

Working with Strings in Scala



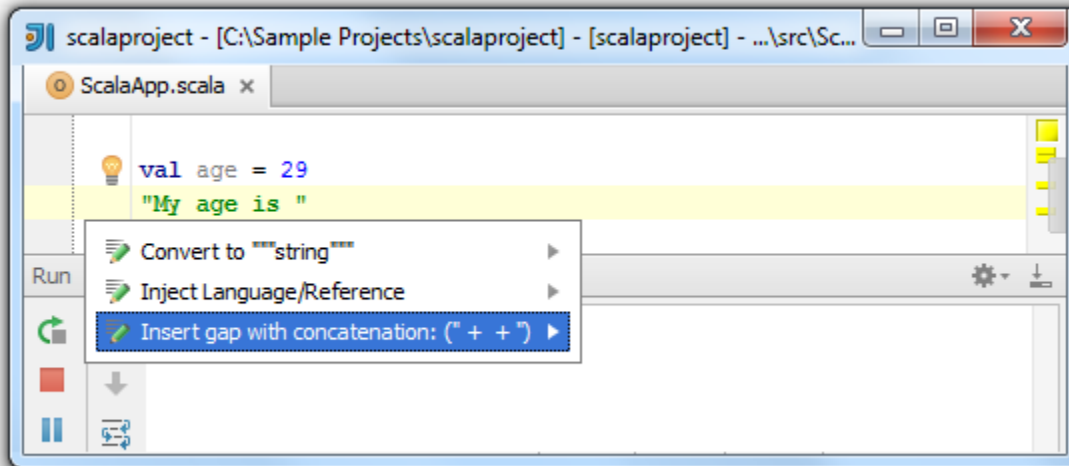
Redirection Notice

This page will redirect to <https://www.jetbrains.com/help/idea/working-with-strings-in-scala.html>.

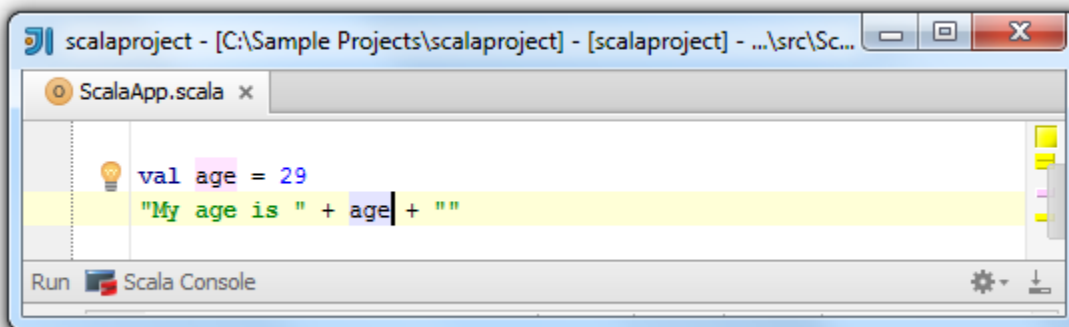
IntelliJ IDEA lets you add different intentions for strings and perform different actions with strings in Scala. IntelliJ IDEA also supports the Scala multi-line strings feature. This feature lets you format literals and helps your code to look more orderly.

