Data types

- Pure
  - Char
- Non Pure
  - String → exceptions
  - Stack, queues

Char

→ Syntax → char ch = 'a'

ch1 = '&'

ch2 = ','

single quote

Memory of char (Java)

2 bytes

1 byte → C / C++

ASCII

A – Z → A → 65, B → 66, C → 67, D → 68 ..... Z → 90

a – z → a → 97, b → 98, c → 99 ..... z → 122

char ch = 'a' → 97

char ch = 'b' → 98 → Binary code

ch = b

```java
int i = 65;

System.out.println("This is char: " + (char)i);
```

Typecast

i = 65

Datatype 1 $\xrightarrow{\text{cast}}$ Datatype 2

Typecast →

int → char

(char) i

i = 65 (Binary code)

$\boxed{65}$ → char → A

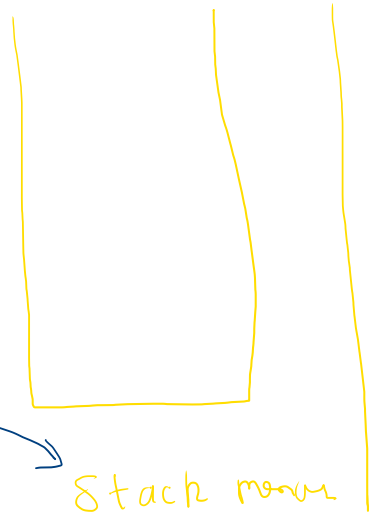
```java
public static void main(String[] args) {

    // Syntax
    char ch = 'a';
    char ch1 = '1';
    char ch2 = '$';

    int i = 65;

    System.out.println("This is char: " + (char)i);

    System.out.println("This is char to int " + (int)ch);

}
```

'a' → 97

ch = 'a'

Memory

↓ Lot of Space



less space

heap memory

Non Prim

String, Array, Stack, Queue

LL,

Object of any class

Primitive Datatype

(int, long, float, char, boolean)

Stack memory

Heap memory

Variables / References Variable of Non Primitive Datatype

new

new keyword → non Primitive Datatype variable name / Ref variable

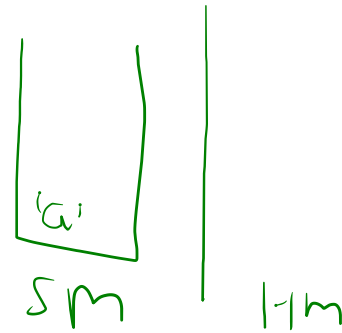
```
// Ques 1 ⟶ take an input from user and
Scanner scn = new Scanner(System.in);
```

→ class in Java

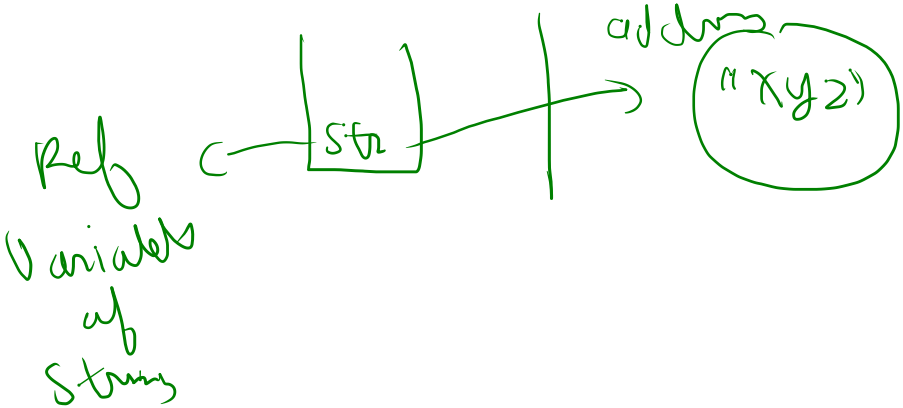
non Prim var datatype name → Ref variable

int a = 5

char ch = 'a'



String str = "xyz"



