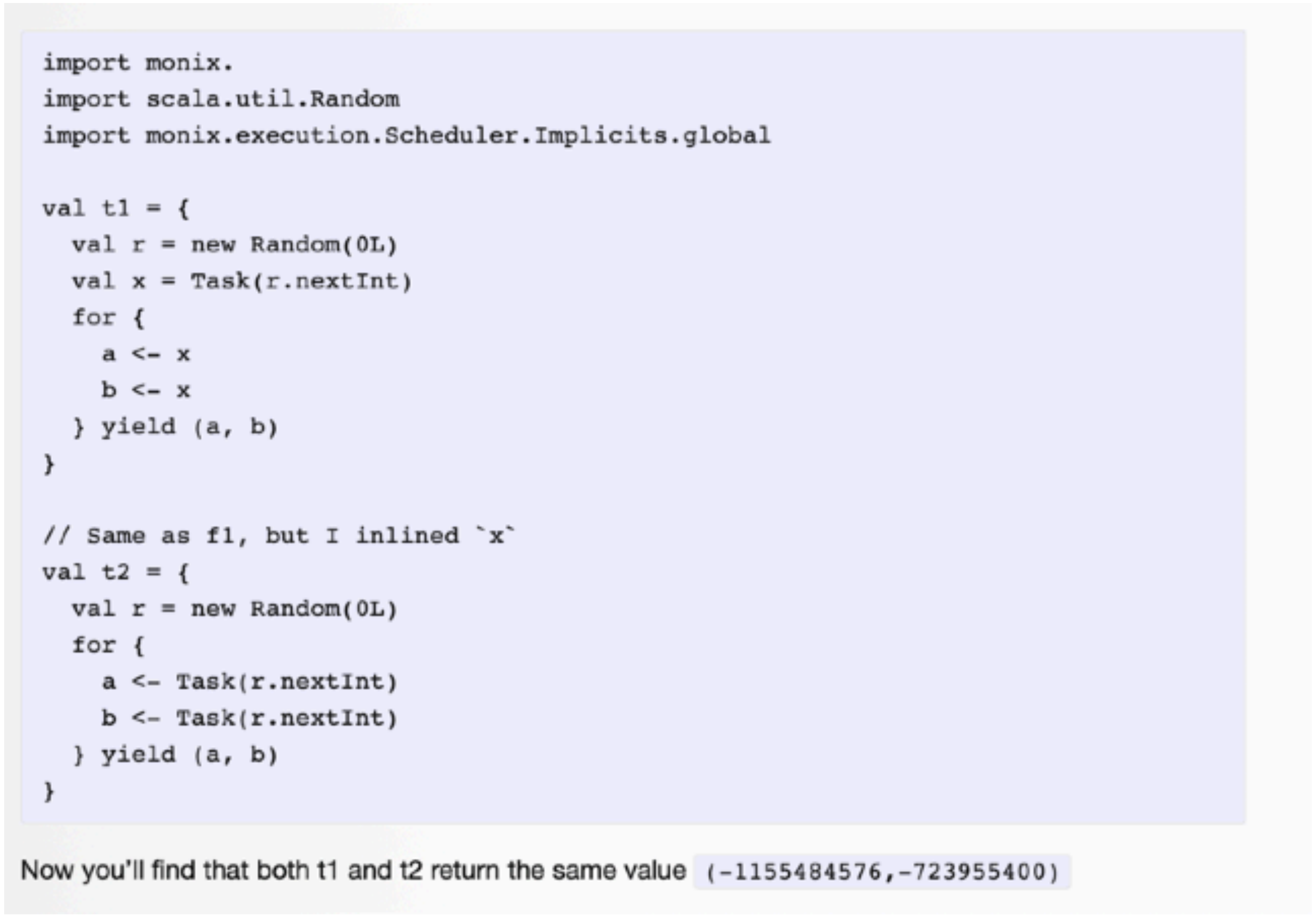
IO Monad - Monix (2.x series)

