

[Project Link](#)

Points to ponder from this app about Kotlin:

Koding Efficiency

Cocise Syntax

Avoids boiler plate code (really slick)

Error reduction

Null Safety

Ability to indicate intentions

Excellent compatibility with Java and Android

Singleton pattern for a class is just done using **object** keyword insted of class (So simple!)

- Automatically instantiated
- All access against same instance
- Member accessed through type name

A class can function as a **data model class** (like POJO in Java)

- Mark class with data keyword
- Kotlin generates standard methods (equals(), hashCode(), toString())
- Primary constructor must contain only properties

Null safety

Null safe operator

?.

Returns member if not null
like

`p?.name`

otherwise return null

Elvis operator

?:

returns first operand if not null
otherwise
second operand

explained in Java way:

```
a != null ? a : b
```

Example Code:

```
class Person{
    val name: String = "Jim"
    var weightLbs: Double = 0.0
    var weightKilos: Double
        get() = weightLbs / 2.2
        set(value){
            weightLbs = value * 2.2
        }
}

val p = Person()
val name = p.name
p.weightLbs = 220
val kilos = p.weightKilos
p.weightKilos = 50.0
val lbs = p.weightLbs

class Person (val name: String, val weightLbs: Double){
    var weightKilos: Double
        get() = weightLbs / 2.2
        set(value){
            weightLbs = value * 2.2
        }

    fun eatDessert(addedIceCream: Boolean = true){
        weightLbs += if (addedIceCream) 4.0 else 2.0
    }

    fun calcGoalWeightLbs(lbsToLose : Double = 10) : Double {
        return weightLbs - lbsToLose
    }
}
```

```
}
```

However, less code in Kotlin could mean less readability and someone coming from Java the syntax seem pretty confusing and obscure!