Strings.

- Non primitive data types }
- Classes in Java

99.9%

~~Array~~. Stream of char. }

"Ritik" → R, i, t, i, K

```
char ch = 'a';
char[] name = {'R', 'i', 't', 'i', 'K'};
```

String {Stream of char}

```
                012345
String name = "Gautam";
```

① `System.out.print(name);` ⇝ Gautam

② `System.out.print(name.charAt(3));` ⇝ t

`name.charAt(4)` ⇝ a

③ `System.out.println(name.length());` ⇝ 6

✓ **Strings are immutable !**

ASCII Values :

'a' $\longrightarrow x$

'b' $\longrightarrow x+1$

$\vdots$

'z' $\longrightarrow x+25$

'A' $\longrightarrow y$

$\vdots$

'Z' $\longrightarrow y+25$

Q.o

char ch = 'm' ⟶ 'M'

'a' ⟷ 'm' : dist

'A' ⟷ 'M' : dist

dist = ch - 'a'

char ch2 = (char)('A' + dist)

'M'

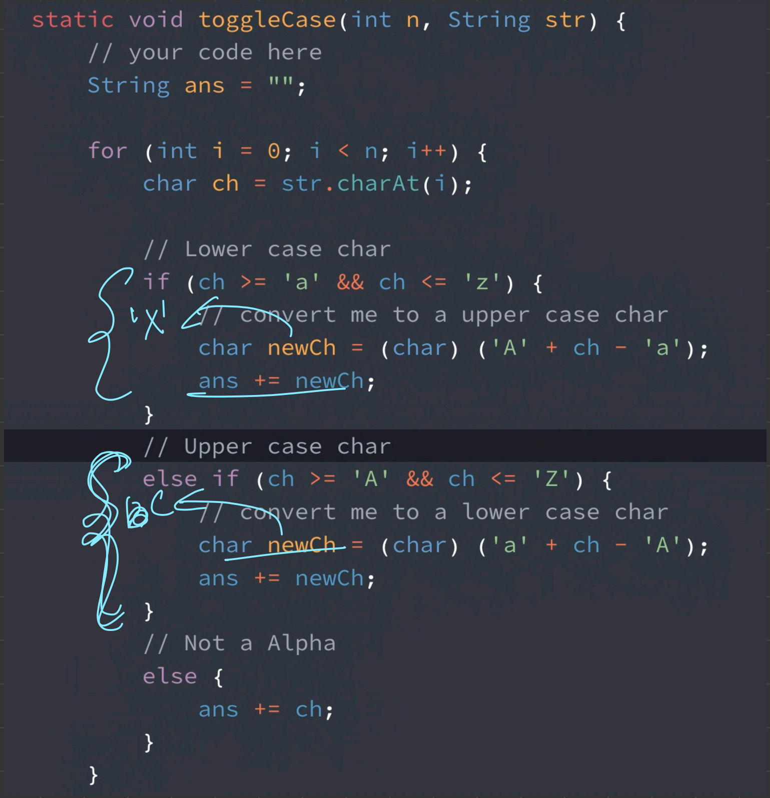
Q

$$\begin{tikzcd}
\text{char ch = 'r'} \arrow[rr, "\text{convert}"] \arrow[d, "\text{dist}"] & & \text{'R'} \arrow[d, "\text{dist}"] \\
\text{'a'} & & \text{'A'}
\end{tikzcd}$$

$\underline{\text{dist = (int)(ch - 'a')}}$

$\underline{\text{ch = (char)('A' + dist);}}$

'R'



```java
static void toggleCase(int n, String str) {
    // your code here
    String ans = "";

    for (int i = 0; i < n; i++) {
        char ch = str.charAt(i);

        // Lower case char
        if (ch >= 'a' && ch <= 'z') {
            // convert me to a upper case char
            char newCh = (char) ('A' + ch - 'a');
            ans += newCh;
        }
        // Upper case char
        else if (ch >= 'A' && ch <= 'Z') {
            // convert me to a lower case char
            char newCh = (char) ('a' + ch - 'A');
            ans += newCh;
        }
        // Not a Alpha
        else {
            ans += ch;
        }
    }
```

String str= "ABCxyz"

ans = 'abc

ans = ans + newch

= abc + 'X'

= abcX

different typing Conventions in Programming

by default we use in Java

1. Camel Casing — javaIsReallyGood

2. Snake Casing — java_is_really_good

3. Kabab Casing — java-is-really-good

Inverse Camel Casing.

str = "IAmJavaProgrammer"

O/P

I
Am
Java
Programmer

1. whenever I see a caps char, I start a new word.
2. whenever I see a caps char again I end the prev and start a new word.

str = "One Piece"

