

Chương trình Go Hello World

Chương trình Go

Trong phần này chúng ta sẽ học cách sử dụng Go bằng cách viết một chương trình quen thuộc là chương trình *Hello World*.

Chúng ta tạo một file text có tên `main.go` với nội dung sau:

```
package main

import "fmt"

// this is a comment

func main() {
    fmt.Println("Hello World")
}
```

Sau đó để dịch và chạy chương trình thì bạn vào Command Prompt (cmd) rồi gõ lệnh `go run <tên file>`. Bạn có thể phải chỉ ra cả đường dẫn đến tên file nếu thư mục hiện tại trong cmd không trùng với thư mục chứa file code. Chẳng hạn file source bạn để trong thư mục **C:/Go/main.go** thì thư mục hiện tại trong cmd phải là **C:/Go**.

```
C:\Go>go run main.go
Hello World
```

Nếu bạn làm đúng các bước trên thì sau khi gõ lệnh màn hình console sẽ in ra dòng chữ *Hello World*. Nếu không phải thì tức là bạn đã thực hiện sai ở bước nào đó, có thể là gõ sai code, trình biên dịch cũng sẽ thông báo lỗi ngay cho bạn. Và cũng giống như bất cứ trình biên dịch khác, chỉ cần trong code của bạn có lỗi thì cả chương trình sẽ không thể chạy được.

Giải thích

Một đoạn code chương trình Go được thực thi từ trên xuống dưới và từ trái sang phải giống như đọc một cuốn sách vậy.

```
package main
```

Dòng đầu tiên là `package main`, đây là câu lệnh khai báo "gói". Tất cả mọi chương trình Go đều phải được bắt đầu bởi câu lệnh khai báo gói. Tính năng Gói có tác dụng giúp chúng ta tổ chức code và tái sử dụng code dễ dàng hơn, chúng ta sẽ tìm hiểu thêm về gói trong các bài sau.

Khi biên dịch code Go thì có 2 loại là biên dịch thành chương trình chạy trực tiếp (**executable**) và biên dịch thành thư viện (**library**). Chương trình chạy trực tiếp là các file khả thi (có đuôi .exe trong Windows) có thể chạy một cách trực tiếp trong terminal (Command Prompt trong Windows). Còn thư viện là tập hợp code được gom lại với nhau và có thể được sử dụng trong các chương trình khác, chúng ta sẽ tìm hiểu về thư viện sau. Hiện tại chỉ bạn chỉ cần biết là cần phải có câu lệnh khai báo package trong code của mình.

Sau dòng khai báo package là một dòng trống, giống như các ngôn ngữ khác, trình biên dịch không quan tâm các khoảng trống này, chúng ta chỉ dùng chúng để đọc code cho dễ hơn thôi.

```
import "fmt"
```

Từ khóa `import` có nghĩa là chúng ta yêu cầu được sử dụng code từ các gói khác trong chương trình của chúng ta. Ở đây là gói `fmt` (viết tắt của *format*), gói này chủ yếu chứa code thực hiện việc định dạng dữ liệu ra/vào.

Khi dùng từ khóa `import` thì tên gói được đặt trong cặp dấu nháy kép. Những kí tự được bọc trong cặp dấu nháy kép đều được gọi chung là chuỗi (hoặc string trong tiếng Anh), chúng ta sẽ tìm hiểu về string sau.

```
// this is a comment
```

Ký tự `//` cho biết những kí tự đứng sau nó là câu bình luận (comment). Các câu bình luận sẽ không được biên dịch. Trong Go có 2 loại comment, `//` là loại comment trên một dòng, tất cả các kí tự phía sau `//` sẽ không được biên dịch, và `/* */` là loại comment có thể sử dụng trên nhiều dòng, tất cả các kí tự nằm trong cặp dấu `/* */` sẽ không được biên dịch.

```
func main() {  
    fmt.Println("Hello World");  
}
```

Tiếp theo là phần khai báo hàm. Hàm là các thành phần xây dựng nên một chương trình. Hàm nhận dữ liệu vào, xử lý dữ liệu và xuất dữ liệu ra. Tất cả các hàm trong Go đều được định nghĩa bởi từ khóa `func`, theo sau là tên hàm (ở đây chúng ta định nghĩa hàm có tên `main`), tiếp theo là cặp dấu `()`, bên trong cặp dấu này chúng ta có thể khai báo một danh sách các tham số, tiếp theo là kiểu dữ liệu trả về (ở đây chúng ta không khai báo), rồi đến phần thân hàm nằm trong cặp dấu

ngoặc nhọn {}, thân hàm chứa các câu lệnh, ở đây chúng ta chỉ có duy nhất một câu lệnh. Chúng ta sẽ tìm hiểu thêm về hàm sau.

Ngoài ra cái tên main là một cái tên đặc biệt, hàm main tự động được hệ điều hành "gọi" đến đầu tiên khi chạy từ file khả thi.

```
fmt.Println("Hello World");
```

Bên trong hàm main chúng ta chỉ có một câu lệnh. Câu lệnh này có 3 phần. Câu lệnh này gọi hàm `Println()` trong gói `fmt.` Hàm này nhận vào tham số là một string có tên *"Hello World"*. Hàm `Println` (viết tắt của print line) thực hiện in chuỗi mà nó nhận được ra màn hình.

Tài liệu của Go được viết rất kỹ nhưng nếu bạn chưa học lập trình bao giờ thì đọc sẽ hơi thấy khó hiểu, ngoài ra tài liệu chủ yếu tiếng Anh là chính.

Revision #7

Created 22 September 2019 15:58:47 by Admin

Updated 22 September 2019 16:31:08 by Admin