1. How to Insert Gap with Concatenation into a String

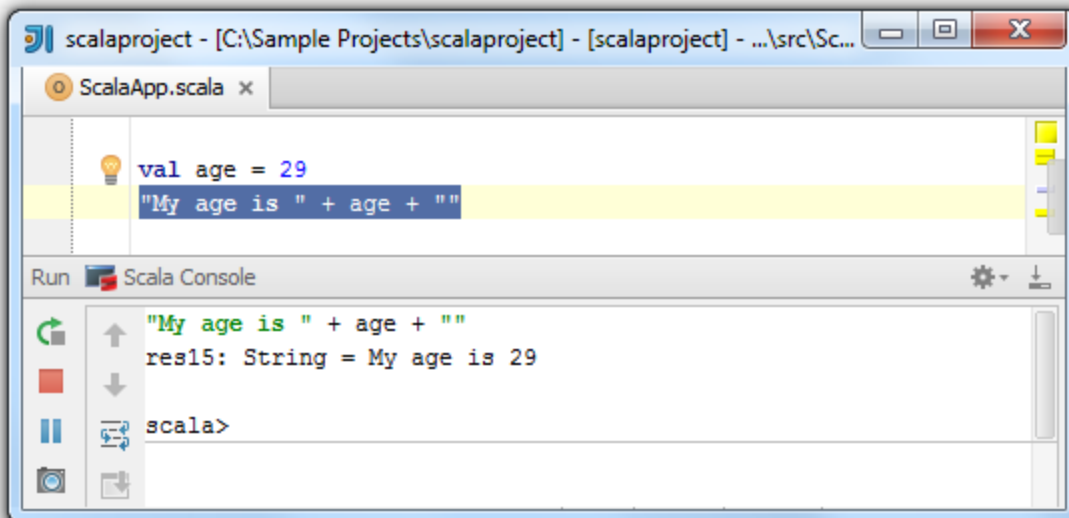
IntelliJ IDEA lets you insert a gap with concatenation intention. To do that, in your editor enter a string and press `Alt+Enter`. The list of appropriate intentions opens.



Select `Insert gap with concatenation (" + + ")` and press `Enter`.

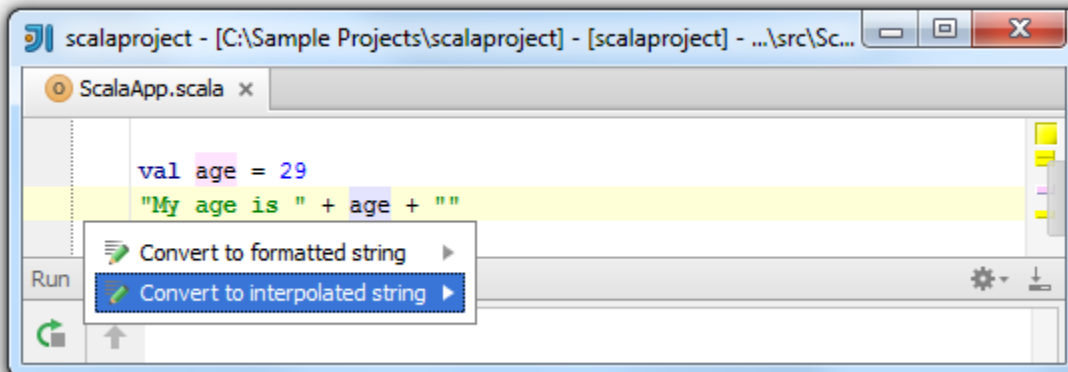


Now you can insert a value into your string. You can evaluate the result in the Scala Console window.