Strings

→ Non Primitive

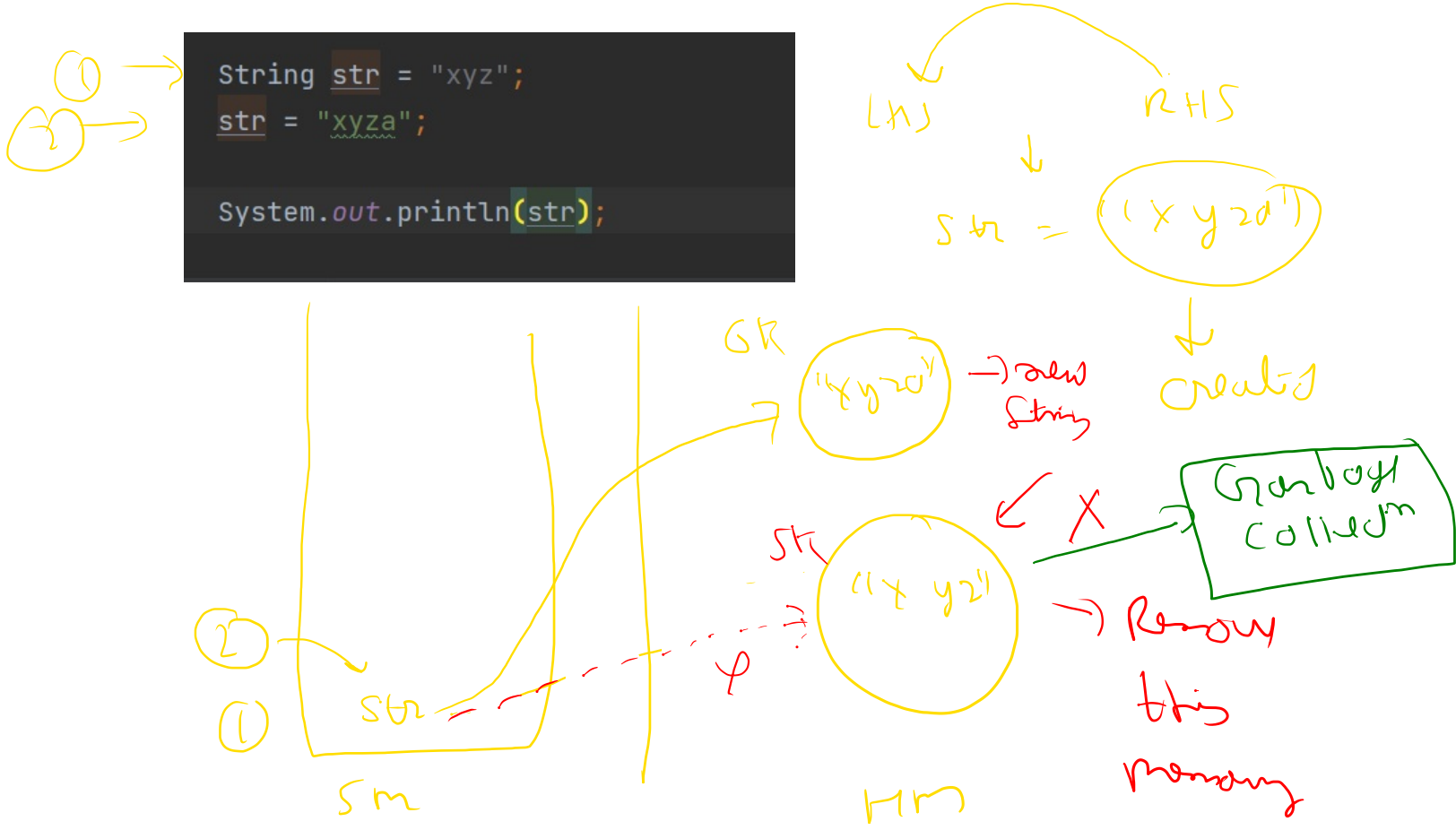
→ Immutable in nature (concat update a string)

String str1 = "XYZ";

str1 = "xyza";

```
String str = "xyz";
str = "xyza";

System.out.println(str);
```

```java
String str = "xyz";
str = "xyza";

System.out.println(str);
```



String str = "xyzabcd"

↳ Collection of char

String idx ↑ 0 out of Bound

index	0	1	2	3	4	5	6	7
str = "	x	y	z	a	b	c	d	"

str → char

```
String str = "xyz";
str = "xyzabcd";
System.out.println(str);

char ch = str.charAt(3);
System.out.println("Char from str is : " + ch);
```

