

What are the limitations on abstract types?

Why do lazy values mesh better with functional programming?

Give the inner class syntax.

How do you access the instance of your outer class from inside the inner class?

What are some features of Enumeration?

What strategies does Scala provide for negating the risk of using uninitialized fields?

What do abstract `vars` create?

How do lazy vals eliminate the risk of using uninitialized values?

What is a common use of abstract types?

What is the main advantage of abstract `vals` over abstract methods?

Describe the singleton type.

Give a Scala shortcut to instantiating inner class that works outside the outer class.

What can substitute for trait parameters
(which Scala does not allow) to create trait
instances?

Why are pre-initialized fields needed?

What is a path-dependent type?

What are they subtypes of?

What is a self type?

What is its syntax?

What is its purpose?

What can be abstract in Scala?

Which need be declared abstract?

What are the types of enum values?

How do you make an enumeration?

What's the limitation of pre-initialized fields?

This is exactly like what?

Give the syntax of pre-initialized fields.

Give the initialization order for class parameters and abstract fields.

What do pre-initialized fields do?

When is this frequently useful?