O/P ≠ One
Piece }

str = "OneTwoThreefourfive"

i

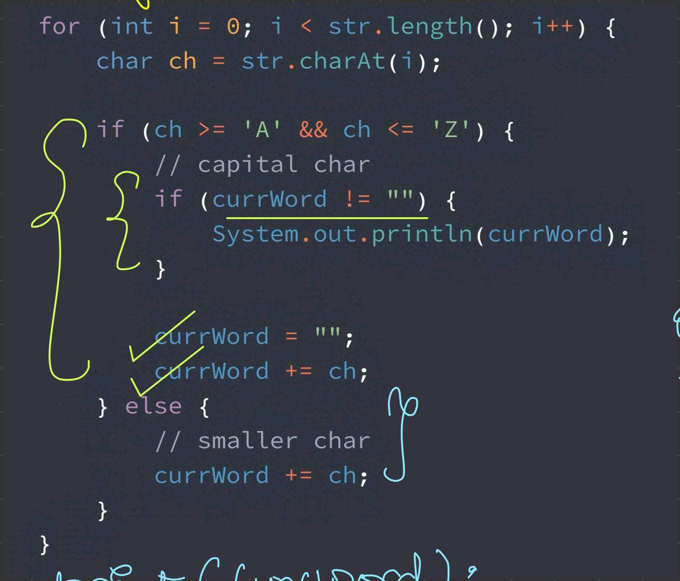
currWord = "T

o/p

One

Two

String currWord = " ";



```
for (int i = 0; i < str.length(); i++) {
    char ch = str.charAt(i);

    if (ch >= 'A' && ch <= 'Z') {
        // capital char
        if (currWord != "") {
            System.out.println(currWord);
        }

        currWord = "";
        currWord += ch;
    } else {
        // smaller char
        currWord += ch;
    }
}
```

print(currWord);

o/p

str = "OneTwoThree"

currWord = ""

Panagram

str = "We promptly judged antique ivory buckles for the next prize"

$\boxed{26}$

0 a
1 b
2 c
3 d
4 e

5 f
6 g
7 h
8 i
9 j

10 k
11 l
12 m
n
o
p

q
r
s
t
u
v

w
x
y
25 z

o/p Panagram

26 - Alphabets

a , A ⟶ 0

b , B ⟶ 1

ch - 'a'

OR

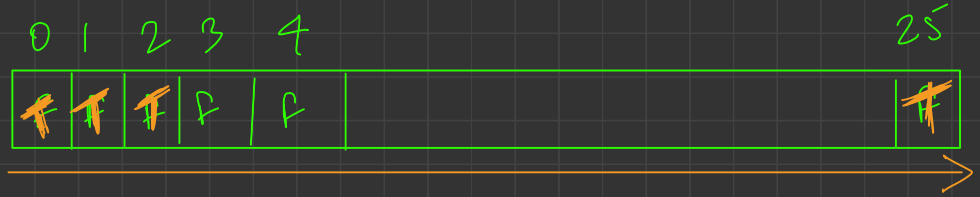
ch - 'A'

Boolean[26]

0 ⟶ 25

0 → 'a' || 'A'

0 → True



str = "AzbC"

Cap A

ch - 'A' = 0

→ 2

Small

→ 1

ch - 'a'

→ 25

Split

str = "Hello This is Beginners 2"

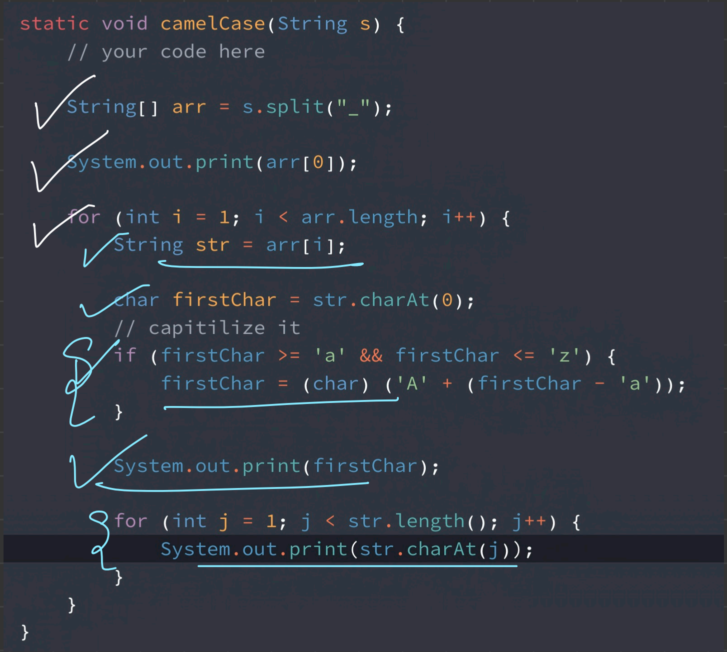
string[] = str.split(" ")

→ {"Hello", "This", "is", "Beginners", "2"}

String str = "Hi, my name is Ritik";

String[] arr = str.split(",");

→ {"Hi", " my name is Ritik"}



```java
static void camelCase(String s) {
    // your code here

    String[] arr = s.split("_");

    System.out.print(arr[0]);

    for (int i = 1; i < arr.length; i++) {
        String str = arr[i];

        char firstChar = str.charAt(0);
        // capitilize it
        if (firstChar >= 'a' && firstChar <= 'z') {
            firstChar = (char) ('A' + (firstChar - 'a'));
        }

        System.out.print(firstChar);

        for (int j = 1; j < str.length(); j++) {
            System.out.print(str.charAt(j));
        }
    }
}
```

str = "abc_xyz_mn"

String[] arr = {"abc", "xyz", "mn"}

0 1 2

✓ $\boxed{abcXyzMn}$

i

str = ~~xyz~~ mn

fc = ~~x~~ ~~X~~ ~~m~~ M