

```
// Declarando Variáveis

val constante: Tipo = valor // Valor imutável (val)

var variavel: Tipo = valor // Variável mutável (var)

// Tipos de Dados Básicos

val inteiro: Int = 42

val duplo: Double = 3.14

val texto: String = "Olá, Kotlin"

val booleano: Boolean = true

// Controle de Fluxo

if (condicao) {

    // Código se a condição for verdadeira

} else {

    // Código se a condição for falsa

}

when (valor) {

    opcao1 -> // Código se o valor corresponder a opcao1

    opcao2 -> // Código se o valor corresponder a opcao2

    else -> // Código se o valor não corresponder a nenhuma opção

}

// Loops

for (i in 1..10) {

    // Código executado 10 vezes com i variando de 1 a 10

}

while (condicao) {

    // Código executado enquanto a condição for verdadeira

}

// Funções

fun nomeDaFuncao(parametro1: Tipo, parametro2: Tipo): TipoDeRetorno {

    // Código da função

    return valorDeRetorno

}

// Classes e Objetos

class MinhaClasse {
```

```
var propriedade: Tipo = valorInicial

fun minhaFuncao() {
    // Código da função
}

val objeto = MinhaClasse()
objeto.propriedade = novoValor
objeto.minhaFuncao()

// Extensões
fun TipoDeDado.minhaFuncaoDeExtensao() {
    // Código da função de extensão
}

variavel.minhaFuncaoDeExtensao()

// Nullable Types (Tipos Nulos)
val valorNulo: Tipo? = null

// Tratamento de Exceções
try {
    // Código que pode lançar uma exceção
} catch (excecao: TipoDeExcecao) {
    // Tratamento da exceção
} finally {
    // Código executado sempre, com ou sem exceção
}

// Arrays e Coleções
val array = arrayOf(1, 2, 3)
val lista = listOf("A", "B", "C")
val mapa = mapOf("chave" to "valor")

// Lambda Expressions
val soma = { a: Int, b: Int -> a + b }
val resultado = soma(2, 3)

// Null Safety (Segurança contra nulos)
val tamanho = texto?.length // Evita uma exceção se 'texto' for nulo
```

```
val tamanhoNaoNulo = texto!!.length // Assume que 'texto' não é nulo

// Operadores

val a = 5

val b = 3

val maximo = if (a > b) a else b // Operador ternário

// Importando Bibliotecas

import pacote.Classe

// Comentários

// Este é um comentário de uma linha

/* Este é um comentário de várias linhas */
```