

Scala STM Cheatsheet

```
import scala.concurrent.stm._

/** From: http://nbronson.github.io/scala-stm/syntax\_cheat\_sheet.html */
object CheatSheet extends App {

  val x = Ref(0) // allocate a Ref[Int]
  val y = Ref.make[String]() // type-specific default
  val z = x.single // Ref.View[Int]

  atomic { implicit txn =>
    val i = x() // read
    y() = "x was " + i // write
    val eq = atomic { implicit txn => // nested atomic
      x() == z() // both Ref and Ref.View can be used inside atomic
    }
    assert(eq)
    y.set(y.get + ", long-form access")
  }

  // only Ref.View can be used outside atomic
  println("y was '" + y.single() + "'")
  println("z was " + z())

  atomic { implicit txn =>
    y() = y() + ", first alternative"
    if (x.getWith { _ > 0 }) // read via a function
      retry // try alternatives or block
  } orAtomic { implicit txn =>
    y() = y() + ", second alternative"
  }

  val prev = z.swap(10) // atomic swap
  val success = z.compareAndSet(10, 11) // atomic compare-and-set
  z.transform { _ max 20 } // atomic transformation
  val pre = y.single.getAndTransform { _.toUpperCase }
  val post = y.single.transformAndGet { _.filterNot { _ == ' ' } }
}
```