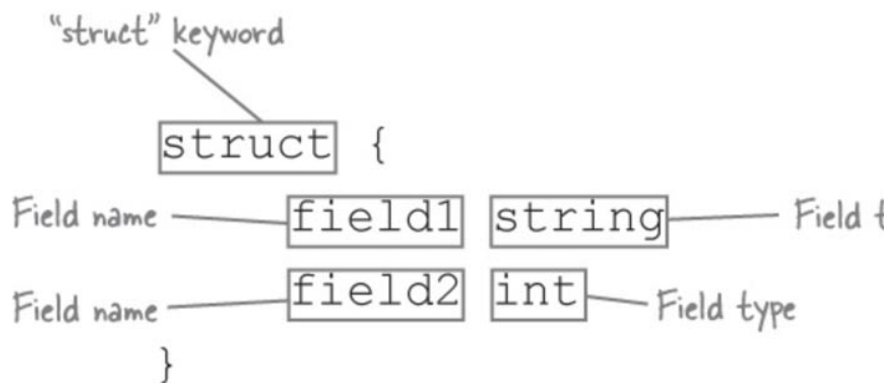


Recommended fields in Golang

In Go structs, fields are encouraged to be the same as anonymous fields, the type of the field is the name of the field.

In Go structs, fields are encouraged to be the same as anonymous fields, the type of the field is the name of the field.



We use this concept in nested structures, where a structure is a field in another structure, just add the name of the structure to another structure and it works like Anonymous Field for the nested structure. And the fields of that structure (other than the nested structure) are part of the nested structure, such type of field is called promoted field. If the anonymous structure or nested structure and the parent structure contain a field with the same name, then that field is not promoted, only fields with different names are promoted to the structure.

Syntax:

```

type x struct{ // Các trường } type y struct{ // Các trường của x
u trúc y x }
  
```

Now let's take a closer look at how to use this recommended field concept in Golang through an example:

For example:

```

// Chương trình Go minh họa // khái niệm các trường lồng nhau
package main import "fmt" // Cú pháp
u trúc type details struct { // Các trường // cấu trúc chi tiết
  
```

```

t name string age int gender string } // C?u tr?c l?
ng nhau type student struct { branch string year int details } func main() { // l
?i t?o các tr??ng c?a // c?u tr?c h?
c sinh values := student{ branch: "CSE", year: 2010, details: details{ name: "Sum
??ng ???c khuy?n kh?ch c?a c?
u tr?c hoc sinh fmt.Println("Name: ", values.name) fmt.Println("Age: ", values.a
??ng bình th??ng c?a // c?u tr?c h?
c sinh fmt.Println("Year: ", values.year) fmt.Println("Branch : ", values.branch

```

Result:

```
Name: Sumit Age: 28 Gender: Male Year: 2010 Branch : CSE
```

Detailed explanation:

In the above example, we have two structures named **details** and **student** . Where **details structure** is a normal structure and **student structure** is a nested structure containing **details structure** as fields which are like anonymous fields. Now, the fields of **details structure** i.e. name, age and gender are promoted to **student structure** and are called promoted fields. Now, you can directly access them with the help of **student structure** like **values.name** , **values.age** and **values.gender** .

You finished reading the article "**Recommended fields 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.