

## Sessão - Funções em Go

### 30. Função:

Uma função é declarada com a palavra-chave `func``.

Exemplo:

```
package main
```

```
import "fmt"
```

```
func saudacao() {  
    fmt.Println("Olá, seja bem-vindo!")  
}
```

```
func main() {  
    saudacao()  
}
```

### 31. Função com Retorno:

Funções podem retornar valores em Go.

Exemplo:

```
package main
```

```
import "fmt"
```

```
func soma(a int, b int) int {  
    return a + b  
}
```

```
func main() {  
    resultado := soma(3, 4)  
    fmt.Println("Resultado da soma:", resultado)  
}
```

## 32. Retorno de Mais de Um Valor:

Go permite que funções retornem múltiplos valores.

Exemplo:

```
package main
```

```
import "fmt"
```

```
func dividir(a int, b int) (int, bool) {  
    if b == 0 {  
        return 0, false  
    }  
    return a / b, true  
}
```

```
func main() {  
    resultado, sucesso := dividir(10, 2)  
  
    if sucesso {  
        fmt.Println("Resultado:", resultado)  
    } else {  
        fmt.Println("Erro: divisão por zero")  
    }  
}
```

### 33. Retorno Nomeado:

Go permite nomear valores de retorno diretamente.

Exemplo:

```
package main
```

```
import "fmt"
```

```
func calculo(a int, b int) (resultado int) {  
    resultado = a * b  
  
    return  
}
```

```
func main() {  
    fmt.Println("Resultado do cálculo:", calculo(5, 3))  
}
```

```
}
```

#### 34. Escopo:

Variáveis só podem ser usadas dentro do escopo onde são declaradas.

Exemplo:

```
package main
```

```
import "fmt"
```

```
func exemploEscopo() {
```

```
    x := 10
```

```
    fmt.Println("Dentro do escopo:", x)
```

```
}
```

```
func main() {
```

```
    exemploEscopo()
```

```
}
```

#### 35. Init:

A função ``init`` é executada automaticamente antes do ``main``.

Exemplo:

```
package main
```

```
import "fmt"
```

```
func init() {  
    fmt.Println("Inicializando o programa...")  
}
```

```
func main() {  
    fmt.Println("Executando o main.")  
}
```