Listing 1: Simple code listing.

```
// this is a simple code listing:
println("hello kotlin from latex")
```

Listing 2: Example kotlin code.

```
/* Block comment */
package hello
import kotlin.collections.* // line comment

/**
 * Doc comment here for `SomeClass`
 * @see Iterator#next()
 */
@Deprecated("Deprecated class")
private class MyClass<out T : Iterable<T>>(var prop1 : Int) {
    fun foo(nullable : String?, r : Runnable, f : () -> Int,
            fl : FunctionLike, dyn: dynamic) {
        println("length\nis ${nullable?.length} \e")
        val ints = java.util.ArrayList<Int?>(2)
        ints[0] = 102 + f() + fl()
        val myFun = { -> "" };
        var ref = ints.size
        ints.lastIndex + globalCounter
        ints.forEach lit@ {
            if (it == null) return@lit
            println(it + ref)
        }
        dyn.dynamicCall()
        dyn.dynamicProp = 5
    }

    val test = """hello
                  world
                  kotlin"""

    override fun hashCode(): Int {
        return super.hashCode() * 31
    }
}

fun Int?.bar() {
    if (this != null) {
        println(message = toString())
    }
    else {
        println(this.toString())
    }
}

var globalCounter : Int = 5
    get = field

abstract class Abstract {
}

object Obj

enum class E { A, B }

interface FunctionLike {
    operator fun invoke() = 1
}
```