Strings -

- Data types (Not objects; represents raw values)
- int, char, double, float ... primitive data types
- String is an example of a non-primitive data type
- Strings in java are objects represented by a char array
- Since, arrays are immutable (cannot grow), strings are immutable as well
- When a change to a string is made, an entirely new string is created

"Hello"

index	0	1	2	3	4	5
Variable	H	e	l	l	o	\0
Address	0x1	0x2	0x3	0x4	0x5	0x6

Syntax -

<String_type> <string_variable> = "<sequence_of_string>";

String name = "Anuj";

String role = "Instructor";

name and role are objects of String class.

Two ways to create a string-

- String literal

```
String siz "Anuj";
```

- Using new keyword-

```
String siz new String ("Anuj");
```

Character encoding-

tells computer how to interpret digital data into letter, numbers and symbols. This is done by assigning a specific numeric value to a letter, number or symbol.

There are a number of character encoding sets in use today, but the most common format in use on the World Wide Web are ASCII, UTF-8, and Unicode.

ASCII-

- American School Code for Information Interchange
- Character encoding standard for electronic communication
- ASCII code represent text in computers, etc.
- "Hello WorLD" should be interpreted by computer? in UTF-8

Decimal 72

\x48 \x65 \x4C \x4C \x6F \x20

Character

H \x57 \x6F \x72 \x6C \x44

You can convert a character, e.g., 'A' to its corresponding ASCII value 65 by just storing it into a numberic data type...

e.g., byte, int or long

```
int asciiOfA = (int) 'A';
```

String methods—

① → charAt(index); → == operator compares references not values

② → concat(s);

③ → equalsIgnoreCase(s)

④ → length();

⑤ → replace('x', 'y');

⑥ → substring(5);

⑦ → substring(5, 8);

⑧ → toLowerCase();

⑨ → toUpperCase();

⑩ → contains(s);

⑪ → String.join("|", name1, name2, ...);

```
s = "Hello, World";
s.split("\\,");
.trim();
startsWith("Sa");
endsWith("na");
String.valueOf(otherType);
```

Searching and sorting in the string-

Searching operations-

- charAt(indexNumber);
- lastIndexOf(c, fromIndex); // Search backward from the specified
- indexOf(c, indexFrom);
- lastIndexOf(c);
- indexOf(c);

Sorting in strings-

Order (lexicographic) is a generalization of ways words are alphabetically ordered based on alphabetic order of their component letters.

s1.compareTo(s2);

s1 > s2	+ve value
s1 == s2	0
s1 < s2	-ve value

Own CompareTo method-

public static int stringCompare(String s1,
String s2) {

```
for (int i=0; i< str1.length() &&
                 i< str2.length();
                 i+=1) {

    if ((int) str1.charAt(i) ==
        (int) str2.charAt(i)) {

            continue;
    } else {

        return (int) str1.charAt(i) -
                     (int) str2.charAt(i);

    }
}

if (str1.length() < str2.length()) {

    return (str1.length() -
            str2.length());

} else if (str1.length() > str2.
                           length()) {

    return str1.length() -
           str2.length();

} else {
    return 0;

}
}
```

StringBuilder –

- create mutable (modifiable) string
- append("some");
- insert(1, "Test"); Hello → HTestello
- replace(1, 3, "JAVA"); Hello → HJAVAlo

  (1 2 3)
- delete(1, 3)
- reverse()

String	StringBuilder
- Immutable - Slow and consumes more memory when we concatenate many strings because every time it creates new instance	- Mutable - fast and consumes less memory when we concatenate strings

s.hashCode() → address of the string in memory

for string and StringBuilder