

Strings

Collections of chars.

ex - "ABC", "abc";

Syntax :

String v = "abc";

Variable name

① Traverse (Iterating)

String v = "Maruti";
0 1 2 3 4 5

length = v.length();

```
for(int i=0; i<length; i++)  
{  
    Print(v.charAt(i));  
}
```

→ To iterate over every character.

Using for-each loop

```
for(type char c : v)  
{  
    print(<>);  
}
```

③ Convert String to
Char array

String s = "Maruti";

char arr = s.toCharArray();

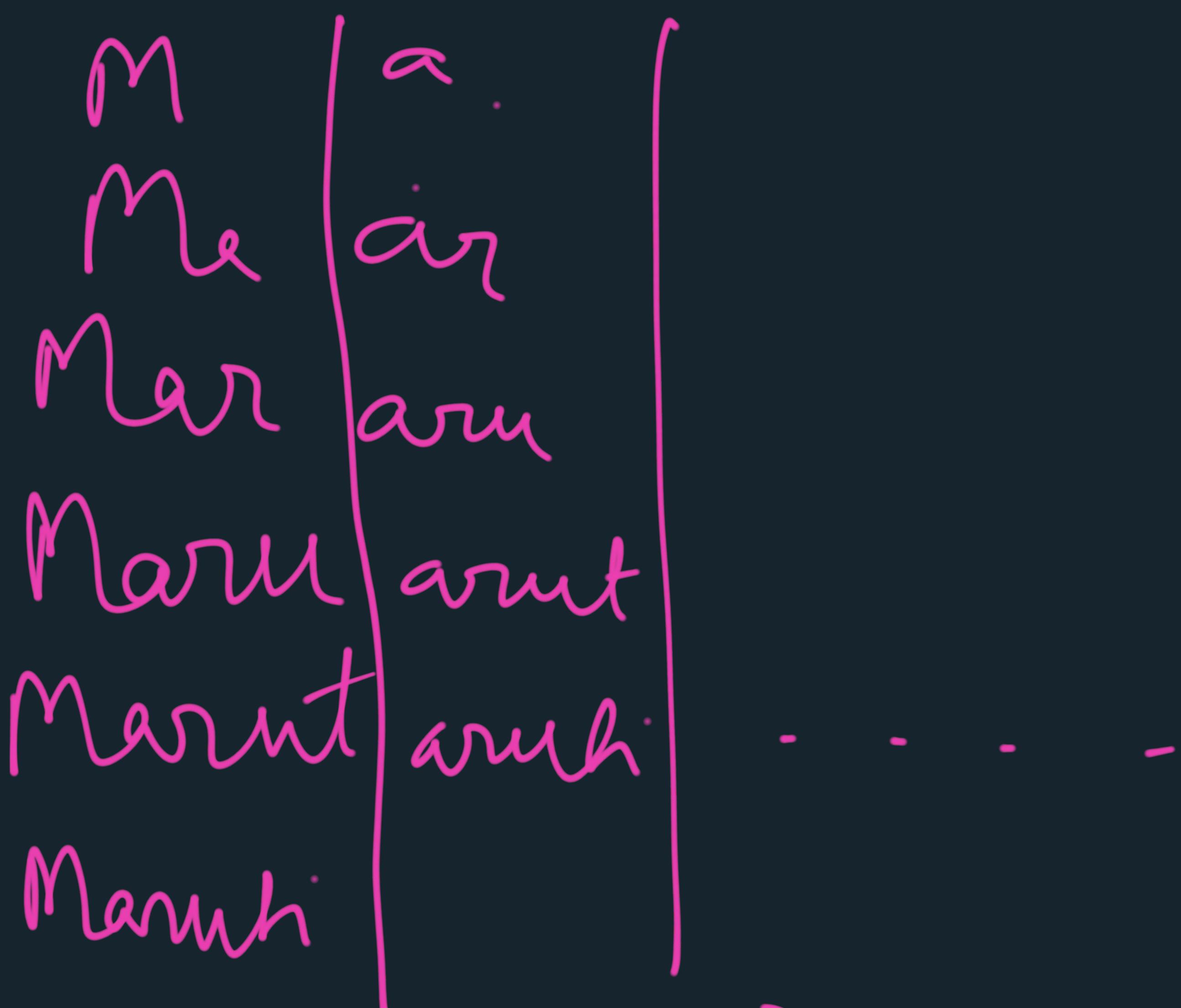


arr = [M | A | R | U | T | I]
 0 1 2 3 4 5

```
for(int i=0; i<arr.length; i++)  
{  
    print(arr[i]);  
}
```

Substring

$S = "Maruti";$



how many ?

