

# Fundamentos de Go

**Tenille Martins**

DIO Tech Education



**bytemartins**



**Tenille10**



**Tenille Martins**

# Objetivo Geral

Este curso foi planejado para programadores sem nenhum conhecimento prévio de Go; ao final deste primeiro Curso, o DEV conseguirá escrever programas simples em Go e será capaz de ler e interpretar um código dentro dessa linguagem.

# Pré-requisitos

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



# 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

# Percurso

## Etapa 7

Expressões e Comandos

## Etapa 8

Variáveis

## Etapa 9

Blocos e Escopos

# Percurso

**Etapa 10**

Arrays, fatias e mapas

**Etapa 11**

Abstrações

**Etapa 12**

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções



# Dúvidas?

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



# O que são Tipos

// Fundamentos de Go

# 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

# Percurso

## Etapa 7

Expressões e Comandos

## Etapa 8

Variáveis

## Etapa 9

Blocos e Escopos

# Percurso

**Etapa 10**

Arrays, fatias e mapas

**Etapa 11**

Abstrações

**Etapa 12**

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# O que são Tipos



- Usamos um Tipo (String) para escrever nosso Hello World!
- Exemplo: Cachorro Tobias
- Linguagem estaticamente tipada



# Dúvidas?

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



# Tipos: Número

// Fundamentos de Go

# 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

# Percurso

## Etapa 7

Expressões e Comandos

## Etapa 8

Variáveis

## Etapa 9

Blocos e Escopos

# Percorso

## Etapa 10

# Arrays, fatias e mapas

## Etapa 11

# Abstrações

## Etapa 12

## Paramètres

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Tipos: Números

- Inteiros;
- Ponto Flutuante;



# Inteiros

- Números sem um componente decimal (-2,-1,0,1)
- Tem tamanho definido
- Tipos inteiros em Go: uint8, uint16, uint32, uint64, int8, int16, int32, int64
- Uint ( unsigned integer – inteiro sem sinal – somente +)
- Int (signed integer – inteiro com sinal - + e -)

# Curiosidade

- Uint8 → byte
- Int32 → rune

# Ponto Flutuante

- Números com componente decimal (1,523; 153,4)
- Tem tamanho definido
- Tipos Ponto Flutuantes em Go: float32, float64
- Em geral, usamos float64

Hands On!

```
package main
```

```
Import "fmt"
```

```
Func main(){  
    fmt.Println(" 2 + 3 =" , 2 + 3)  
}
```

# 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

# Percurso

## Etapa 7

Expressões e Comandos

## Etapa 8

Variáveis

## Etapa 9

Blocos e Escopos

# Percurso

**Etapas**

**Etapas 10**

Arrays, fatias e mapas

**Etapas 11**

Abstrações

**Etapas 12**

Paramêtros



# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Dúvidas?

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



# Tipos: String

// Fundamentos de Go

# 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

# Percurso

## Etapa 7

Expressões e Comandos

## Etapa 8

Variáveis

## Etapa 9

Blocos e Escopos

# Percurso

**Etapa 10**

Arrays, fatias e mapas

**Etapa 11**

Abstrações

**Etapa 12**

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções



# Tipos: String

- Representa um texto
- Sequência de caracteres
- Tamanho definido
- Vem entre “ ” ou ` `

# Operações comuns

- `Len("Hello World")` – Descobre o tamanho da String
- `"Hello World"[2]` – Acessa um caractere específico
- `"Hello" + " World"` – Concatena duas Strings

Hands On!

```
package main
```

```
import "fmt"
```

```
func main(){  
    fmt.Println(len(" Hello World"))  
    fmt.Println("Hello World" [2])  
    fmt.Println("Hello" + "World")  
}
```

# 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

# Percurso

## Etapa 7

Expressões e Comandos

## Etapa 8

Variáveis

## Etapa 9

Blocos e Escopos

# Percurso

**Etapa 10**

Arrays, fatias e mapas

**Etapa 11**

Abstrações

**Etapa 12**

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções



# Dúvidas?

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



# Tipos: Booleanos

// Fundamentos de Go

# 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

# Percurso

## Etapa 7

Expressões e Comandos

## Etapa 8

Variáveis

## Etapa 9

Blocos e Escopos

# Percurso

**Etapa 10**

Arrays, fatias e mapas

**Etapa 11**

Abstrações

**Etapa 12**

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# 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**

# Percurso

## Etapa 7

Expressões e Comandos

## Etapa 8

Variáveis

## Etapa 9

Blocos e Escopos

# Percurso

**Etapas**

**Etapas 10**

Arrays, fatias e mapas

**Etapas 11**

Abstrações

**Etapas 12**

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Dúvidas?