https://monix.io/docs/2x/eval/task.html#design-summary **Alex Nedelcu**



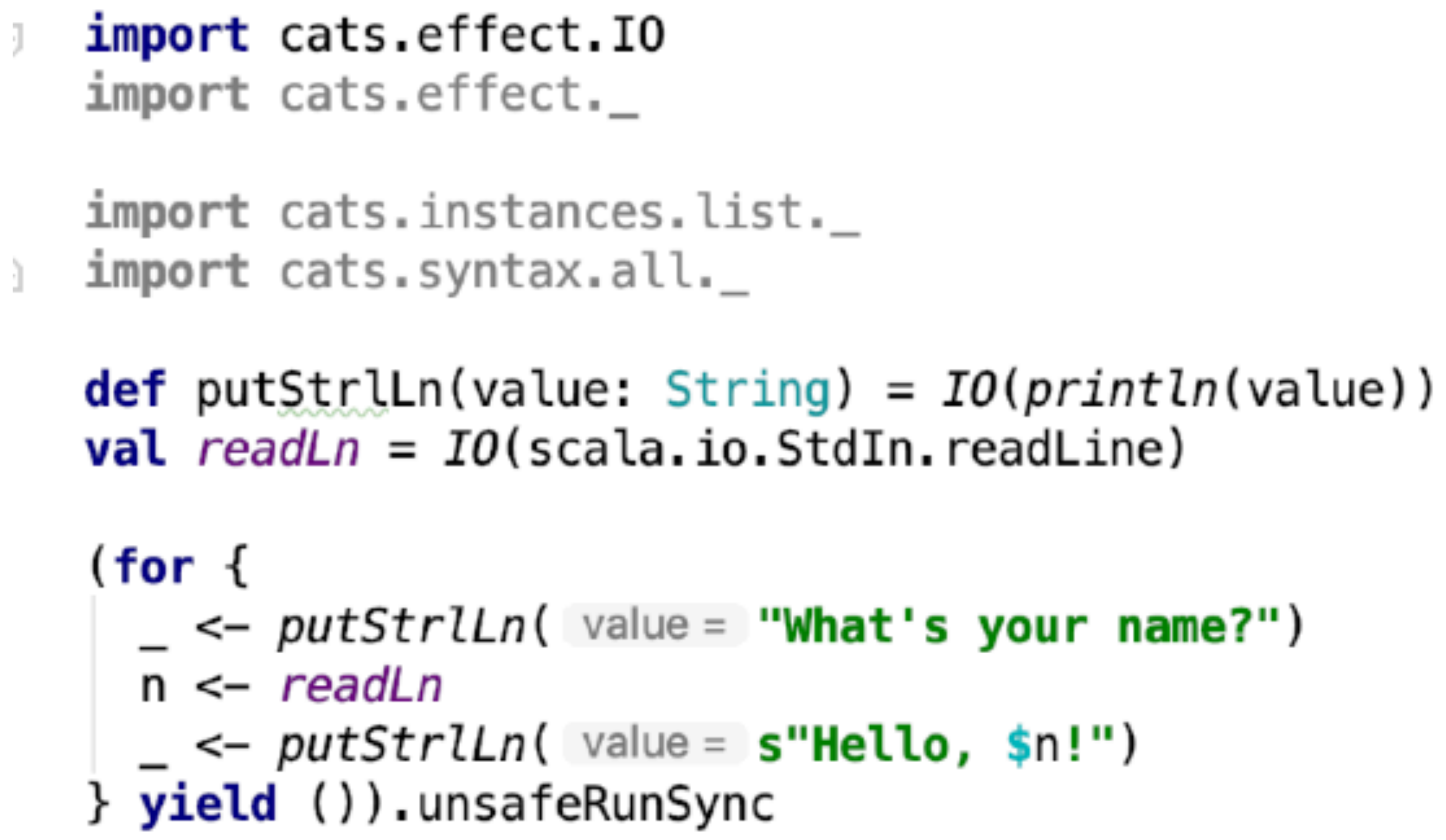
```
import monix.
import scala.util.Random
import monix.execution.Scheduler.Implicits.global

val t1 = {
  val r = new Random(0L)
  val x = Task(r.nextInt)
  for {
    a <- x
    b <- x
  } yield (a, b)
}

// Same as f1, but I inlined `x`
val t2 = {
  val r = new Random(0L)
  for {
    a <- Task(r.nextInt)
    b <- Task(r.nextInt)
  } yield (a, b)
}
```

Now you'll find that both t1 and t2 return the same value `(-1155484576,-723955400)`

IO Monad - Cats Effect

Monix Task, Scalaz IO/Task became Cats Effect

Includes cancelation, stack safe trampolining, parallel execution and a green threads system (fibres),



```scala
import cats.effect.IO
import cats.effect._

import cats.instances.list._
import cats.syntax.all._

def putStrlLn(value: String) = IO(println(value))
val readLn = IO(scala.io.StdIn.readLine)

(for {
  _ <- putStrlLn( value = "What's your name?")
  n <- readLn
  _ <- putStrlLn( value = s"Hello, $n!")
} yield ()).unsafeRunSync
```

https://www.youtube.com/watch?v=g_jP47HFpWA

Daniel Spiewak - The Making of an IO