$$\rightarrow \frac{n \times (n+1)}{2}$$

Recursive Substring

G

```
for(int i = 0; i < len; i++)
```

```
{  
    for(j → 0 → i)
```

```
{  
    for(k → i - (o j))
```

```
{  
    s(i, j)}
```

```
}
```

Using In-built method

s = "Mahut";
0 1 2 3 4 5

s.substring(start, end)
↓
[start, end]

s.substring(2, 6);
↓ "rut"

s.substring(1);
↓ arut

Using the Substring method

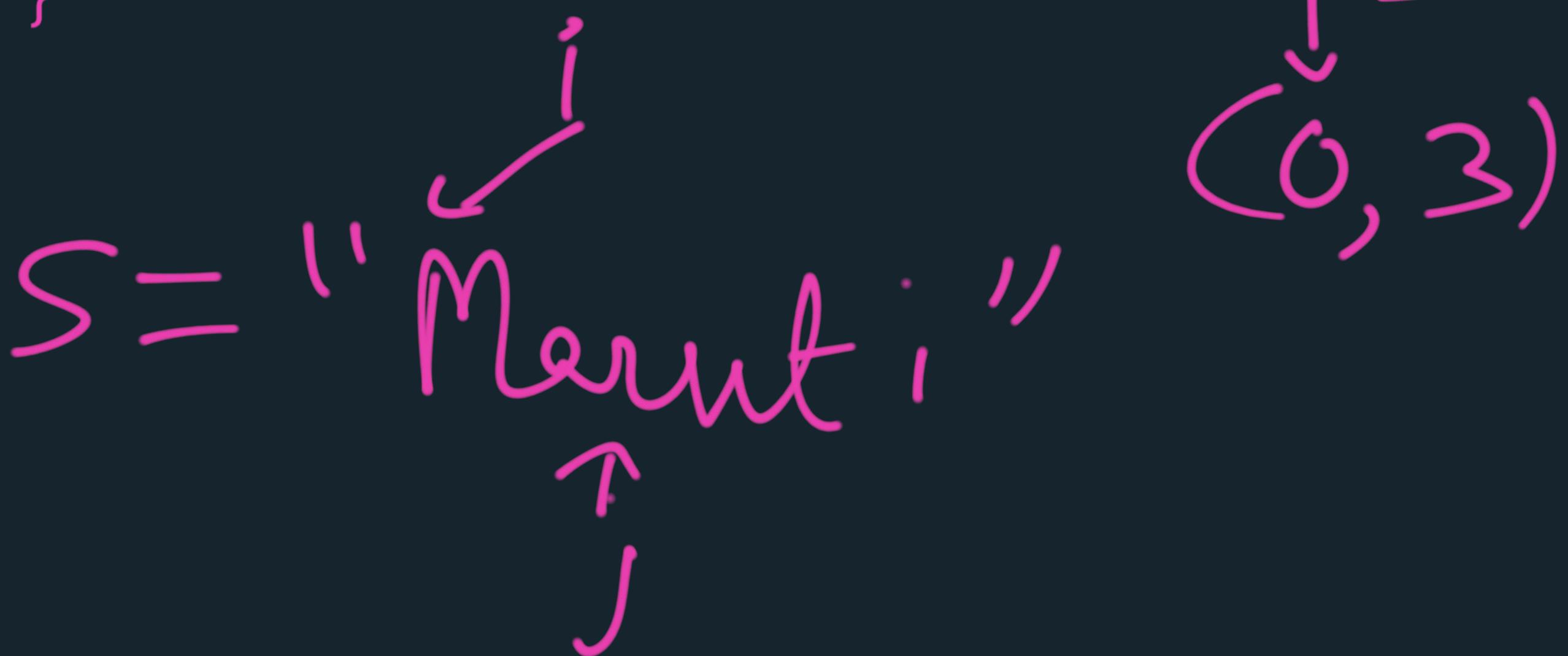
```
for(int i = 0; i < str.length(); i++) {  
    for(int j = i; j <= str.length(); j++) {  
        System.out.println(str.substring(i, j+1));  
    }  
}
```

$s = "Meraut;"$

↑
 j

i

$(0, 3)$



Types of string

G

" ABCAA " → X

" AA CA A " → ✓

Reverse of string

= Original string

↓
Palindrome

②

Reverse String

```
String rev = "";
```

```
for (int i = s.length() - 1; i >= 0; i--) {
    rev = rev + s.charAt(i);
}
```

③

```
char[] arr = s.toCharArray();
```



```
arr =
```

M	A	R	U	T	I
---	---	---	---	---	---



```
arr =
```

E	T	U	R	A	M
---	---	---	---	---	---



```
String rev = String.valueOf(arr);
```

Used to get string back from char array.

Check Palindrome

G

```
if (rev. equals (orig))  
{  
    print ("Palindrome")  
}
```