

# Introdução ao Go

## Aula 10: Tipos em Go & Valores Nulos

Nesta aula, o professor introduz os **tipos básicos** em Go, que são os tipos de dados fundamentais na linguagem. Além disso, ele explica o conceito de **valores nulos** em Go e como os valores padrão são atribuídos a variáveis de diferentes tipos quando não recebem um valor explícito.

---

### Tipos Básicos em Go

Go possui alguns **tipos básicos** embutidos, que são usados para representar valores comuns. Aqui estão os principais:

#### 1. `int`:

- Representa números **inteiros** (sem casas decimais).
- Exemplos: `-5`, `10`, `12`.

#### 2. `float64`:

- Representa números com **casas decimais**.
- Exemplos: `-5.2`, `10.123`, `12.9`.

#### 3. `string`:

- Representa **textos**.
- Exemplos: `"Olá, Mundo!"`, `Hi there!`.
- Pode ser criada com aspas duplas ou backticks.

#### 4. `bool`:

- Representa valores **verdadeiros ou falsos**.
  - Exemplos: `true`, `false`.
- 

### Tipos Menos Comuns

Existem também **tipos específicos** que você pode encontrar em situações menos frequentes:

1. **uint:**

- Um **inteiro sem sinal**, ou seja, sempre **não-negativo**.
- Exemplos: `0`, `10`, `255`.

2. **int32:**

- Um **inteiro assinado de 32 bits**.
- Faixa de valores: de `-2,147,483,648` a `2,147,483,647`.

3. **rune:**

- Um **alias para int32**, representando **pontos de código Unicode** (usado para caracteres, como letras e símbolos).
- Exemplos: `'a'`, `'ñ'`, `'世'`.

4. **uint32:**

- Um **inteiro sem sinal de 32 bits**.
- Faixa de valores: de `0` a `4,294,967,295`.

5. **int64:**

- Um **inteiro assinado de 64 bits**.
- Faixa de valores: de `-9,223,372,036,854,775,808` a `9,223,372,036,854,775,807`.

6. **outros tipos:**

- Existem tipos menores como **int8** e **uint8** que têm um intervalo de valores mais restrito.

---

## Valores Nulos

Em Go, cada tipo possui um **valor nulo** associado, que é o valor padrão atribuído a uma variável quando ela é declarada, mas não inicializada explicitamente.

- **int:** O valor nulo é `0`.
- **float64:** O valor nulo é `0.0`.
- **string:** O valor nulo é uma **string vazia** (`""`).

- **bool:** O valor nulo é `false`.

Exemplo:

go

```
var age int // 0 valor de age será 0
var name string // 0 valor de name será ""
var active bool // 0 valor de active será false
```

Isso é útil, pois quando você declara uma variável sem inicializá-la, ela já tem um valor válido (o "valor nulo") para evitar comportamentos inesperados no código.

---

## Resumo dos Valores Nulos

Aqui está uma lista dos valores nulos para os tipos mais comuns:

- **int:** `0`
- **float64:** `0.0`
- **string:** `""` (uma string vazia)
- **bool:** `false`

---

## Exemplo Prático

go

```
package main

import "fmt"

func main() {
    // Declarando variáveis sem inicialização explícita
    var age int // 0 valor padrão é 0
    var height float64 // 0 valor padrão é 0.0
    var name string // 0 valor padrão é uma string vazia ""
    var isActive bool // 0 valor padrão é false
```

```
// Exibindo os valores padrão
fmt.Println("Age:", age)
fmt.Println("Height:", height)
fmt.Println("Name:", name)
fmt.Println("Active:", isActive)
}
```

Saída:

vbnet

```
Age: 0
Height: 0
Name:
Active: false
```

Esses conceitos são essenciais para trabalhar com Go, pois ajudam a entender como a linguagem lida com tipos de dados e valores nulos. Se tiver mais dúvidas ou quiser explorar algum ponto específico, é só falar!