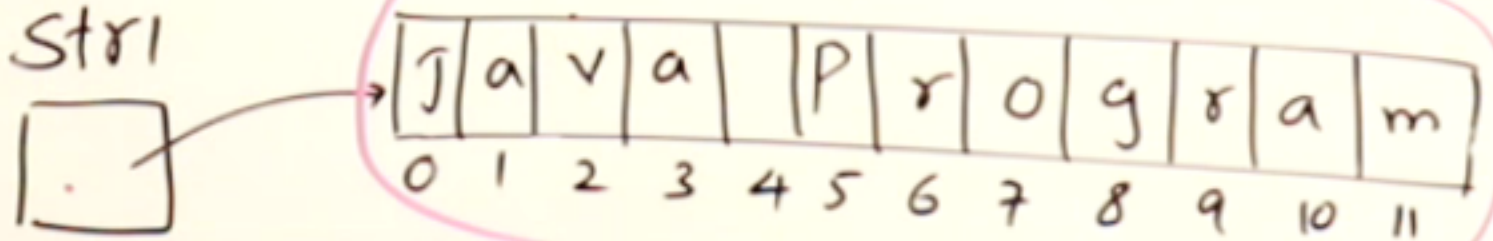
String

literal

String str1 = "Java Program";

1 Reference.

string object



Constructors

- String(char[])
- String(byte[])
- String(String)

o. String is basically an object that represents sequence of char values. An array of characters works same as Java string.

Ex: Char c = { 'j', 'a', 'v', 'a' };

.

Same as; String c = "java";

➢ jo naam ham String ko deta hai usa "reference" bolta hai ha which is used for pointing objects.

➢ String object is referred to as a literal.

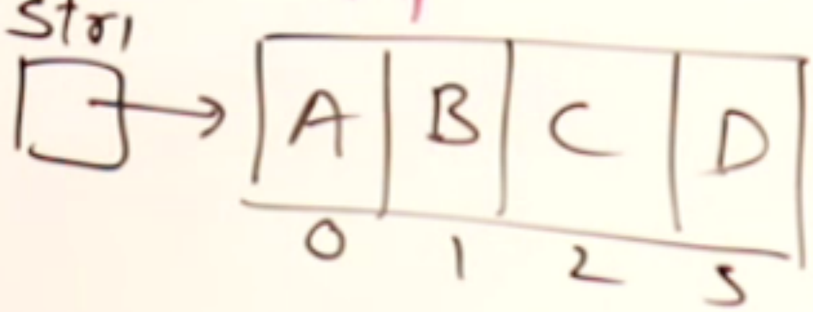
➢ ***Way to declare string known as 'Constructor'.***

Constructors

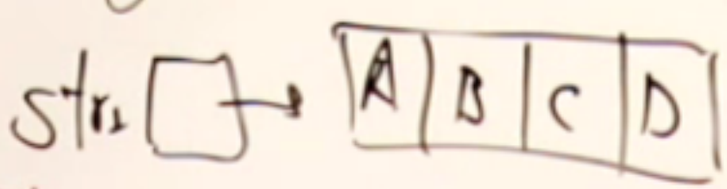
- String(char[])
- String(byte[])
- String(String)

```
char c[] = {'A', 'B', 'C', 'D'};

String str1 = new String(c);
```

Heap



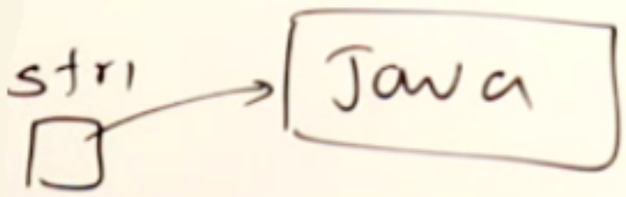

```
byte b[] = {65, 66, 67, 68};

String str2 = new String(b);
```