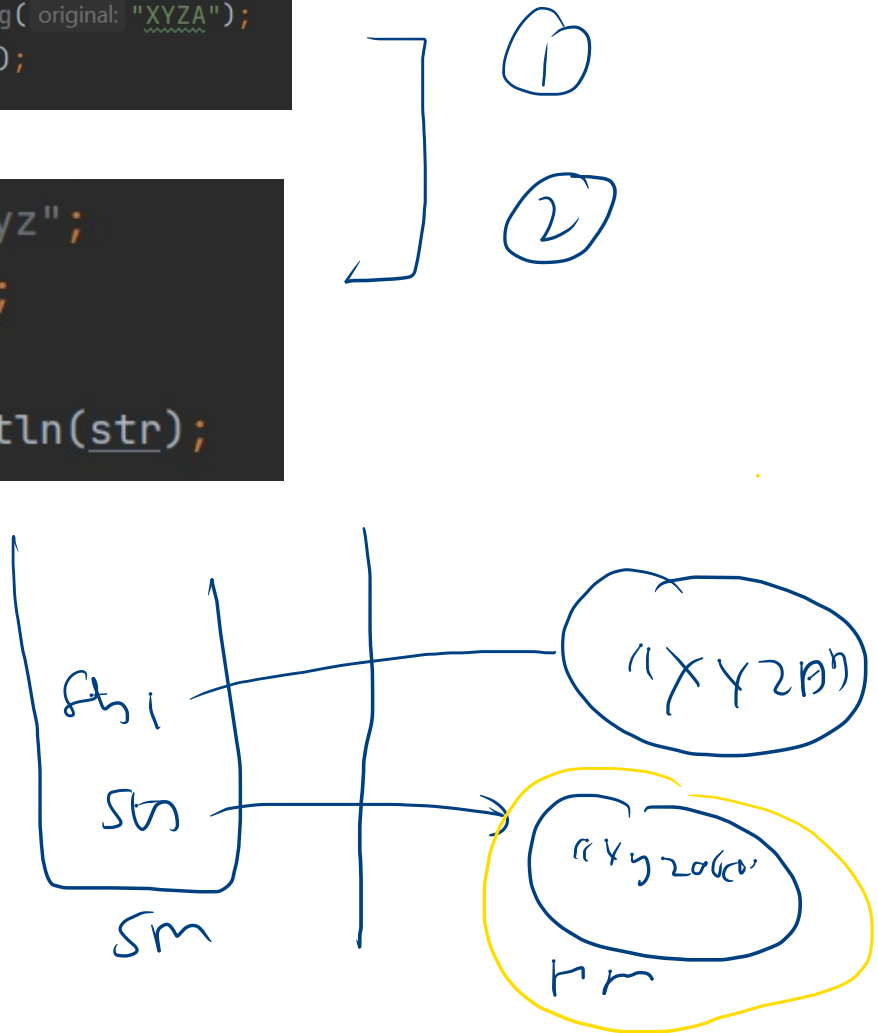
(0 1 2 3)

(1) str.charAt(idx) → str.charAt(3) → a

(2) str.length() → len of String → 7

```java
// Other type
String str1 = new String( original: "XYZA");
System.out.println(str1);
```

```java
String str = "xyz";
str = "xyzabcd";

System.out.println(str);
```



Loops

① for ② while ③ do-while ④ for each

T / F

① for loops

Syntax → for (initial value ; breaking cond^n ; increment / Decrement cond^n)

```
for (int i = 1; i < 10; i++) {
    // body
}
```

i++

i = i + 1

i = 1
2
3
4
5
6
7
8
9

```java
for (int i = 1; i < 10; i++) {
    // body
    System.out.println(i);
}
```

```java
// Syntax
for (int i = 1; i < 10;) {
    System.out.println(i);
    i = i + 2; // i += 2;c
}
```

i = 1 ; i < 10 → 1 → i++

i = 2 ; i < 10 → 2 → i++

i = 3 ; i < 10 → 3 → i++

⋮

i = 9 ; i < 10 → 9 → i++

i = 10 ; i < 10 ✗ → break

i++ → i = i + 1

i += 2 → i = i + 2

i -- → i = i - 1

i -= 2 → i = i - 2

i *= 2 → i = i × 2

i++ → Post increment

++i → Pre increment

i -- → Post decrement --i → Pre decrement

```java
for (int i = 1; i < 10; i--) {
    System.out.println(i);
}
```

