

[Scala 3 Reference](#) / [Experimental](#) / [Named Type Arguments](#)**LEARN**

INSTALL

PLAYGROUND

FIND A LIBRARY

COMMUNITY

BLOG

Named Type Arguments

[✎ Edit this page on GitHub](#)

Note: This feature is implemented in Scala 3, but is not expected to be part of Scala 3.0.

Type arguments of methods can now be specified by name as well as by position.

Example:

```
def construct[Elem, Coll[_]](xs: Elem*): Coll[Elem] = ???

val xs1 = construct[Coll = List, Elem = Int](1, 2, 3)
val xs2 = construct[Coll = List](1, 2, 3)
```

Similar to a named value argument `(x = e)`, a named type argument `[X = T]` instantiates the type parameter `X` to the type `T`. Named type arguments do not have to be in order (see `xs1` above) and unspecified arguments are inferred by the compiler (see `xs2` above). Type arguments must be all named or un-named, mixtures of named and positional type arguments are not supported.

Motivation

The main benefit of named type arguments is that unlike positional arguments, you are allowed to omit passing arguments for some parameters, like in the definition of `xs2` above. A missing type argument is inferred as usual by local type inference. This is particularly useful in situations where some type arguments can be easily inferred from others.

[More details](#)[< Erased ...](#)[Named... >](#)