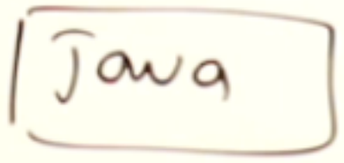
String

String str1 = new String("Java");

Heap



pool

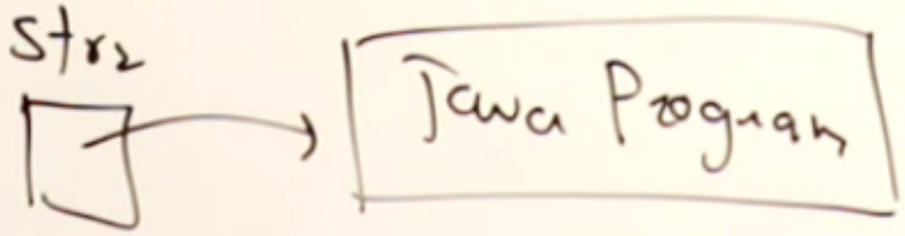


Constructors

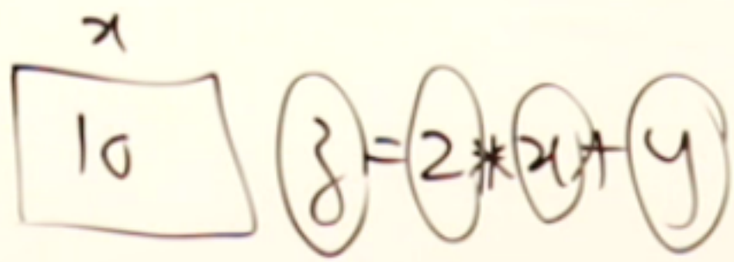
- String(char[])
- String(byte[])
- → String(String)

String str2 = "Java Program";

Pool



int x=10;



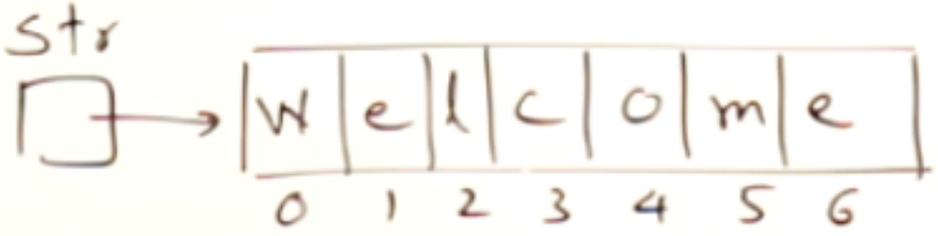
➢ *Whenever "new" is applied the object is created in heap memory and the memory occupied by the object literal is in pool.*
Iska mtlb hai ki jab ham "new" ka help sai String define karta hai to wo phela heap mai memory create karta hai then wo pool mai bhi same data store karta hai.

➢ *Jab ham "without new" String declare karta hai then wo data only costant pool mai hi save hota hai.*

➢ *Agar ham same value ka constant ko 2 alag - alag variable sai declare karta hai to wo memory mai sirf ek hi place par store hota hai aur reference uska taraf point karta hai. Lekin both refernce aapas mai independant hota hai mtlb agar ham ek ki value change kara to dusra mai koi bhi change nahi hoga bas jis mai change kia hai wo new value ki taraf point karaga*

String

```
String str="Welcome";
```




```
str = str.replace('e','k');
```

W	k	l	c	o	m	k

↠ *In Java, a 'String' object is immutable, which means once it is created, its value cannot be changed.*

↠ *Agar ham yaha dia hua kisi bhi "string class method" sa kisi string mai koi bhi change karta hai to wo use type ka new string bna ka show karta hai jis type ka input diya jata hai. Aur purana wala as it is rahta hai.*

