Contents

Methods of String

Strings is a data type that use in programming. It represent the text rather than number. Include number, text, word and special characters. And also include white spaces. String can't fixed. The original string can't fix. String array was use if you want to store string data. So in string, it has many methods and types. Here is some method of strings and its functions with real world example.

Int hashcode()

It has the return value a hash code for a string. It return a inter value using by hashing algorithm. Object which is equal length with different character must return different hash code. But same character and same length but different variable will also take the same return hash value of the string. Because of its hashing methods.

```
s[0]*31^(n - 1) + s[1]*31^(n - 2) + ... + s[n - 1]
```

Because of that hashing method. Here is testing code.

```
*/
public class HashCode {

  public static void main(String[] args) {

    String fb="FB";
    System.out.println("Hashcode for FB:" + fb.hashCode());

  }

}
```

So I announce the string with data. And display output with the hash code of the announce string. If you run the output will see as an integer because, it returns integer value as shown in figure.

```
Hashcode for FB:2236
BUILD SUCCESSFUL (total time: 0 seconds)
```

2236 is the hash code for FB. It is the return value of 'FB' as an integer. So I hope the working of the hash code. One more thing. It doesn't need any parameter and return value.

As a real world example, it is only use in technical term. It is just to know that java has hashing function. Mostly it doesn't use in programming. Because programs are using combine like database, eclipse and so on. That kind of program has their own authorization and their own encryption method. But if you can have an external security. You can try it. Even the database security is break out. Here is code sample.

```
public static void main(String[] args) {

  Scanner sc= new Scanner (System.in);
   System.out.println("Are You authorize?");
   String yes=sc.nextLine();
   String pass="qwert";

   if(yes.equals("yes"))
   {
     System.out.println("Your Password is " +pass);
   }
   else
   {
     System.out.println("Your Password is " + pass.hashCode());
   }


}
```

If you are authorize and type yes. You will see a password if not you can't see the password, you will only see hash code of the password.

```
Are You authorize?
yes
Your Password is qwert
```

```
Are You authorize?
no
Your Password is 108003713
```

You will see here difference of authorize. I hope you understand about hash code of string.

String replace (char oldChar, char newChar)

This method returns as a string which is replacing all the old character to new characcter. If the oldChar does not present in the character which represented by this String object, then a reference to this String object is returned. Or a new String object is created that symbolize a character to the new character represented by this String object, except that every occurrence of oldChar is replaced by an occurrence of newChar.

This method return new character to replace and it need parameter of old character and new character to replace.

```
public String replace(char oldChar, char newChar)
```

Here is a sample of replace method.

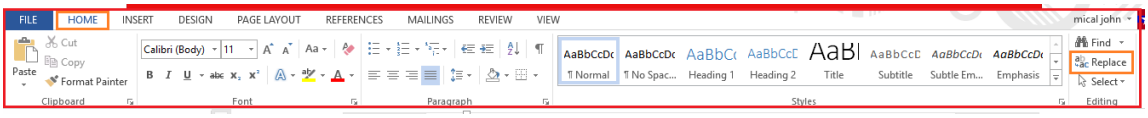
```
public class Replace {

  public static void main(String[] args) {

    String test="I am Mg Mg";
    System.out.println("Old Character = " + test);
    System.out.println("Replace Result = " + test.replace("Mg", "Ag"));

  }

}
```

You will see a string that is announce “I am Mg Mg”. So that person want to change his name. so we don’t fix the original string. We just use replace method. So old character of the first is “Mg” to replace with a new character is “Ag”. If you run, you will see the changes as shown in figure.

```
Old Character = I am Mg Mg
Replace Result = I am Ag Ag
BUILD SUCCESSFUL (total time: 0 seconds)
```

“Mg” was replace by “Ag”. It is very simple to use. It return the the replacement new character as a string object and overwride it in the memory.

As an real world example microsoft word is the best example for using that method. So in microsoft word, in “Home” tab at “Editing“ place. You will find a “Replace” button. You will find a box. in that box you will see two input first is the old character second is the new character that you want to replace.



