

# Nested Structure in Golang

Go language allows nested structures. A structure that is a field of another structure is called Nested Structure.

A structure or struct in Golang is a user-defined type that allows us to create a group of elements of different types into a single unit. Any real-world entity that has some set of properties or fields can be represented as a struct.



## How to use nested structures in Golang

**Go language allows nested structures. A structure that is a field of another structure is called a Nested Structure.** In other words, a structure inside another structure is called a Nested Structure. **Syntax:**

```
type struct_name_1 struct{ // Các tr??
ng } type struct_name_2 struct{ variable_name struct_name_1 }
```

**Consider the following example to understand nested structure in Golang:**

### Example 1:

```
// Ch??ng tr??nh Go minh ho??c // c??u tr??c l?
ng nhau package main import "fmt" // T??o c?
u tr??c type Author struct { name string branch string year int } // T??o c?
u tr??c lồng nhau type HR struct { // c??u tr??c là m?t tr??
ng details Author } func main() { // Kh??i t??o các tr??ng // c?a c?
u tr??c result := HR{ details: Author{"Sona", "ECE", 2013}, } // Hi??n giá tr
? fmt.Println("nDetails of Author") fmt.Println(result) }
```

Result:

Details of Author {{Sona ECE 2013}}

### Example 2:

```
// Chương trình Golang minh họa // cấu trúc lồng nhau package main import "fmt" // Tạo cấu trúc type Student struct { name string branch string year int } // Tạo cấu trúc type Teacher struct { name string subject string exp int details Student } // Tạo các trường // Cấu trúc result := Teacher{ name: "Suman", subject: "Java", exp: 5, details: Student{ name: "Sona", branch: "ECE", year: 2013 } } // In giá trị fmt.Println("Details of the Teacher") fmt.Println("Teacher's name: ", result.name)
```

Result:

Details of the Teacher Teacher's name: Suman Subject: Java Experience: 5 Details of the Teacher

### Example 3:

In Go, a structure can have fields that are themselves structures, called nested structures. Here is an example of a structure that has a nested structure:

```
package main import ( "fmt" ) type Address struct { Street string City string State string PostalCode int } type Person struct { Name string Age int Address Address } // Tạo các trường // Cấu trúc result := Person{ Name: "John Doe", Age: 30, Address: Address{ Street: "123 Main St", City: "Anytown", State: "CA", PostalCode: 90210 } } // In giá trị fmt.Println("John Doe Age: 30 Address: Street: 123 Main St City: Anytown State: CA Postal Code: 90210")
```

Result:

John Doe Age: 30 Address: Street: 123 Main St City: Anytown State: CA Postal Code: 90210

Here, we define two struct types: Person and Address. Person has a nested struct field called Address. In the main function, we create a new instance of Person with an Address field. Then, we print the values of various fields of the Person and Address structs using dot notation to access the nested fields.

You finished reading the article "**Nested Structure in Golang**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.