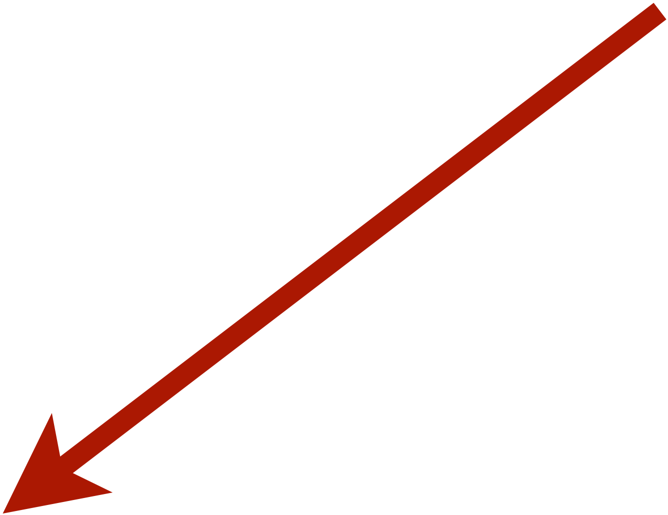
KOTLIN-MOCKITO

```
class PresenterTest {

    val view: View = mock()

    @Test
    fun `given non-empty list when presenter start then display elements on view`() {
        val elements = listOf(
                Element(1, "first"),
                Element(2, "second")
        )

        val dataProvider: DataProvider = mock {
            on { getAll() } doReturn elements
        }

        val presenter = Presenter(view, dataProvider)

        presenter.start()

        verify(view).displayItems(elements)

    }
}
```

inline, reifeid



better syntax

KOTLIN-MOCKITO

```
class PresenterTest {

    val view: View = mock()

    @Test
    fun `given non-empty list when presenter start then display elements on view`() {
        val elements = listOf(
                Element(1, "first"),
                Element(2, "second")
        )

        val dataProvider: DataProvider = mock {
            on { getAll() } doReturn elements
        }

        val presenter = Presenter(view, dataProvider)

        presenter.start()

        verify(view).displayItems(elements)

    }
}
```

inline, reifeid

better syntax

MOCKK

```
class PresenterTest {

    val view: View = mockk(relaxUnitFun = true)

    @Test
    fun `given non-empty list when presenter start then display elements on view`() {
        val elements = listOf(
                Element(1, "first"),
                Element(2, "second")
        )

        val dataProvider = mockk<DataProvider>()

        every { dataProvider.getAll() } returns elements

        val presenter = Presenter(view, dataProvider)

        presenter.start()

        verify { view.displayItems(elements) }

    }
}
```