

# Kotlin Cheatsheet

Kotlin is a statically-typed programming language that runs on the Java Virtual Machine (JVM). It is designed to be concise, expressive, and safe, making it a popular choice for Android app development, server-side programming, and web development.

## Unique Features

- Null safety
- Extension functions
- Data classes
- Coroutines
- Operator overloading
- Type inference
- Interoperability with Java

## Variables

Variables in Kotlin are declared using the `var` or `val` keyword. Kotlin supports type inference, so you don't need to specify the type of the variable.

```
var name = "John"
val age = 30
val pi = 3.14
```

## Functions

Functions in Kotlin are declared using the `fun` keyword followed by the function name and parameters. Kotlin supports lambda expressions, which are anonymous functions that can be assigned to variables and passed as arguments to other functions.

```
fun greet(name: String) {
    println("Hello, $name!")
}

greet("John")

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

println(add(2, 3))
```

## Loops

Kotlin supports `for` and `while` loops, as well as the enhanced `for` loop, which can iterate over collections.

```
val numbers = listOf(1, 2, 3, 4, 5)

for (number in numbers) {
    println(number)
}

var i = 0
while (i < 5) {
    println(i)
    i++
}
```

## Conditionals

Kotlin supports `if`, `else if`, and `else` statements, as well as the ternary operator.

```
val age = 30

if (age < 18) {
    println("You are too young to vote.")
} else if (age < 21) {
    println("You can vote, but not drink.")
} else {
    println("You can vote and drink.")
}

val result = if (age >= 18) "You are an adult" else "You are not an adult"
println(result)
```

## File Manipulation

Kotlin provides several ways to read and write files. You can use the `File` class to create, read, write, and delete files.

```
val file = File("example.txt")

// Write to file
file.writeText("Hello, world!")

// Read from file
val content = file.readText()
println(content)

// Delete file
file.delete()
```

## Resources

- [Kotlin Documentation](#)
- [Kotlin Style Guide](#)
- [Kotlin Koans](#)
- [Kotlin Playground](#)