

# SUPORTE\_

Baseado em  
história real  
presenciada  
por mim mesmo.

## Entrevista de emprego

AH! ENTÃO VOCÊ VIU  
JAVA NA FACULDADE?



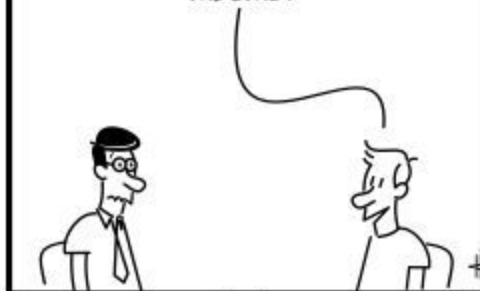
SIM.

JÁ SABE FAZER UM  
"HELLO WORLD"?  
HAHAHASHAHASHASS



HEHEHHE NÃO,  
NÃO...

SÓ APRENDI O  
BÁSICO  
MESMO.



[vidadesuporte.com.br](http://vidadesuporte.com.br)



Aula 04

# Introdução a Kotlin e Estrutura Básica

# Tipos de dados numéricos

Tipos de Dados		
<b>Int</b>	Número inteiro	32 bits
<b>Long</b>	Número inteiro muito grande	64 bits
<b>Float</b>	Número com ponto flutuante(Ex.: 1.5)	32 bits
<b>Double</b>	Número com ponto flutuante e muitos decimais (Ex.: 10065676677.0)	64 bits
<b>Byte</b>		8 bits

# Operadores

## Operadores Matemáticos

+	Adição
-	Subtração
/	Divisão
*	Multiplicação
%	Resto da divisão
++	Adicionar 1
--	Subtrair 1

## Comparações

>	Maior
<	Menor
==	Igual
<=	Menor ou igual
>=	Maior ou igual
!=	Diferente

# Tipos de dados texto

Tipos de Dados	
Char	Um caractere (Ex.: 'a')
String	Cadeia de caracteres (textos)

# Operadores

Operadores	
+	Concatenação

# Tipos de dados lógicos

Tipos de Dados		
Boolean	Verdadeiro ou falso	8 bits

# Operadores

## Operadores Lógicos

Operadores Lógicos	
<b>&amp;&amp;</b>	E (retorna verdadeiro se as duas avaliações forem verdadeiras)
<b>  </b>	OU (retorna verdadeiro se uma das duas avaliações for verdadeira)
<b>!</b>	NÃO (nega uma avaliação)



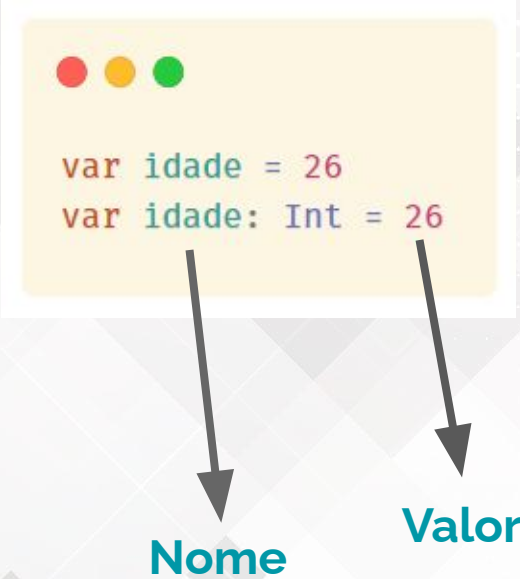
# Função main



```
fun main() {  
    print("Hello, World")  
}
```

## Declarando variáveis de leitura e escrita

```
var idade = 26  
var idade: Int = 26
```




The diagram shows two code snippets. The first snippet, 'var idade = 26', has an arrow pointing from 'idade' to the word 'Nome' and another arrow pointing from '26' to the word 'Valor'. The second snippet, 'var idade: Int = 26', has an arrow pointing from 'idade' to 'Nome' and another arrow pointing from '26' to 'Valor'.

Nome

Valor

## Declarando variáveis somente de leitura (constantes)



```
val pi = 3.14
```

The diagram shows an arrow pointing from the first code snippet to the second code snippet, indicating a transition or comparison between the two types of variable declarations.

