Methods of Strings

(1) length():

```
String s = "Good Morning";
System.out.println(s.length());
```

o/p: 12

Good Morning

↓

Char Array

G	o	o	d		m	o	r	n	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11

s.indexOf("i"); // 9

ASCII Values

↓

Characters

- Alphabets
  - Uppercase: 65-90
  - lowercase: 97-122
- Numeric Chars: 48-57
- Special chars

Case 1

```
    97 98 99
S1 = a  b  c
    97 98 91
S2 = a  b  c
    -------
     0  0  0
```

S.compareTo(S2) = 0

Case 2

S1 = abc

S2 = ab

3 - 2 = 1

a

abcd

1 - 4 = -3

a = a

abcdef - abcdef

✓ (1) different

```
    97 98 99
c1 = a  b  c
    97 97 99
s2 = a  a  c
    -------
     0  1 →
```

(2) toLowerCase() toUpperCase()

(3) contains()

(4) startsWith() endsWith()

(5) indexOf()

(6) trim()

⑦ compareTo

⑧ compareToIgnoreCase

⑨ equalsToIgnoreCase

⑩ charAt()

⑪ subString()

⑫ toCharArray()

⑬ split()

⑭ replace

① WAP to reverse a string

② WAP to check palindrome or not given string

Exercises

Input: Jack and jill

Output: (1) jill and jack

(2) kcaj dna llij

(3) jack4 and3 jill4

(4) Jack And JIll

(5) jIll aND jACK

(6) llij dna kcaj

(7) j#ck a** j#ll

Parsing String to primitive types:

String n = "100"

Integer.parseInt(n)

Byte parseByte("120");

Short.parseShort("555")

Float.parseFloat("125.67")

125.6

Boolean.parseBoolean("True")

Note:

1) In case of parseBoolean. if we pass any thing apart "true" it returns false

2) In case if we pass any invalid type in the above case it would given an exception called "java.lang.NumberFormatException"

eg: Integer.parseInt("A")