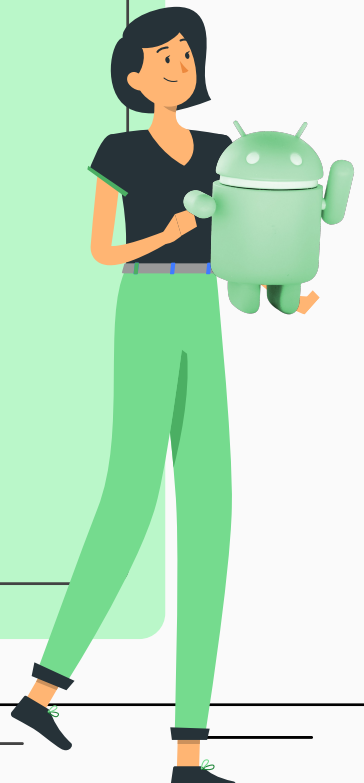
AULA 4

Kotlin



01 KOTLIN

Entendimento básico da linguagem

02 ALGORITMOS

Praticar o que aprendemos de kotlin

Qual a diferença entre:

```
var personName: String - > Tipo non null
var animalName: String? - > Tipo nullable
```

DOJO - ELVIS OPERATOR

Abrir **Kotlin REPL (read evaluate print loop)**

```
var stringNaoNula: String = null
var stringNula: String? = null
stringNula.length
stringNula!!.length
stringNulla?.legth

val stringLength = stringNulla?.length :? 0
```

DOJO - CONDITIONALS

Abrir **Kotlin REPL (read evaluate print loop)**

```
val i = 21
If (i < 15) {
   print("pequeno")
} else {
   print("grande")
}

If (i < 15) {
   print("pequeno")
} else if (i >= 15 &&  i < 25) {
   print("ok")
} else {
   print("grande")
}

X = If (i < 15) {
   print("pequeno")
} else if (i >= 15 &&  i < 25) {
   print("ok")
} else {
   print("grande")
}
```

```
X
X = If (i < 15) {
   print("pequeno")
"pequeno"
} else if (i >= 15 &&  i < 25) {
   print("ok")
"ok"
} else {
   print("grande")
"grande"
}

X

val texto: String? = "Kotlin"
if(text != null) {
          String.lenght
}
```

DOJO - CONDITIONALS

Abrir **Kotlin REPL (read evaluate print loop)**

```
val preco = 50
when(preco) {
        0 -> print("hoje é gratis")
        25 -> print("promoção")
        26..30 -> print("desconto fidelidade")
        31..49 -> print("desconto funcionarias")
        else -> ("preço regular")
}
```

```
val preco = 30
when(preco) {
        0 -> print("hoje é gratis")
        25 -> print("promoção")
        10 + 20 -> print("desconto fidelidade")
        31..49 -> print("desconto funcionarias")
        else -> ("preço regular")
}
```

```
val preco = 30
when(preco) {
        0 -> print("hoje é gratis")
        !25 -> print("promoção")
        else -> ("preço regular")
}
```

```
val preco = 10
 when {
     preco < 5 -> "promoção"
     preco >= 6 && preco < 8 -> "desconto"
     else -> "regular"
 }
```

DOJO - CONDITIONALS

Abrir **Kotlin REPL (read evaluate print loop)**

```
val preco = 50
when(preco) {
        0 -> print("hoje é gratis")
        25 -> print("promoção")
        26..30 -> print("desconto fidelidade")
        31..49 -> print("desconto funcionarias")
        else -> ("preço regular")
}
```

```
val preco = 30
when(preco) {
        0 -> print("hoje é gratis")
        25 -> print("promoção")
        10 + 20 -> print("desconto fidelidade")
        31..49 -> print("desconto funcionarias")
        else -> ("preço regular")
}
```

```
val preco = 30
when(preco) {
        0 -> print("hoje é gratis")
        !25 -> print("promoção")
        else -> ("preço regular")
}
```

```
val preco = 10
 when {
     preco < 5 -> "promoção"
     preco >= 6 && preco < 8 -> "desconto"
     else -> "regular"
 }
```

DOJO - COLLECTIONS

```
List<Int>
Set<Int>
Map<Int>

MutableList<Int>

Val array = arrayOf(1, 3, 4, 11)
array.joinToString()

val list = listOf(1,2,3)
val mutableList = mutableListOf(1, 2, 3)
mutableList[0] = 99


val set = setOf(1,1,2,3,4,5)

val mutableSet = mutableSetOf(1,2,2,3,4,5,5)
```

```
val map = mapOf(Pair(1, "Android"), Pair(2, "Kotlin"))
 val mutableMap = mutableMapOf(1 to "Android", 2 to "Kotlin", 3 to "Java")
```

DOJO - LOOPS

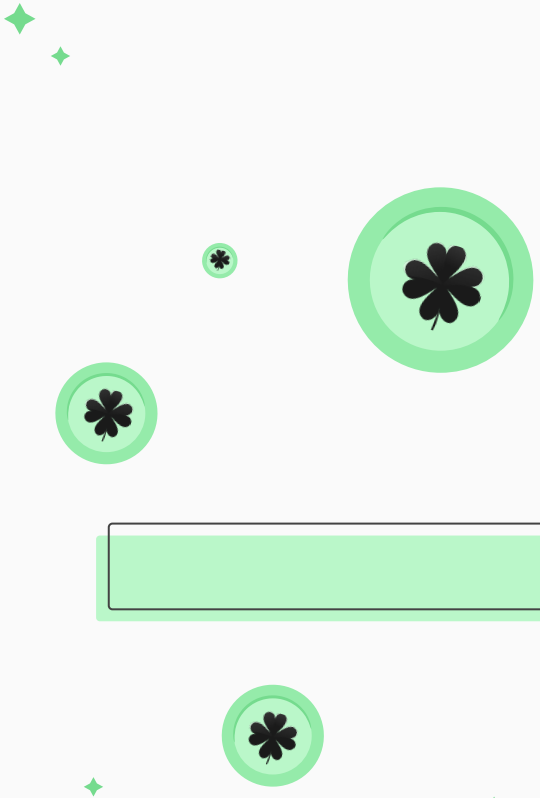
```
for(i in 1..10) {
     println(i)
 }

for(i in 1..10) {
     println("$i ")
 }

for(c in "Kotlin") {
     println("$c ")
 }

val list = listOf(1,2,3,4,5,6,7,8,9)
 for (i in list) {
     print(i)
 }
```

```
val listOfCities = listOf("Belo Horizonte", "São Paulo", "Rio de Janeiro")
 for (city in listOfCities) {
     print("$city é legal")
 }

val listOfCities = listOf("Belo Horizonte", "São Paulo", "Rio de Janeiro")
 for (city in listOfCities) {
     print("$city é legal\n")
 }

for(i in 10 downTo 1){
     print("$i " )
 }
```

DOJO - FUNCTIONS

```
fun permitirEntrada(idade: Int): Boolean {
    return idade >= 18
 }

 permitirEntrada(7)
res11: kotlin.Boolean = false

permitirEntrada(18)
res12: kotlin.Boolean = true

fun permitirEntrada(idade: Int): Boolean = idade >= 18

permitirEntrada(24)
res14: kotlin.Boolean = true

val permitido = permitirEntrada(12)

permitido
res16: kotlin.Boolean = false
```

```
fun permitirEntrada(vararg idades: Int): Boolean {
    return idades.any {idade -> idade >= 18}
 }

 permitirEntrada(12, 18, 13, 10)
res17: kotlin.Boolean = true
```



RESOLVER PROBLEMA DO TRIPLETS