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





# Inferência de Tipos

// Fundamentos de Go

# 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**

# Percurso

## Etapa 7

Expressões e Comandos

## Etapa 8

Variáveis

## Etapa 9

Blocos e Escopos

# Percurso

**Etapa 10**

Arrays, fatias e mapas

**Etapa 11**

Abstrações

**Etapa 12**

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# 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**

# Percurso

## Etapa 7

Expressões e Comandos

## Etapa 8

Variáveis

## Etapa 9

Blocos e Escopos

# Percurso

**Etapas**

**Etapas 10**

Arrays, fatias e mapas

**Etapas 11**

Abstrações

**Etapas 12**

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Dúvidas?

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



# Palavras Reservadas

// Fundamentos de Go

# 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**

# Percurso

## Etapa 7

Expressões e Comandos

## Etapa 8

Variáveis

## Etapa 9

Blocos e Escopos

# Percurso

**Etapa 10**

Arrays, fatias e mapas

**Etapa 11**

Abstrações

**Etapa 12**

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Palavras Reservadas

- Palavras reservadas **são componentes da própria linguagem e não podem ser redefinidas, ou seja, denominar elementos criados pelo programador.**
- Exemplo: CONST, palavra reservada para indicar uma constante, não pode ser usada para quaisquer outro identificadores no programa.

# Palavras Reservadas

<code>break</code>	<code>default</code>	<code>func</code>	<code>interface</code>	<code>select</code>
<code>case</code>	<code>defer</code>	<code>go</code>	<code>map</code>	<code>struct</code>
<code>chan</code>	<code>else</code>	<code>goto</code>	<code>package</code>	<code>switch</code>
<code>const</code>	<code>fallthrough</code>	<code>if</code>	<code>range</code>	<code>type</code>
<code>continue</code>	<code>for</code>	<code>import</code>	<code>return</code>	<code>var</code>

# 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~~**



# Percurso

## Etapa 7

Expressões e Comandos

## Etapa 8

Variáveis

## Etapa 9

Blocos e Escopos

# Percorso

## Etapa 10

# Arrays, fatias e mapas

## Etapa 11

# Abstrações

## Etapa 12

## Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Dúvidas?

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



# Expressões e Comandos

// Fundamentos de Go

# 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~~**

# Percurso

## Etapa 7

Expressões e Comandos

## Etapa 8

Variáveis

## Etapa 9

Blocos e Escopos



# Percorso

## Etapa 10

# Arrays, fatias e mapas

## Etapa 11

# Abstrações

## Etapa 12

## Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Expressões e Comandos

- São comandos (informações) fornecidas em forma de código para que o programe execute uma determinada função.

# Expressões e Comandos

- Comando de repetição: **for**
- Comando atribuição: **:=**
- Comando declaração: **var**
- Comando Condicionais: **if, switch**

# 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~~**

# Percurso

~~Etapa 7~~

~~Expressões e Comandos~~

Etapa 8

Variáveis

Etapa 9

Blocos e Escopos

# Percurso

**Etapa 10**

Arrays, fatias e mapas

**Etapa 11**

Abstrações

**Etapa 12**

Paramêtros



# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Dúvidas?

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



# Variáveis

// Fundamentos de Go

# 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~~**

# Percurso

**Etapa 7**

~~Expressões e Comandos~~

**Etapa 8**

**Variáveis**

**Etapa 9**

**Blocos e Escopos**

# Percorso

## Etapa 10

# Arrays, fatias e mapas

## Etapa 11

# Abstrações

## Etapa 12

## Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções



# Variáveis

- Área de armazenagem de um TIPO específico;
- Tem um nome associado;

Hands On!

```
package main
```

```
import "fmt"
```

```
func main(){  
    var x string = "Hello, World!"  
    fmt.Println(x)  
}
```

Hands On!

```
package main
```

```
import "fmt"
```

```
func main(){  
    var x string  
    x = "Hello, World!"  
    fmt.Println(x)  
}
```

# 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~~**

# Percurso

~~Etapa 7~~

~~Expressões e Comandos~~

~~Etapa 8~~

~~Variáveis~~

Etapa 9

Blocos e Escopos

# Percurso

**Etapa 10**

Arrays, fatias e mapas

**Etapa 11**

Abstrações

**Etapa 12**

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções



# Dúvidas?

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



# Blocos e escopos

// Fundamentos de Go

# 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~~**

# Percurso

~~Etapa 7~~

~~Expressões e Comandos~~

~~Etapa 8~~

~~Variáveis~~

Etapa 9

Blocos e Escopos

# Percurso

**Etapas**

**Etapas 10**

Arrays, fatias e mapas

**Etapas 11**

Abstrações

**Etapas 12**

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Blocos

- os **blocos** são a unidade fundamental e podem representar comandos, condições, objetos e muitas outras variáveis que fazem parte da construção de um programa.



