

KOTLIN MECH

A ERA DOS ANDROIDES



LEONARDO DIAS

INTRODUÇÃO AO KOTLIN

Uma linguagem moderna para desenvolvedores

Kotlin é uma linguagem de programação moderna, concisa e segura, desenvolvida pela JetBrains. Ela foi criada para interoperar totalmente com Java, permitindo que desenvolvedores Java possam adotar Kotlin de forma gradativa e aproveitar os benefícios de uma linguagem moderna sem reescrever todo o código existente. Kotlin é usada tanto para desenvolvimento Android quanto para backend, aplicativos de desktop e até mesmo desenvolvimento web com Kotlin/JS.



01

TIPOS BASICOS

Kotlin é uma linguagem estaticamente tipada, o que significa que os tipos das variáveis são conhecidos em tempo de compilação. No entanto, Kotlin permite que você omita a declaração de tipos quando o compilador pode inferi-los automaticamente.

Inteiros (Int, Long, Short, Byte)

Os tipos de dados inteiros em Kotlin são usados para representar números inteiros, seja positivos, negativos ou zero. Aqui estão alguns dos tipos inteiros disponíveis em Kotlin:

Int: representa números inteiros padrão, geralmente de 32 bits.

Long: usado para números inteiros longos, de 64 bits.

Short: para números inteiros curtos, de 16 bits.

Byte: representa números inteiros muito pequenos, de 8 bits.

```
val idade: Int = 30
val populacaoMundial: Long = 78000000000L
val temperatura: Short = -10
val numeroPequeno: Byte = 5
```



02

FUNÇÕES EM KOTLIN

As funções são blocos de código reutilizáveis que executam uma tarefa específica. Em Kotlin, funções são declaradas usando a palavra-chave `fun`. As funções podem receber parâmetros, retornar valores, e podem ser definidas em diversos contextos, como em classes, objetos, ou de forma top-level (no nível superior do arquivo).

Declaração de Funções

Uma função em Kotlin é declarada usando a palavra-chave `fun`, seguida pelo nome da função, a lista de parâmetros entre parênteses, o tipo de retorno (se houver), e o corpo da função.

```
fun saudacao(nome: String): String {  
    return "Olá, $nome!"  
}  
  
fun main() {  
    println(saudacao("Mundo"))  
}
```



03

ESTRUTURAS DE CONTROLE

Kotlin oferece diversas estruturas de controle que permitem controlar o fluxo de execução do programa de maneira eficiente e expressiva. As principais estruturas de controle incluem `if`, `when`, `for`, e `while`. Vamos explorar cada uma dessas estruturas com mais detalhes e exemplos práticos.

Estrutura "if"

A estrutura "if" é usada para tomar decisões com base em condições. Em Kotlin, "if" pode ser usado tanto como uma expressão (retornando um valor) quanto como uma instrução (executando um bloco de código).

```
fun maiorDeIdade(idade: Int): Boolean {  
    return if (idade >= 18) {  
        true  
    } else {  
        false  
    }  
}  
  
fun main() {  
    val idade = 20  
    val isAdulto = maiorDeIdade(idade)  
    println("Maior de idade: $isAdulto")  
}
```

Neste exemplo, a função "maiorDeIdade" usa "if" como uma expressão para retornar "true" se a idade for 18 ou mais, e "false" caso contrário.

