**String handling in java**

Generally, String is a sequence of characters. But in Java, string is an object that represents a sequence of characters. The java.lang.String class is used to create a string object.

**Creating a string object**

There are two ways to create String object:

1. By string literal
2. By new keyword

### 1) String Literal

Java String literal is created by using double quotes. For Example:

### String str1 = "BeginnersBook";

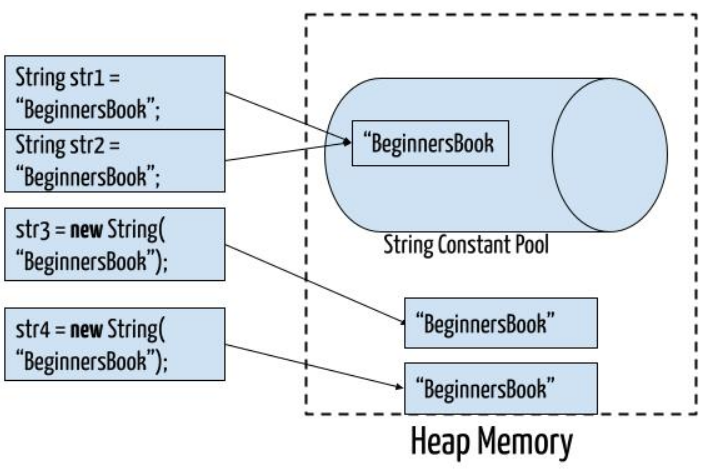
### String str2 = "BeginnersBook";

### 

### 2) By new keyword

String str3 = new String("BeginnersBook");

String str4 = new String("BeginnersBook");



**Java String vs new String**

class JavaExample

{

public static void main(String args[])

{

//creating string using string literal

String s1 = "SKBRCOLLEGE";

String s2 = " SKBRCOLLEGE ";

//creating strings using new keyword

String s3 = new String("SKBRCOLLEGE ");

String s4 = new String("SKBRCOLLEGE ");

if(s1 == s2){

System.out.println("String s1 and s2 are equal");

}else{

System.out.println("String s1 and s2 are NOT equal");

}

if(s3 == s4){

System.out.println("String s3 and s4 are equal");

}else{

System.out.println("String s3 and s4 are NOT equal");

}

}

}

**Example:**

public class JavaExample {

public static void main(String[] args) {

String str = "Hello"; //creating using literal

String str2 = new String("Hello"); //using new keyword

if(str.equals(str2)){

System.out.println("Strings str and str2 are equal");

}else{

System.out.println("Strings str and str2 are NOT equal");

}

}

}

**StringExample.java**

public class StringExample{

public static void main(String args[]){

String s1="java";

char ch[]={'s','t','r','i','n','g','s'};

String s2=new String(ch);//converting char array to string

String s3=new String("example");//creating Java string by new keyword

System.out.println(s1);

System.out.println(s2);

System.out.println(s3);

}}

**Example: Displaying first and last character of a String**

public class JavaExample {

public static void main(String[] args) {

String str = "this is skbr college";

//finding length of the string using length() method.

int len = str.length();

// First character of the string

System.out.println("First character: "+ str.charAt(0));

// Last character

System.out.println("Last character: "+ str.charAt(len-1));

}

}

**Example 5: String concatenation**

public class JavaExample {

public static void main(String[] args) {

String str = "Welcome";

String str2 = "Home";