# Escopo

- **Escopo** é a acessibilidade de objetos, variáveis e funções em diferentes partes do código. Em outras palavras, o que determina quais são os dados que podem ser acessados em uma determinada parte do código é o **escopo**.

Hands On!

```
package main
```

```
import "fmt"
```

```
func main(){  
    var x string = "Hello, World!"  
    fmt.Println(x)  
}
```

## Hands On!

```
package main
```

```
import "fmt"
```

```
var x string = "Hello, World!"
```

```
func main(){  
    fmt.Println(x)  
}
```

# 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~~**

# Percurso

**~~Etapa 7~~**

**~~Expressões e Comandos~~**

**~~Etapa 8~~**

**~~Variáveis~~**

**~~Etapa 9~~**

**~~Blocos e Escopos~~**

# Percurso

## Etapa 10

Arrays, fatias e mapas

## Etapa 11

Abstrações

## Etapa 12

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções



# Dúvidas?

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



# Arrays, fatias e mapas

// Fundamentos de Go

# 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~~**

# Percurso

**~~Etapa 7~~**

**~~Expressões e Comandos~~**

**~~Etapa 8~~**

**~~Variáveis~~**

**~~Etapa 9~~**

**~~Blocos e Escopos~~**

# Percorso

## Etapa 10

# Arrays, fatias e mapas

## Etapa 11

# Abstrações

## Etapa 12

## Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Arrays

- Sequência numerada;
- Tem um único TIPO;
- Tamanho fixo;



# Fatias

- É uma parte (fatia) do Array;
- Tem um único TIPO;
- Tamanho variável;

# Mapas

- Buscar um valor de acordo com a palavra associada;
- Pode ser chamado de tabelas hash, arrays associativos ou dicionários;

# 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~~**

# Percurso

**~~Etapa 7~~**

**~~Expressões e Comandos~~**

**~~Etapa 8~~**

**~~Variáveis~~**

**~~Etapa 9~~**

**~~Blocos e Escopos~~**

# Percurso

**Etapa 10**

~~Arrays, fatias e mapas~~

**Etapa 11**

Abstrações

**Etapa 12**

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Dúvidas?

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





# Abstrações

// Fundamentos de Go

# 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~~**

# Percurso

**~~Etapa 7~~**

**~~Expressões e Comandos~~**

**~~Etapa 8~~**

**~~Variáveis~~**

**~~Etapa 9~~**

**~~Blocos e Escopos~~**

# Percurso

**~~Etapa 10~~**

~~**Arrays, fatias e mapas**~~

**Etapa 11**

**Abstrações**

**Etapa 12**

**Paramêtros**

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Abstrações

- O conceito de abstração consiste em esconder os detalhes de algo, no caso, os detalhes desnecessários.

# 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~~**

# Percurso

**~~Etapa 7~~**

**~~Expressões e Comandos~~**

**~~Etapa 8~~**

**~~Variáveis~~**

**~~Etapa 9~~**

**~~Blocos e Escopos~~**

# Percurso

~~Etapa 10~~

~~Arrays, fatias e mapas~~

~~Etapa 11~~

~~Abstrações~~

Etapa 12

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Dúvidas?

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



# Parâmetros

// Fundamentos de Go

# 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~~**



# Percurso

**~~Etapa 7~~**

**~~Expressões e Comandos~~**

**~~Etapa 8~~**

**~~Variáveis~~**

**~~Etapa 9~~**

**~~Blocos e Escopos~~**

# Percurso

~~Etapa 10~~

~~Arrays, fatias e mapas~~

~~Etapa 11~~

~~Abstrações~~

Etapa 12

Paramêtros

# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Parâmetros

- é um valor, proveniente de uma variável ou de uma expressão mais complexa, que pode ser passado para uma função interna (sub-rotina), que utiliza os valores atribuídos aos parâmetros para alterar o seu comportamento em tempo de execução.

# 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~~**

# Percurso

**~~Etapa 7~~**

**~~Expressões e Comandos~~**

**~~Etapa 8~~**

**~~Variáveis~~**

**~~Etapa 9~~**

**~~Blocos e Escopos~~**

# Percurso

**~~Etapa 10~~**

**~~Arrays, fatias e mapas~~**

**~~Etapa 11~~**

**~~Abstrações~~**

**~~Etapa 12~~**

**~~Paramêtros~~**



# Percurso

Etapa 13

Objetos e Classes

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Dúvidas?