// photo of replacement in word

It is easy to use for the changes. For example an announcement of Gusto Institude. That announcement is too long and a owner want to fix the instutide to Collage. It is not possible to find and fix each word. So using replace method is the good way to solve the proble faster and very effective to use it. it can have less time and less worry. So I hope you will understand about the replace method.

String replaceAll()

This method replace the character with all strings by checking with the regular expression. It return the new string of the replacement with the regax. It is the opposite of replace method. Here is a clear example.

```
import java.io.*;
public class Test {

   public static void main(String args[]) {
      String Str = new String("Welcome to Tutorialspoint.com");

      System.out.print("Return Value :" );
      System.out.println(Str.replaceAll("(.*)Tutorials(.*)", "AMROOD"));
   }
}
```

So a string is announce and it use the old character with the regax and the new character. If you run that the whole string is replaced with as shown in figure. Because of the regax.

```
Return Value :AMROOD
```

So that will replaced instead of the old string.

As an real-world example, this is same with the replacement method (). It replace all with the regular expression. So I hope you will understand about the replaceAll method.

String [] split (String regax)

This method is use to split the string into it sub strings which is based on the given parameter of regular expression. It return array of strings enusing splid as an input string on the regular expression. If the expression does not match any part of the string then the resulting array has just one element, namely this string. One more things we need to remember. We can't make out put easily by using print. It will show you the memory address. Because it store in a array in a memory. So I hve to use moderanize loop to display out put. So here is clear example.

```
public static void main(String[] args) {

  String test="Once upon a time, there has a king in the America. One day, he goes hunts in the forest\n";

  System.out.println("Before Split = " +test);

 for (String retval: test.split("\\.")) {
  System.out.println("After Split = "+retval);
 }
```

So here is a string that include with many texts that inclue full stop. So if you want to split you have to find a expression. In this case the expression will be full stop. So it will place in the regex in the split method by using modernize loop to deiplay instead of memory address. So it will split every full stop and display each sentences. So here is the out put of that expression.

```
Before Split = Once upon a time, there has a king in the America. One day, he goes hunts in the forest

After Split = Once upon a time, there has a king in the America
After Split =  One day, he goes hunts in the forest

BUILD SUCCESSFUL (total time: 0 seconds)
```

It will split the string as shown photos. The regex can use the regular expression like \\D,\\W and others regular expression.

As an realworld example, also with microsoft word. You will see the total word count on the left button. There is a word count to know for the user exactly.know the total. It is not possible to count each word by user. And we have to split the spaces in every words. So in the regular expression we have to place the "\s" to check the white space. It will split all word for each spaces. And we have to use string tonkenizer which is allow to break string into tokens. It return integer value for each words. So I use that object form split and it will add to count and display total coudt of words.

```
public static void main(String[] args) {

  String test="Once upon a time, there has a king in the America. One day, he goes hunts in the forest\n";
 System.out.println("Before Split = " +test);
         int count=0;
 for (String retval: test.split("\\s" )) {

    StringTokenizer tokens = new StringTokenizer(retval);

     count= count+tokens.countTokens();


 }

  System.out.println("Total Word Count = "+count);

}
```

```
Before Split = Once upon a time, there has a king in the America. One day, he goes hunts in the forest

Total Word Count = 19
BUILD SUCCESSFUL (total time: 0 seconds)
```

In Microsoft word it also ust that kind of logics to count word. I hope you will understand for split method of string.

Int length()

It is a method which returns the length of string that is accept for string onjects of length and returns the integer or number of each character which has in string. It include white spaces which has in the string value. And also include the special character because they can accept as string in string value. So here is a example code.

