Sir C.R.Reddy (A) College

Aided & Autonomous

ELURU-534 007, W.G.DIST. A.P., INDIA

Affiliated to Adikavi Nannaya University, Rajamahendravaram

THRICE ACCREDITED AT 'A' LEVEL BY NAAC, BANGALURU :: COLLEGE WITH POTENTIAL FOR EXCELLENCE

An ISO 9001:2015 Certified Institution

Core JAVA

Siva Krishna Siripurapu

Siripurapusivakrishna29@gmail.com

Department of Computer Science

JAVA – STRINGS

Strings, which are widely used in Java programming, are a sequence of characters. In Java programming language, strings are treated as objects.

The Java platform provides the String class to create and manipulate strings.

Creating Strings

The most direct way to create a string is to write −

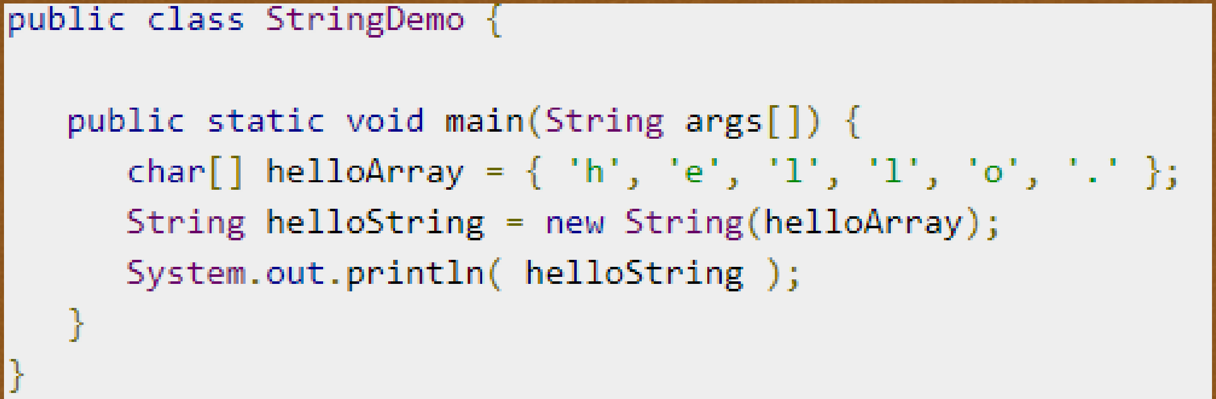
→ String greeting = "Hello world!";

Whenever it encounters a string literal in your code, the compiler creates a String object with its value in this case, "Hello world!'.

As with any other object, you can create String objects by using the new keyword and a constructor. The String class has 11 constructors that allow you to provide the initial value of the string using different sources, such as an array of characters.

JAVA – STRINGS

Example



```java
public class StringDemo {

   public static void main(String args[]) {
      char[] helloArray = { 'h', 'e', 'l', 'l', 'o', '.' };
      String helloString = new String(helloArray);
      System.out.println( helloString );
   }
}
```

This will produce the following result –

Output → hello.

Note – The String class is immutable, so that once it is created a String object cannot be changed. If there is a necessity to make a lot of modifications to Strings of characters, then you should use String Buffer & String Builder Classes.

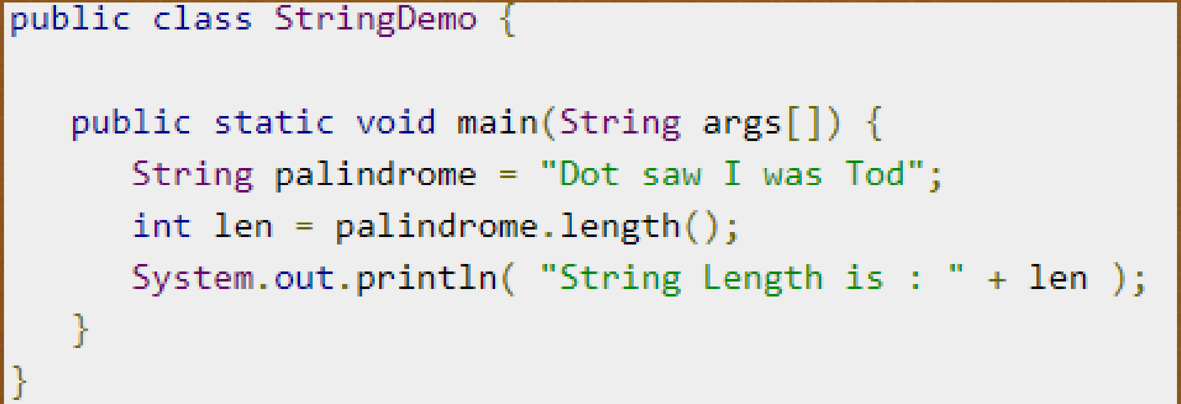
JAVA – STRINGS

String Length

Methods used to obtain information about an object are known as accessor methods. One accessor method that you can use with strings is the length() method, which returns the number of characters contained in the string object.

The following program is an example of length(), method String class.

Example



```
public class StringDemo {

   public static void main(String args[]) {
      String palindrome = "Dot saw I was Tod";
      int len = palindrome.length();
      System.out.println( "String Length is : " + len );
   }
}
```

Output String Length is : 17

JAVA – STRINGS

Concatenating Strings

The String class includes a method for concatenating two strings −

```
string1.concat(string2);
```

This returns a new string that is string1 with string2 added to it at the end.