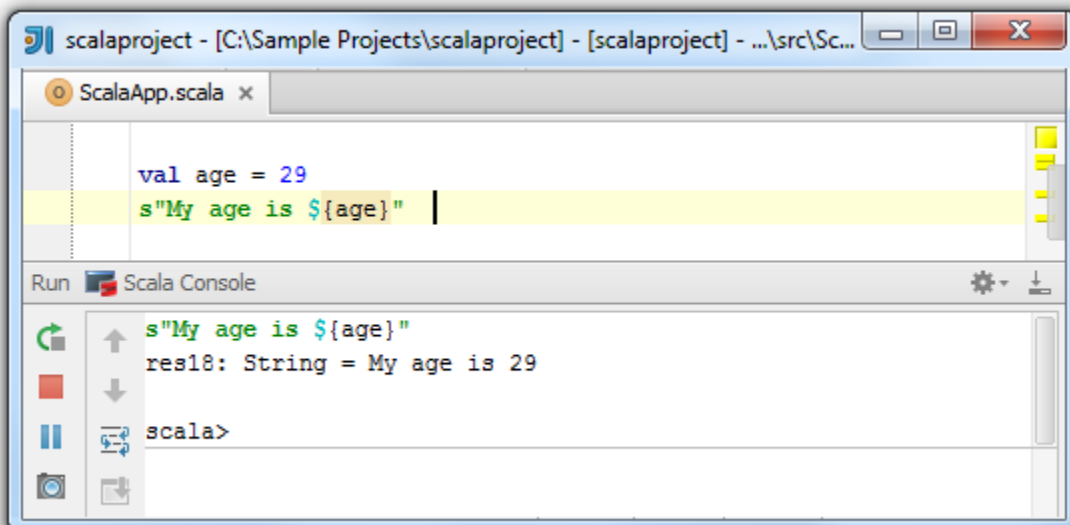


2. How to Convert a String Concatenation into Interpolated String

For the Scala version 2.10 and later you have the ability to convert a string concatenation into the interpolated string. To do that, select a value with concatenation in your string and press **Alt+Enter**. The list of appropriate intentions opens.



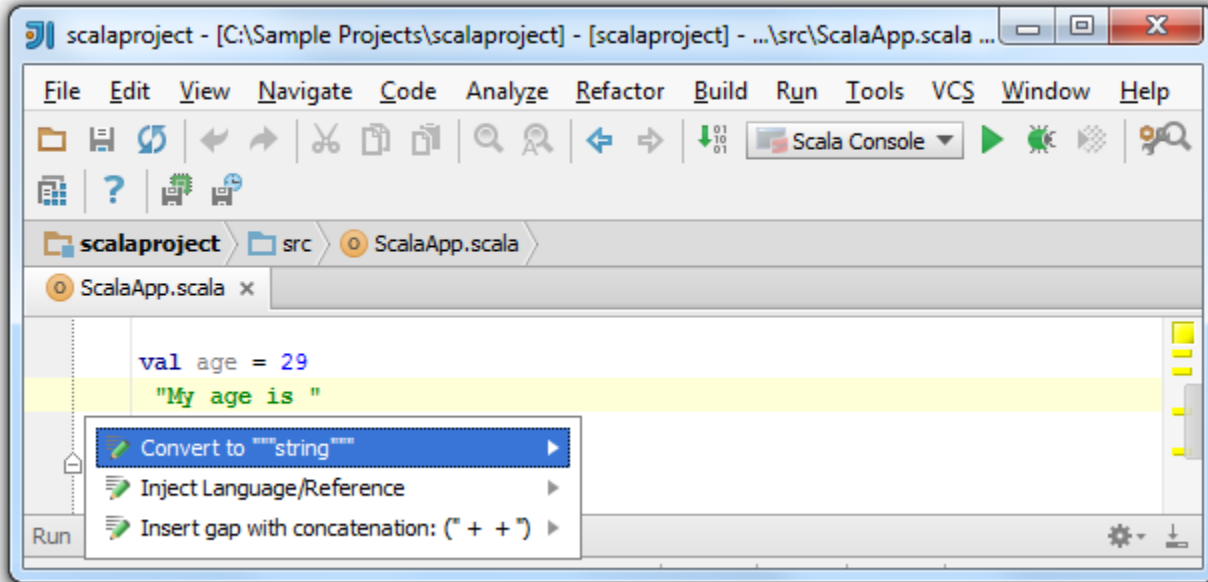
Select **Convert to interpolated string**, press **Enter** and view the result.



You can also choose an option Convert to formatted string and get your basic Java formatted string.

