

Introdução ao Go

Aula 25: Uma Sintaxe Alternativa para Retorno de Valores

Nesta aula, exploramos uma sintaxe alternativa para definir e retornar valores em funções Go. Além da abordagem tradicional, onde usamos o comando `return` seguido de variáveis ou valores, é possível dar **nomes** aos valores de retorno diretamente na definição da função.

A Sintaxe Alternativa

Como Funciona?

1. Ao definir os **tipos de retorno** da função, você pode atribuir **nomes** a eles.
2. Go cria automaticamente **variáveis** com esses nomes.
3. Você pode **atribuir valores diretamente** a essas variáveis, sem precisar declará-las.
4. Para retornar os valores, você pode usar o comando `return` sem especificar as variáveis, já que Go retorna automaticamente os valores atribuídos.

Exemplo Comparativo

Forma Tradicional

```
go

package main

import (
    "fmt"
    "math"
)

// Forma tradicional
func calcularValoresFuturosTradicional(investimento float64, taxa float64, anos
float64) (float64, float64) {
    FV := investimento * math.Pow(1+taxa/100, anos)
```

```

    RFV := FV / 1.05
    return FV, RFV
}

func main() {
    futuro, futuroReal := calcularValoresFuturosTradicional(1000, 5, 10)
    fmt.Println("Forma Tradicional:")
    fmt.Println("Valor futuro:", futuro)
    fmt.Println("Valor futuro ajustado pela inflação:", futuroReal)
}

```

Saída:

plaintext

```

Forma Tradicional:
Valor futuro: 1628.89
Valor futuro ajustado pela inflação: 1552.27

```

Sintaxe Alternativa

go

```

package main

import (
    "fmt"
    "math"
)

// Sintaxe alternativa
func calcularValoresFuturos(investimento float64, taxa float64, anos float64) (FV float64, RFV float64) {
    // Variáveis FV e RFV já existem
    FV = investimento * math.Pow(1+taxa/100, anos)
    RFV = FV / 1.05
    // Retorno automático das variáveis nomeadas
    return
}

```

```
func main() {  
    futuro, futuroReal := calcularValoresFuturos(1000, 5, 10)  
    fmt.Println("Sintaxe Alternativa:")  
    fmt.Println("Valor futuro:", futuro)  
    fmt.Println("Valor futuro ajustado pela inflação:", futuroReal)  
}
```

Saída:

plaintext

```
Sintaxe Alternativa:  
Valor futuro: 1628.89  
Valor futuro ajustado pela inflação: 1552.27
```

Quando Usar?

Vantagens da Sintaxe Alternativa

- **Código mais curto:** Não há necessidade de declarar explicitamente as variáveis.
- **Mais semântica:** Os nomes dos valores de retorno adicionam clareza sobre o significado dos dados que a função retorna.

Desvantagens da Sintaxe Alternativa

- **Menos explícita:** Em funções longas, pode ser confuso identificar o que está sendo retornado, já que você precisa verificar os nomes definidos na assinatura da função.
- **Complexidade em manutenção:** Caso a função seja alterada, localizar as variáveis nomeadas pode exigir mais esforço.

Recomendações

- **Para funções curtas e simples:** A sintaxe alternativa pode ser útil para clareza e concisão.

- **Para funções longas ou complexas:** Considere usar a forma tradicional para explicitar os valores retornados, evitando confusões.
-

Resumo

A sintaxe alternativa permite nomear os valores de retorno diretamente na assinatura da função. Isso elimina a necessidade de declarar variáveis separadamente e possibilita o uso do comando `return` de forma implícita. No entanto, essa abordagem deve ser usada com cuidado, especialmente em funções maiores, para evitar ambiguidade.

Com essa técnica, você tem mais flexibilidade na definição de funções e pode escolher o estilo que melhor se adapta ao seu projeto!