

String Interpolation with 'raw'

In the following lesson, you will be introduced to the 'raw' string interpolator.

We'll cover the following ^

- The raw String Interpolator
 - Syntax

The `raw` string interpolator is similar to the `s` interpolator. The only difference is that `raw` doesn't recognize [character literal](#) escape sequences.

The `raw` String Interpolator

For string interpolation with `raw`, we prepend the word **raw** to any string literal. This allows us to print characters symbols within strings.

This code requires the following environment variables to execute: ^

LANG C.UTF-8

```
println("Without Raw:\nFirst\nSecond")
println(raw"With Raw:\nFirst\nSecond")
```



Syntax

`raw"OptionalStringEscapeSequenceOptionalString"`

In the example above, `raw"With Raw:\nFirst\nSecond"` is the processed string literal. `\n` is the escape sequence which is used to take the string to the next line. When we don't use the `raw` interpolator, the output is as expected; the word **First** and **Second** are on different lines. However, when we do use the `raw` interpolator, the compiler prints the escape sequence `\n` as is.

You can try the above code with any escape sequence you want.

This brings our discussion on string interpolation to an end. Let's now look at some common `String` problems and see how we can solve them using Scala.