You can also use the concat() method with string literals, as in −

```
"My name is ".concat("Zara");
```

Strings are more commonly concatenated with the + operator, as in −

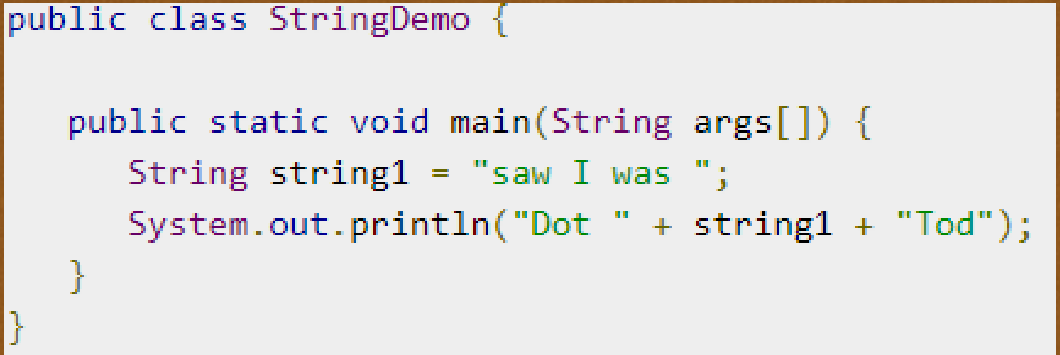
```
"Hello," + " world" + "!“
```

which results in −

```
"Hello, world!"
```

JAVA – STRINGS

Example



```
public class StringDemo {

   public static void main(String args[]) {
      String string1 = "saw I was ";
      System.out.println("Dot " + string1 + "Tod");
   }
}
```

This will produce the following result −

Output → Dot saw I was Tod

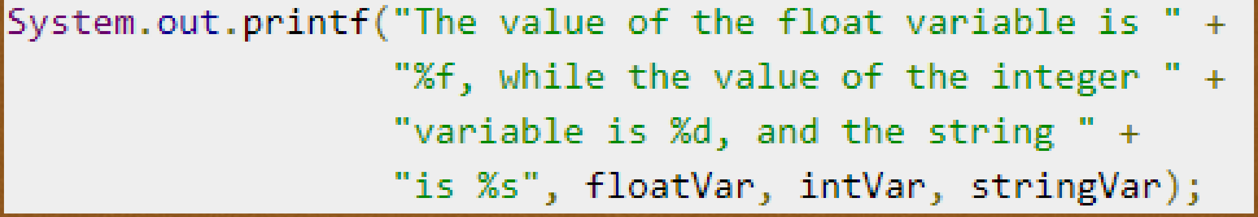
Creating Format Strings

You have printf() and format() methods to print output with formatted numbers. The String class has an equivalent class method, format(), that returns a String object rather than a PrintStream object.

Using String's static format() method allows you to create a formatted string that you can reuse, as opposed to a one-time print statement. For example, instead of −

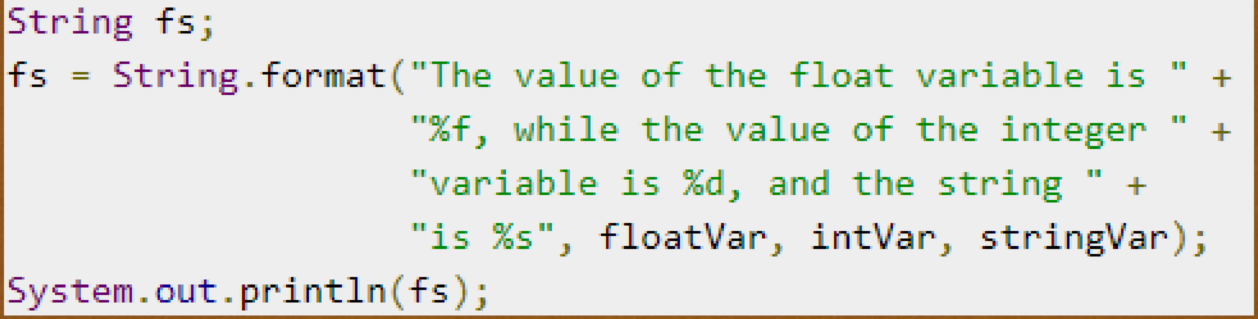
JAVA – STRINGS

Example



```
System.out.printf("The value of the float variable is " +
                  "%f, while the value of the integer " +
                  "variable is %d, and the string " +
                  "is %s", floatVar, intVar, stringVar);
```

You can write –



```
String fs;
fs = String.format("The value of the float variable is " +
                   "%f, while the value of the integer " +
                   "variable is %d, and the string " +
                   "is %s", floatVar, intVar, stringVar);
System.out.println(fs);
```

JAVA – STRINGS

String Methods

Here is a few list of methods supported by String class –

S.No.	Method & Description
1	char charAt(int index) Returns the character at the specified index.
2	int compareTo(Object o) Compares this String to another Object.
3	int compareTo(String anotherString) Compares two strings lexicographically.
4	int compareToIgnoreCase(String str) Compares two strings lexicographically, ignoring case differences.
5	String concat(String str) Concatenates the specified string to the end of this string.

Thank You !