☰   🔍

Scala 3 Reference / Soft Keywords

LEARN   **INSTALL**   **PLAYGROUND**   **FIND A LIBRARY**   **COMMUNITY**

BLOG

Soft Keywords

✎ Edit this page on GitHub

A soft modifier is one of the identifiers `infix` , `inline` , `opaque` , `open` and `transparent` .

A soft keyword is a soft modifier, or one of `as` , `derives` , `end` , `extension` , `throws` , `using` , `|` , `+` , `-` , `*`

A soft modifier is treated as potential modifier of a definition if it is followed by a hard modifier or a keyword combination starting a definition (`def` , `val` , `var` , `type` , `given` , `class` , `trait` , `object` , `enum` , `case class` , `case object`). Between the two words there may be a sequence of newline tokens and soft modifiers.

Otherwise, soft keywords are treated specially in the following situations:

- `inline` , if it is followed by any token that can start an expression.
- `derives` , if it appears after an extension clause or after the name and possibly parameters of a class, trait, object, or enum definition.
- `end` , if it appears at the start of a line following a statement (i.e. definition or toplevel expression)
- `extension` , if it appears at the start of a statement and is followed by `(` or `[` .
- `using` , if it appears at the start of a parameter or argument list.
- `as` , in a renaming import clause
- `|` , if it separates two patterns in an alternative.
- `+` , `-` , if they appear in front of a type parameter.
- `*` , in a wildcard import, or it follows the type of a parameter, or if it appears in a vararg splice `x*` .

Everywhere else a soft keyword is treated as a normal identifier.

Scaladoc