

(Transcript) Java Strings

Rex
Jones II

Java Strings

In Java, a String is class that represents a sequence of characters. Therefore, the word or the phrase is surrounded by double quotes and contains several methods. We can create a string by writing `String username = new String ()` then pass in a literal like `"SoftwareEngineer34"`. However, most engineers create their own string by writing `String` with an object like `name = to a value such as "Rex Jones II"`.

Let's print both String values.

```
sysout("Username: " + username + "\n" + "Name: " + name)  
}
```

```
public class StringDemo {  
  
    public static void main(String[] args) {  
        String username = new String ("SoftwareEngineer34");  
        String name = "Rex Jones II";  
  
        System.out.println("Username: " + username + "\n" + "Name: " + name);  
    }  
}
```

Let's Run. We see Username: SoftwareEngineer34 and Name: Rex Jones II

Ok. Let's look at some of the methods for String. Hover over the String class then click Open Attached Javadoc in a Browser. First thing we see is Class String. Scroll to the methods section. There are a lot of methods: charAt, concat, contains, equals, equalsIgnoreCase, format, indexOf, lastIndexOf, length, replace.

Go back to our program and use some of these methods. To convert all of the letters for name to lowercase. We write

sysout("Name (Lowercase): " + name.) and we see all of the same methods we just saw in JavaDocs. Select toLowerCase.

There's also a method to convert all of the letters to uppercase.

Copy and Paste and change Lower Case to UpperCase. Change .toLowerCase to .toUpperCase.

```
System.out.println("Name (Lowercase): " + name.toLowerCase());  
System.out.println("Name (Uppercase): " + name.toUpperCase());
```

Let's Run. The Console shows my name in all lowercase letters and uppercase letters.

Next, let's look at the length method. It will return the number of characters in my name. Should be 12. Copy and paste and change Lower Case to length. Change name.toLowerCase to name.length()

```
System.out.println("Name (Length): " + name.length());
```

Let's Run. We see 12 for the length.

There's a way for us to take a portion of a string using the substring method. For example, I can print only the First Name by writing substring. There are 2 methods. One method receives 1 argument and the other method receives 2 arguments. Let's select 2 arguments: Begin index will start at 0. The end

Rex Jones II

index is tricky. A person may think the end index should be 2 because x is the 2nd index. Rex – 0, 1, 2 but we need the end of the index which is 3. Therefore, we enter 3 for the ending index.

```
System.out.println("Name (substring): " + name.substring(0, 3));
```

Let's Run.

A popular method is charAt. It returns the value of our index. Let's get the first letter. sysout("Name (First Character): " + name.charAt(0)) and Run.

```
System.out.println("Name (First Character): " + name.charAt(0));
```

You know how the length is 12. If we enter an incorrect index such as 13 then we will get an exception: StringIndexOutOfBoundsException – String index out of range: 13. It's out of range because 12 is the length. Change it back to 0.

Let's look at a few more examples. How about contains? Contains returns true or false if the string has a certain value. sysout("Name (Contains Jones): " + name.contains("Jones")).

```
System.out.println("Name (Contains Jones): " + name.contains("Jones"));
```

The Console returns true

Another popular method is equals

sysout("Name (Equals REX JONES II): ") In all capital letters + name.equals("REX JONES II")).

```
System.out.println("Name (Equals REX JONES II): " + name.equals("REX JONES II"));
```

Let's Run. This time, the Console returns false because the string is not in all capital letters. If you want to ignore the case then we can write equalsIgnoreCase would have returned true.

One more, how about String.format? This method returns a formatted string using a string and an argument. The argument will be String nickname = "James". I was actually born James Allen Jones but it was changed to Rex Allen Jones II about 2 or 3 days after I was born. Sometimes, I still go by James Jones.

sysout(String.format()). The string is "My Nickname Is %s". Percent s, let's the compiler know to print out a string and the argument is nickname.

```
String nickName = "James";  
System.out.println(String.format("My Nickname Is %s", nickName));
```

Our value will be concatenated because 2 strings are joined together. The plus signs are a form of concatenation because it also joins 2 or more strings. Know what, that's it for String and we see My Nickname Is James.

```
Username: SoftwareEngineer34  
Name: Rex Jones II  
Name (Lowercase): rex jones ii  
Name (Uppercase): REX JONES II  
Name (Length): 12  
Name (substring): Rex  
Name (First Character): R  
Name (Contains Jones): true  
Name (Equals REX JONES II): false  
My Nickname Is James
```

That's it for Strings. I'll make sure to add a link to the transcript and source code in the description. You can follow me on Twitter, connect with me on LinkedIn, and subscribe to my YouTube channel. I still have more content to release so I will see you next time.

Social Media Contact

- YouTube <https://www.youtube.com/c/RexJonesII/videos>
- Twitter <https://twitter.com/RexJonesII>
- LinkedIn <https://www.linkedin.com/in/rexjones34/>
- GitHub <https://github.com/RexJonesII/Free-Videos>
- Facebook <http://facebook.com/JonesRexII>