

Introdução ao Go

Resumo da Aula 28 - Introdução ao "if" e Booleans em Go

Nesta aula, foi introduzida a estrutura condicional "if" e o tipo de dado Boolean (`bool`) na linguagem Go. O objetivo foi utilizar essas estruturas para tomar decisões dentro do programa bancário iniciado na aula anterior.

Pontos principais da aula:

1. Necessidade do "if" no Código

- O programa precisa executar ações diferentes com base na escolha do usuário.
- O `if` permite verificar essa escolha e executar o código correspondente.

2. Sintaxe do "if"

- Usa a palavra-chave `if`, seguida de uma condição dentro de parênteses.
- A condição deve resultar em `true` ou `false`.
- Exemplo:

```
go

if choice == 1 {
    fmt.Println("Você escolheu verificar o saldo.")
}
```

- **Importante:** O operador `==` é usado para comparar valores, enquanto `=` é usado para atribuições.

3. Operadores Lógicos

- `&&` (E lógico): ambas as condições devem ser verdadeiras.
- `||` (OU lógico): basta que uma das condições seja verdadeira.
- Exemplo:

```
go

if choice == 1 || choice == 2 {
    fmt.Println("Você escolheu uma opção válida.")
}
```

```
}
```

4. Introdução ao Tipo Booleano (`bool`)

- O resultado de uma comparação (`==`, `<`, `>`, etc.) é um `bool` (`true` ou `false`).
- É possível armazenar esse resultado em uma variável:

```
go

wantsCheckBalance := choice == 1
if wantsCheckBalance {
    fmt.Println("Verificando saldo...")
}
```

- O tipo `bool` é fundamental para estruturas de controle e tomadas de decisão.

5. Exemplo Prático: Exibir o Saldo da Conta

- Criada a variável `accountBalance` inicializada com `1000.0` (`float64`, permitindo valores decimais).
- Se o usuário escolher a opção `1`, o saldo será exibido.
- Exemplo:

```
go

accountBalance := 1000.0
if choice == 1 {
    fmt.Println("Seu saldo é:", accountBalance)
}
```

- O programa agora responde à entrada do usuário de forma dinâmica.

Próximos Passos

- Implementar mais condições para **depósito e saque**.
- Adicionar **tratamento de erros** (ex.: valores inválidos).
- Explorar **estruturas de controle mais avançadas**, como `else` e `switch`.

Esta aula marcou o início do uso de **decisões lógicas** no código, tornando o programa mais interativo!