

Kotlin Quick Notes

1. Basics

- **File Extension:** .kt
 - **Entry Point:**
 - **fun main() {**
 - **println("Hello, Kotlin!")**
 - **}**
 - **Variables:**
 - **val** → **Immutable (read-only)**
 - **var** → **Mutable**
 - **val name = "Minu"**
 - **var age = 22**
-

2. Data Types

Type Example

Int **10**

Double **10.5**

Boolean **true / false**

Char **'A'**

String **"Hello"**

Type inference means Kotlin automatically detects the type.

3. Conditional Statements

if (age > 18) {

println("Adult")

} else {

println("Minor")

```
}
```

When Expression (like switch):

```
when (day) {  
    1 -> println("Monday")  
    2 -> println("Tuesday")  
    else -> println("Other day")  
}
```

4. Loops

```
for (i in 1..5) println(i) // 1 to 5  
  
for (i in 5 downTo 1) println(i)  
  
while (x > 0) x--
```

5. Functions

```
fun add(a: Int, b: Int): Int {  
    return a + b  
}
```

// Single-expression function

```
fun multiply(a: Int, b: Int) = a * b
```

Default and named arguments:

```
fun greet(name: String = "User") = println("Hello, $name")  
  
greet("Minu")
```

6. Classes & Objects

```
class Person(val name: String, var age: Int)
```

```
val p1 = Person("Minu", 22)
```

```
println(p1.name)
```

Inheritance:

```
open class Animal { open fun sound() = println("Sound") }

class Dog: Animal() { override fun sound() = println("Bark") }
```

7. Null Safety

```
var name: String? = null

println(name?.length) // Safe call

println(name ?: "Unknown") // Elvis operator
```

8. Collections

```
val list = listOf("Apple", "Banana") // Immutable

val mutableList = mutableListOf(1, 2, 3)

mutableList.add(4)

val map = mapOf(1 to "One", 2 to "Two")
```

9. Lambdas & Higher-Order Functions

```
val sum = { a: Int, b: Int -> a + b }

println(sum(5, 3))

fun operate(a: Int, b: Int, op: (Int, Int) -> Int) = op(a, b)

println(operate(2, 3, sum))
```

Key Features

- Interoperable with Java
- Null safety
- Smart casts

- **Data classes**
- **Coroutines (for async programming)**