3. How to Convert a String into a Multi-line String and Vice Versa

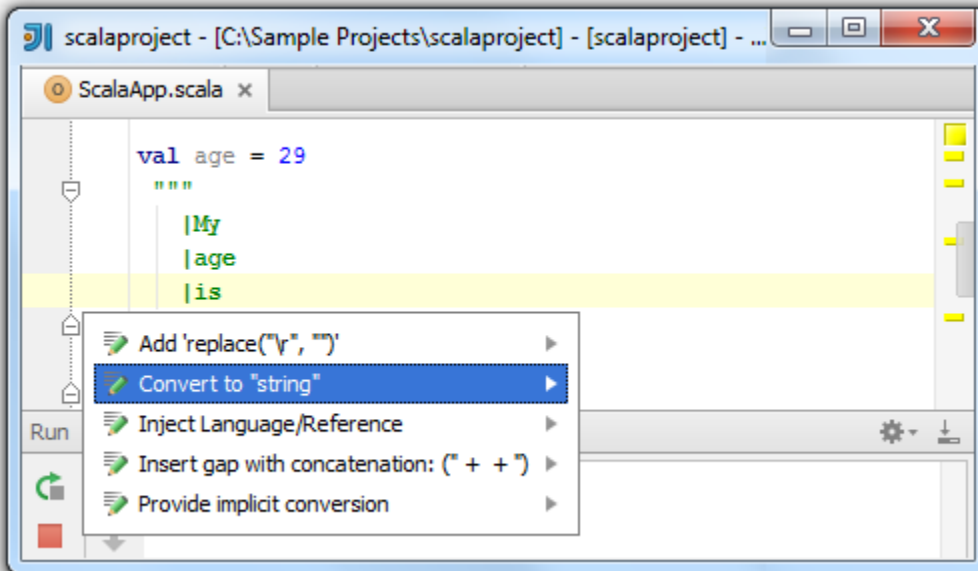
You can convert your string into a multi-line string. Simply enter your string, pressAlt+Enterand select Convert to ""string"" option in the drop-down list of intentions.



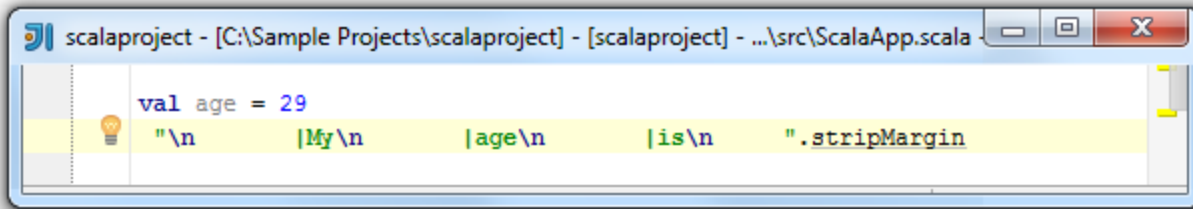
Converting to multi-line strings removes escaped sequences such as `\\` or `\n`.

You also can convert the multi-line string into the regular string.

PressAlt+Enter to open the drop-down list of intentions. Select Convert to "string" and pressEnter.

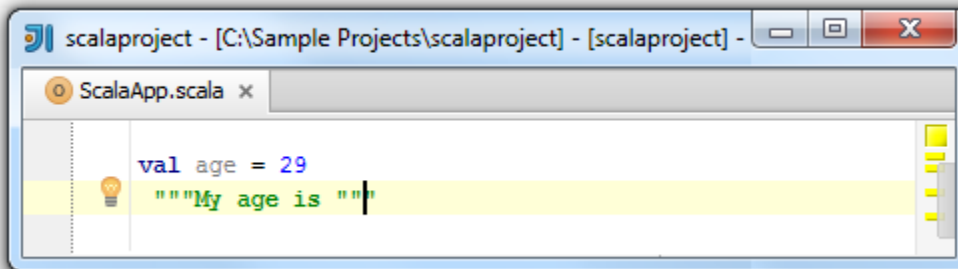


View the result.