# Prática I

# E se... ?



# Operações condicionais

Em Kotlin, if é uma expressão, ou seja, pode retornar um valor

```
fun main() {  
    val numero = 2  
  
    if (numero == 1) {  
        print("É pouco")  
    } else if (numero == 2) {  
        print("É muito")  
    } else if (numero == 3 || numero == 4) {  
        print("É demais")  
    } else {  
        print("Caiu no else")  
    }  
}
```

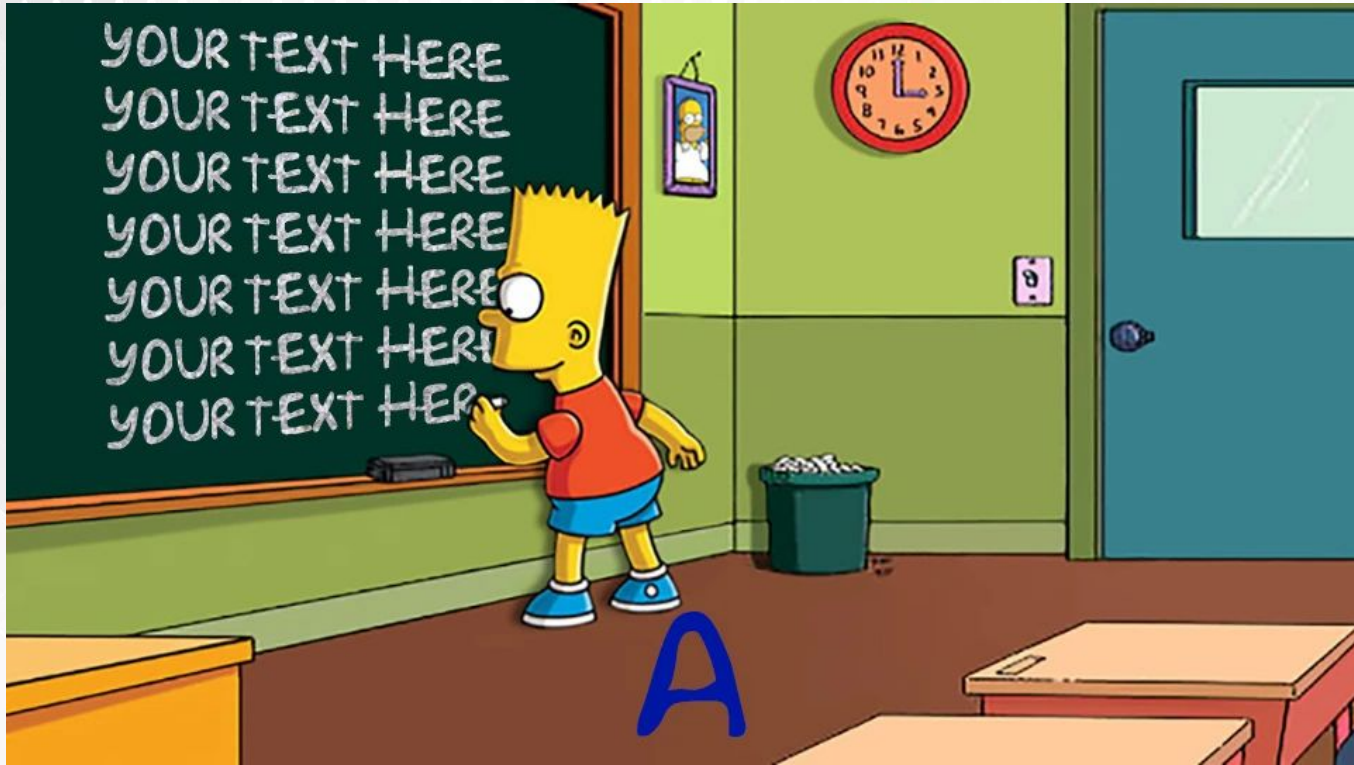
# Simplificando com when

Expressão When, pode substituir if/else complexos

```
fun main() {  
    val numero = 2  
  
    when(numero) {  
        1 → print("É pouco")  
        2 → print("É muito")  
        3, 4 → print("É demais")  
        else → print("Caiu no else")  
    }  
}
```



# Prática II





# Estruturas de repetição

## for

```

// Um loop que vai de 1 a 5

for (num in 1..5) {
    println("numero: $num")
}
```

## while

```

var x = 2

while (x > 0) {
    x--
}
```

# Estruturas de repetição

## for

```

// Um loop que vai de 1 a 5

for (num in 1..5) {
    println("numero: $num")
}
```

## while

```

var x = 2

while (x > 0) {
    x--
}
```

# Break and continue

**break** - Sai do loop

**continue** - Volta para o começo do loop

# Prática III