# **Kotlin Cheat Sheet**



## 1. Kotlin Basics

- `var` vs `val`
- Data Types
- String Interpolation
- Default Argumentsz
- Named Arguments
- Elvis Operator (`?:`)
- Safe Calls (`?.`)
- Null Safety (`!!`)
- Type Casting (`as`, `is`)
- Extension Functions
- Companion Object
- `lateinit` vs `lazy`

# 4. Collections

- Immutable vs Mutable Collections
- · List, Set, Map
- · Filtering and Mapping
- `flatMap` and `groupBy`
- Lazy Collections (`sequence`)
- · Sorting and Reversing
- `forEach` vs for Each Indexed

# 8. Exception Handling

- Try-Catch-Finally
- Custom Exceptions
- require, check, assert

# Interoperability

- @JvmStatic
- @JvmField
- @Throws
- Annotations (@Nullable, @NonNull)
- SAM Conversions

# 2. Functions

- Top-Level Functions
- Single-Expression **Functions**
- Inline Functions
- Higher-Order Functions ((map, filter, for Each, reduce))
- Lambda Functions
- Anonymous Functions
- Tail-Recursive Functions
- **Function Overloading**

# 3.Classes and Objects

- Primary and Secondary Constructors
- Data Classes
- Sealed Classes
- **Abstract Classes**
- Interfaces
- Enum Classes
- Object Keyword
- Inner Classes
- **Nested Classes**

# 5. Coroutines

- launch vs async
- CoroutineScope
- Structured Concurrency
- runBlocking

- suspend Functions

- withContext
- Dispatchers (IO, Main, Default)
- · Flows (Flow, StateFlow, SharedFlow)

# Delegation Inline Classes

Generics

- Type Aliases Operator Overloading
- Destructuring Declarations

6. Advanced Kotlin

- Variance (in, out)
- Reified Types

- Inline Lambdas
- when Expression

10.Miscellaneous

- Type Inference
- Kotlin Scripting (.kts)
- by Keyword
- Reflection (KClass, call)
- Kotlin Multiplatform (KMM)
- Enum Constants and Enum **Functions** (valueOf, values)



# 7. Kotlin Standard Library

- apply, let, run, also, with
- String Functions (substring, split, replace)
- Collections Operations (filter, map, reduce)
- Ranges (.., until, downTo)
- takeIf and takeUnless
- repeat

# S

# 11. Object-Oriented **Programming (OOP)**

- Classes & Objects
- Constructors (Primary & Secondary)
- Inheritance (open, super)
- Abstract & Sealed Classes
- Interfaces



/@pixeldesigndeveloper









Github/Dinesh2510

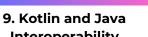
| Github/Dinesh2510











- @JvmOverloads
- Nullability