i = 1 → i < 10 → 1 → i--

i = 0 → i < 10 → 0 → i--

i = -1 → i < 10

zero hit

infinite

```java
for (int i = 10; i > 0; i--) {
    System.out.println(i);
}
```

```java
for (int i = 10; i > 0;) {
    System.out.println(i);
    i--;
}
```

i = 10 → i > 0 → 10 → i--

i = 9 → i > 0 → 9 → i--

i = 8 → i > 0 → 8 → i--

⋮

i = 1 → i > 0 → 1 → i--

i = 0 → i > 0 → break

```java
        // infinite for loop
//        for (int i = 1; true; i++) {
//            System.out.println(i);
//        }

//        int i = 1000;
//
//        for (i = 1; i < 10; i++) {
//            System.out.println(i);
//        }

        // Scope of a is only in for loops
        for (int a = 1; a < 10; a++) {
            System.out.println(a);
        }

        System.out.println(a);


    }
```

```java
for (char ch = 'a'; ch ≤ 'z'; ch++) {
    System.out.print(ch + " ");
}
```

ASCII

ch = 'a' → 97 (Binary code)

↓

ch = 'z' → 122 (Binary code)

↓

97 < 122 → true

↓

97 → char → 'a'

↓

ch+1 → 97+1 = 98 ('b')

Nested for loop

```java
for (int i = 1; i ≤ 10; i++) {

    System.out.println("****************************");
    System.out.println("i: " + i);
    for (int j = 1; j ≤ 10; j++) {
        System.out.println("j: " + j);
    }

}
```

$i \times j \rightarrow$ Total iterations

$\downarrow$

$10 \times 10 = 100$

$i = 1 \rightarrow j = 1$

$j = 2$

$\vdots$

$j = 10$

$i = 2 \rightarrow j = 1$

$j = 2$

$\vdots$

$j = 10$

$\vdots$

$i = 10 \rightarrow j = 1$

$j = 2$

$\vdots$

$j = 10$

```java
    for (int i = 1; i ≤ 10; i++) {

        System.out.println("****************************");
        System.out.println("i: " + i);
        for (int j = 1; j ≤ 10; j++) {
            System.out.println("j: " + j);
            for (int k = 1; k < 5; k++) {
                System.out.println("k: " + k);
            }
        }

    }

}
```

$i \times j \times k$

$10 \times 10 \times 4 \rightarrow$ 400 iterations

```
for (int i = 5; i > -1; i--) {
    for (int j = 10; j > 0; j--) {
        for (int k = 1; k < 5; k++) {
            }
        for (int k = 1; k < 5; k++) {

            }
    }
}
```

k scope

k scope end

Parallel

```
for (int i = 5; i > -1; i--) {
    for (int j = 10; j > 0; j--) {
        for (int k = 1; k < 5; k++) {

        }

        for (int k = 1; k < 5; k++) {

        }
    }
}
```

1 → 3.1

2 → 3.2

i = 5, 4, 3, 2, 1, 0

j = 10 ⋮ 1

3.1 → k = 1 ⋮ 4 → 4

3.2 → k → 1 ⋮ 5 → 5

Flow → 1 × 2 × max(3.1, 3.2)

6 × 10 × 5

= 60 × 5 = 300

$n = 2$

output $\Rightarrow$ $2 \times 1 = 2$

$2 \times 2 = 4$

$2 \times 3 = 6$

$2 \times 4 = 8$

$2 \times 5 = 10$

$\vdots$

$2 \times 10 = 20$

$n = 4$

$4 \times 1 = 4$

$4 \times 2 = 8$

$4 \times 3 = 12$

$4 \times 4 = 16$

$4 \times 5 = 20$

$\vdots$

$4 \times 10 \rightarrow 40$

```java
Scanner scn = new Scanner(System.in);
int n = scn.nextInt();
for (int i = 1; i <= 10; i++) {
    System.out.println("2 x " + i + " = " + n*i);
}
```

take i/p from user

$n = 10$

→ Sum of n natural numbers

$Sum = 1 + 2 + 3 + \dots n$

$= 55$

```
int sum = 0;
for (int i = 1; i ≤ n; i++) {
    sum += i;
}
System.out.println(sum);
```

Sum = 0

$i = 1 \rightarrow i \leq 10$

$\downarrow$

$Sum = Sum + i$ (Sum += i)