string actually creates a new string.

- StringBuilder ⇒

- StringBuilder is mutable, allowing you to modify the content of the string without creating a new object
- It is more efficient for situations where you need to perform a lot of string manipulations, such as concatenation or insertion.

Eg:

```
  public class StringBuilderOperations {
        public static void main (String[] args) {
 StringBuilder stringBuilder = new StringBuilder("Hello");
 stringBuilder.append("World");
stringBuilder.insert(5, "Java");
stringBuilder.delete(5, 10);
stringBuilder.replace(0, 5, "Hi");
  int length = stringBuilder.length();
  int capacity = stringBuilder.capacity();
  S.o.pln("Result:" + stringBuilder);  → Hi World
  S.o.pln("Length:" + length);  → 8
  S.o.pln("Capacity:" + capacity);  → 21
 }
}
```