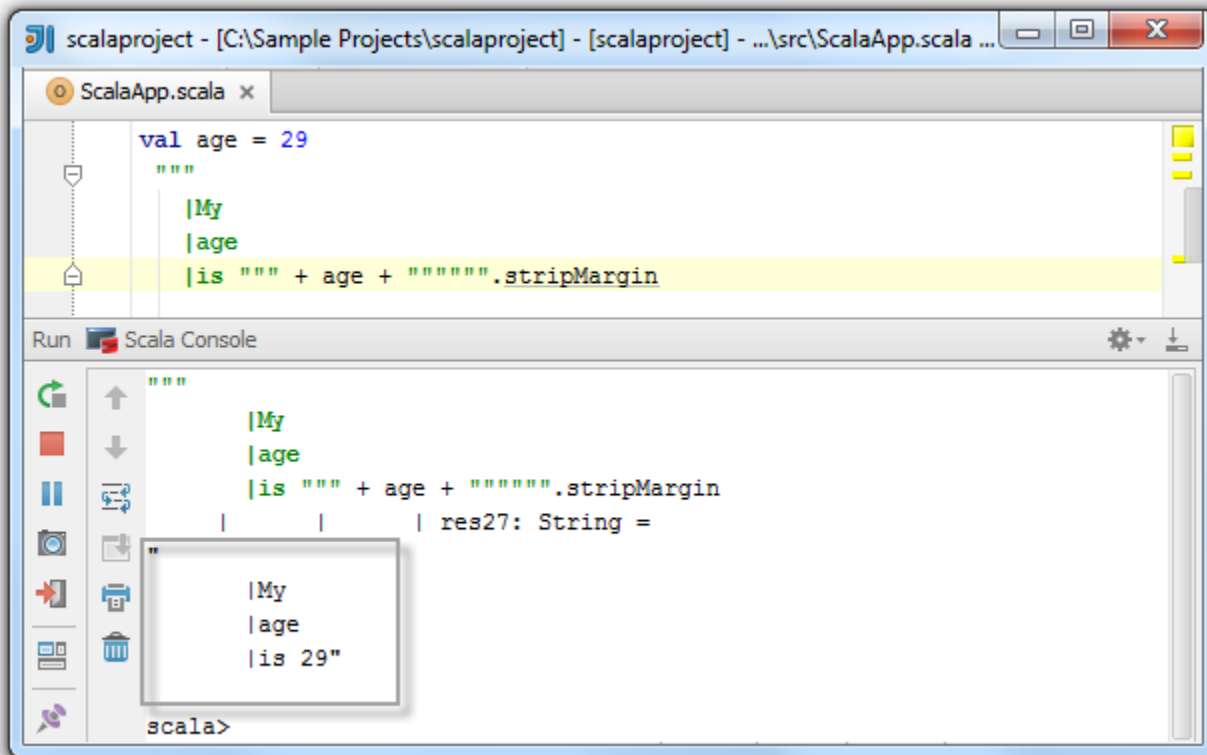
```
scalaproject - [C:\Sample Projects\scalaproject] - [scalaproject] - ...src\ScalaApp.scala
val age = 29
"""\n |My\n |age\n |is\n """.stripMargin
```

To start entering a multi-line string, simply type triple quotes in your editor.



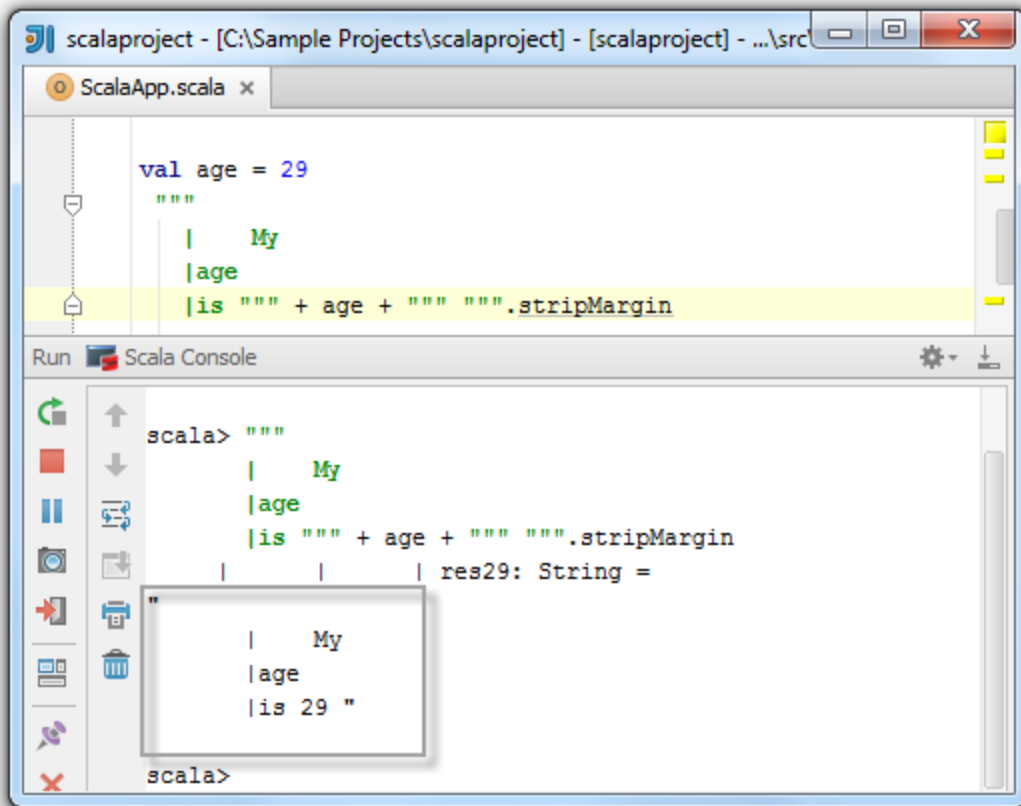
```
scalaproject - [C:\Sample Projects\scalaproject] - [scalaproject] - ...
ScalaApp.scala x
val age = 29
"""My age is """
```

