

GO Programming Language

Presentation by Christina Marcial



Table of contents

01

Introduction

What is GO?

02

Basics

What makes GO special?

03

Hello World

How does this basic program look on GO?

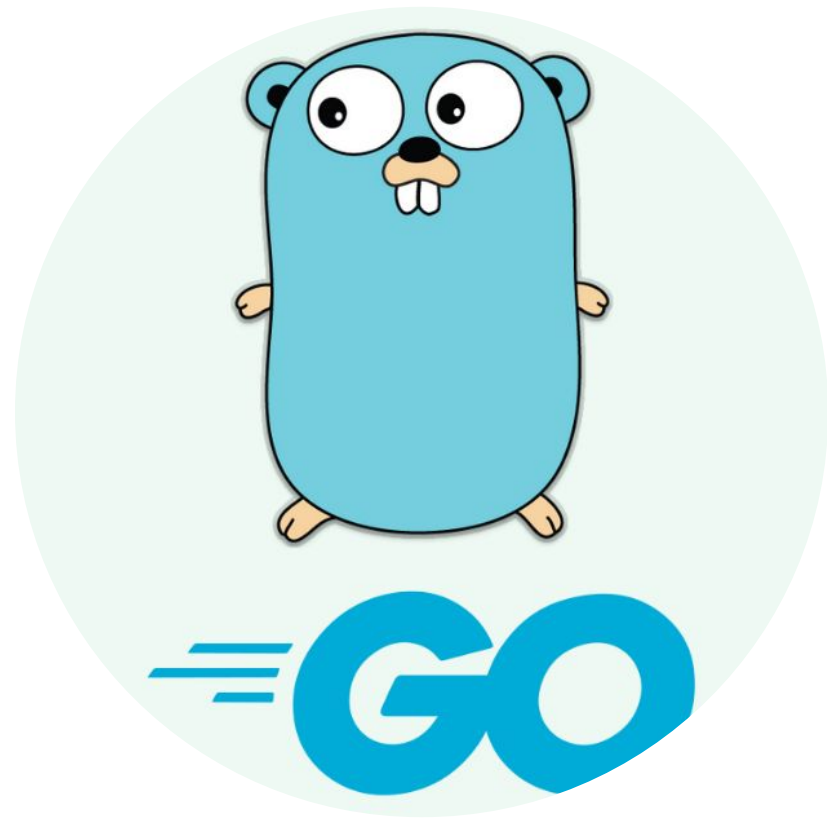
04

Implementation

What is GO's main idea?

Introduction

GO is procedural (similar to C) and designed by Google, using functions to make up a program. GO is unique because it runs fast, is simpler, and easy to learn. It is useful for making applications faster since it has enhanced memory performance. The mascot is a (go)pher.



Basic Structure

GO is structured a lot like C, and is capable of achieving things other languages can do, but is made simpler. GOroutines only take 2kb memory, allowing for many processes to run at once. This makes GO useful for light-weight microservices and fast CLIs. GO compiles very fast and isn't as complex as other compiled languages, so it almost feels like a scripting language instead.

functions.go

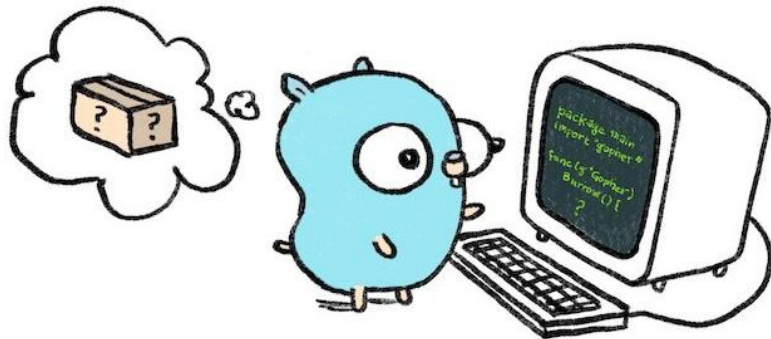
```
1 package main
2
3 import "fmt"
4
5 func add(x int, y int) int {
6     return x + y
7 }
8
9 func main() {
10     fmt.Println(add(42, 13))
11 }
12
```

Hello World

hello.go

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Println("Hello World!")
7 }
8
```

It looks similar to C++/C and runs in a terminal with “go run hello.go.” Go programs are made up of packages (collection of source files). “FMT” is the format package that allows for input and output.



Implementation

The main idea of GO is a focus on productivity and letting users get what they want done in a straightforward way. The language has automatic garbage collection, so the programmer doesn't have to worry about memory management. It may not be a replacement for C++/C, but it is fast and easy to learn and use for applications that need to be made simple and quickly.



```
package main
```

```
import (  
    "fmt"
```

```
)
```

```
func main () {
```

```
    fmt.Println("OK Let's Go!")
```

```
}
```