New Control Syntax

Edit this page on GitHub

Scala 3 has a new "quiet" syntax for control expressions that does not rely on enclosing the condition in parentheses, and also allows to drop parentheses or braces around the generators of a `for` -expression. Examples:

```
if x < 0 then
  "negative"
else if x == 0 then
  "zero"
else
  "positive"

if x < 0 then -x else x

while x >= 0 do x = f(x)

for x <- xs if x > 0
yield x * x

for
  x <- xs
  y <- ys
do
  println(x + y)

try body
catch case ex: IOException => handle
```

The rules in detail are:

- The condition of an `if` -expression can be written without enclosing parentheses if it is followed by a `then` .
- The condition of a `while` -loop can be written without enclosing parentheses if it is followed by a `do` .

- The enumerators of a `for`-expression can be written without enclosing parentheses or braces if they are followed by a `yield` or `do`.
- A `do` in a `for`-expression expresses a `for`-loop.
- A `catch` can be followed by a single case on the same line. If there are multiple cases, these have to appear within braces (just like in Scala 2) or an indented block.

Rewrites

The Scala 3 compiler can rewrite source code from old syntax to new syntax and back. When invoked with options `-rewrite -new-syntax` it will rewrite from old to new syntax, dropping parentheses and braces in conditions and enumerators. When invoked with options `-rewrite -old-syntax` it will rewrite in the reverse direction, inserting parentheses and braces as needed.

‹ The @t... Option... ›