If you press `Enter`, it will automatically invoke `stripMargin` method. The `stripMargin` method removes the left-hand part of a multi-line string up to a specified delimiter.



```
scalaproject - [C:\Sample Projects\scalaproject] - [scalaproject] - ...src\ScalaApp.scala ...
ScalaApp.scala x
val age = 29
"""
|My
|age
|is """ + age + """".stripMargin
Run Scala Console
"""
|My
|age
|is """ + age + """".stripMargin
| | | res27: String =
|My
|age
|is 29"
scala>
```

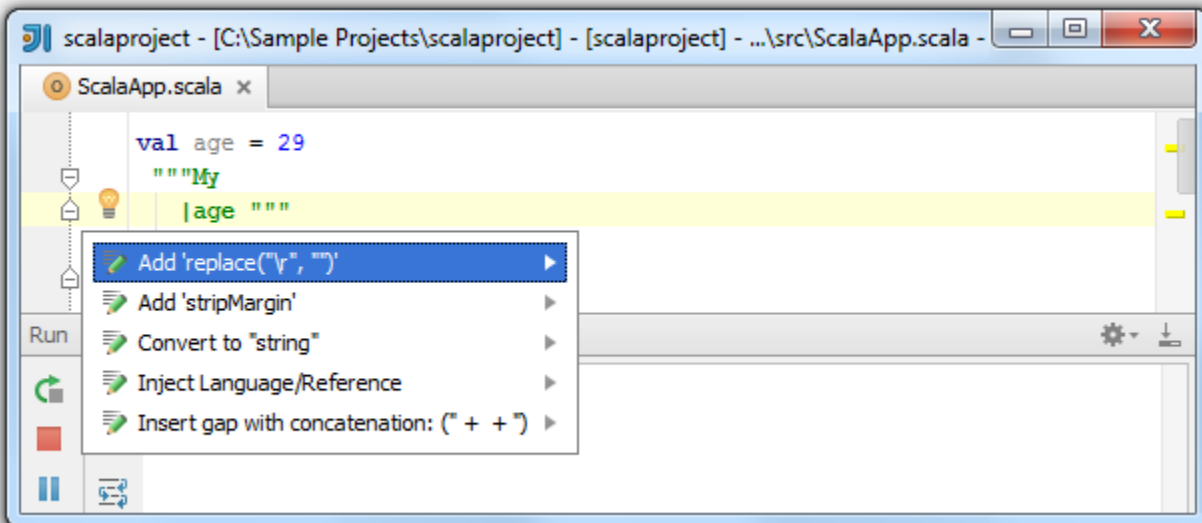
The white spaces are also preserved. Please see the following example:

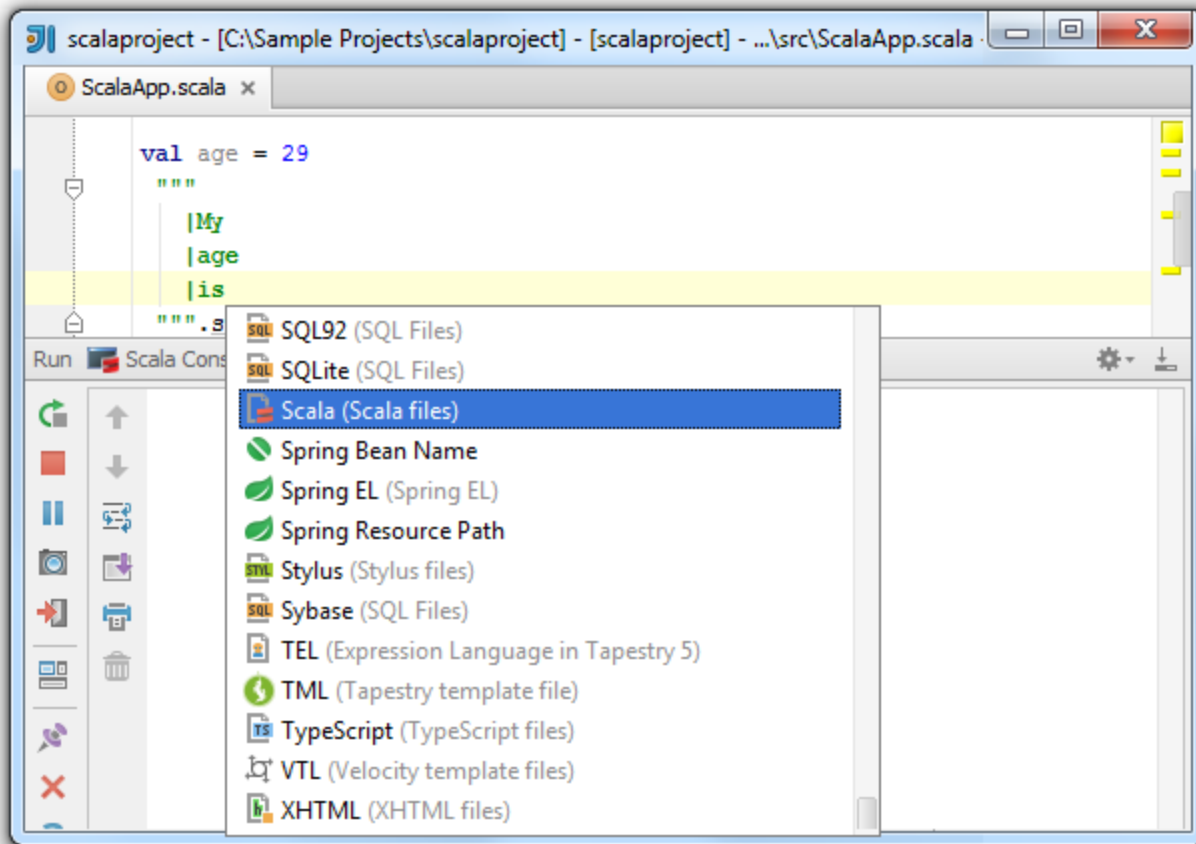


4. How to Insert a Replace Intention

The `.replace("\r", " ")` intention lets you keep the caret in the correct place on the next line in the multi-line strings regardless of what operating system you have at the moment.

Enter a multi-line string, press `Alt+Enter` and select the appropriate intention from the drop-down list.





To cancel the language injection, simply choose the Un-Inject Language/Reference intention.

6. How to Edit Multi-line Strings Settings

You can set a different format for multi-line strings' options such as Margin char indent or Margin Char. You can also disable a multi-line strings support.

To open the Multi-line strings settings, in the main menu select `File > Settings > Project Settings > Code Style > Scala`. On the Scala page select the Multi-line strings tab.

Settings

- Project Settings]
- Code Style
 - General
 - Scala**
 - Java
 - ActionScript
 - CFML
 - CoffeeScript
 - CSS
 - Gherkin
 - Groovy
 - GSP
 - HAML
 - HTML
 - JavaScript
 - JSP
 - JSPX
 - Yaml
 - ColdFusion
 - Gradle

Code Style ▶ Scala

Scheme: Default (2) [v] Manage...

Default (2)

Multi-line strings support: Insert margin char [v]

Margin char: []

Margin char indent: [2]

- Opening quotes on new line
- Keep opening quotes
- Strip/Insert margin on copy/paste

OK Cancel