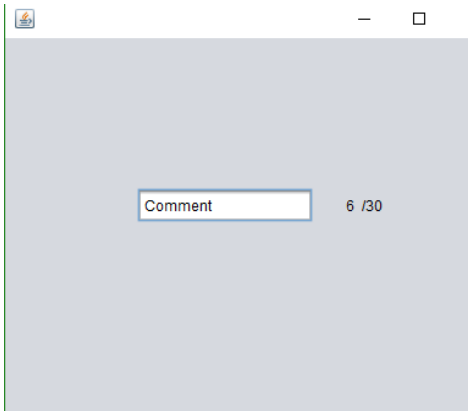
```
import java.io.*;
public class Test {

   public static void main(String args[]) {
      String Str1 = new String("Welcome to Tutorialspoint.com");

      System.out.print("String Length :" );
      System.out.println(Str1.length());


   }
}
```

So you will see a string which is announce. So if you out put it with length(). It will show the total length of that string which will including with spaces.

```
String Length :29
```

There is a out put of the string with return the integer value of the string length.

As a realworld example, it use to check the length of the string of the text box. We can assume that the tex box length is 0. There is no data. It means blank. If the length is 0 we have to assume it is blank. So user name can't be blank in the realworld program. So if the staff skip it will save blank data of the user name. so it is a kind of technical term useage. But some of the text boxes have limitation. So it can show the limitation of text box data. For the limitation of the text box data it can show easily.



So user can know easily the limitation of letters. It can show the letter count to user if there has the limitaion of the letter. It use in comment session or the other place in the short description place.

String concat()

This method is to combine the string value each other. And return the value of string with the value of string which placed in it's parameter to the end of the first string.

```
public class Test {

    public static void main(String args[]) {
        String s = "Strings are immutable";
        s = s.concat(" all the time");
        System.out.println(s);
    }
}
```

There is an example of code. You will see a first strin which announce with the variable. The string is doesn't finish. So if you want to add use concat method to combine with the first string. If you use concat you need parameter of the string. And if you out put string with concat you will see combine of each string as shown in photo.

```
Strings are immutable all the time
```

So if you want to combine each string you can use concat method.

As a realworld example, it can use with database. It is impossible of each string to combine from the database table in the program. It can use concat mentod to combine data from the database as a string. You can accept database data with string from the different table. So those data will combine as one strin with different database table. So it can combine database data in a progrma by using concat method.

String matches()

It check the string value with the regular expression with will placed in the patameter. It returns the true of the the string is matches with regular expression. if not it return false. It use to check the string with the regular expression like for the name only accept alpabetic not for both numbers and special character. So here is an example.

```
import java.io.*;
public class Test {

   public static void main(String args[]) {
      String Str = new String("Welcome to Tutorialspoint.com");

      System.out.print("Return Value :" );
      System.out.println(Str.matches("(.*)Tutorials(.*)"));

      System.out.print("Return Value :" );
      System.out.println(Str.matches("Tutorials"));

      System.out.print("Return Value :" );
      System.out.println(Str.matches("Welcome(.*)"));

   }
}
```

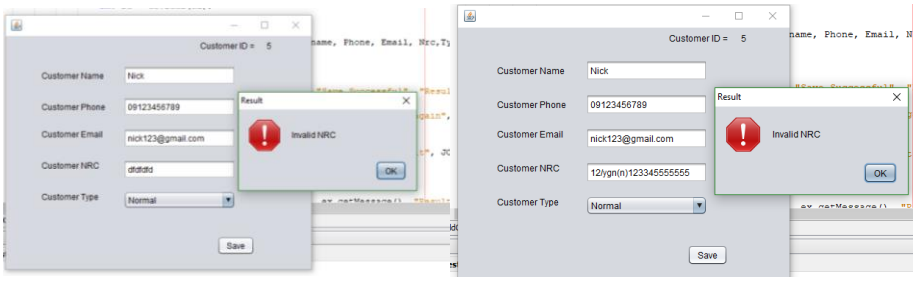
There is a string with include the matches method with the regax of (.*). That meanks it can match with the word with has in the regular expression. The parameter must contain regular expression in this match method. Because it need to check with the regex. If the String contain and match with the regular expression. It will return ture as shown in picture.

```
Return Value :true
Return Value :false
Return Value :true
```

As a realworld example it use to check the input value of the test field in the program. For example NRC input text box. It can input number, characters and special character. It should be “12/YGN(n)123456” with the correct format. So it accept all format but it wrong. So we have to check with the real world format as I shown. So here is an example.

```
        if(CNrc.trim().length()>0&&CNrc.matches("\\d{1,2}/\\D{1,3}\\(n\\)\\d{6}")){
    this.CNrc = CNrc;
}
else{
throw new NameException("Invalid NRC");
}
```

There is a regular expression in the parameter. If they did match with that it will show as shown in figure.