System.out.println(str.concat(" ").concat(str2)); //

}

}

**Methods in Strings:**

|  |  |
| --- | --- |
| 1 | [char charAt(int index)](https://www.javatpoint.com/java-string-charat) |
| 2 | [int length()](https://www.javatpoint.com/java-string-length) |
| 3 | [static String format(String format, Object... args)](https://www.javatpoint.com/java-string-format) |
| 4 | [static String format(Locale l, String format, Object... args)](https://www.javatpoint.com/java-string-format) |
| 5 | [String substring(int beginIndex)](https://www.javatpoint.com/java-string-substring) |
| 6 | [String substring(int beginIndex, int endIndex)](https://www.javatpoint.com/java-string-substring) |
| 7 | [boolean contains(CharSequence s)](https://www.javatpoint.com/java-string-contains) |
| 8 | [static String join(CharSequence delimiter, CharSequence... elements)](https://www.javatpoint.com/java-string-join) |
| 9 | [static String join(CharSequence delimiter, Iterable<? extends CharSequence> elements)](https://www.javatpoint.com/java-string-join) |
| 10 | [boolean equals(Object another)](https://www.javatpoint.com/java-string-equals) |
| 11 | [boolean isEmpty()](https://www.javatpoint.com/java-string-isempty) |
| 12 | [String concat(String str)](https://www.javatpoint.com/java-string-concat) |
| 13 | [String replace(char old, char new)](https://www.javatpoint.com/java-string-replace) |
| 15 | [static String equalsIgnoreCase(String another)](https://www.javatpoint.com/java-string-equalsignorecase) |
| 16 | [String[] split(String regex)](https://www.javatpoint.com/java-string-split) |
| 17 | [String[] split(String regex, int limit)](https://www.javatpoint.com/java-string-split) |
| 18 | [String intern()](https://www.javatpoint.com/java-string-intern) |
| 19 | [int indexOf(int ch)](https://www.javatpoint.com/java-string-indexof) |
| 20 | [int indexOf(int ch, int fromIndex)](https://www.javatpoint.com/java-string-indexof) |
| 21 | [int indexOf(String substring)](https://www.javatpoint.com/java-string-indexof) |
| 22 | [int indexOf(String substring, int fromIndex)](https://www.javatpoint.com/java-string-indexof) |
| 23 | [String toLowerCase()](https://www.javatpoint.com/java-string-tolowercase) |
| 24 | [String toUpperCase()](https://www.javatpoint.com/java-string-touppercase) |
| 25 | [String trim()](https://www.javatpoint.com/java-string-trim) |
| 26 | [static String valueOf(int value)](https://www.javatpoint.com/java-string-valueof) |

**public** **class** CharAtExample{

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

String name="SKBR COLLEGE";

**char** ch=name.charAt(4);//returns the char value at the 4th index

System.out.println(ch);

}}

**public** **class** StringTrimExample{

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

String s1="  hello string   ";

System.out.println(s1+"SKBR COLLEGE");//without trim()

System.out.println(s1.trim()+"SKBR COLLEGE ");//with trim()

}}

**public** **class** StringUpperExample{

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

String s1="hello string";

String s1upper=s1.toUpperCase();

System.out.println(s1upper);

}}

**public** **class** StringLowerExample{

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

String s1=" SKBR COLLEGE”

String s1lower=s1.toLowerCase();

System.out.println(s1lower);

}}

**public** **class** IndexOfExample{

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

String s1="this is index of example";

//passing substring

**int** index1=s1.indexOf("is");//returns the index of is substring

**int** index2=s1.indexOf("index");//returns the index of index substring

System.out.println(index1+"  "+index2);//2 8

//passing substring with from index

**int** index3=s1.indexOf("is",4);

**int** index4=s1.indexOf('s');//returns the index of s char value

System.out.println(index4);//3

}}

**public** **class** ReplaceExample1{

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

String s1="skbr is a very good college";

String replaceString=s1.replace('r','e');

System.out.println(replaceString);

}}

**public** **class** ConcatExample{

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

String s1="java string";

s1.concat("is immutable");

System.out.println(s1);

s1=s1.concat(" is immutable so assign it explicitly");

System.out.println(s1);

}}

**class** ContainsExample{

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

String name="what do you know about me";

System.out.println(name.contains("do you know"));

//System.out.println(name.append(“varma”));// strings are immutable so it cannot //be changed

System.out.println(name.contains("hello"));

}}

**public** **class** SubstringExample{

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

String s1="javatpoint";

