

Variáveis, Valores e Tipos em Go

Tenille Martins

DIO Tech Education



bytemartins



Tenille10



Tenille Martins

Objetivo Geral

Este curso foi planejado para programadores com conhecimento básico de Go; ao final deste Curso, o DEV conseguirá escrever programas com variáveis em Go e será capaz de ler e interpretar um Código, intermediário, dentro dessa linguagem.

Pré-requisitos

- Um computador com acesso a internet;
- Muita vontade de aprender;
- Curso Introdução a Linguagem Go
- Fundamentos em Go



Percurso

Etapa 1

Declaração, inicialização e atribuição

Etapa 2

Operador Curto

Etapa 3

Explorando Tipos

Percurso

Etapa 4

Conversão de Tipo

Etapa 5

O pacote func

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Declaração, inicialização e atribuição

// Variáveis, valores e tipos

Percurso

Etapa 1

Declaração, inicialização e atribuição

Etapa 2

Operador Curto

Etapa 3

Explorando Tipos

Percurso

Etapa 4

Conversão de Tipo

Etapa 5

O pacote func

Declaração, inicialização e atribuição

- Variável: local que armazena uma informação;
- Forma: var (**palavra reservada**) + nome (**identificação**) + string (**tipo**) = expressão
- Principais declarações: var, const, type, func;

Nomeando a variável algo engraçado



Esquecendo o que a variável é



Hands On!

```
package main
```

```
import "fmt"
```

```
const ebulicaoF = 212.0
```

```
func main() {
```

```
    var F = ebulicaoF
```

```
    var C = (F - 32) * 5 / 9
```

```
    fmt.Println("Ponto de ebulição da água em °F é:", F)
```

```
    fmt.Println("Ponto de ebulição da água em °C é:", C)
```

```
}
```

Percurso

Etapa 1

~~Declaração, inicialização e atribuição~~

Etapa 2

Operador Curto

Etapa 3

Explorando Tipos

Percurso

Etapa 4

Conversão de Tipo

Etapa 5

O pacote func

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Operador Curto

// Variáveis, valores e tipos

Percurso

Etapa 1

~~Declaração, inicialização e atribuição~~

Etapa 2

Operador Curto

Etapa 3

Explorando Tipos

Percurso

Etapa 4

Conversão de Tipo

Etapa 5

O pacote func

Operador Curto

- = para atribuir valores a variáveis já existentes
- Gopher := utilizado pra criar novas variáveis, dentro de code blocks.



Percurso

~~Etapa 1~~

~~Declaração, inicialização e atribuição~~

~~Etapa 2~~

~~Operador Curto~~

Etapa 3

Explorando Tipos

Percurso

Etapa 4

Conversão de Tipo

Etapa 5

O pacote func

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Explorando Tipos

// Variáveis, valores e tipos

Percurso

~~Etapa 1~~

~~Declaração, inicialização e atribuição~~

~~Etapa 2~~

~~Operador Curto~~

Etapa 3

Explorando Tipos

Percurso

Etapa 4

Conversão de Tipo

Etapa 5

O pacote func

Exemplos de Tipos

- Primitivos (básicos) : int, float, string, bool
- Compostos (criados pelo usuário): arrays, fatia, struct, mapas

Percurso

~~Etapa 1~~

~~Declaração, inicialização e atribuição~~

~~Etapa 2~~

~~Operador Curto~~

~~Etapa 3~~

~~Explorando Tipos~~

Percurso

Etapa 4

Conversão de Tipo

Etapa 5

O pacote func

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Tipos: Booleanos

// Variáveis, valores e tipos

Percurso

Etapa 1

~~O que são tipos~~

Etapa 2

~~Tipos: Número~~

Etapa 3

~~Tipos: String~~

Percurso

Etapa 4

Tipos: Booleanos

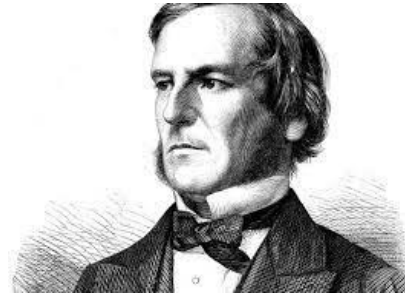
Etapa 5

Inferência de tipos

Etapa 6

Palavras reservadas

Tipos: Booleanos



- Homenagem ao matemático George Boole
- Representa Verdadeiro ou Falso (Ligado ou Desligado)
- Operadores lógicos:
 - && (e)
 - || (ou)
 - ! (negação)

Tabela Verdade

Expressão	Valor
true && true	true
true && false	false
false && true	false
false && false	false

Expressão	Valor
true true	true
true false	true
false true	true
false false	false

Expressão	Valor
!true	false
!false	true

Hands On!

```
package main
```

```
Import "fmt"
```

```
Func main(){  
    fmt.Println(true && true)  
    fmt.Println(true && false)  
    fmt.Println(true || true)  
    fmt.Println(true || false)  
    fmt.Println(!true)  
}
```

Percurso

Etapa 1

~~O que são tipos~~

Etapa 2

~~Tipos: Número~~

Etapa 3

~~Tipos: String~~

Percurso

Etapa 4

~~Tipos: Booleanos~~

Etapa 5

Inferência de tipos

Etapa 6

Palavras reservadas

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Inferência de Tipos

// Variáveis, valores e tipos

Percurso

~~Etapa 1~~

~~O que são tipos~~

~~Etapa 2~~

~~Tipos: Número~~

~~Etapa 3~~

~~Tipos: String~~

Percurso

Etapa 4

~~Tipos: Booleanos~~

Etapa 5

Inferência de tipos

Etapa 6

Palavras reservadas

Inferência de Tipos

- Go consegue inferir (decifrar) o tipo dessas variáveis. Ele consegue entender que, se a variável começa e termina com aspas, ela é uma string. Da mesma forma, se temos um número inteiro, sem casa decimal, o Go entenderá que a variável é do tipo inteiro

Hands On!



```
package main
```

```
Import "fmt"
```

```
Func main(){
```

```
    var nome string = Tenille
```

```
    var idade int = 36
```

```
    var versao float32 = 3.2
```

```
    fmt.Println( "Meu nome é", nome " e minha idade é",  
idade)
```

```
    fmt.Println("Estou usando o programa de versão:", versao
```

```
}
```

Percurso

~~Etapa 1~~

~~O que são tipos~~

~~Etapa 2~~

~~Tipos: Número~~

~~Etapa 3~~

~~Tipos: String~~

Percurso

~~Etapa 4~~

~~Tipos: Booleanos~~

~~Etapa 5~~

~~Inferência de tipos~~

Etapa 6

Palavras reservadas

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Links Úteis

- **Repositório no GitHub:**
<https://github.com/Tenille10/Introdu-o-a-linguagem-Go.git>
- **Documentação Oficial: https:**
[//go.dev/doc/](https://go.dev/doc/)
- **Referências:**
 - **Introdução a linguagem Go/ Caleb Doxsey**
 - **A linguagem de programação Go/ Alan Donovan e Brian Kernighan**

Para saber mais

- **Site na Web:** <https://go.dev/doc/>
- **Um curso da Dio:** Denilson Bonati <https://web.dio.me/course/logica-de-programacao-essencial-1/learning/e667ddc6-ca15-4f37-9291-dea944593898>
- **Livro:** Introdução à linguagem Go

Hands On!

"A questão não é saber os códigos ou linhas de comando, é saber o que fazer com eles."

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)

