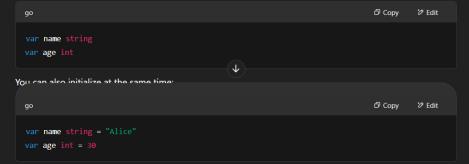


variables in go

In Go (Golang), variables are used to store data. Here's a breakdown of how you declare and use variables in Go:

◆ 1. Declaring Variables

A. Using var Keyword

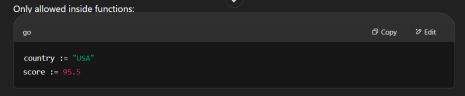


B. Type Inference

Go can infer the type from the value:



C. Short Declaration (:=)



2. Multiple Variable Declaration

```
go

Var x, y, z int = 1, 2, 3

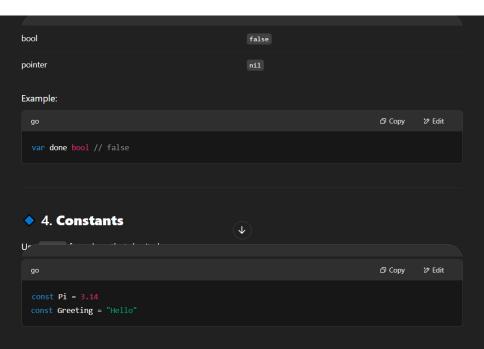
Or using short declaration:

a, b := 4, "hello"
```

3. Zero Values

If you declare a variable without initializing it, Go assigns a zero value:

Туре	Zero Value
int	
float64	0.0
string	···



5. Variable Scope

- Variables declared outside functions are package-level (global within the package).
- Variables declared inside functions are local to that function.

Let me know if you want a hands-on example or explanation of global vs local scope, constants, or variable naming rules.

add some examples

Sure! Here are some practical examples of how variables are declared and used in Go:

1. Basic Variable Declaration

```
go

Copy Deficit

puccope mean

import "fmt"

func main() {
   var name string = "John"
   var age int = 25
   fmt.Println("Name:", name)
   fmt.Println("Age:", age)
}
```

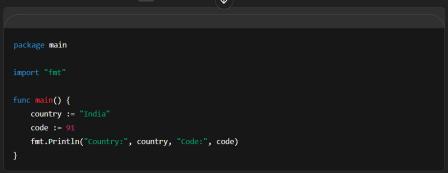
```
✓ 2. Type Inference

package main

import "fmt"
```

```
runc main() {
   var city = "London" // Go infers string
   var population = 8900000 // Go infers int
   fmt.Println(city, "has population", population)
}
```

✓ 3. Short Declaration (:=)



4. Multiple Variable Declaration



 $\overline{\Psi}$

y o. Constants

```
go
package main

import "fmt"

func main() {
    const Pi = 3.14159
    const Language = "Go"
    fmt.Println("Pi:", Pi)
    fmt.Println("Language:", Language)
}
```

6. Zero Values (default values)