- int length()
- string toLowerCase()
- string toUpperCase()
- string trim()
- string substring(int begin)
- string substring(int begin, int end)
- string replace(char old, char New)

```
boolean startsWith(String s)
boolean endsWith(String s)
char    charAt(int index)
int     indexOf(String s)
int     lastIndexOf(String s)
boolean equals(String s)
boolean equalsIgnoreCase(String s)
int     compareTo(String s)
String  valueOf(int i)
```

Matching Symbols

Regular Expression	Description
.	Any character
[abc]	Exactly given letters
[abc][vz]	Either first or second set
[^abc]	Except abc
[a-z1-7]	a-z or 1-7
AIB	A or B
XZ	Exactly XZ

String str="m"

Method to run this

```
import java.util.Scanner;
public class regularexpression
{
    Run | Debug
    public static void main(String[] args)
    {
        Scanner sc=new Scanner (System.in);
        String str1;
        System.out.print(s:"Enter your input = ");
        str1=sc.next();
        System.out.println(str1.matches(regex:"[^xyz]"));
        sc.close();

    }

}
```

Meta Characters

Regular Expression	Description
\d	Digits
\D	Not digits
\s	Space
\S	Not space
\w	Alphabets or digit
\W	Neither alphabet or digit

Quantifiers

Regular Expression	Description
*	0 or more time
+	One or more
?	0 or 1 time
{X}	X times
{X,Y}	Between X and Y time

Methods of string class:

- The methods creates a new string before giving the results.
- The new object is then created in heap memory.

#1.

- str.length(): it returns the length of the string mentioned.
- str.toLowerCase(): it converts the given characters of string into lower case.
- str.toUpperCase(): it converts the given characters of string into upper case.
- str.trim(): it is used to remove the leading and tailing spaces from the array if there are any.
- str.substring(int begin): it returns a new string by giving the part of a string from where the index is given.
- str.substring(int begin, int end): it works same as the above but the starting and ending index both can be given in the substring.
- str.replace(char old, char new): it replaces the old character with the new character.

- str.startsWith(string s): to find the particular starting word of a the string/to find the initials.
- Str.endsWith(string s): to find particular ending word of the string.
- str.charAt(int index): to find the particular character present on the index given.
- str.indexOf(String s): to find the index of the given character.
- str.lastIndexOf(string s): to find the index of the given character from the end of the array.

- str.equals(string s): to check whether the contents of two strings are equal or not.
- str.equalsIgnoreCase(string s): to check whether the contents of two strings are equal or not but it does not depend upon the case of characters.
- str.compareTo(string s): to compare two strings according to the dictionary order(in accordance with the ASCII codes for cases of characters).
- str.valueOf(int i): to convert different types of data into string data type.

Regular Expressions:

They are used to define the symbols.

Matching symbols:

- these symbols are meant for single alphabets.
- '.' it means any letter or the symbol from the keyboard.i.e for single alphabet it is true.
- [abc]: range or set of characters/ the string is true if the alphabet is either a or b or c.
- [abc][vz]: range of multiple symbol/the string is true if first symbol is among a,b &c and second symbol is among v & z.
- [^abc]: the string is true if the symbol is other than a, b, & c.
- [a-z 1-7]: the string is true if the symbol is from the range a-z or 1-7.
- A|B: it is true for single alphabets fro either A or b.
- XZ: to check whether the string maybe either a single or multiple alphabets.

Meta characters:

- \d: it will be true if it is a digit among 0-9.
- \D: it will be true if it is any symbol other than digits.
- \s: it will be true if there is just a space.
- \S: it will be true if there are any symbols other than space.
- \w: it will be true if it is either alphabet or digits.
- \W: it will be true if it is any symbol other than alphabets or digits.

String matching with regular expressions:

- Quantifiers : these are used to define the number of symbols you want.
  - '*' it represents number of occurrences of any of the characters for zero or more times.
  - '+' it represents number of occurrences of any of the character for one or more times.
  - '?' it represents number of occurrences of any of the characters for zero or one time.
  - {X} it represents any of the characters for the size of X value given.
  - {X, Y} it represents any of the characters for the min and max size given.