Java String class functions

The methods specified below are some of the most commonly used methods of the String class in Java.

- charAt() method

String charAt() function returns the character located at the specified index.

```java
public class Demo {

   public static void main(String[] args) {

      String str = "Chitkara";

      System.out.println(str.charAt(2));

   }

}
```

Output: i

NOTE: Index of a String starts from 0, hence str.charAt(2) means third character of the String str.

- equalsIgnoreCase() method

String equalsIgnoreCase() determines the equality of two Strings, ignoring their case (upper or lower case doesn't matter with this method).

```java
public class Demo {

   public static void main(String[] args) {

      String str = "java";

      System.out.println(str.equalsIgnoreCase("JAVA"));

   }

}
```

It returns true if strings are equal, false if not equal.

- indexOf() method

String indexOf() method returns the index of first occurrence of a substring or a character. indexOf() method has four override methods:

int indexOf(String str): It returns the index within this string of the first occurrence of the specified substring.

int indexOf(int ch, int fromIndex): It returns the index within this string of the first occurrence of the specified character, starting the search at the specified index.

int indexOf(int ch): It returns the index within this string of the first occurrence of the specified character.

int indexOf(String str, int fromIndex): It returns the index within this string of the first occurrence of the specified substring, starting at the specified index.

Example:

```
public class Study {

    public static void main(String[] args) {

        String str="ChitkaraUniversity";

        System.out.println(str.indexOf('U'));

        System.out.println(str.indexOf('t', 5));

        String subString="Uni";

        System.out.println(str.indexOf(subString));

        System.out.println(str.indexOf(subString,9));        }

}
```

Output

8

16

8

-1

NOTE: -1 indicates that the substring/Character is not found in the given String.

- length() method

String length() function returns the number of characters in a String.

```java
public class Demo {
    public static void main(String[] args) {
        String str = "Count me";
        System.out.println(str.length());
    }
}
```

Output

8

- replace() method

String replace() method replaces occurances of character with a specified new character.

```java
public class Demo {
    public static void main(String[] args) {
        String str = "Change me";
        String x = str.replace('C','c');
        System.out.println(str.replace('m','M'));
        System.out.println(x);
    }
}
```

Output

Change Me

- substring() method

String substring() method returns a part of the string. substring() method has two override methods.

1. public String substring(int begin);

2. public String substring(int begin, int end);

The first argument represents the starting point of the subtring. If the substring() method is called with only one argument, the subtring returns characters from specified starting point to the end of original string.

If method is called with two arguments, the second argument specify the end point of substring.

```
public class Demo {
    public static void main(String[] args) {
        String str = "Java Programming";
        System.out.println(str.substring(5));
        System.out.println(str.substring(5,7));
    }
}
```

Output

Programming

Pr   // index ends at -1

- toLowerCase() method

String toLowerCase() method returns string with all uppercase characters converted to lowercase.

```
public class Demo {
    public static void main(String[] args) {
        String str = "CHITKARA";
        System.out.println(str.toLowerCase());
```

```
    }
}
```
Output

chitkara

- toUpperCase() method

This method returns string with all lowercase character changed to uppercase.

```
public class Demo {
    public static void main(String[] args) {
        String str = " university";
        System.out.println(str.toUpperCase());
    }
}
```
Output

UNIVERSITY

- valueOf() method

String class uses overloaded version of valueOf() method for all primitive data types and for type Object.

NOTE: valueOf() function is used to convert primitive data types into Strings.

```
public class Demo {
    public static void main(String[] args) {
        int num = 35;
        String s1 = String.valueOf(num);    //converting int to String
        System.out.println(s1);
        System.out.println("type of num is: "+s1.getClass());    }
```

}

Output

35

type of num is: java.lang.String

- toString() method

String toString() method returns the string representation of an object. It is declared in the Object class, hence can be overridden by any java class. (Object class is super class of all java classes).

```
public class Car {

    public static void main(String args[])

    {

        Car c = new Car();

        System.out.println(c);

    }

    public String toString()

    {

        return "This String will return to car object";

    }

}
```

Output

This String will return to car object

Whenever we will try to print any object of class Car, its toString() function will be called.

NOTE: If we don't override the toString() method and directly print the object, then it would print the object id that contains some hashcode.

- trim() method

This method returns a string from which any leading and trailing whitespaces has been removed.

```java
public class Demo {

    public static void main(String[] args) {

        String str = "   Remove WhiteSpaces  ";

        System.out.println(str.trim());

    }

}
```

Output

Remove WhiteSpaces

- contains()Method

String contains() method is used to check the sequence of characters in the given string. It returns true if a sequence of string is found else it returns false.

```java
public class Demo {

    public static void main(String[] args) {

        String a = "Hello welcome to Chitkara University";

        boolean b = a.contains("Chitkara");

        System.out.println(b);

        System.out.println(a.contains("university"));

    }

}
```

Output

true

false

- startsWith() Method

String startsWith() is a string method in java. It is used to check whether the given string starts with given prefix or not. It returns true when prefix matches the string else it returns false.

```
public class Demo {

    public static void main(String[] args) {

        String str = "Chitkara";

        System.out.println(str.startsWith("C"));

        System.out.println(str.startsWith("t"));

        System.out.println(str.startsWith("k",4));     }

}
```

Output

true, false, true

- endsWith() Method

String endsWith() method is used to check whether the string ends with the given suffix or not. It returns true when suffix matches the string else it returns false.

```
public class Demo {

    public static void main(String[] args) {

        String a="Hello welcome to Chitkara University";

        System.out.println(a.endsWith("University"));

        System.out.println(a.endsWith("y"));

    }

}
```

Output

true, true

- format() Method

String format() is a string method. It is used to the format of the given string.