So if it's match, it will save successfully. So it use to check the input with the realworld example. So I hope you will understand about the matches method.

String trim()

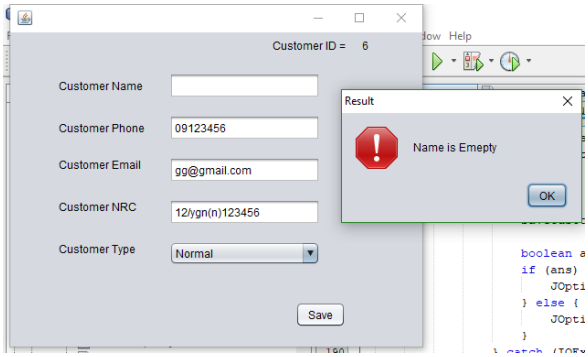
This method delete the whitespaces form beginning of the string and the end of the string. It delete the whitespaces in the leading and endd of the string. So here is clear example.

```
import java.io.*;
public class Test {

   public static void main(String args[]) {
      String Str = new String("   Welcome to Tutorialspoint.com   ");

      System.out.print("Return Value :" );
      System.out.println(Str.trim() );
   }
}
```

There is a white spaces in the beninning and at the end. So if you use that string with trim method. It will automtically delete the white spaces automatically and you will not see the whitespaces in the out put as shown in figure.

```
Return Value :Welcome to Tutorialspoint.com
```

As a realworld example it can use to check the white spaces in the input value. The input value doesn't contain spaces. For exampel in the name input text box. It shouldn't contain whitespaces from the beginning and at the end. So if it placed spaces it shows errors.



It is use to check the input values like name or others input values. So I hope you will understand about the trim method.

boolean equalsIgnoreCase(String another String)

This method check the strin with another string by ignore the characters. It mean same length same character but only difference captical letter and small letters. i only ignore capital or small letter if the string is same length and same letter.

```
public class Test {

   public static void main(String args[]) {
      String Str1 = new String("This is really not immutable!!");
      String Str2 = Str1;
      String Str3 = new String("This is really not immutable!!");
      String Str4 = new String("This IS REALLY NOT IMMUTABLE!!");
      boolean retVal;

      retVal = Str1.equals( Str2 );
      System.out.println("Returned Value = " + retVal );

      retVal = Str1.equals( Str3 );
      System.out.println("Returned Value = " + retVal );

      retVal = Str1.equalsIgnoreCase( Str4 );
      System.out.println("Returned Value = " + retVal );
   }
}
```

Here is an example for it's method. It return true if the statement are correct. equalIgnoreCase is to check the same length but only ignore the string with the capital or small or letter.

```
Returned Value = true
Returned Value = true
Returned Value = true
```

As the realworld example, in the searching place, it use. It select the data from the condition of the search box. The data can be the samll or capital. So staff or customer can't know it easily. So in this case it can use equalIgnoreCase to check the data. It will ignore the small and capital letter of each letter. So user an type small or capitial to search.

boolean equals(Object an object)

It check with the string with another string obje. It check same character, same length. It is the opposite of the equalIgnoreCase. it check with the capital and small petter. If all are correct it return ture if wrong, return false.

```
public class Test {

   public static void main(String args[]) {
      String Str1 = new String("This is really not immutable!!");
      String Str2 = Str1;
      String Str3 = new String("This is really not immutable!!");
      boolean retVal;

      retVal = Str1.equals( Str2 );
      System.out.println("Returned Value = " + retVal );

      retVal = Str1.equals( Str3 );
      System.out.println("Returned Value = " + retVal );
   }
}
```

So there is an example, which contain the two strin with the letter. If it check the same length with the same letter condition as shown in upper fingure. If it's true it return the true as shown in figure.

```
Returned Value = true
Returned Value = true
```

As an real world it use to match the password like retype- password or comfirm password. Those must have same length and same character. To idendify correct or not. So if you use equalIgnoreCase user can't login any more. This case must use only equals method to check. It use password to check for the retype password.

```
Enter Password
ggwp
Retype Password
GGwp
Password Doesn't Match
BUILD SUCCESSFUL (total time: 7 seconds)
```

```
Enter Password
ggwp
Retype Password
ggwp
Password Match
BUILD SUCCESSFUL (total time: 5 seconds)
```

So I hope you will understand about the equals method.