System.out.println(s1.substring(2,4));//

System.out.println(s1.substring(,7));//

}}

**public** **class** StringJoinExample2 {

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

        String date = String.join("/","25","06","2018");

        System.out.print(date);

        String time = String.join(":", "12","10","10");

        System.out.println(" "+time);

    }

}

**public class** Main {

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

String myStr1 = "Hello";

String myStr2 = "";

System.out.println(myStr1.isEmpty());

System.out.println(myStr2.isEmpty());

}

}

**String Buffer class in java:**

Stringbuffer is a class in java that represents a mutable sequence of characters. It provides an alternative to the immutable string class , allowing you to modify the contents of a string without creating a new object every time.

Example:

Public class StringBufferExample{

Public static void main(String args[]){

StringBuffer sb = new StringBuffer(“varma”);

Sb.append(“ hello“);

Sb.append(“ “);

Sb.append(“ world“);

String msg = sb.toString();

System.out.println(msg);

}}

What is a mutable String?

A String that can be modified or changed is known as mutable String. StringBuffer and StringBuilder classes are used for creating mutable strings.

1) StringBuffer Class append() Method

The append() method concatenates the given argument with this String.

**StringBufferExample.java**

Play Video

**class** StringBufferExample{

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

StringBuffer sb=**new** StringBuffer("Hello ");

sb.append("Java");//now original string is changed

System.out.println(sb);//prints Hello Java

}

}

**Output:**

Hello Java

2) StringBuffer insert() Method

The insert() method inserts the given String with this string at the given position.

**StringBufferExample2.java**

**class** StringBufferExample2{

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

StringBuffer sb=**new** StringBuffer("Hello ");

sb.insert(1,"Java");//now original string is changed

System.out.println(sb);//prints HJavaello

}

}

**Output:**

HJavaello

3) StringBuffer replace() Method

The replace() method replaces the given String from the specified beginIndex and endIndex.

**StringBufferExample3.java**

**class** StringBufferExample3{

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

StringBuffer sb=**new** StringBuffer("Hello");

sb.replace(1,3,"Java");

System.out.println(sb);//prints HJavalo

}

}

**Output:**

HJavalo

4) StringBuffer delete() Method

The delete() method of the StringBuffer class deletes the String from the specified beginIndex to endIndex.

**StringBufferExample4.java**

**class** StringBufferExample4{

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

StringBuffer sb=**new** StringBuffer("Hello");

sb.delete(1,3);

System.out.println(sb);//prints Hlo

}

}

**Output:**

Hlo

5) StringBuffer reverse() Method

The reverse() method of the StringBuilder class reverses the current String.

**StringBufferExample5.java**

**class** StringBufferExample5{

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

StringBuffer sb=**new** StringBuffer("Hello");

sb.reverse();

System.out.println(sb);//prints olleH

}

}

**Output:**

olleH

### 6) StringBuffer capacity() method

### //default capacity 16

## public class JavaExample{

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

## StringBuffer sb=new StringBuffer();

## //default capacity 16

## System.out.println("Current Capacity: "+sb.capacity());

## //appended string is 13 chars, can be adjusted in current

## //capacity so no capacity increase

## sb.append("BeginnersBook");

## System.out.println("Current Capacity: "+sb.capacity());

## Sb.ensureCapacity(18);

## System.out.println("Current Capacity: "+sb.capacity()); //34

## }

## }

**Other examples:**

## **Convert String to int using Integer.parseInt(String)**

public class JavaExample{

public static void main(String args[]){

String str="123";

int inum = 100;

int inum2 = Integer.parseInt(str);

int sum = inum+inum2;

System.out.println("Result is: "+sum);

}

}

## **Convert int to String using String.valueOf()**

public class JavaExample {

public static void main(String args[]) {

int ivar = 111;

String str = String.valueOf(ivar);

System.out.println("String is: "+str);

System.out.println(“555”+str);//555111

}

}

### convert String to double using parseDouble(String)

Convert double to string using String.valueOf(double) method