

[Scala 3 Reference](#) / [Language Versions](#) / [Source Compatibility](#)**LEARN****INSTALL****PLAYGROUND****FIND A LIBRARY****COMMUNITY****BLOG**

Source Compatibility

[✎ Edit this page on GitHub](#)

Scala 3 does NOT guarantee source compatibility between different minor language versions (e.g. some syntax valid in 3.x might get deprecated and then phased out in 3.y for $y > x$). There are also some syntax structures that were valid in Scala 2 but are not anymore in Scala 3. However the compiler provides a possibility to specify the desired version of syntax used in a particular file or globally for a run of the compiler to make migration between versions easier.

The default Scala language syntax version currently supported by the Dotty compiler is `3.0`. There are also other language versions that can be specified instead:

- `3.0-migration`: Same as `3.0` but with a Scala 2 compatibility mode that helps moving Scala 2.13 sources over to Scala 3. In particular, it
 - flags some Scala 2 constructs that are disallowed in Scala 3 as migration warnings instead of hard errors,
 - changes some rules to be more lenient and backwards compatible with Scala 2.13
 - gives some additional warnings where the semantics has changed between Scala 2.13 and 3.0
 - in conjunction with `-rewrite`, offer code rewrites from Scala 2.13 to 3.0.
- `future`: A preview of changes introduced in the next versions after 3.0. In the doc pages here we refer to the language version with these changes as `3.1`, but it might be that some of these changes will be rolled out in later `3.x` versions.

Some Scala 2 specific idioms will be dropped in this version. The feature set supported by this version will be refined over time as we approach its release.

- `future-migration`: Same as `future` but with additional helpers to migrate from `3.0`. Similarly to the helpers available under `3.0-migration`, these include migration warnings and optional rewrites.



There are two ways to specify a language version :

- with a `-source` command line setting, e.g. `-source 3.0-migration` .
- with a `scala.language` import at the top of a source file, e.g:

```
package p
import scala.language.`future-migration`

class C { ... }
```

Language imports supersede command-line settings in the source files where they are specified. Only one language import specifying a source version is allowed in a source file, and it must come before any definitions in that file.

Note: The [Scala 3 Migration Guide](#) gives further information to help the Scala programmer moving from Scala 2.13 to Scala 3.

[< Language...](#)

[Binary ... >](#)

