

TIPOS BÁSICOS:

- Int: Inteiro
- Double: Ponto flutuante
- Boolean: Verdadeiro/Falso
- String: Texto
- Char: Caractere

DECLARAÇÃO DE VARIÁVEIS:

```
var nome: String = "Alice" // Mutável  
val idade: Int = 30        // Imutável
```

CONDICIONAIS:

```
if (idade >= 18) {  
    println("Maior de idade")  
} else {  
    println("Menor de idade")  
}
```

LOOPS:

```
for (i in 1..5) {  
    println(i)  
}
```

COLLECTIONS:

```
val lista = listOf(1, 2, 3)  
val mapa = mapOf("a" to 1, "b" to 2)
```

FUNÇÕES:

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

```
}
```

```
saudacao("João")
```

EXTENSÕES DE CLASSE:

```
-----
```

```
fun String.contarPalavras(): Int {  
    return this.split(" ").size  
}
```

```
val texto = "Isso é um exemplo"  
println(texto.contarPalavras()) // Saída: 4
```

CLASSES:

```
-----
```

```
class Pessoa(val nome: String, val idade: Int)
```

```
val pessoa = Pessoa("Alice", 30)  
println(pessoa.nome) // Acessando propriedade
```

HERANÇA:

```
-----
```

```
open class Animal(val nome: String) {  
    open fun fazerSom() {  
        println("Animal fazendo som")  
    }  
}
```

```
class Cachorro(nome: String) : Animal(nome) {  
    override fun fazerSom() {  
        println("Cachorro latindo")  
    }  
}
```

EXCEÇÕES:

```
-----
```

```
try {  
    // Código que pode gerar exceção
```

```

} catch (e: Exception) {
    println("Ocorreu uma exceção: ${e.message}")
} finally {
    // Bloco opcional
}

```

--

NULL SAFETY:

```

var nome: String? = null // Permite valores nulos
nome?.length           // Operador de chamada segura
nome ?: "Nome Padrão"  // Operador Elvis

```

EXTENSÕES DE FUNÇÃO:

```

fun Int.dobro(): Int {
    return this * 2
}

```

```

val resultado = 5.dobro() // 10

```

INTERFACE:

```

interface Imprimivel {
    fun imprimir()
}

```

```

class Documento : Imprimivel {
    override fun imprimir() {
        println("Imprimindo documento")
    }
}

```

ENUM CLASSES:

```

enum class DiaSemana {

```

```
    SEGUNDA, TERÇA, QUARTA, QUINTA, SEXTA, SÁBADO, DOMINGO  
}
```

```
val dia = DiaSemana.SEGUNDA
```

SMART CAST:

```
fun imprimirTamanho(valor: Any) {  
    if (valor is String) {  
        println("Tamanho da string: ${valor.length}")  
    }  
}
```

RANGE:

```
for (i in 1 until 5) {  
    println(i) // 1, 2, 3, 4  
}
```