Following are the format specifiers and their datatypes:

Format Specifier	Conversion Applied
%%	Inserts a % sign
%x %X	Integer hexadecimal
%t %T	Time and Date
%s %S	String
%n	Inserts a newline character
%o	Octal integer
%f	Decimal floating-point
%e %E	Scientific notation
%g	Causes Formatter to use either %f or %e, whichever is shorter
%d	Decimal integer
%c	Character
%b %B	Boolean
%a %A	Floating-point hexadecimal

```
public class Demo {
    public static void main(String[] args) {
        String a1 = String.format("%d", 125);

        String a2 = String.format("%s", "JAVA");

        String a3 = String.format("%f", 125.00);

        String a4 = String.format("%x", 125);
```

```java
        String a5 = String.format("%c", 'a');

        System.out.println("Integer Value: "+a1);

        System.out.println("String Value: "+a2);

        System.out.println("Float Value: "+a3);

        System.out.println("Hexadecimal Value: "+a4);

        System.out.println("Char Value: "+a5);

    }

}
```

Output

Integer Value: 125

String Value: JAVA

Float Value: 125.000000

Hexadecimal Value: 7d

Char Value: a

- getBytes() Method

String getBytes() method is used to get byte array of the specified string.

```java
public class Demo {

    public static void main(String[] args) {

        String a="abcd";

        //String a="ABCD";

        byte[] b=a.getBytes();

        for(int i=0;i<b.length;i++)

        {

            System.out.print(b[i]+" ");

        }
```

```
    }
}
```

Output

97 98 99 100

- getChars() Method

String getChars() method is used to copy the content of the string into a char array.

```
public class Demo {
    public static void main(String[] args) {
        String a= new String("Hello Welcome to Chitkara");
        char[] ch = new char[15];
        a.getChars(6, 16,ch, 0);
        System.out.println(ch);
    }
}
```

Output

Welcome to

- isEmpty() Method

String isEmpty() method is used to check whether the string is empty or not. It returns true when length string is zero else it returns false.

```
public class IsEmptyDemo1{
    public static void main(String args[]){
        String a="";
        String b="Chitkara";
        System.out.println(a.isEmpty());
```

```
        System.out.println(b.isEmpty());    }
}
```

Output

true

false

- join() Method

String join() method is used to join strings with the given delimiter. The given delimiter is copied with each element

```
public class JoinDemo1
{
    public static void main(String[] args)
    {
        String s = String.join("*","Welcome to Chitkara");
        System.out.println(s);
        String date1 = String.join("/","04","05","2023");
        System.out.println("Date: "+date1);
        String time1 = String.join(":", "02","05","10");
        System.out.println("Time: "+time1);
    }
}
```

Output

Welcome to Chitkara

Date: 04/05/2023

Time: 02:05:10

- List of methods of String class in JAVA:-

Method	Description
char charAt(int index)	It returns char value for the particular index
int length()	It returns string length
static String format(String format, Object... args)	It returns a formatted string.
static String format(Locale l, String format, Object... args)	It returns formatted string with given locale.
String substring(int beginIndex)	It returns substring for given begin index.
String substring(int beginIndex, int endIndex)	It returns substring for given begin index and end index.
boolean contains(CharSequence s)	It returns true or false after matching the sequence of char value.
static String join(CharSequence delimiter, CharSequence... elements)	It returns a joined string.
static String join(CharSequence delimiter, Iterable<? extends CharSequence> elements)	It returns a joined string.
boolean equals(Object another)	It checks the equality of string with the given object.
boolean isEmpty()	It checks if string is empty.
String concat(String str)	It concatenates the specified string.

String replace(char old, char new)	It replaces all occurrences of the specified char value.
String replace(CharSequence old, CharSequence new)	It replaces all occurrences of the specified CharSequence.
static String equalsIgnoreCase(String another)	It compares another string. It doesn't check case.
String[] split(String regex)	It returns a split string matching regex.
String[] split(String regex, int limit)	It returns a split string matching regex and limit.
String intern()	It returns an interned string.
int indexOf(int ch)	It returns the specified char value index.
int indexOf(int ch, int fromIndex)	It returns the specified char value index starting with given index.
int indexOf(String substring)	It returns the specified substring index.
int indexOf(String substring, int fromIndex)	It returns the specified substring index starting with given index.
String toLowerCase()	It returns a string in lowercase.
String toLowerCase(Locale l)	It returns a string in lowercase using specified locale.

String toUpperCase()	It returns a string in uppercase.
String toUpperCase(Locale l)	It returns a string in uppercase using specified locale.
String trim()	It removes beginning and ending spaces of this string.
static String valueOf(int value)	It converts given type into string. It is an overloaded method.