



Kotlin

Rafael Vieira Coelho
rafaelvc2@gmail.com

<https://kotlinlang.org>

Variáveis

- val: apenas leitura
- var: mutável

Obs: objetos armazenados em uma val podem ser alterados internamente.

Variável de Apenas Leitura (val)

```
val question: String =  
    "life, the universe, " +  
        "and everything"  
println("$question")
```

life, the universe, and everything?

Variável de Apenas Leitura (val)

```
val question: String =  
    "life, the universe, " +  
        "and everything"  
println("$question")
```

question = "sure?"

Compiler error:
val cannot be reassigned

Read-only variable: val

```
val question: String =  
    "life, the universe, " +  
        "and everything"  
println("$question")  
  
question = "sure?"
```

Compiler error:
val cannot be reassigned

corresponds to a final variable in Java

Read-only variab...
val question: String =

Variável Mutável (var)

```
var answer = 0  
answer = 42  
println(answer)      // 42
```

Inferência de Tipo Local

- Não precisamos definir o tipo da variável em Kotlin.
- O tipo da variável é inferido a partir do tipo de informação armazenado inicialmente:

```
val greeting = "Hi!"
```

```
var number = 0
```

Inferência de Tipo Local

- Nos exemplos abaixo, os tipos inferidos são String e Int.

```
: String  
val greeting = "Hi!"  
  
: Int  
var number = 0
```

Inferência de Tipo Local

- O tipo não pode ser alterado após o primeiro dado ser armazenado:

```
: Int  
var string = 1  
string = "abc"
```

Compiler error: Type mismatch:
inferred type is String
but Int was expected

Listas Mutáveis x Listas Apenas Leitura

- Não podemos alterar uma lista que foi criada com o método `listOf()`

```
val mutableList = mutableListOf("Java")
mutableList.add("Kotlin")
```

```
val readOnlyList = listOf("Java")
readOnlyList.add("Kotlin")
```

Aula03Kotlin > src > Ex1Variaveis.kt

1: Project

```

Project
Aula03Kotlin ~/Dropbox/Livros
  .idea
  out
  src
    Aula03Kotlin.iml
External Libraries
Scratches and Consoles

```

Ex1Variaveis.kt

```

1 fun main() {
2     var a: String = "initial" // 1
3     println(a)
4     val b: Int = 1           // 2
5     val c = 3                // 3
6
7     var e: Int = 3           // 1
8     println(e) // 2
9
10    val d: Int // 1
11
12    if (e == 1) {
13        d = 1 // 2
14    } else {
15        d = 2 // 2
16    }
17    println(d) // 3
18}

```

2: Favorites

Run: Ex1VariaveisKt

```

Run: Ex1VariaveisKt
  "/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" "-javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/ideaagent.jar"
  initial
  3
  2
  Process finished with exit code 0

```

3: Structure

4: Terminal

5: Messages

6: Run

7: TODO

8: Event Log

Build completed successfully with 2 warnings in 8 s 384 ms (moments ago)

1:13 LF UTF-8 4 spaces

Variáveis

Variáveis Não-inicializadas

- Quando tentamos usar uma variável que não foi inicializada, ocorre um erro de compilação.

```
var e: Int // 1
println(e) // 2
```

Exemplos de Conteúdo Nulo

1. Declara uma String não-nula
2. Ocorre um erro de compilação
3. O símbolo ? permite valor null
4. Inicializa com null a variável
5. Tipo inferido na variável
6. Mesmo erro de compilação do passo 2
7. Declara uma função com um parâmetro não-nulo
8. Não ocorre erro pois a variável é não-nula
9. Ocorre um erro de compilação pois é variável pode ser nula

```
var neverNull: String = "This can't be null"          // 1
neverNull = null                                     // 2

var nullable: String? = "You can keep a null here"   // 3
nullable = null                                       // 4

var inferredNonNull = "The compiler assumes non-null" // 5
inferredNonNull = null                                // 6

fun strLength(notNull: String): Int {
    return notNull.length
}                                                       // 7

strLength(neverNull)                                 // 8
strLength(nullable)                                  // 9
```