String toUpperCase()

This method is to change the all string value to the capital letter. It returns the strings by converting strings to the uppercase. It is easy to understand and use. Becaue it change the characters to the uppercase. Here is an coding example.

```java
import java.io.*;
public class Test {

   public static void main(String args[]) {
      String Str = new String("Welcome to Tutorialspoint.com");

      System.out.print("Return Value :" );
      System.out.println(Str.toUpperCase() );
   }
}
```

There is an example. A string is announce and it will used by toUpperCase method. So that that string will convert to the all uppercase character expect numberic and special character.

```
Return Value :WELCOME TO TUTORIALSPOINT.COM
```

As an realworld example, if you want to change same data but for the capital letter, you can use this method. For example, NRC input. A policy change, from they use samll letter to reprresent their citizen. But now as a new policy, all business should change to upper case as a standard. So they don't need to change all system. So with the program they select and accept as strings. And they can use toUpperCase and that new string will update in database. So they don't need to change whole system with the new polocy.

String toLowerCase()

This method is opposite of the toUpperCase method. Because it convert to the lowercase. And it returs a string by converting to lowercase. So here is an clear example.

```
import java.io.*;
public class Test {

    public static void main(String args[]) {
        String Str = new String("Welcome to Tutorialspoint.com");

        System.out.print("Return Value :" );
        System.out.println(Str.toUpperCase() );
    }
}
```

There is an example. A string is announce and it will used by toLowerCase method. So that that string will convert to the all lowercase character expect numberic and special character.

```
Return Value :WELCOME TO TUTORIALSPOINT.COM
```

As an realworld example is with the toUppercase method example. As they convert string to lower case., so they will have same example in this session.

String toString()

This method convert other variable to the string object. It doesn't have any parameter. It's return value is return string itself.

```
import java.io.*;
public class Test {

   public static void main(String args[]) {

            int n=123;
            String nn=Integer.toString(n);
            System.out.println(nn);
   }
}
```

If you want to change integer value to string value. You can use toString. It will change integer to string object. So it can accept string value.

As an realworld example, you want to save data in the text file you ncan save data as a string. In your data, it might contain integer or other values like float, or boolean. In this case you can’t save data in save file. So by using toString method. All of the data are convert to the string value. So you can save it in the text file.

String format()

It returns a string value as a format string with the agreement of their rules like \t,\d\n. It make the formatted output of the string. So here is a clear example with coding.

```
String name="sonoo";
String sf1=String.format("name is %s",name);
String sf2=String.format("value is %f",32.33434);
String sf3=String.format("value is %32.12f",32.33434);/


System.out.println(sf1);
```

There are string with the formatted strings. Unlink print method. Print method can't use that formatted sign. So if you want to use that signs. You should use format method.

As a real world example, it is a short form. You can use it to output with the formative like tabs, next line to clear to see the strings values. If you want to put the output with formatted, you can use format method.

String join ()

It returns string value with join of each string in the parameter. It has parameter which will be a character to join each string value. The first parameter will be a character which make know that string is join. Here is clear code example.

```
public class StringJoinExample{
public static void main(String args[]){
String joinString1=String.join("-","welcome","to","javatpoint");
System.out.println(joinString1);
```

It will join which has in the parameter and separate with the first value of the parameter. If it's output, it will join together. With using separating value of the parameter. As shown in photo.

```
welcome-to-javatpoint
```

As an real world example, it use to joint data from the database. It is same with concat method. But unlike method, it can easily know the separating because of the parameter. So it is an same example with the concat method.