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



# Objetos e Classes

// Fundamentos de Go

# 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~~**

# Percurso

**~~Etapa 7~~**

**~~Expressões e Comandos~~**

**~~Etapa 8~~**

**~~Variáveis~~**

**~~Etapa 9~~**

**~~Blocos e Escopos~~**

# Percurso

**~~Etapa 10~~**

**~~Arrays, fatias e mapas~~**

**~~Etapa 11~~**

**~~Abstrações~~**

**~~Etapa 12~~**

**~~Parâmetros~~**

# Percurso

**Etapa 13**

**Objetos e Classes**

**Etapa 14**

**Sistemas de tipos**

**Etapa 15**

**Interface e exceções**



# Objetos e Classes

- Go não tem objetos e classes, porém tem Estrutura e Funções que simulam ambas em outras linguagens.

# Estrutura

- Estrutura guarda apenas estado e não tem comportamento.

```
type Pessoa struct {  
    Nome string  
    Sobrenome string  
    Idade int8  
}
```

```
pessoa1 := Pessoa{"Tiago", "Temporin", 31}  
  
pessoa2 := Pessoa{Nome: "João", Sobrenome: "Silva", Idade:  
35}  
  
pessoa3 := Pessoa{Nome: "Aline", Idade: 25}
```

# 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~~**

# Percurso

**~~Etapa 7~~**

**~~Expressões e Comandos~~**

**~~Etapa 8~~**

**~~Variáveis~~**

**~~Etapa 9~~**

**~~Blocos e Escopos~~**

# Percurso

**~~Etapa 10~~**

**~~Arrays, fatias e mapas~~**

**~~Etapa 11~~**

**~~Abstrações~~**

**~~Etapa 12~~**

**~~Paramêtros~~**

# Percurso

~~Etapa 13~~

~~Objetos e Classes~~

Etapa 14

Sistemas de tipos

Etapa 15

Interface e exceções

# Dúvidas?

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





# Sistemas de Tipos

// Fundamentos de Go

# 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~~**

# Percurso

**~~Etapa 7~~**

**~~Expressões e Comandos~~**

**~~Etapa 8~~**

**~~Variáveis~~**

**~~Etapa 9~~**

**~~Blocos e Escopos~~**

# Percurso

~~Etapa 10~~

~~Arrays, fatias e mapas~~

~~Etapa 11~~

~~Abstrações~~

~~Etapa 12~~

~~Parâmetros~~

# Percurso

**Etapa 13**

~~Objetos e Classes~~

**Etapa 14**

**Sistemas de tipos**

**Etapa 15**

**Interface e exceções**

# Sistemas de Tipos

- é um conjunto de regras que atribuem uma propriedade chamada de **tipo** para as várias construções - tais como variáveis, expressões, funções - que um programa de computador é composto.

# 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~~**

# Percurso

**~~Etapa 7~~**

**~~Expressões e Comandos~~**

**~~Etapa 8~~**

**~~Variáveis~~**

**~~Etapa 9~~**

**~~Blocos e Escopos~~**

# Percurso

**~~Etapa 10~~**

**~~Arrays, fatias e mapas~~**

**~~Etapa 11~~**

**~~Abstrações~~**

**~~Etapa 12~~**

**~~Parâmetros~~**

# Percurso

~~Etapa 13~~

~~Objetos e Classes~~

~~Etapa 14~~

~~Sistemas de tipos~~

Etapa 15

Interface e exceções

# Dúvidas?

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



# Interfaces e Exceções

// Fundamentos de Go

# 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~~**



# Percurso

**~~Etapa 7~~**

**~~Expressões e Comandos~~**

**~~Etapa 8~~**

**~~Variáveis~~**

**~~Etapa 9~~**

**~~Blocos e Escopos~~**

# Percurso

**~~Etapa 10~~**

**~~Arrays, fatias e mapas~~**

**~~Etapa 11~~**

**~~Abstrações~~**

**~~Etapa 12~~**

**~~Paramêtros~~**

# Percurso

**Etapa 13**

**~~Objetos e Classes~~**

**Etapa 14**

**~~Sistemas de tipos~~**

**Etapa 15**

**Interface e exceções**

# Interface e Exceções

- Uma interface define um comportamento de um tipo;
- Go não possui exceções;

# 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~~**

# Percurso

**~~Etapa 7~~**

**~~Expressões e Comandos~~**

**~~Etapa 8~~**

**~~Variáveis~~**

**~~Etapa 9~~**

**~~Blocos e Escopos~~**

# Percurso

**~~Etapa 10~~**

**~~Arrays, fatias e mapas~~**

**~~Etapa 11~~**

**~~Abstrações~~**

**~~Etapa 12~~**

**~~Paramêtros~~**



# Percurso

~~Etapa 13~~

~~Objetos e Classes~~

~~Etapa 14~~

~~Sistemas de tipos~~

~~Etapa 15~~

~~Interface e exceções~~

# 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)

